

UNIVERSIDAD DE EL SALVADOR
FACULTAD DE CIENCIAS NATURALES Y MATEMÁTICA
ESCUELA DE MATEMÁTICA



Una Caracterización del Desarrollo del Pensamiento
Algorítmico de los Estudiantes de las carreras de Licenciatura
en Matemática y Licenciatura en Estadística de la sede central
de la Universidad de El Salvador en el período 2018-2020

TRABAJO DE GRADUACIÓN PRESENTADO POR:
NAVAS LÓPEZ, EDUARDO ADAM

PARA OPTAR AL GRADO DE:
MAESTRO EN DIDÁCTICA DE LA MATEMÁTICA

CIUDAD UNIVERSITARIA, SAN SALVADOR,
EL SALVADOR, CENTROAMÉRICA
13 DE DICIEMBRE DE 2021

AUTORIDADES

UNIVERSIDAD DE EL SALVADOR

RECTOR:

MSc. ROGER ARMANDO ARIAS ALVARADO

SECRETARIO GENERAL:

Ing. FRANCISCO ANTONIO ALARCÓN SANDOVAL

FACULTAD DE CIENCIAS NATURALES Y MATEMÁTICAS

DECANO:

Lic. MAURICIO HERNÁN LOVO CÓRDOVA

SECRETARIO:

Lic. JAIME HUMBERTO SALINAS ESPINOZA

ESCUELA DE MATEMÁTICA

DIRECTOR:

PhD. DIMAS NOÉ TEJADA TEJADA

SECRETARIO:

MSc. CARLOS ERNESTO GÁMEZ RODRÍGUEZ

TRIBUNAL EVALUADOR

MSc. PEDRO ARMANDO RAMOS ALBERTO (ASESOR INTERNO)

PhD. MIGUEL CRUZ RAMÍREZ (ASESOR EXTERNO)

PhD. LÁZARO HUMBERTO VIERA HERNÁNDEZ (VOCAL)

CIUDAD UNIVERSITARIA
13 DE DICIEMBRE DE 2021

Dedicatoria

Le dedico este trabajo a todos aquellos que no reconocen a la Didáctica de la Matemática como una ciencia.

Agradecimientos

Al Maestro Pedro Ramos, amigo y colega, que me escuchó, me animó, me aconsejó, me apresuró y me hizo correr.

A Amanda Mármol, ya que gracias a ella pude avanzar mucho en la realización de este trabajo durante un tiempo.

Al Doctor Miguel Cruz, que nos enseñó, nos motivó y nos inspiró durante y después de los cursos de Maestría.

A los Doctores Lázaro Viera, Jesús Valle y Karina Guardado, por su buena disposición de revisar este trabajo.

A todos mis maestros y maestras, que de una u otra forma me transmitieron el deseo de enseñar.

A todos los que quieren ayudar a construir el «Otro Mundo Posible».

Al tiuj kiuj eĉ kontraŭ flue kunhelpas fari pli bonan mondon.

A Blaĉjo, Brunjo, Bandito y Capuchina, porque no se apartaron de mi lado durante las largas horas que invertí en este documento.



A la comunidad de desarrollo de Software Libre, ya que este trabajo se ha realizado enteramente usando software libre.



Y finalmente, gracias a la pandemia de COVID-19, porque debido a ella pude dedicar las largas horas que antes invertía para desplazarme al trabajo, en enriquecer esta investigación y alimentarme más sanamente en mi casa.

Índice de Contenidos

Índice de Cuadros	X
Índice de figuras	XIII
Índice de Listados de Código	XIV
Abstract	XVI
Resumen	XVIII
Resumo	XX
1. Introducción	1
1.1. Fundamentación del problema	1
1.2. Planteamiento del problema	4
1.3. Objetivos	4
1.3.1. Objetivo General	4
1.3.2. Objetivos Específicos	5
1.4. Justificación	5
1.5. Hipótesis	7
2. Marco Teórico	8
2.1. Definiciones	8
2.1.1. Algoritmo, Algoritmizar y Algoritmización	8
2.1.2. Pensamiento Computacional y Pensamiento Algorítmico	9
2.1.3. Ciencias de la Computación	11
2.1.4. Problemas, Resolución de Problemas y Diagramas tipo Schoenfeld	13
2.1.5. Rúbricas	15
2.1.6. Enfoque Sociocultural	16
2.2. La Habilidad Matemática de Algoritmizar	16
2.2.1. Diferencias de género	17
2.2.2. Etnomatemáticas	17
2.2.3. Resolución de problemas y demostraciones matemáticas	17
2.2.4. Desarrollo ontogenético de las operaciones aritméticas	18
2.2.5. Aprendizaje de la Matemática por medio de Calculadoras avanzadas y Computadoras	19
2.2.6. Aprendizaje de la Matemática por medio de Programación de Compu- tadoras	20
2.2.7. Modelamiento matemático	22

2.2.8. Desarrollo de habilidades profesionales	24
2.2.9. Educación inclusiva	26
2.2.10. Política educativa	26
2.3. Pensamiento Matemático y Pensamiento Algorítmico	27
2.3.1. Pensamiento Matemático	27
2.3.2. Orígenes del término Pensamiento Algorítmico y su relación con la Matemática	28
2.3.3. Caracterización y componentes operacionales del Pensamiento Algo- rítmico	30
2.3.4. Diversas investigaciones sobre Pensamiento Algorítmico	35
2.3.5. Beneficios o aportes del Pensamiento Algorítmico en la formación de matemáticos	36
2.3.6. Dificultades del Desarrollo del Pensamiento Algorítmico	38
2.4. Pensamiento Algorítmico y Pensamiento Computacional	41
2.4.1. Posibles confusiones de los términos	41
2.4.2. Orígenes y definición operacional del término Pensamiento Compu- tacional	43
2.4.3. Beneficios del Desarrollo del Pensamiento Computacional	49
2.4.4. Diferencias entre el Pensamiento Algorítmico y el Pensamiento Compu- tacional	50
2.5. Pensamiento Computacional y Matemática	51
3. Metodología	54
3.1. Tipo, diseño y alcance de la investigación	54
3.2. Población objeto de estudio y muestra	57
3.2.1. Población y muestra para la parte cuantitativa	57
3.2.2. Población y muestra para la parte cualitativa	57
3.3. Definición operacional de Pensamiento Algorítmico	58
3.4. Variables de la parte cuantitativa	66
3.5. Instrumento de recolección de datos de la parte cuantitativa	66
3.5.1. Indicaciones generales	67
3.5.2. Problemas	68
3.5.3. Aplicación del instrumento	72
3.5.4. Confiabilidad y validez	72
3.6. Operacionalización de las variables de la parte cuantitativa	73
3.6.1. Detalle de la Rúbrica de medición de la variable Pensamiento Algorít- mico de un Principiante	77
3.6.2. Rúbrica de medición de la variable Habilidad de modelamiento y re- solución de problemas de referencia	99
3.7. Extracción de datos de la parte cuantitativa	102
3.8. Archivo de datos de la parte cuantitativa	106
3.9. Análisis estadísticos aplicados a los datos de la parte cuantitativa	107
3.10. Recolección de datos de la parte cualitativa	107
3.11. Categorías para la modelación de los procesos cognitivos durante la resolu- ción de problemas de la parte cualitativa	110

3.12. Un diseño ampliado de Gráficos tipo Schoenfeld para representación de procesos cognitivos durante la resolución de problemas	111
3.13. Archivo de datos de la parte cualitativa	117
4. Resultados y Análisis de resultados	119
4.1. Resultados y Análisis de la Parte Cuantitativa	119
4.1.1. Análisis descriptivo de las variables	119
4.1.2. Análisis de normalidad	131
4.1.3. Análisis de correlación	133
4.1.4. Gráficos de correlación entre las diferentes variables de la parte cuantitativa	139
4.2. Resultados y Análisis de la Parte Cualitativa	167
4.2.1. Características generales de los sujetos	167
4.2.2. Resultados Gráficos	167
4.2.3. Análisis de los gráficos de la parte cualitativa	183
5. Conclusiones, Discusión y Recomendaciones	190
5.1. Conclusiones	190
5.1.1. Conclusiones de la parte cuantitativa	190
5.1.2. Interpretación de resultados de la parte cualitativa	192
5.1.3. Metainferencias (Triangulación) de los Resultados	192
5.2. Discusión y recomendaciones	193
5.2.1. Sobre la revisión bibliográfica	193
5.2.2. Sobre los resultados de la parte cuantitativa	193
5.2.3. Sobre los resultados de la parte cualitativa	195
5.2.4. Sobre la evaluación por rúbricas en la parte cuantitativa	196
Referencias Bibliográficas	198
A. Instrumentos de recolección de datos de la parte cuantitativa	216
B. Lista de problemas usados para la parte cualitativa	221
C. Planes de estudio	224
D. Scripts para la generación de los gráficos de la parte cuantitativa	227
E. Script para la generación de los Diagramas tipo Schoenfeld en la parte cualitativa	271
F. Meta-análisis cuantitativo de las Referencias Bibliográficas	280
G. Código fuente de los scripts para generar los gráficos del análisis de las Referencias Bibliográficas	284
H. Software Utilizado	293

Índice de cuadros

3.1. Población de para la parte cuantitativa	57
3.2. Población de para la parte cualitativa	58
3.3. Propuesta propia de Operacionalización genérica del <i>Pensamiento Algorítmico de un Principiante</i>	60
3.4. Propuesta de Rúbrica de evaluación genérica de las sub-variables del constructo <i>Pensamiento Algorítmico de un Principiante</i>	61
3.5. Tipología de las variables de la parte cuantitativa.	67
3.6. Operacionalización de la Variable PAP.	74
3.7. Operacionalización de la Variable RP.	76
3.8. Variantes básicas del problema 2.	80
3.9. Rúbrica de evaluación de las sub-variables de la variable PAP.	91
3.10. Rúbrica de evaluación de las sub-variables de la variable RP.	100
3.11. Columnas del archivo de datos cuantitativos.	106
3.12. Tareas cognitivas y anotaciones consideradas en esta investigación.	110
3.13. Ejemplo de registros en el archivo de datos cualitativos del ejemplo de la figura 3.17 en la página 116.	117
3.14. Ejemplo de registros en el archivo de datos cualitativos.	118
4.1. p-valores de las Pruebas de normalidad para los datos de las variables cuantitativas no dicotómicas.	133
4.2. Coeficientes de correlación biserial puntual.	137
4.3. Características de los sujetos de la parte cualitativa.	167
4.4. Relación entre cantidad de intervenciones de ayuda y la resolución exitosa.	184
4.5. Sujetos que dedicaron mucho o poco tiempo de Análisis del problema.	186
4.6. Sujetos que dedicaron mucho o poco tiempo de Planificación de la Solución.	186
4.7. Sujetos que realizaron Cambios abruptos de actividad cognitiva.	187
4.8. Sujetos que expresaron o no, no poder resolver su problema.	188

Índice de figuras

2.1. Gráfico de línea de tiempo de un estudiante típico intentando resolver un problema no familiar.	14
2.2. Gráfico de línea de tiempo de un matemático trabajando un problema difícil.	14
2.3. Episodios clave de Shoenfeld para la resolución de problemas.	15
2.4. Ciclo de modelamiento clásico	23
2.5. Ciclo de modelamiento extendido usando tecnología	24
2.6. Intersección del Pensamiento Computacional y del Pensamiento Matemático.	52
2.7. Resumen de la relación entre los principales conceptos de esta investigación.	53
3.1. Pasos de la parte cuantitativa de esta investigación.	55
3.2. Pasos de la parte cualitativa de esta investigación.	56
3.3. Diseño mixto concurrente de esta investigación.	56
3.4. Grafo que representa la información del Problema 1.	68
3.5. Triángulo rectángulo que hay que identificar en el Problema 3.	69
3.6. Figura del Problema 3.	70
3.7. Figura del Problema 4.	71
3.8. Vista lateral del bus del Problema 4.	84
3.9. Vista lateral de los asientos del Problema 4.	84
3.10. Solución realizada por el sujeto I-30 del problema 1 (a la izquierda), y evaluación y ponderación de esa solución por parte del investigador (a la derecha).	103
3.11. Solución realizada por el sujeto I-32 del problema 3 (a la izquierda), y evaluación y ponderación de esa solución por parte del investigador (a la derecha).	104
3.12. Solución realizada por el sujeto D-05 del problema 4 (a la izquierda), y evaluación y ponderación de esa solución por parte del investigador (a la derecha).	105
3.13. Capturas de pantalla de dos de las grabaciones de las sesiones de recolección de datos de la parte cualitativa.	109
3.14. Representación de una resolución de un ejercicio cuyo procedimiento es conocido. Equivalente al episodio A de la Figura 2.3 en la página 15.	113
3.15. Representación de un proceso creativo de resolución de problema sin errores. Equivalente al episodio B de la Figura 2.3 en la página 15.	114
3.16. Representación de una resolución creativa de problema que incluye dos errores. Equivalente al episodio C de la Figura 2.3 en la página 15.	115
3.17. Representación de una resolución fallida de un problema, con una falla técnica, una petición de ayuda y una expresión del alumno de incapacidad para resolver el problema.	116

4.1. Esquema de las relaciones examinadas entre las variables de la parte cuantitativa.	120
4.2. Histogramas de la variable edad.	123
4.3. Histogramas de la variable cum.	124
4.4. Histogramas de la variable año de ingreso.	125
4.5. Gráfico de proporciones de la variable gen (Género).	126
4.6. Gráfico de proporciones de la variable alg.	126
4.7. Gráfico de proporciones de la variable matProg.	127
4.8. Histogramas de la variable matTot.	128
4.9. Histograma de la variable PAP.	129
4.10. Histograma de la Variable RP.	130
4.11. Correlación entre las principales variables continuas y ordinales.	134
4.12. Mapa de Calor de los Coeficientes de Correlación de Spearman y Kendall entre las principales variables continuas y ordinales.	135
4.13. Esquema de las correlaciones entre las variables de la parte cuantitativa.	139
4.14. Correlación entre matProg y matTot.	139
4.15. Correlación entre las variables cum y edad.	140
4.16. Correlación entre PAP y RP.	141
4.17. Variable PAP según matProg.	142
4.18. Detalle de las sub-variables de PAP según matProg — Parte 1.	143
4.19. Detalle de las sub-variables de PAP según matProg — Parte 2.	144
4.20. Detalle de las sub-variables de PAP según matProg — Parte 3.	145
4.21. Detalle de las sub-variables de PAP según matTot — Parte 1.	146
4.22. Detalle de las sub-variables de PAP según matTot — Parte 2.	147
4.23. Detalle de las sub-variables de PAP según matTot — Parte 3.	148
4.24. Variable PAP según la variable alg.	149
4.25. Variable PAP según matProg y alg.	149
4.26. Variable PAP según cum.	150
4.27. Variable PAP según edad.	151
4.28. Variable PAP según gen.	152
4.29. Variable PAP según matProg y gen.	152
4.30. Variable PAP según gen en orden ascendente.	153
4.31. Variable PAP según gen en orden ascendente con corrección en la escala horizontal de mujeres.	154
4.32. Variable PAP según matTot.	155
4.33. Dispersión de la variable PAP según matTot.	156
4.34. Variable RP según alg.	157
4.35. Variable RP según cum.	158
4.36. Variable RP según edad.	159
4.37. Variable RP según gen.	160
4.38. Variable RP según gen en orden ascendente.	160
4.39. Variable RP según gen en orden ascendente con corrección en la escala horizontal de mujeres.	161
4.40. Variable RP según matProg.	162
4.41. Detalle de las sub-variables de RP según matProg.	163

4.42. Detalle de las sub-variables de RP según matTot.	164
4.43. Variable RP según matTot.	165
4.44. Dispersión de la variable RP según matTot.	166
4.45. Representación gráfica de la resolución del Sujeto S01.	168
4.46. Representación gráfica de la resolución del Sujeto S02.	169
4.47. Representación gráfica de la resolución del Sujeto S03.	170
4.48. Representación gráfica de la resolución del Sujeto S04.	171
4.49. Representación gráfica de la resolución del Sujeto S05.	172
4.50. Representación gráfica de la resolución del Sujeto S06.	173
4.51. Representación gráfica de la resolución del Sujeto S07.	174
4.52. Representación gráfica de la resolución del Sujeto S08.	175
4.53. Representación gráfica de la resolución del Sujeto S09.	176
4.54. Representación gráfica de la resolución del Sujeto S10.	177
4.55. Representación gráfica de la resolución del Sujeto S11.	178
4.56. Representación gráfica de la resolución del Sujeto S12.	179
4.57. Representación gráfica de la resolución del Sujeto S13.	180
4.58. Representación gráfica de la resolución del Sujeto S14.	181
4.59. Representación gráfica de la resolución del Sujeto Exp (experto).	182
F.1. Tipos de Referencias Bibliográficas.	281
F.2. Cantidad de Referencias por año.	282
F.3. Cantidad de Autores por Género por año.	283

Listados de código

D.1. Columnas del archivo de datos	227
D.2. Configuración de fuente del documento	228
D.3. Código de regresión lineal para usar en varios gráficos	228
D.4. Código de tabla de coeficientes de Correlación de Spearman y de Kendall	229
D.5. Histogramas de las variables PAP y RP	231
D.6. Histogramas de matTot, ingreso, edad y cum	233
D.7. Gráficos de pastel de matProg, alg y gen	235
D.8. Correlación entre cum y edad	237
D.9. Correlación entre matProg y matTot	239
D.10.Relaciones entre las variables PAP, gen, alg y matProg	240
D.11.Detalle de las sub-variables de PAP según matProg	243
D.12.Detalle de las sub-variables de PAP según matTot	247
D.13.Correlación entre PAP y RP	250
D.14.Relación entre PAP, RP y alg	251
D.15.Relación entre PAP, RP y gen	253
D.16.Relación entre PAP, RP y gen en orden ascendente	256
D.17.Relación entre PAP, RP y gen en orden ascendente con corrección de escala	258
D.18.Relación entre PAP, RP y matProg	260
D.19.Relaciones entre PAP, RP y matTot, edad y cum	261
D.20.Dispersión de PAP, RP según matTot	264
D.21.Detalle de las sub-variables de RP según matProg	266
D.22.Detalle de las sub-variables de RP según matTot	268
E.1. Script para generación de gráficos de Actividades Cognitivas tipo Schoenfeld	271
G.1. Script para generación de gráfico de Referencias por Tipo	284
G.2. Script para generación de gráfico de Referencias por año	286
G.3. Script para generación de gráfico de Autores por Género	289

Todos los programas y fragmentos de código en este documento son de elaboración propia.

Enlace a los archivos de datos

Los archivos de datos, la lista de referencias, todos los scripts, la presentación digital, y la versión más reciente de este documento se encuentra en la siguiente carpeta compartida:

<https://drive.google.com/drive/folders/142xLK7tHUWCoUyoZ41PbM6ghP6aeP0mz>



Abstract

Title: An Algorithmic Thinking Development Characterization for Students of the Bachelor in Mathematics and Bachelor in Statistics at the central campus of the University of El Salvador between 2018-2020.

This correlational and mixed-method —quantitative and qualitative— research constitutes an approach to the cross-sectional characterization of the *Algorithmic Thinking* development in Mathematics and Statistics Bachelor students at central campus of University of El Salvador, in the period from 2018 to 2020.

The quantitative part consists of a survey to measure the development level of *Algorithmic Thinking* from four groups of students (80 students in total) of the mentioned study programs, as the students increase their academic level according to the curriculum, specifically through the courses that require and are related to computer programming.

A moderate correlation was found between this advance in the academic level and the development level of *Algorithmic Thinking*. In addition, no significant correlation was found between the development level of *Algorithmic Thinking* and the global grades of the students, nor their age, nor their gender.

The qualitative part includes an in-depth study of the cognitive processes carried out by a group of students from the same study programs (14 students in total) during the resolution of mathematical problems whose solution is an algorithm.

It was found that students develop *Algorithmic Thinking* very slowly and with many obstacles, especially in the process of analyzing problems. Much variation was also found in the development of this type of thinking among students. Some of them have developed it acceptably and others have developed it very little.

The theoretical products of this work are: (a) the development of an operational definition of *Algorithmic Thinking* (section 3.3), (b) its corresponding generic evaluation rubric (table 3.4), (c) a specific rubric (table 3.9), and (d) a new expanded design of the Schoenfeld-like graphs (section 3.12).

Keywords: Algorithmic Thinking, Algorithmization, Computational Thinking, Schoenfeld Diagrams, Mathematician Training.

Mathematics Subject Classification 2020¹:

97P10 — Comprehensive works on computer science (educational aspects)

97C30 — Cognitive processes, learning theories (aspects of mathematics education)

97-02 — Research exposition (monographs, survey articles) pertaining to mathematics education

68W01 — General topics in the theory of algorithms

68-02 — Research exposition (monographs, survey articles) pertaining to computer science

ZDM Subject Classification Scheme²:

M15 — Mathematical modelling, applications of mathematics — Mathematization, its nature and its use in education.

Q35 — Psychological, pedagogical and didactical aspects of teaching and learning computer science — Cognitive processes (Concept formation, thought processes, problem solving. Learning)

¹<https://mathscinet.ams.org/mathscinet/msc/msc2020.html>

²http://www.imvibl.org/dmbl/novosti/zdm_didactical_mathematics_mlassification.pdf

Resumen

Título: Una Caracterización del Desarrollo del Pensamiento Algorítmico de los Estudiantes de las carreras de Licenciatura en Matemática y Licenciatura en Estadística de la sede central de la Universidad de El Salvador en el período 2018-2020.

Esta investigación correlacional de enfoque de métodos mixtos, cuantitativo y cualitativo, constituye un acercamiento a la caracterización transversal del desarrollo del *Pensamiento Algorítmico* en los estudiantes de las carreras de Licenciatura en Matemática y Licenciatura en Estadística de la sede central de la Universidad de El Salvador, en el período de 2018 a 2020.

La parte cuantitativa consiste de aplicar un instrumento de medición del nivel de desarrollo del *Pensamiento Algorítmico* a cuatro grupos de alumnos (80 alumnos en total) de las carreras mencionadas, a medida que los educandos de estas carreras van aumentando su nivel académico de acuerdo al pensum, específicamente a través de las asignaturas que requieren y están relacionadas con la programación de computadoras.

Se encontró una correlación moderada entre este avance en el nivel académico y el nivel de desarrollo del *Pensamiento Algorítmico*. Así mismo no se encontró una correlación significativa entre el nivel de desarrollo del *Pensamiento Algorítmico* y las notas globales de los estudiantes, ni su edad, ni su género.

La parte cualitativa comprende un estudio profundo de los procesos cognitivos que realiza un grupo de educandos de esas mismas carreras (14 alumnos en total) durante la resolución de problemas matemáticos cuya solución es un algoritmo.

Se encontró que los alumnos desarrollan el *Pensamiento Algorítmico* muy lentamente y con muchos obstáculos, especialmente en el proceso de analizar los problemas. También se encontró mucha variación en el desarrollo de este tipo de pensamiento entre los alumnos. Unos lo han desarrollado aceptablemente y otros lo han desarrollado muy poco.

Los productos teóricos de este trabajo son: (a) el desarrollo de una definición operacional propia de *Pensamiento Algorítmico* (sección 3.3), (b) su correspondiente rúbrica de evaluación genérica (cuadro 3.4), (c) una rúbrica específica (cuadro 3.9), y (d) un nuevo diseño ampliado de los gráficos tipo Schoenfeld (sección 3.12).

Palabras clave: Pensamiento Algorítmico, Algoritmización, Pensamiento Computacional, Diagramas de Schoenfeld, Formación de Matemáticos.

Estructura de este trabajo

En el capítulo 1 se presenta la fundamentación del problema, los objetivos, la justificación del problema de investigación, y las hipótesis de la parte cuantitativa de este trabajo de investigación.

El capítulo 2 incluye las definiciones conceptuales de los términos relacionados al tema de investigación, además de breves estados del arte de los tres principales, la Habilidad Matemática de Algoritmizar, del Pensamiento Algorítmico y del Pensamiento Computacional, y su relación con la matemática y la formación de matemáticos.

El capítulo 3 describe todos los métodos y técnicas aplicados en esta investigación, tanto para la parte cuantitativa como para la parte cualitativa. Todo lo relativo al diseño de la investigación, población, muestra, instrumento, operacionalización de variables, recolección y digitalización de datos, se encuentra en este capítulo.

En el capítulo 4 se presentan explícitamente los resultados de la parte cuantitativa y de la parte cualitativa de esta investigación.

El capítulo 5 incluye las conclusiones y la discusión de los resultados de esta investigación, tanto de la parte cuantitativa como de la cualitativa.

Posteriormente a la lista de Referencias bibliográficas se incluyen los apéndices. El primero (página 217) presenta el instrumento que se usó en la parte cuantitativa de esta investigación. El segundo lista los enunciados de los problemas usados para la parte cualitativa (página 221). El tercero (página 225) incluye la malla curricular de los planes de estudio vigentes de las carreras que cursan los sujetos analizados en esta investigación.

Los siguientes dos apéndices presentan los programas de computadora creados para la construcción de los gráficos del capítulo de Resultados. El apéndice D en la página 227 para la parte cuantitativa, y el apéndice E en la página 271 para la parte cualitativa.

Hacia el final, el apéndice F en la página 280, incluye un breve análisis cuantitativo de la lista de referencias de este documento. Específicamente se presenta la proporción de tipos de documentos referenciados, entre artículos de revistas científicas, artículos de memorias de congresos, capítulos de libro, tesis, libros, etc. Luego se analiza la antigüedad relativa de las referencias. Y finalmente se analiza la proporción de autores según su género y fecha de su publicación.

Finalmente, el apéndice H en la página 293, lista los principales programas de computadora utilizados para la realización de esta investigación y el correspondiente documento.

Resumo

Titolo: Karakterizo de disvolviĝo de Algoritma Pensmaniero de gestudentoj de Bakalaŭro pri Matematiko kaj Bakalaŭro pri Statistiko ĉe la centra sidejo de la Universitato de Salvadoro inter 2018 kaj 2020.

Jena interrilat-cela scienc-esplorado kun miksita alproksimiĝo, kvanteca kaj kvaliteca, konsistas el aliro al laŭlarĝa karakterizo de disvolviĝo de algoritma pensmaniero de gestudentoj de Bakalaŭro pri Matematiko kaj Bakalaŭro pri Statistiko ĉe la centra sidejo de la Universitato de Salvadoro inter 2018 kaj 2020.

La kvanteca parto konsistas el aplikado de enketo por mezuri la nivelon je disvolviĝo de algoritma pensmaniero de kvar grupoj da studentoj (80 gestudentoj entute) de la menciitaj bakalaŭroj laŭgrade kiel ili sukcesas tra la kursoj de la studprogramo, pli precize tra la kursoj rilataj al komputila programado.

Oni trovis mezgradan interrilaton inter la kvanto da finitaj kursoj kaj la nivelo je disvolviĝo de algoritma pensmaniero. Samtempe oni ne trovis gravan interrilaton inter la nivelo je disvolviĝo de algoritma pensmaniero kaj la meznotoj, aĝoj kaj genroj de gestudentoj.

La kvaliteca parto konsistas el profunda esploro de kognitivaj procezoj realigitaj de unu grupo da studentoj de la samaj bakalaŭroj (14 gestudentoj entute) dum ili solvas matematikajn problemojn kies solvo estas algoritmo.

Oni trovis ke la gestudentoj disvolviĝas algoritman pensmanieron tre malrapide kaj tra multaj obstakloj, speciale dum la procezo analizi problemojn. Ankaŭ oni trovis multan varion je la disvolviĝo de ĉi tiu pensmaniero inter la gestudentoj. Kelkaj el ili ĝin disvolviĝis akcepteble kaj aliaj disvolviĝis ĝin tre malmulte.

Teori-produktoj de la jena verko estas: (a) plenumo de propra operaci-difino por la termino Algoritma Pensmaniero (sekcio 3.3), (b) ĝia responda ĝenerala kvalifik-tabelo (tabelo 3.4), (c) specifa kvalifik-tabelo (tabelo 3.9), kaj (d) nova pliampleksigita desegno por Schoenfeld-similaj diagramoj (sekcio 3.12).

Ŝlosvortoj: Algoritma Pensmaniero, Peralgoritma Solvado, Komputika Pensmaniero, Schoenfeld-diagramoj, Instruado de Matematikistoj.

1. Introducción

En este capítulo se describe el problema científico en el que está enmarcada esta investigación. Aquí se plantea también los objetivos tanto general como específicos, y además la correspondiente justificación científica que plantea la pertinencia de esta investigación. Finalmente se enuncian las hipótesis de la parte cuantitativa de esta investigación.

1.1. Fundamentación del problema

Pyzara (2012) plantea que la matemática posee una naturaleza dual, que es al mismo tiempo conceptual y al mismo tiempo algorítmica. Estas dos caras de la misma moneda no pueden separarse, constantemente se permean la una a la otra y son igual de importantes (Sfard, 1991). Aún más, en la opinión de Davis y McGowen (2001, p. 311), «si las definiciones son la base de nuestra fundamentación matemática, entonces los algoritmos son los ladrillos en el puente de nuestra ruta matemática». Así que el estudio de los algoritmos es de alta importancia en la matemática y consecuentemente el estudio de los algoritmos es de alta importancia en la didáctica de la matemática.

Además, desde hace ya varias décadas se sabe que el uso de algoritmos en la enseñanza contribuye al desarrollo de habilidades de pensamiento lógico (Coscarelli, Visscher, y Schwen, 1976), que es muy valorado en la matemática.

Crear algoritmos en la matemática es tan importante que la *algoritmización* forma parte del Sistema Básico de Habilidades Generales Matemáticas de Hernández (1989, 1993) (como se citó en Delgado Rubí, 1999, p. 64), compuesto por las habilidades de Identificar, Interpretar, Calcular, Graficar, Recodificar, *Algoritmizar*, Definir y Demostrar.

Existen múltiples investigaciones que reportan la relación entre escribir y manipular algoritmos con el aprendizaje de la matemática. Estas se presentan con detalle en la sección 2.2 en la página 16. Pero ya a mediados de los años 70s Dijkstra (1974) reflexionaba que la Programación de Computadoras es una disciplina de naturaleza matemática. En la actualidad está bien reconocido que las Ciencias de la Computación son una de las principales áreas de estudio de la matemática. Esto puede comprobarse en el sistema de clasificación de publicaciones que utilizan la American Mathematical Society¹ como la European Mathematical Society² (Mathematical Reviews y zbMATH, 2020).

En algunos lugares del mundo desarrollado, los profesionales capacitados en TICs y con habilidades relacionadas con la programación de computadoras y las ciencias de la computación en general son los más demandados y los mejor pagados (Bureau of Labor Statistics,

¹<https://mathscinet.ams.org/msnhtml>

²<https://zbmath.org/classification/>

2020; Shefer, Nosova, y Lebedeva, 2018). Más aún, de entre las habilidades consideradas necesarias en estos profesionales, destaca explícitamente la habilidad de algoritmizar tareas (Shefer y cols., 2018).

Existe un cierto modo de pensamiento de estilo algorítmico, conocido como *Pensamiento Algorítmico*, fuertemente relacionado con las habilidades matemáticas, entre otras con la *Habilidad de Algoritmizar*, muy importante desde el punto de vista educativo (Doğan, 2020) como desde el punto de vista laboral y económico (Hurlburt, 2018).

Este *Pensamiento Algorítmico* tiene gran importancia no sólo en la matemática sino en muchos aspectos de la vida diaria en la actualidad (Sadykova y Il'bahtin, 2019; Sadykova y Usolzev, 2018). Kátai (2015), Sadykova y Il'bahtin (2019) y Doğan (2020) afirman que el *Pensamiento Algorítmico* es una habilidad importante en una sociedad basada en la información, que todos deberían poseer.

Este *Pensamiento Algorítmico* es a su vez un componente fundamental de otro modo de pensamiento conocido como *Pensamiento Computacional* (Kalelioğlu, Gülbahar, y Kukul, 2016; Korkmaz, Çakir, y Özden, 2017; Selby y Woollard, 2013). Y de modo similar, Wing (2006) y Grover y Pea (2018) señalan que el *Pensamiento Computacional* es una habilidad fundamental para todos, no sólo para los computólogos³. Según Wing (2006), a la lectura, la escritura y la aritmética, se debe agregar el *Pensamiento Computacional* a la capacidad analítica de cada niño.

Según algunos científicos de la computación, como Denning (2017), la sociedad como un todo adoptará eventualmente el *Pensamiento Computacional*, no por mandato político, sino por la implementación de ofertas educativas que ayuden a la gente a pensar a ser más efectiva en su propio dominio a través de la computación. Muchos científicos como Wing (2006, 2008, 2017), Barr, Harrison, y Conery (2011) y Riesco y cols. (2014) abogan por la inclusión del *Pensamiento Computacional* (más allá de la simple Ofimática⁴ o de la Alfabetización Digital básica) como materia esencial obligatoria desde la primaria para preparar a la población a las demandas del mundo tecnológico en el que ya vivimos.

En esta línea de pensamiento, D'Ambrosio (1999) propone un nuevo *Trivium* (un nuevo tipo de currículum) para la educación de las nuevas generaciones de ciudadanos de un mundo que se mueve hacia una civilización planetaria. Este nuevo currículum está compuesto por las competencias que él llama *Literacy*, *Matheracy* y *Technoracy*, entendiendo por *Literacy* la habilidad de entender e interpretar la información, gráficos, tablas, códigos, etc. que están ahora más relacionados con pantallas y botones que con papel y lápiz; por *Matheracy* la habilidad de llegar a conclusiones a partir de datos y cálculos, hacer inferencias y proponer hipótesis; y por *Technoracy* la habilidad de entender las ideas básicas, posibilidades y peligros de los dispositivos tecnológicos así como los asuntos morales subyacentes de su uso.

Wing (2006) menciona cómo las Ciencias de la Computación han cambiado y seguirán cambiando profundamente diversas ciencias como la Estadística, la Biología, la Economía, la Química y la Física. Lee y Malyn-Smith (2019) también mencionan esta influencia en la Astrofísica, la Arqueología, la Neurociencia, las Ciencias Sociales y la Historia.

³Nos referimos a «*computer scientists*». Ver más sobre este término en la sección 2.1.3.

⁴<https://es.wikipedia.org/wiki/Ofimática>

En general se espera que los profesionales que se gradúan de las carreras de Matemática, Física, Biología, Ingeniería Mecánica, Biomédica, Eléctrica y otras, se hayan apropiado de los principales avances científicos y técnicos relacionados a las Ciencias de la Computación⁵ (Salgado Castillo y Alonso Berenguer, 2017). Particularmente en las carreras de Matemática esto es importante ya que el *Pensamiento Algorítmico* es una arista importante del pensamiento matemático (Lockwood, DeJarnette, Asay, y Thomas, 2016) y de la matemática en sí misma (Lovász, 1988; Pyzara, 2012; Sfard, 1991). Después de todo, «la matemática es una ciencia de patrones y orden, [y] su dominio no son moléculas o células, sino los números, el azar, la forma, los *algoritmos*, y el cambio» (National Research Council, 1989, p. 31).

Gravemeijer, Stephan, Julie, Lin, y Ohtani (2017) ponen especial consideración a la posición de la matemática en una sociedad digitalizada y a las formas matemáticas específicas de argumentación y comunicación para hacer la pregunta de qué matemáticas se deberían enseñar cuando las tecnologías digitales pueden hacer la mayoría del trabajo matemático que se enseña en este momento. Estos autores argumentan que la digitalización de la sociedad del siglo 21 requiere un cambio de enfoque, y pasar de enfatizar en las competencias matemáticas que compiten con lo que las computadoras pueden hacer, para pasar a enfatizar en las competencias matemáticas que complementen el trabajo de las computadoras.

Por otro lado, la investigación de Lockwood y cols. (2016) concluye que la noción de *Pensamiento Computacional* no está clara en el lenguaje cotidiano de los matemáticos, pero la noción de *Pensamiento Algorítmico* sí tiene presencia entre las habilidades que los matemáticos valoran en la práctica de hacer matemática. Sin embargo, la investigación de Buteau, Jarvis, y Lavicza (2014) muestra que a pesar de que en la formación académica de los matemáticos el aprendizaje de la programación de computadoras está ampliamente aceptado, hay ciertos vacíos en lo relativo al desarrollo de la habilidad de escribir algoritmos.

En la encuesta desarrollada por Buteau y cols. (2014) se hace un análisis de la relación existente entre la enseñanza universitaria de ciertas tecnologías computacionales y su uso en la investigación profesional de los matemáticos. En este estudio se descubrió que mientras que el 43 % de los matemáticos encuestados reportó usar programación de computadoras en sus investigaciones, sólo el 18 % indicó que la usan en su práctica docente. Es decir que escribir algoritmos forma parte de las habilidades profesionales de casi la mitad de los matemáticos, sin embargo es algo que se usa en menos de una quinta parte de la formación de los nuevos matemáticos.

Cada vez se da mayor importancia al uso de la tecnología en la matemática para ser usada en sus diferentes aplicaciones, en las ciencias naturales y también en la enseñanza de la matemática (ver secciones 2.2.5, 2.2.6, 2.3.5 y 2.4.3). Pero hay que considerar investigaciones como la de Maass, Geiger, Romero Ariza, y Goos (2019) que reporta que entre el 15 % y el 20 % de los estudiantes de países desarrollados carecen de habilidades en matemática y ciencia suficientes para incrementar la competitividad internacional. En América Latina y

⁵Aquí los autores usan la palabra «informática», pero para mejorar la fluidez se ha utilizado como sinónimo el término «ciencias de la computación».

el Caribe, los jóvenes no están siendo preparados de manera apropiada para contar con las herramientas en matemáticas y ciencias naturales necesarias en una economía mundial cada vez más interconectada (Treviño y cols., 2015; Valverde y Näslund-Hadley, 2010).

En lo que respecta a los resultados en cursos introductorios de programación de computadoras a nivel mundial, estos son en general también muy bajos en términos de las aprobaciones y aprovechamiento. Guzdial y Soloway (2002) nos hablan de porcentajes de reprobación de 15%–30%. Kurland, Clement, Mawby, y Pea (2002) presentan hasta 79% de falla en algunos problemas de su instrumento de evaluación. En la investigación de Gayo Avello y cols. (2003) se reporta el 40% de reprobación. El estudio de Kinnunen y Malmi (2006) nos habla de reprobaciones de 30%–50% durante muchos años. McCracken y cols. (2001) realizaron una exploración en 216 estudiantes de segundo año de carreras que incluyen programación de computadoras en su primer año, de cuatro universidades de tres países de primer mundo, y lo que se encontró fue una nota promedio de 22,89 de 110 puntos, que equivale a un promedio de 20,81 en escala de 0 a 100 puntos. Similarmente en un reporte del Centro de Innovación y Calidad de la Educación (2012) (como se citó en Vargas, Blanco, Pérez, y Rodríguez, 2013, en Blanco Hamad, Salgado Castillo, y Alonso Berenguer (2016) y en Blanco Hamad, 2016), en el que se evaluó a 1128 estudiantes que ya habían aprobado las asignaturas de la disciplina algorítmica y matemática, indica que el 79,4% aún tiene dificultades en matemática y algoritmización. En uno de los instrumentos de la investigación de Blanco Hamad (2016, p. 38) «el 92,5% de los estudiantes no llegó a expresar una representación adecuada del problema y sólo el 7,5% resolvió correctamente el problema, realizando un correcto análisis e interpretación». Arévalo-Mercado, Estrada-Rentería, y Muñoz-Andrade (2019) indican que históricamente sus reprobaciones son de 30% a 40%.

1.2. Planteamiento del problema

El problema científico en el que se enmarca esta investigación es:

¿Cómo se desarrolla o evoluciona el *Pensamiento Algorítmico* en los estudiantes de las carreras de Licenciatura en Matemática y Licenciatura en Estadística, impartidas por la Escuela de Matemática de la Facultad de Ciencias Naturales y Matemática de la Universidad de El Salvador?

1.3. Objetivos

1.3.1. Objetivo General

Caracterizar el desarrollo del *Pensamiento Algorítmico* alcanzado por los estudiantes de las carreras de Licenciatura en Matemática y Licenciatura en Estadística, impartidas por la Escuela de Matemática de la Facultad de Ciencias Naturales y Matemática de la Universidad de El Salvador en el período de 2018 a 2020, por medio de una investigación de

enfoque mixto, con una parte cuantitativa y una parte cualitativa, para lograr un mayor entendimiento de cómo se desarrolla este tipo de pensamiento.

1.3.2. Objetivos Específicos

1. Realizar una sistematización de los antecedentes, fundamentos y referentes conceptuales del *Pensamiento Algorítmico* y de otros constructos relacionados, como la *Habilidad Matemática de Algoritmizar* y el *Pensamiento Computacional*, por medio de una búsqueda profunda y análisis selectivo de la literatura científica disponible.
2. Encontrar una definición conceptual y una definición operacional del *Pensamiento Algorítmico*, por medio de la identificación y valoración de las definiciones publicadas en la literatura científica correspondiente.
3. Construir un instrumento de recolección de información, y su correspondiente rúbrica, que cuantifique el nivel de desarrollo del *Pensamiento Algorítmico* alcanzado por estudiantes de las carreras de Licenciatura en Matemática y Licenciatura en Estadística, impartidas por la Escuela de Matemática de la Facultad de Ciencias Naturales y Matemática de la Universidad de El Salvador.
4. Realizar un análisis de correlación estadística múltiple entre el nivel de desarrollo de *Pensamiento Algorítmico* alcanzado por los estudiantes de las carreras de Licenciatura en Matemática y Licenciatura en Estadística, impartidas por la Escuela de Matemática de la Facultad de Ciencias Naturales y Matemática de la Universidad de El Salvador, y la cantidad de materias relacionadas con la programación de computadoras que han aprobado, la cantidad total de materias de la carrera que han aprobado, su Coeficiente de Unidades de Mérito (CUM), edad, y género.
5. Profundizar el entendimiento sobre los procesos cognitivos durante la resolución de problemas cuya solución es un algoritmo, de un grupo de estudiantes de las carreras de Licenciatura en Matemática y Licenciatura en Estadística, impartidas por la Escuela de Matemática de la Facultad de Ciencias Naturales y Matemática de la Universidad de El Salvador, durante su aprendizaje de la programación de computadoras, por medio de una exploración cualitativa de tales procesos.

1.4. Justificación

Modeste (2016) defiende que la investigación epistemológica sobre las interacciones entre las matemáticas y las ciencias de la computación⁶ es necesaria para alimentar la investigación didáctica sobre la enseñanza-aprendizaje de las matemáticas de hoy⁷, impactadas por el desarrollo de las ciencias de la computación⁸.

⁶Modeste utiliza en inglés el término «*Informatics*» como traducción del término francés «*Informatique*» que utiliza en (2012a; 2012b; 2016), pero que en inglés se refiere al término «*Computer Science*».

⁷mucho más sobre este tema en Durand-Guerrier, Meyer, y Modeste (2019)

⁸En la publicación de Modeste (2016) aquí ocurre de nuevo el mismo uso de la palabra «*Informatics*» en lugar de «*Computer Science*» debido al origen francés de los autores.

«En la transición hacia la sociedad de la información, en las condiciones de interacción constante con los sistemas informáticos, el estilo algorítmico de pensamiento [*Pensamiento Algorítmico*] es una base necesaria para el desempeño de toda persona moderna.» (Grozdev y Terzieva, 2015, p. 471)

Además, «muchos campos de la vida moderna envuelven procesos como: seguir procedimientos, aplicar protocolos, o implementar técnicas; todas estas tareas pueden ser vistas como algoritmos procesados por humanos. [...] Por otro lado, una de las principales características de la era digital es que el control detrás de la tecnología que ha impregnado todos los sectores de la sociedad está implementado con algoritmos procesados por computadoras.» (Kátai, 2015, p. 287)

Y como se mencionó en la sección 1.1, según algunos computólogos (*computer scientists*), la sociedad como un todo adoptará poco a poco el *Pensamiento Computacional* a través de ofertas educativas que ayuden a la gente a pensar a ser más efectiva en su propio dominio a través de la computación (Denning, 2017). En ese mismo sentido algunos investigadores como Barr y cols. (2011) y Riesco y cols. (2014) abogan por la inclusión del *Pensamiento Computacional* como materia esencial obligatoria desde la primaria, para preparar a la población a las demandas del mundo tecnológico en el que ya vivimos.

Schoenfeld (2013) nos dice que la presencia de las herramientas computacionales tiene el potencial de cambiar radicalmente la forma del conocimiento al que los estudiantes tienen acceso en las aulas de clase, y los modos en los que ellos pueden operar con ese conocimiento.

Existe además una fuerte tendencia en la investigación de la enseñanza integrada del *Pensamiento Computacional* con las ciencias en general (Waterman y cols., 2018; Weintrop y cols., 2016), y particularmente con la matemática (Broley, Buteau, y Muller, 2017; Broley, Caron, y Saint-Aubin, 2018; Freiman, Broley, Buteau, y Vasilyeva, 2018; Israel y Lash, 2019; Kilhamn y Bråting, 2019)⁹. Sin embargo, Tang, Yin, Lin, Hadad, y Zhai (2020, p. 6) reportan en su revisión sistemática de literatura sobre las investigaciones sobre *Pensamiento Computacional* que sólo el 15,45% están orientadas a la educación universitaria.

diSessa (2018) afirma que ve que la computación, potencialmente, provee una alfabetización nueva, profunda y con gran influencia, la alfabetización computacional, y que impacta todas las disciplinas STEM¹⁰ en su núcleo mismo, pero más especialmente en términos de aprendizaje.

En el caso particular de los estudiantes de carreras universitarias relacionadas con la matemática, ya se mencionó cómo Coscarelli y cols. (1976), Sfard (1991), Pyzara (2012), Buteau y cols. (2014) y Lockwood y cols. (2016) enfatizan la importancia del *Pensamiento Algorítmico* en la Matemática en sí misma y por consiguiente en la formación de futuros matemáticos y de futuros Profesores de Matemática.

Sin embargo, en diferentes partes del mundo se presenta un bajo rendimiento en las materias relacionadas con la programación de computadoras, tal como reportan McCracken y cols. (2001), Guzdial y Soloway (2002), Kurland y cols. (2002), Gayo Avello y cols. (2003),

⁹Sobre este tema se habla más ampliamente en la sección 2.5.

¹⁰Science, Technology, Engineering and Mathematics

Kinnunen y Malmi (2006), Futschek (2006), Vargas y cols. (2013), Blanco Hamad y cols. (2016) y Arévalo-Mercado y cols. (2019). Este bajo rendimiento en las materias relacionadas con la programación de computadoras también ocurre entre los estudiantes de la Licenciatura en Matemática y la Licenciatura en Estadística de la Escuela de Matemática de la Universidad de El Salvador. Los porcentajes de reprobación son de 45% al 51% en los últimos años (2016–2019) en la Escuela de Matemática de la Facultad de Ciencias Naturales y Matemática de la Universidad de El Salvador.

A pesar de los intentos, durante las etapas iniciales de esta investigación, la búsqueda bibliográfica arrojó nulos resultados sobre términos como «*Algoritmización*» y «*Pensamiento Algorítmico*» en publicaciones sobre la realidad salvadoreña. Sin embargo, en El Salvador se comparten similares problemáticas y se realizan debates informales similares en torno al desarrollo del *Pensamiento Algorítmico* manifestado en los bajos resultados de los cursos introductorios de programación de computadoras.

Es así como en base a la breve revisión bibliográfica presentada en este capítulo se considera de gran importancia estudiar el desarrollo del *Pensamiento Algorítmico*, en particular entre los estudiantes de las carreras de Licenciatura en Matemática y Licenciatura en Estadística, impartidas por la Escuela de Matemática de la Facultad de Ciencias Naturales y Matemática de la Universidad de El Salvador.

1.5. Hipótesis

Para esta investigación se plantean las siguientes hipótesis en relación a los estudiantes de las carreras de Licenciatura en Matemática y Licenciatura en Estadística, impartidas por la Escuela de Matemática de la Facultad de Ciencias Naturales y Matemática de la Universidad de El Salvador:

1. El nivel de desarrollo de *Pensamiento Algorítmico* de los estudiantes está positivamente correlacionado con la cantidad de materias relacionadas con la programación de computadoras que han aprobado.
2. El nivel de desarrollo de *Pensamiento Algorítmico* de los estudiantes está positivamente correlacionado con la cantidad total de materias universitarias que han aprobado.
3. El nivel de desarrollo de *Pensamiento Algorítmico* de los estudiantes está positivamente correlacionado con su Coeficiente de Unidades de Mérito.
4. El nivel de desarrollo de *Pensamiento Algorítmico* de los estudiantes está positivamente correlacionado con su edad.
5. El nivel de desarrollo de *Pensamiento Algorítmico* de los estudiantes es independiente de su género (es decir que es equivalente en hombres y en mujeres).
6. El nivel de desarrollo de *Pensamiento Algorítmico* de los estudiantes se manifiesta diferente cuando se les solicita desarrollar un algoritmo que cuando se les solicita desarrollar unas instrucciones.

2. Marco Teórico

En este capítulo se describe, explica y analiza los conceptos y definiciones más relevantes para esta investigación.

2.1. Definiciones

A continuación se presentan las principales definiciones conceptuales que se utilizarán y asumirán en este trabajo.

2.1.1. Algoritmo, Algoritmizar y Algoritmización

Lo primero que hay que mencionar es el hecho de que los algoritmos son objetos matemáticos distinguibles de su implementación como programa de computadora (Lovász, 1988).

Para este trabajo se asumirá la definición de Algoritmo de Pyzara (2012, p.53): Un **Algoritmo** es «un esquema o descripción precisa de la solución a un problema expresado con operaciones que un ejecutante entiende y es capaz de llevar a cabo». Se prefiere debido a que se refiere más ampliamente a la ejecución por parte de humanos tanto como por máquinas.

Por otra parte, Knuth (1974, p.323) define Algoritmo como «una secuencia precisa y definida de reglas [indicaciones] que dicen cómo producir información de salida especificada a partir de información de entrada dada en un número finito de pasos.» La definición de Kátai (2015, p.287) es más simple, dice que el término algoritmo esencialmente se refiere a una secuencia de pasos lógicos orientados a realizar una tarea bien definida. En su artículo sobre aprendizaje de la matemática por medio de Programación de Computadoras, Mifseldt y Ejsing-Duun (2015, p.2526) definen algoritmo como una descripción sistemática de estrategias de construcción y de resolución de problemas, de relaciones causa-efecto, y de eventos. De manera similar, Modeste (2012a) dice que «un algoritmo es un procedimiento para resolver problemas, que puede ser aplicado a un conjunto de instancias del problema y que produce, en un número finito de pasos constructivos, claros y organizados, la respuesta al problema para alguna de sus instancias.»

En esta investigación se asumirá la definición de la acción de **Algoritmizar** de Delgado Rubí (1998) (como se citó en Vargas y cols. 2013, p. 1628): «Plantear una sucesión estricta de operaciones matemáticas que describan un procedimiento conducente a la solución de determinado problema.». De forma similar, la **Algoritmización** se entenderá como la construcción de un algoritmo para resolver algún problema o familia de problemas.

2.1.2. Pensamiento Computacional y Pensamiento Algorítmico

Otra definición importante es **Pensamiento Computacional**. Para Aho (2012, p. 832), el *Pensamiento Computacional* es «el proceso de pensamiento involucrado en la formulación de problemas tal que su solución puede ser representada como pasos computacionales y algoritmos». Voogt, Fisser, Good, Mishra, y Yadav (2015, p. 726) opinan que «el *Pensamiento Computacional* se refiere a los procesos de pensamiento que están involucrados cuando se resuelven problemas complejos y cuando se generalizan y transfieren estos procesos de resolución de problemas a una amplia variedad de problemas.»

Mientras que para Wing (2017, p. 8), «*Pensamiento Computacional* son los procesos de pensamiento involucrados en la formulación de problemas y en la expresión de sus soluciones de tal forma que una computadora —humana o mecánica— pueden llevar a cabo efectivamente.». De forma similar, Waterman y cols. (2018, p. 283) considera que «*Pensamiento Computacional* es una forma de pensar que involucra la formulación y descomposición de problemas, y la estructuración y comunicación de las soluciones tal que pueden ser entendidas por humanos y procesadas por máquinas».

Las cuatro definiciones conceptuales son en esencia equivalentes, pero en esta investigación se prefiere la de Waterman y cols. (2018) ya que es más explícita que la de Aho (2012) y la de Voogt y cols. (2015), y más fácil de entender que la de Wing (2017).

Aho (2012) utiliza el término «paso computacional» que es explicado en su artículo, pero que no es de uso común. En ese sentido es menos explícita que la de Waterman y cols. (2018). La definición de Voogt y cols. (2015) menciona «problemas complejos», pero eso parece muy extremo, ya que los problemas abordados en entornos escolares no tienen por qué ser «complejos». Además, la «amplia variedad de problemas» de los que hablan, es muy vago. Por otro lado, Wing (2017) habla de «computadoras humanas» refiriéndose a cuando una persona ejecuta un algoritmo concreto con precisión y sin errores. Por ello, aunque tal idea es clara, no es tan fácil de entender como la de Waterman y cols. (2018). Claro que la definición de Waterman y cols. (2018) no aclara de qué clase de humanos y qué clase de máquinas se trata, por lo que la definición queda en función de los humanos disponibles y las máquinas disponibles.

Sin embargo, la definición de Wing (2017) indica, tal como ella misma lo explica, tres cosas importantes: Primero, que las personas pueden computar; segundo, que las personas pueden aprender *pensamiento computacional* sin una máquina; y tercero, que el *pensamiento computacional* no trata solamente de resolución de problemas, sino también de (re)formulación de problemas.

Diferentes definiciones operacionales de *Pensamiento Computacional* se analizan más adelante, en la sub-sección 2.4.2 en la página 43.

Por otro lado, se asumirá que **Pensamiento Algorítmico** es «Una forma lógica y organizada de pensar usada para descomponer un objetivo complicado en una serie de pasos (ordenados) o etapas usando las herramientas disponibles» (Lockwood y cols., 2016, p. 1591). «Este [modo de pensamiento] involucra planeamiento y diseño de los pasos, tener una idea general de lo que puede hacer el algoritmo, así como tener los detalles en su lugar para implementarlo con éxito.» (Lockwood y cols., 2016, p. 1594)

Existen otras definiciones más sencillas e informales de *Pensamiento Algorítmico*. Por ejemplo Kátai (2015, p.287) afirma que el *Pensamiento Algorítmico* está relacionado al concepto de crear y procesar algoritmos. Mientras que Misfeldt y Ejsing-Duun (2015, p.2526) definen *Pensamiento Algorítmico* como la habilidad de trabajar con algoritmos; similar a la investigación de Vera Olivera (2019, p.15) según la cual el *Pensamiento Algorítmico* es «La Capacidad de entender, ejecutar, evaluar y crear algoritmos».

Está también la definición de Mingus y Grassl (1998, p. 34) (como se citó en Modeste, 2012b, p. 472): «El *Pensamiento Algorítmico* es un método de pensar y guiar los procesos de pensamiento que utiliza procedimientos paso a paso, requiere entradas y produce salidas, requiere decisiones sobre la calidad y la idoneidad de la información que entra y de la que sale, y monitorea los procesos de pensamiento como un medio para controlar y dirigir el proceso de pensamiento. En esencia, el pensamiento algorítmico es al mismo tiempo un método de pensamiento y un medio para pensar en el propio pensamiento.»

Modeste (2012b, p. 472) está de acuerdo con que «el *pensamiento algorítmico* sería entonces una forma de abordar un problema tratando de sistematizar su resolución, de cuestionar la forma en que los algoritmos podrían o no resolverlo.», y enfatiza que el *Pensamiento Algorítmico* no se trata sólo de aprender e implementar algoritmos, sino que también abarca su producción, comprensión y estudio.

Grozdev y Terzieva (2011) (como se citó en Grozdev y Terzieva, 2015, p.471), afirma que el *Pensamiento Algorítmico* es «una forma de pensar, que proporciona una solución para una tarea específica a través de una sucesión de acciones elementales». Similar a esta definición, para Ziatdinov y Musa (2012, p.1109) el *Pensamiento Algorítmico* es «la habilidad de resolver problemas de diversos orígenes, que requieren un plan de acción para lograr el resultado deseado.»

Para Hoyles y Noss (2015) (como se citó en Sinclair y Patterson, 2018, p. 63) el *Pensamiento Algorítmico* es «la propensión a ver las tareas en términos de pasos discretos conectados más pequeños».

Las definiciones de Grozdev y Terzieva (2011) y Hoyles y Noss (2015) tienen el problema de que hablan de «acciones elementales» y «pasos discretos», pero tales operaciones atómicas dependen de la tecnología (o lenguaje) subyacente en el que se está expresando el algoritmo. Se debe observar que lo que en un lenguaje de programación de alto nivel es una operación atómica (por ejemplo una instrucción de suma de matrices) requiere de muchas instrucciones en un lenguaje de bajo nivel.

En la investigación de Sadykova y Usolzev (2018, p.5), el *Pensamiento Algorítmico* se define como «un sistema de operaciones mentales y procedimientos con componentes reales y abstractos basados en objetos que conducen al desarrollo de un algoritmo optimizado para la solución de un problema dado».

Para Sadykova y Il'bahtin (2019, p.421), «el *Pensamiento Algorítmico* es un sistema de métodos de pensamiento que es necesario para construir una secuencia de resultados intermedios, planificando la estructura de acciones y su implementación, conduciendo al logro de una meta».

Más recientemente, Juškevičienė (2020, p. 148) define sucintamente el *Pensamiento Algorítmico* como la «habilidad de formular problemas que transforman una entrada en la salida deseada usando algoritmos.»

En esta investigación se prefiere la definición de Lockwood y cols. (2016), es decir que se entenderá que el *Pensamiento Algorítmico* es una forma de pensar usada para descomponer un objetivo complicado en una serie de pasos (ordenados) que deben diseñarse usando las herramientas disponibles. Se prefiere ya que no implica que la persona ejecutante deba conocer y comprender el concepto de *Algoritmo* previamente, es decir, es más cercana a lo que se esperaría de un estudiante que todavía no sabe programar o que está comenzando a aprender a programar, a diferencia de la de Mingus y Grassl (1998) que parece describir más a un experto en *Pensamiento Algorítmico*. La definición de Lockwood y cols. (2016) también aclara el tema de las acciones elementales de las que hablan Grozdev y Terzieva (2011) y Hoyles y Noss (2015), refiriéndose en su lugar a «las herramientas disponibles», lo cual es más apropiado ya que deja claro que los pasos dependen de la tecnología concreta en la que se expresan esos pasos.

Debe recordarse que en esta sección se está hablando sólo de definiciones conceptuales, mientras que la definición operacional se discutirá más adelante en la sub-sección 3.3 en la página 58.

Por otro lado, esta investigación explora el «Desarrollo del Pensamiento Algorítmico», y según Andreev (1996), como se citó en Grozdev y Terzieva (2015, p. 471), «el **desarrollo del pensamiento** en los procesos de aprendizaje significa la formación y perfeccionamiento de todos los tipos, formas y operaciones de pensamiento, desarrollo de habilidades y hábitos de aplicar las leyes de pensamiento en las actividades cognitivas y de aprendizaje, tanto como los hábitos para transferir los métodos de actividad intelectual de un área de conocimiento a otra».

2.1.3. Ciencias de la Computación

En los inicios de las Ciencias de la Computación Donald Knuth (1985, p. 170) reconoce que su definición no estaba todavía bien definida. Afirma que en Estados Unidos se le llamaba Ciencias de la Computación a esta «ciencia», lo cual enfatiza el hecho de que los algoritmos son ejecutados por máquinas. Para él, esta «ciencia» es principalmente el estudio de los algoritmos y las estructuras de dato. De hecho, él afirma que él la llamaría *Informatics* si pudiera. También explica que en Francia y Alemania se le conocía precisamente como *Informatique* e *Informatik* respectivamente, enfatizando que trata más que sólo de los procesos en sí mismos. De hecho, aún ahora se le llama así en el mundo francófono, ver Modeste, Gravier, y Ouvrier-Buffet (2010, p. 8). Knuth (1985) también explica que en el mundo rusohablante de la época se le llamaba «кибернетика» (/Kibernetika/) enfatizando el control sobre los procesos y la utilidad de esta «ciencia».

Posteriormente, Aho y Ullman (1992) definen las **Ciencias de la Computación** como la ciencia de la abstracción, cuyo propósito es crear el modelo correcto para pensar sobre los problemas e inventar las técnicas mecanizables apropiadas para resolverlos.

La definición de Ciencias de la Computación de la ACM (Association for Computing Machinery¹) en 2003, específicamente para educación básica y media (K-12) argumenta que las ciencias de la computación no es Programación [de Computadoras] ni Alfabetización Digital², sino que es el estudio de las computadoras y los procesos algorítmicos incluyendo sus principios, el diseño de su hardware y su software, sus aplicaciones, y su impacto en la sociedad, incluyendo: programación [de computadoras], hardware, diseño, redes [de computadoras], gráficos, bases de datos y recuperación de información, seguridad computacional, diseño de software, lenguajes y paradigmas de programación, lógica, traducción entre diferentes niveles de abstracción, inteligencia artificial, los límites de la computación (lo que las computadoras no pueden hacer), aplicaciones en tecnología de la información y sistemas de información, y cuestiones sociales (seguridad en internet, privacidad, propiedad intelectual, etc.). (Tucker y cols., 2003, p. 6).

De forma más escueta, Wing (2006, p.34) define a las Ciencias de la Computación como el estudio de la *computación*, es decir, de lo que puede ser computado y cómo computarlo. Aquí, *computar* se debe entender en el sentido del modelo de la Máquina de Turing.

Las Ciencias de la Computación frecuentemente son confundidas con el uso cotidiano de las computadoras, como aprender cómo usar el Internet y crear presentaciones de diapositivas digitales. Padres, profesores, estudiantes y administradores educativos pueden compartir esta confusión. Una encuesta muestra que la mayoría de los estudiantes creen que crear documentos y presentaciones (78%) y buscar en Internet (57%) son actividades de las Ciencias de la Computación (Google y Gallup, 2015). Los padres, profesores y directores son casi tan malos en delinear la diferencia entre las actividades tradicionales de Alfabetización Digital y las Ciencias de la Computación, y en realidad más padres que estudiantes creen que hacer una búsqueda en Internet es Ciencias de la Computación. Y tal confusión se extiende a los administradores educativos (Google y Gallup, 2015).

Esta investigación no trata propiamente sobre Ciencias de la Computación, pero se considera que la descripción de Tucker y cols. (2003) es bastante clara sobre la gran amplitud de campos que abarcan las modernas Ciencias de la Computación y la definición de Aho y Ullman (1992) es bastante precisa.

Otro término importante para distinguir es el de **Computer Literacy**, que según la Association for Computing Machinery, Code.org, Computer Science Teachers Association, Cyber Innovation Center, y National Math and Science Initiative (2016, p. 13) se refiere al uso general de computadoras y programas, como software de productividad, ofimática, lo que incluye ser capaz de hacer búsquedas en internet y crear presentaciones digitales.

Yendo más allá, Jun, Han, Kim, y Lee (2014) describen los términos *ICT literacy* (educación en TICs) y *Computational Literacy* que no se discutirán en esta tesis.

¹<https://www.acm.org/>

² *Computer Literacy*

2.1.4. Problemas, Resolución de Problemas y Diagramas tipo Schoenfeld

Por «**Problema**» en este trabajo se entenderá como lo resume [Companioni Madeu \(2005\)](#) después de hacer una revisión de literatura al respecto: Un problema es «un reto provocado por una situación que no tiene solución en lo inmediato y que además de incertidumbre crea [la] necesidad de ser abordada» (p. 24). Y tiene las siguientes características:

1. El sujeto que recibe la información asume que efectivamente tiene ante sí una interrogante, y que en lo inmediato “no está capacitado para resolverla”. Este elemento es de suma importancia y de hecho constituye el punto de partida de un problema, pues la situación existe independientemente de su conciencia, y es sólo, en el momento en que el sujeto recibe la información, que determina asumirla o dejarla a un lado, lo que depende de las necesidades o de determinadas influencias externas, como puede ser su presencia en una clase de Matemáticas.
2. Seguidamente debe ocurrir algo no menos importante y es que el sujeto debe sentir que requiere de cierto “esfuerzo cognoscitivo”, encaminado a encontrar los medios para obtener la vía de la solución.
3. La persona que ya asumió estar ante un problema, siente la necesidad de resolverlo impulsado por el interés que le provoca el propio cuestionamiento o por una exigencia exterior.
4. Otro elemento que caracteriza a los problemas se relaciona con el carácter individual y el carácter relativo de estos, pues cada situación no es asumida como problema por todas las personas por igual, aún estando estas agrupadas por determinadas características semejantes, lo que depende de los conocimientos y experiencias precedentes que tienen los sujetos que interactúan con la situación.
5. Esta experiencia es la que determina en general la necesidad de asumir un problema, además de otras exigencias de orden externo que se dan fundamentalmente en el proceso de enseñanza y aprendizaje. ([Companioni Madeu, 2005](#), p. 24)

[Ferrer Vicente \(2000\)](#) resume qué es **Resolver un Problema** de la siguiente forma: «Resolver un problema es hacer lo que se hace cuando no se sabe qué hacer, pues si se sabe lo que hay que hacer [entonces] ya no hay problema» (p. 13).

En el libro de [Schoenfeld \(1985\)](#), citado en [Schoenfeld \(2016\)](#), se presentan un tipo de diagrama que muestra un gráfico de un intento de resolución de un problema por parte de una determinada persona. En el eje horizontal se representa el tiempo transcurrido, y en el eje vertical se presentan las diferentes etapas de trabajo cognitivo de la persona. Los diagramas de [Schoenfeld \(1985, 2016\)](#) reflejan el tiempo que una persona dedica a cierta actividad durante la resolución de un problema no familiar (es decir que no tiene solución inmediata). Las figuras 2.1 y 2.2 en la página siguiente muestran comportamientos típicos de un estudiante principiante y de un experto, respectivamente. Los triángulos invertidos en la figura 2.2 representan intervenciones verbales del resolutor.

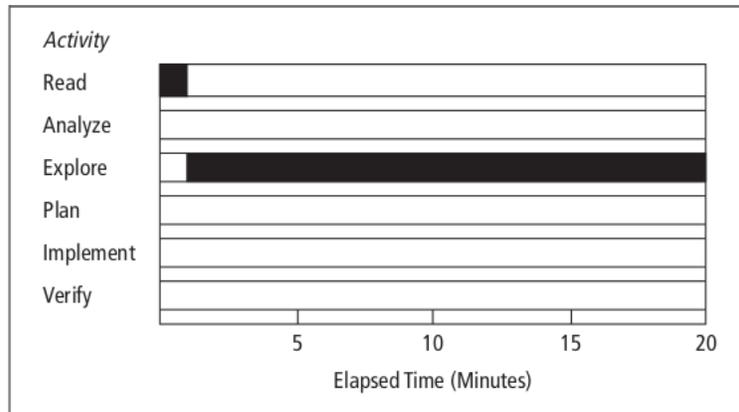


Figura 2.1.: Gráfico de línea de tiempo de un estudiante típico intentando resolver un problema no familiar.

Tomado de la figura 15.3 de Schoenfeld (2016, p. 24)

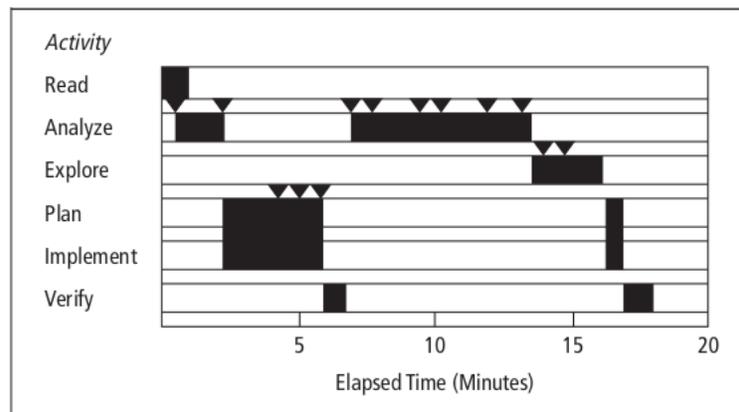


Figura 2.2.: Gráfico de línea de tiempo de un matemático trabajando un problema difícil. Tomado de la figura 15.4 de Schoenfeld (2016, p. 24).

El principiante lee el problema y luego procede a explorar el problema sin establecer una estrategia definida y sin replantearse su proceder hasta que se acaba el tiempo disponible. Schoenfeld (2016, p. 24) concluye que esta primera, rápida y equivocada decisión, si no se reconsidera y/o se revierte, garantiza el fracaso en la resolución del problema.

Schoenfeld (2016) explica que a diferencia del alumno, el matemático dedica más de la mitad del tiempo permitido a intentar darle sentido al problema. En lugar de comprometerse a sí mismo en alguna dirección particular, realiza una cantidad significativa de análisis y exploración estructurada, y no la implementa hasta que está seguro de que está trabajando en la dirección correcta. Monitoreando su solución con cuidado, persiguiendo pistas interesantes y abandonando rutas que no parecen dar frutos, logra solucionar el problema.

Más recientemente, Granberg (2016) presenta en su artículo un tipo diferente de diagramas, similares a los de Schoenfeld (1985), pero con más detalle sobre las etapas cognitivas

del proceso de resolución de problemas, y con las tareas cognitivas en orden inverso. La figura 2.3 muestra tres ejemplos prototípicos de resolución de problema utilizando estos diagramas: La línea A representa un ejercicio cuyo procedimiento es conocido. La línea B describe un proceso creativo de resolución de problema sin errores. La línea C describe una resolución creativa de problema que incluye dos errores.

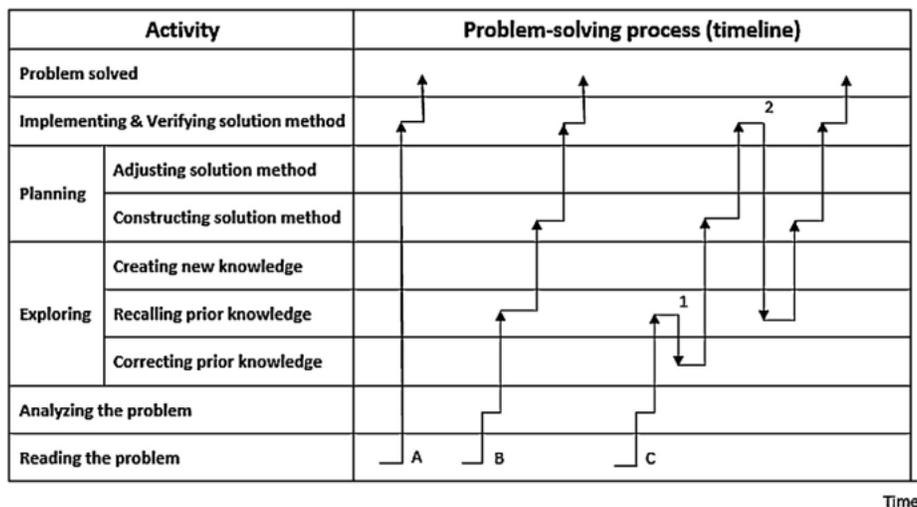


Figura 2.3.: Episodios clave de Shoefeld para la resolución de problemas. Tomado de la figura 1 de Granberg (2016, p. 35). Todos los derechos reservados 2016 por Elsevier Inc.

Así que a este tipo de diagramas cuyo eje horizontal es el tiempo, cuyo eje vertical son las tareas cognitivas, y que muestran la secuencia de cambios de tareas cognitivas de una persona mientras resuelve un problema, los llamaremos **Diagramas tipo Schoefeld**.

2.1.5. Rúbricas

Las **Rúbricas** principalmente se componen de indicadores y evidencias ponderadas a los que se asignan criterios (Cebrián de la Serna y Monedero Moya, 2014, p.83). Una rúbrica puede partir desde un diseño de una o varias tareas, como también desde la concepción de una o varias competencias. En cualquier caso suele tener por lo general un conjunto de elementos relacionados con un objetivo de aprendizaje. En la primera columna suelen situarse las categorías de la tarea o los diferentes indicadores de la competencia, desde la que se elige para cada uno de estos elementos el número de niveles de desempeño, los logros con un ranking de los criterios en los que pueden presentarse las evidencias, siendo éstas últimas las descripciones de las respuestas (las conductas, los productos, los pensamientos, los procesos, etc.) claros y concretos que puede mostrar un estudiante. (Cebrián de la Serna y Monedero Moya, 2014, p.84)

Se tiene evidencia científica de que la puntuación confiable de las evaluaciones de desempeño se puede mejorar mediante el uso de rúbricas (Jönsson y Svingby, 2007). Por ello

se utilizarán rúbricas analíticas (ver Mertler, 2000) para puntuar o ponderar las variables cuantitativas complejas de esta investigación.

Andrade (2005) presenta un análisis sobre las ventajas, las desventajas y los problemas de usar rúbricas.

2.1.6. Enfoque Sociocultural

Como parte de un programa de investigación en curso más extenso que se centra en el pensamiento computacional y su relación con el aprendizaje de la matemática, el enfoque que se utilizará en esta investigación es el **Enfoque Sociocultural**: «Desde el punto de vista del enfoque sociocultural los procesos psicológicos son concebidos como el resultado de la interacción mutua entre el individuo y la cultura.» (Martínez Rodríguez, 1999, p. 18)

«Un punto de vista situado/sociocultural considera el conocimiento y el aprendizaje no principalmente en términos de representaciones en la cabeza, aunque no hay necesidad de negar que tales representaciones existen y juegan un papel importante. Por el contrario, considera el conocimiento y el aprendizaje en términos de una relación entre un individuo con una mente y un cuerpo y un entorno en el que el individuo piensa, siente, actúa e interactúa.» (Gee, 2008, p.81)

2.2. La Habilidad Matemática de Algoritmizar

Como se mencionó previamente, el Sistema Básico de Habilidades Matemáticas de Hernández (1989) y su grupo de investigación (como se citó en Delgado Rubí, 1999, p. 64), está compuesto por las habilidades de Identificar, Interpretar, Calcular, Graficar, Re-codificar, *Algoritmizar*, Definir y Demostrar.

Para saber más sobre la historia del Sistema Básico de Habilidades Matemáticas, consúltese Vargas y cols. (2013).

La *algoritmización* es fundamental en la Teoría Cognitiva del Procesamiento de la Información debido a la analogía del programa de computadora (Gallagher, 1979) y particularmente en la Ciencia Cognitiva Computacional (Eysenck y Keane, 2010).

Rasmussen, Zandieh, King, y Teppo (2005) afirman que las actividades de Simbolizar (es decir, Re-codificar), *Algoritmizar* y Definir son ejemplos de prácticas sociales o culturales que representan un conjunto útil de prácticas centrales que abarcan todos los dominios matemáticos.

La Habilidad Matemática de *Algoritmizar* tiene diferentes facetas y puede estudiarse desde diferentes enfoques, según su relación con el aprendizaje de la matemática en general. Estas facetas se presentan en las siguientes sub-secciones.

2.2.1. Diferencias de género

El estudio del pensamiento matemático puede ser según el género, como en la investigación de Horne (2003), en la que se realiza un estudio longitudinal a gran escala que muestra que hay diferencias en las estrategias empleadas por niños y niñas para resolver problemas de adición y sustracción previo al aprendizaje de los algoritmos estándares escolares.

2.2.2. Etnomatemáticas

Micelli y Crespo (2012) describen los modos de aplicación de los ábacos de la América prehispánica, resaltando la manera en la que estas culturas aplicaban técnicas y algoritmos relacionados con los sistemas de numeración utilizados.

El trabajo realizado por Lipka y cols. (2015) en el que se identifica un enfoque para la enseñanza de los números racionales por medio de los procedimientos ancestrales de los esquimales Yuk'ip es un ejemplo de la valoración que le debe dar a los estilos no occidentales de hacer matemática, y en este caso, de expresar y transmitir algoritmos.

Vásquez Hernández y Moreno Gatica (2016) hablan sobre nudos y patrones en Guatemala y Costa Rica, como los *Quipus* de Sudamérica. Exponen cómo estas confecciones artesanales, para las que existen técnicas (algoritmos) para ser construidos, forman parte de la salvaguardia del patrimonio cultural intangible latinoamericano.

2.2.3. Resolución de problemas y demostraciones matemáticas

Lovász (2008, p. 3) enfatiza la necesidad de preservar la resolución de problemas como la principal característica de la enseñanza de la matemática. Además, Sfard (1991) nos dice que los procesos de aprendizaje y resolución de problemas consisten en una intrincada interacción entre las concepciones operacionales (algoritmos) y estructurales (objetos) de las mismas nociones.

Hay investigaciones orientadas a comprender los procesos de resolución de problemas que se basan en la algoritmización. Esto debido a que «resolver» como Habilidad General Matemática no básica *puede* consistir de construir o encontrar un algoritmo para resolver algún problema, es decir, «algoritmizar» (Delgado Rubí, 1999, p. 72). A continuación se presentan algunas de esas investigaciones.

La presentación de Klein y Hoffmann (2004) muestra que el uso de algoritmos implementados en hoja de cálculo provee múltiples beneficios en la resolución de problemas cuya solución requiere recursión, ya que ayuda a predecir cómo continúan los patrones, ayuda a organizar la información sistemáticamente, y a desarrollar generalizaciones.

Modeste (2012b) nos dice que en matemáticas, la actividad algorítmica se puede definir como un paso en la resolución de una determinada categoría de problemas en los que se busca un método de resolución sistemático y eficaz.

Hay experiencias como la de Weber (2003) en la que se describe cómo estudiantes de grado pueden desarrollar su entendimiento del concepto de demostración viendo el acto de demostrar por inducción como una prescripción mecánica paso a paso, primero como *algoritmo*, luego como *proceso* y finalmente como *argumento*. Como continuación está la investigación de Weber (2004) en la que se ofrece un marco de referencia para categorizar y describir diferentes tipos de procesos que los alumnos usan para construir demostraciones en general.

Para Futschek (2006, p.160), un algoritmo es «un método para resolver problemas que consiste de instrucciones exactamente definidas». Esto coincide con la noción de algoritmo de Pyzara (2012) en su relación con la resolución de problemas.

Modeste (2012a) realiza un análisis epistemológico del concepto de *Algoritmo* ligado a las Ciencias de la Computación pero también a las matemáticas y particularmente a la demostración, para estudiar su transposición y construir situaciones didácticas en el contexto del currículum de secundaria de Francia. En esa tesis doctoral se subraya que los algoritmos y las demostraciones están enlazadas de diferentes formas. Lo primero que hay que apuntar es que un algoritmo es un método efectivo para resolver un problema general.

La definición de *Algoritmo* asumida en esa investigación refleja esta importante relación entre los algoritmos y la resolución de problemas: «Un algoritmo es un procedimiento para resolver problemas, que puede ser aplicado a un conjunto de instancias del problema y que produce, en un número finito de pasos constructivos, claros y organizados, la respuesta al problema para alguna de sus instancias.»(Modeste, 2012a)

Modeste (2016) nos recuerda el resultado teórico llamado Correspondencia de Curry-Howard, que muestra que cualquier demostración puede ser vista como un algoritmo y cualquier algoritmo como una demostración. «Por lo tanto, la resolución algorítmica de problemas incluye una dimensión importante de prueba y se han desarrollado muchas herramientas y teorías para probar algoritmos y estudiar sus propiedades. Por otro lado, cualquier demostración matemática constructiva implica un algoritmo implícito que puede hacerse explícito. En particular, es el caso de cualquier prueba por inducción, donde se puede detallar un proceso de construcción subyacente.» (Modeste, 2016, p. 247)

2.2.4. Desarrollo ontogenético de las operaciones aritméticas

Boufi y Skaftourou (2004) presentan una interesante experiencia sobre intentar la transformación del entendimiento de los algoritmos de multiplicación y división en niños de quinto grado de primaria, de un aprendizaje memorístico del algoritmo (que es lo normal) hacia una construcción del algoritmo basado en la indagación de los mismos alumnos, es decir que los mismos alumnos construyan su propio algoritmo. Este es el tipo de enfoque instruccional que Pyzara (2012) sostiene que produce múltiples beneficios en los estudiantes.

En la misma línea, Bulgar (2003) presenta las estrategias que un grupo de niños *inventó* para resolver problemas de divisiones de fracciones en ausencia de instrucción algorítmica previa.

Heirdsfield (2003) presenta un análisis entre dos grupos de alumnos que resuelven un mismo conjunto de problemas numéricos. En un grupo los alumnos ejecutan algoritmos rígidos aprendidos frente al otro grupo de alumnos que ejecutan cómputo mental «flexible», es decir que utilizan diferentes estrategias dependiendo de cada caso. En general el resultado es que los alumnos que construyen sus propios algoritmos eligen estrategias de cálculo numérico más eficientes.

La investigación conducida por Kieran y Guzmán (2003) muestra que se puede propiciar que los alumnos desarrollen conceptos teóricos sobre los números al implementar actividades con calculadora que realicen sucesivas ejecuciones del algoritmo de división de Euclides para determinar divisibilidad entre números enteros.

En medio oriente, Barash y Klein (1996) aplican un marco teórico para analizar el conocimiento de los números racionales de niños de séptimo grado estudiando la imagen exhaustiva de los algoritmos implementados por los niños. En el mismo sentido, las investigaciones iniciadas de forma independiente por Gallardo Romero (2004) y Castaño-García (2008) buscan la comprensión del conocimiento matemático de niños a partir de su ejecución de los algoritmos para multiplicar números.

Por otro lado, Anghileri (2001) se cuestiona el propósito de enseñar en el siglo XXI los algoritmos estándares clásicos en lugar de propiciar la construcción personalizada de estrategias de solución más efectivas por parte de los alumnos. Estas estrategias son más efectivas porque usan el sentido que cada alumno tiene de los números. En la misma línea de ideas Gravemeijer y cols. (2017) enfatizan en la importancia del desarrollo de competencias matemáticas que complementen el trabajo de las computadoras en lugar de enfatizar en el desarrollo de competencias matemáticas que compiten con lo que las computadoras pueden hacer.

2.2.5. Aprendizaje de la Matemática por medio de Calculadoras avanzadas y Computadoras

En aquellos contextos socio-económicos donde las personas se lo pueden permitir, se han utilizado computadoras para la enseñanza y el aprendizaje de la matemática desde que ha sido materialmente posible.

Schoenfeld (2013, p. 31) nos dice que «la presencia de herramientas computacionales [...] tiene el potencial de cambiar la forma del conocimiento al que los estudiantes tienen acceso en las aulas de clase de matemática, y las formas en las que ellos pueden operar con ese conocimiento.»

Hoffmann (2001) presenta una experiencia en sexto grado de primaria utilizando la construcción de una simulación tipo *Monte Carlo* para la introducción del concepto de área de un círculo unitario (que equivale también a la aproximación del número π). En esta experiencia los alumnos crearon el algoritmo y lo implementaron en hoja de cálculo.

Serhan (2006) muestra que el uso de la calculadora graficadora puede ayudar a entender conceptos matemáticos complejos, así como Büyükköroglu y cols. (2006) comprueban los beneficios de usar computadoras para asimilar conceptos de Cálculo.

Ng y Hu (2006) investigan el impacto de hacer que los alumnos de noveno grado experimenten con simulaciones web y participen en discusiones en línea, para aprender a bosquejar transformaciones de curvas trigonométricas.

Más recientemente Dos Santos y Batista Trocado (2016) argumenta cómo con sus poderosas capacidades, el software GeoGebra se ha transformado en un verdadero Ambiente Matemático de Aprendizaje, lleno de funcionalidades colaborativas entre estudiantes y profesores como los GeoGebraBooks³ y los GeoGebraGroups.

Rojas Ayala (2018) desarrolla una propuesta de programa basado en software de juegos matemáticos que mejora las capacidades matemáticas en las operaciones básicas en estudiantes de primer año de carreras de ingeniería.

De manera similar, Congacha Aushay, Santillán Castillo, Guerra Salazar, y Barba Vera (2018) realizaron una investigación que tiene por objetivo observar el impacto que produce el uso de un software educativo con estrategias didácticas dentro del campo de la aptitud académica concerniente al razonamiento numérico, verbal y abstracto en la educación, de 175 estudiantes que cursan el último año de educación media. Como resultado de la investigación, usando pre-prueba y post-prueba, se observó que 155 estudiantes incrementaron su puntuación luego de la preparación con el software, lo cual representó un crecimiento del 94,12% en el puntaje obtenido, lo que permite concluir que las tecnologías informáticas son herramientas de apoyo para el desarrollo de aptitudes académicas.

Sin embargo, el uso de calculadora o computadora no debe confundirse con la *algoritmización*. Si bien estas máquinas permiten la ejecución de algoritmos (que pueden ser muy extensos y complicados) muy rápidamente, su uso no guarda ninguna relación con el entendimiento de los algoritmos que ejecuta. Por ejemplo, el estudio de casos realizado por Hershkowitz y Kieran (2001) y la consecuente reacción de Sutherland (2001) ilustran cómo el uso de una calculadora graficadora puede de hecho atenuar la habilidad de resolver problemas cuando esta se usa sin una estrategia clara de resolución y sin el entendimiento de los algoritmos que esta implementa. Lo mismo puede decirse del uso de la computadora para ejecutar los algoritmos: Que el uso de la computadora en cursos introductorios de programación de computadoras no garantiza que los alumnos aprendan a escribir algoritmos. Esto está respaldado en el informe de Association for Computing Machinery y cols. (2016).

2.2.6. Aprendizaje de la Matemática por medio de Programación de Computadoras

Yendo más allá del simple uso de computadoras para enseñar y aprender matemática, hay una gran variedad de investigaciones sobre el uso de la Programación de Computadoras, y por consiguiente de la *Algoritmización*, para contribuir con la enseñanza y el aprendizaje de la matemática.

Los intentos serios de utilizar la programación de computadoras en la enseñanza de las matemáticas en la escuela primaria y secundaria comenzaron con Seymour Papert (1980).

³https://wiki.geogebra.org/es/Creando_un_Libro_GeoGebra

La idea de Papert era: crear un micromundo interactivo al que los niños accedan a través de las matemáticas, lo que los impulsa a pensar matemáticamente al incrustar pepitas de conocimiento matemático en el micromundo con el que los alumnos se topan jugando mientras desarrollan pequeños proyectos.

«Como una ciencia de objetos abstractos, la matemática se basa en la lógica en lugar de la observación como su medida de verdad, [pero] también emplea la observación, la *simulación*, e incluso la *experimentación* como medio para descubrir la verdad» (National Research Council, 1989, p. 31). De ahí que la programación de computadoras sea un recurso útil para el aprendizaje de la matemática.

Bubnó y Takács (2014) describen muchas analogías entre la resolución de problemas matemáticos de texto⁴ y de problemas de programación de computadoras. En ambos casos hay que comprender el problema, trabajar con variables, construir un modelo, diseñar un plan, ejecutar el plan, realizar pruebas, contestar la pregunta inicial, etc. También identifican una diferencia fundamental: la notación y las reglas sintácticas. Entonces, para poder reducir al mínimo la interferencia de estas diferencias, y quedarse sólo con la algoritmización, utilizan el lenguaje Blockly⁵ en su investigación, en la que buscan ayudar a los niños a resolver problemas matemáticos de texto por medio de la algoritmización con dicho lenguaje.

Misfeldt y Ejsing-Duun (2015) investigan el potencial de aprendizaje matemático en actividades de programación de computadoras. Ellos describen dos situaciones educativas en aula de clase usando las actividades de programación de niños de quinto grado para aprender matemática. Estas actividades pueden ser vistas genuinamente con potencial de aprendizaje matemático en el sentido que involucran mediación epistémica hacia conceptos matemáticos.

Está la experiencia de DeJarnette (2016) en la que los estudiantes utilizan el entorno de programación Scratch para comprender cómo las distancias recorridas por ciertos objetos son función del tiempo. DeJarnette (2016) estudia la reinención de las representaciones del concepto «distancia» por medio del uso de un entorno de programación visual.

Navas-López (2017) presenta una propuesta artístico-didáctica para mejorar la imagen de concepto de los Conjuntos de Julia⁶ por medio de programación de computadoras, usando un modelo algorítmico-computacional cuyo enfoque integra la dimensión computacional, la dimensión matemática, la dimensión lúdica y la dimensión artística.

Baranauskas y Carbajal (2017) realizaron una investigación situada socialmente en la que argumentan que la inclusión de un entorno de programación en un contexto educacional no debería ser desconectada del significado que la gente involucrada (niños, profesores, administradores, padres) tienen de él. También indican que tal inclusión debe realizarse apoyando a las partes interesadas, aquellas personas afectadas directa o indirectamente por los correspondientes proyectos de inclusión.

Redón y Serrano Aroca (2018) reporta una experiencia positiva de aprendizaje de la matemática a través del lenguaje R, particularmente en el tema de resolución de ecuaciones

⁴ *word problems*, es decir, aquellos problemas cuyo enunciado es completamente un bloque de texto.

⁵ <https://blockly.games/?lang=es>

⁶ https://es.wikipedia.org/wiki/Conjunto_de_Julia

polinómicas en educación secundaria.

DeJarnette (2018) realiza una investigación en la que los estudiantes usan un entorno de programación visual y mejoran sus razonamientos sobre las funciones trigonométricas.

La investigación de Kilhamn y Bråting (2019) muestra cómo las actividades de programación de computadoras pueden interactuar en diferentes formas con el desarrollo del pensamiento algebraico en los estudiantes. Entre otras interacciones, muestra que el aprendizaje del álgebra puede presentar dificultades cuando se realiza después de actividades introductorias a la programación, principalmente debido a la diferente simbología. Esto no indica que sea especialmente una mala idea aprender a programar antes de aprender álgebra, ya que Modeste (2012b, 2016) describe que lo opuesto también ocurre. Simplemente tienen diferentes notaciones y reglas sintácticas, como describen Bubnó y Takács (2014), y por lo tanto generan interferencias la una a la otra.

Los análisis cualitativos de Broley (2015a) y Broley y cols. (2018) nos revelan algunas de las formas en las que los profesores e investigadores matemáticos usan la programación de computadoras para la resolución de problemas tanto de matemática pura como de matemática aplicada. Con la programación de computadoras logran una variedad de tareas como: (a) modelar fenómenos complejos para resolver problemas del mundo real, (b) realizar cálculos y simulaciones imposibles de hacer a mano para realizar «matemática experimental», (c) construir demostraciones, (d) desarrollar herramientas tecnológicas, etc.

La forma en que los profesores universitarios utilizan la algoritmización o la programación de computadoras para la enseñanza es muy diversa. Pero la investigación de Broley (2015b) (como se citó en Broley, 2016) y Broley y cols. (2018) permite identificar seis tipos o niveles de interacción que los profesores ofrecen a sus estudiantes:

1. Estricta observación de los resultados de la ejecución de un programa de computadora realizada generalmente por parte del profesor.
2. Manipulación (ex-aula) de una interface de un programa existente.
3. Observación (y tal vez interpretación) del código fuente de un programa.
4. Modificación de un código fuente existente para lograr algo nuevo o diferente.
5. Construcción de código fuente, con características especificadas por el profesor (incluso el algoritmo mismo).
6. Realización completa de creación de programas de computadora por parte de los alumnos, incluyendo el desarrollo de los algoritmos, codificación y validación.

2.2.7. Modelamiento matemático

El artículo de Pyzara (2014) expone un experimento en el que se pide a los estudiantes que construyan un modelo para un problema matemático cotidiano. A la mitad de ellos les pide que construyan un modelo tipo *Esquema* y a la otra mitad que construya un modelo tipo *Algoritmo*. El modelo natural para el problema planteado es un algoritmo y muchos

alumnos que participan en el estudio fracasan en el modelamiento correcto del problema precisamente porque evaden (por falta de comprensión o por falta de formación elemental) la construcción de un algoritmo. Aquellos alumnos que tienen formación elemental previa en la representación gráfica de algoritmos (diagramas de flujo o pseudo-código) y a los que además se les pidió explícitamente que hicieran un algoritmo para el problema son los que obtuvieron mejores resultados.

Yurkovych, Seben, y Mar'yan (2017) reportan una sistemática mejora en las notas obtenidas por sus estudiantes de Enseñanza de la Física al utilizar un enfoque de modelamiento fractal para la enseñanza de ciertos conceptos físicos.

Gravemeijer y cols. (2017) discuten sobre lo que está involucrado con hacer matemáticas en el mundo real. Dicen que están involucrados los siguientes pasos: (1) Reconocer dónde las matemáticas son aplicables, (2) Traducir el problema real en problema matemático [Modelamiento Matemático propiamente], (3) Resolver el problema matemático, y (4) Interpretar y evaluar el resultado.. Ver figura 2.4.

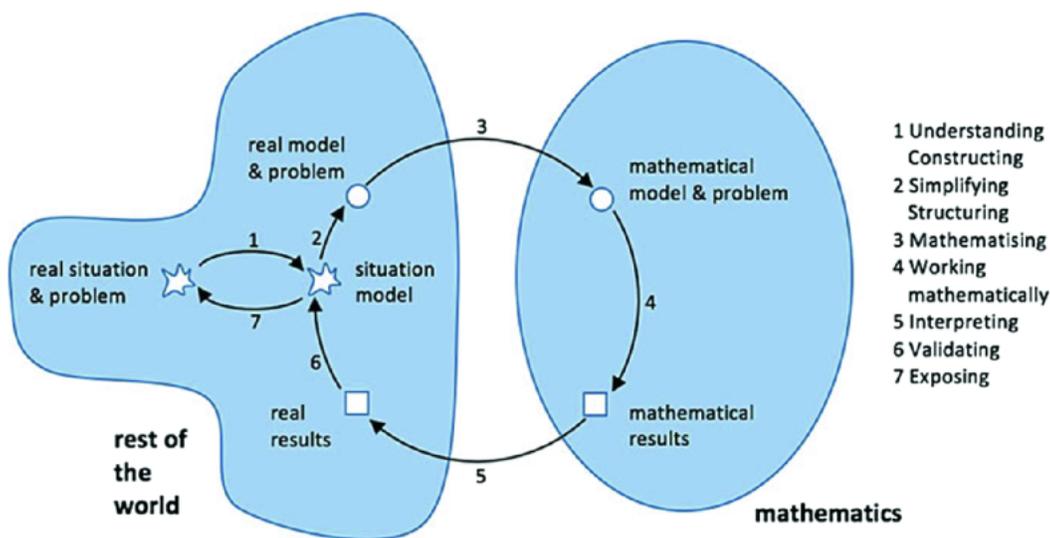


Figura 2.4.: Ciclo de modelamiento clásico

Tomado de la figura 4.5 de Vorhölter, Greefrath, Ferri, Leiß, y Schukajlow (2019)

Y tal como se había dicho antes, la habilidad matemática de «resolver» un problema *puede* consistir de construir o encontrar un algoritmo para resolver algún problema (Delgado Rubí, 1999, p. 72), es decir, «algoritmizar». Como por ejemplo en el problema de las Torres de Hanoi, en el que el modelo de solución es un algoritmo.

Esta algoritmización puede incluso requerir que la ejecución no sea por parte de humanos, sino que requiera de tecnología informática concreta. Ver figura 2.5 en la página siguiente

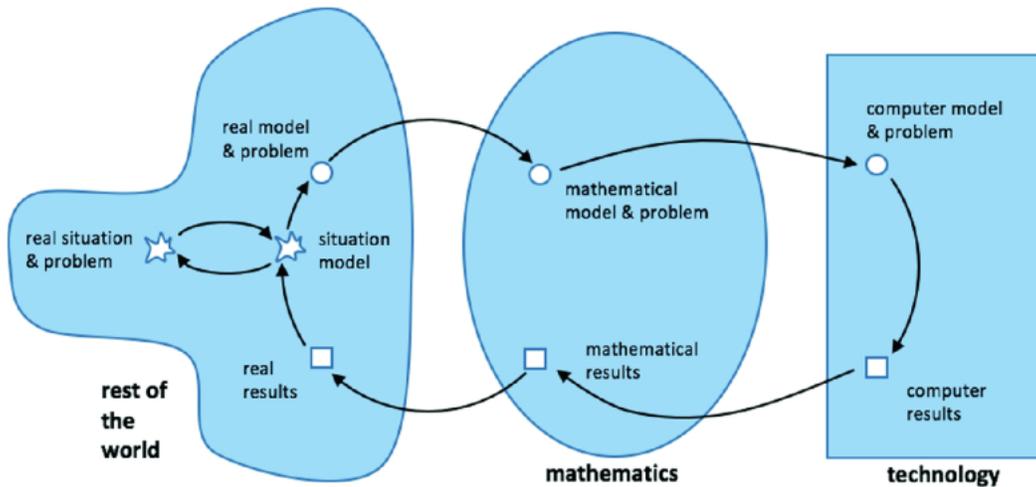


Figura 2.5.: Ciclo de modelamiento extendido usando tecnología
Tomado de la figura 4.7 de Vorhölter y cols. (2019)

A la inversa, Grozdev y Terzieva (2015) nos dicen que crear un algoritmo es construir un modelo, y se trata de un modelo matemático porque los algoritmos son objetos matemáticos (Lovász, 1988).

Futschek (2006) incluso nos dice explícitamente que para mejorar o modificar algoritmos es necesario tener modelos abstractos y definiciones precisas de las propiedades. De nuevo, aparece la dualidad algoritmo-concepto de Pyzara (2012); Sfard (1991).

En resumen, el Modelamiento Matemático puede requerir construcción y ejecución de algoritmos, así como también la manipulación de algoritmos requiere de modelos matemáticos.

2.2.8. Desarrollo de habilidades profesionales

Lockwood y cols. (2016) expone cómo la noción de *Pensamiento Algorítmico* tiene presencia importante entre las cosas que los matemáticos profesionales valoran en la práctica de hacer matemática, y tal como encontraron Buteau y cols. (2014) en su investigación, el 43% de los matemáticos que encuestaron reportó usar programación de computadoras en sus investigaciones, es decir que escribir algoritmos forma parte de las habilidades profesionales de casi la mitad de los matemáticos.

La algoritmización no sólo es de interés en la matemática (y en la educación matemática) sino en áreas profesionales específicas como por ejemplo la programación de PLC's (Programmable Logic Controllers) por medio del modelamiento de problemas con Automatas Finitos, Máquinas de Moore y Máquinas de Mealy (Shalyto, 2001). Y en general es de interés en muchas áreas de la ingeniería (Porto Solano y cols., 2017).

Más aún, de entre las habilidades consideradas necesarias para los profesionales en Tecnologías de la Información, destaca explícitamente la habilidad de Algoritmizar Tareas (Shefer y cols., 2018).

En las profesiones asociadas a la enseñanza también es valiosa la habilidad de algoritmizar. Borovik (2011) nos presenta una extensa reflexión sobre la relación de los matemáticos y profesores de matemática con las Tecnologías de la Información, y cómo han sido principales pioneros de su uso didáctico-pedagógico.

Desde hace muchas décadas Knuth (1974) estaba convencido del valor pedagógico de un enfoque algorítmico en la enseñanza ya que este ayuda a entender conceptos de todo tipo. «Se suele decir que una persona no entiende algo realmente hasta que se lo enseña a alguien más. Realmente una persona no entiende algo realmente hasta que puede enseñárselo a una computadora, lo que se hace expresándolo como un algoritmo.» (Knuth, 1974, p. 327)

«Además, [...], deberíamos [los profesores] poner más énfasis a algunas actividades matemáticas no tradicionales como el diseño de algoritmos [es decir, algoritmizar], modelamiento [matemático], experimentación y exposición.» (Lovász, 2008, p. 3)

Leonard, Buss, Unertl, y Mitchell (2016) indican que se esperan en el futuro cercano un aumento en la demanda de personal formado en áreas CTIM (Ciencia, Tecnología, Ingeniería y Matemática) [*STEM - Science, Technology, Engineering, and Mathematics*] y realizaron una investigación para evaluar y mejorar el *pensamiento computacional* de los niños (junto con los constructos de auto-eficacia en tecnología y actitud hacia la ingeniería y tecnología) de un programa de verano especializado en tecnología, usando y programando equipos robóticos.

En El Salvador, Joma (2020) reporta gran potencial en el desarrollo de habilidades relacionadas a la robótica, cuando un grupo de niñas ganó un concurso regional de robótica educativa.

En palabras de Tall (2009), «Puede que ya no necesitemos preparar a los niños a usar rutinas matemáticas regulares como una habilidad central de su futuro empleo, sino que ellos necesitarán crecer en una forma que los habilite a sobrevivir en un nuevo mundo tecnológico».

Hay muchos investigadores que reflexionan acerca de la importancia de las competencias (o habilidades) relacionadas con las TICs para el desarrollo profesional de las nuevas generaciones. Por ejemplo, D'Ambrosio (1999) propone desechar los viejos sistemas curriculares por un nuevo *Trivium*, un nuevo tipo de currículum al estilo del *Trivium* (gramática, dialéctica, retórica) y *Quadrivium* (aritmética, geometría, astronomía, música) de la edad media para la educación de las nuevas generaciones de ciudadanos del mundo. Tal como se mencionó antes, este nuevo currículum está compuesto por las competencias que él llama *Literacy*, *Matheracy* y *Technoracy*. En esta concepción, *Literacy* engloba la habilidad de entender e interpretar la información, gráficos, tablas, códigos, etc., mucho más allá de simplemente saber leer y escribir; *Matheracy*, que engloba la habilidad de llegar a conclusiones a partir de datos y cálculos, independientemente de saber hacer manualmente tales cálculos; y *Technoracy*, que engloba la habilidad de entender las ideas, posibilidades y peligros de los dispositivos tecnológicos así como los asuntos morales subyacentes de su uso, lo que requiere evidentemente no sólo la habilidad de saber usar la tecnología disponible con cierta fluidez.

2.2.9. Educación inclusiva

El estudio de la algoritmización, que en general se puede usar para comprender la cognición de los estudiantes, también se puede usar en casos de estudio para comprender (y eventualmente ayudar) individuos específicos. Un ejemplo está en la investigación de DeWindt-King y Goldin (2001) en la que investigan las representaciones visuales e imaginativas internas de dos niños, exploran sus imágenes dinámicas, imágenes pictóricas, patrones, etc. por medio del análisis de los algoritmos que utilizan para resolver problemas.

De forma parecida, Ruiz Ledesma y Valdemoros Álvarez (2004), realizan un estudio cualitativo cuyo propósito inicial era evaluar una propuesta de enseñanza en el tema de razones y proporciones en sexto grado de primaria pero que derivó en el análisis de una de las alumnas, Paulina, que implementaba algoritmos que no tenían sentido, y posteriormente a la intervención se logró el enriquecimiento de la comprensión de la niña y mejoró su manejo de los algoritmos relativos a problemas de razones y proporciones.

En otra perspectiva, la investigación de Rowland, Huckstep, y Thwaites (2004) analiza la práctica docente de una profesora de quinto grado, Laura, y entre otras cosas describe su conocimiento de la materia (Subject Matters Knowledge, SMK) y su conocimiento del contenido pedagógico (Pedagogical Content Knowledge, PCK) por medio del análisis de sus clases y de los algoritmos enseñados por ella (multiplicaciones simples de enteros) y la ejecución de dichos algoritmos por parte de sus alumnos.

Con estos tres breves ejemplos se puede concluir que el estudio de la algoritmización puede ayudar en la inclusión educativa de estudiantes con diferentes tipos de problemas de aprendizaje.

2.2.10. Política educativa

No es casualidad que en los inicios del siglo XXI en muchos países el estudio y aprendizaje de los algoritmos matemáticos estándares en la educación primaria sea parte explícita de los contenidos oficiales a evaluar, como en Estados Unidos de América (Romberg, 2001), o incluso como en Alemania en que uno de los objetivos de la educación matemática para la educación general es «Proveer dominio de las técnicas, algoritmos, y conceptos necesarios para la vida cotidiana en la sociedad» (Kaiser, 2001).

En el caso local, el enfoque oficial del Ministerio de Educación, Ciencia y Tecnología de El Salvador (MINEDUCyT) para la asignatura Matemática en educación primaria es la resolución de problemas: «En términos de enseñanza, el docente debe generar situaciones en las que los estudiantes exploren, apliquen, argumenten y analicen los conceptos, procedimientos, algoritmos u otros tópicos matemáticos acerca de los cuales deben aprender» (2008; 2018). Y sobre la competencia del Razonamiento Lógico-Matemático a desarrollar el Ministerio de Educación, Ciencia y Tecnología de El Salvador (MINEDUCyT) dice que «Esta competencia promueve que los estudiantes identifiquen, nombren, interpreten información, comprendan procedimientos, utilicen algoritmos y relacionen conceptos.» (2008; 2018)

Más allá de los programas concretos de cada país, el National Council of Teachers of Mathematics (NCTM), como institución de influencia internacional en la enseñanza de la matemática, plantea entre sus estándares que los currículos de matemática «enfaticen el entendimiento conceptual, representaciones múltiples y conexiones múltiples, *modelación matemática*, y *resolución de problemas*» (1989, p. 125). Y como hemos mencionado antes, «el proceso de modelamiento matemático» y «resolución de problemas» en matemática puede consistir de construir o encontrar un algoritmo para resolver algún problema (Delgado Rubí, 1999; Futschek, 2006).

2.3. Pensamiento Matemático y Pensamiento Algorítmico

2.3.1. Pensamiento Matemático

Existen diferentes modos de pensamiento. «Todos los tipos de pensamiento son aspectos individuales de habilidades cognitivas de las personas que aparecen en diferentes esferas de actividades mentales y actividades basadas en objetos» (Sadykova y Usolzev, 2018, p. 1).

Los siguientes pueden considerarse modos de *Pensamiento Matemático* aunque no son exclusivos de la Matemática:

- Pensamiento Cuantitativo (ver Ruiz Ledesma y Valdemoros Álvarez, 2004)
- Pensamiento Lógico (ver Coscarelli y cols., 1976; y Voskoglou y Buckley, 2012)
- Pensamiento Crítico (ver Voskoglou y Buckley, 2012; Kalelioğlu y Gülbahar, 2014; y Doleck, Bazelais, Lemay, Saxena, y Basnet, 2017)
- Pensamiento Creativo (ver Doleck y cols., 2017; y Hidayat, Susilaningsih, y Kurniawan, 2018)
- Pensamiento Numérico y Aritmético (ver Gallardo Romero, 2004; Gallardo Romero y González, 2007; y Congacha Aushay y cols., 2018)
- Pensamiento Algebraico (ver Kilhamn y Bråting, 2019)
- Pensamiento Geométrico (ver Thom y McGarvey, 2015)
- Pensamiento Combinatorio (ver Fernández Millán, 2013)
- Pensamiento Matemático Avanzado (ver Tall, 2002; Dubinsky y Tall, 2002; y Rasmussen y cols., 2005)
- Pensamiento Variacional (ver Martínez-López y Gualdrón-Pinto, 2018; y Ordóñez-Ortega, Gualdrón-Pinto, y Amaya-Franky, 2019)
- Pensamiento Estadístico (ver Batanero Bernabeu, 2013)
- Pensamiento Algorítmico (ver Futschek, 2006; Grozdev y Terzieva, 2015; Lockwood y cols., 2016; Sadykova y Usolzev, 2018; y Sadykova y Il'bahtin, 2019)

Muchos de estos modos o tipos de pensamiento están asociados entre sí en diferentes áreas de la matemática y en otros contextos.

2.3.2. Orígenes del término Pensamiento Algorítmico y su relación con la Matemática

Como se mencionó en la Introducción (en la página 1), ya en los años 70s se reflexionaba sobre que la Programación [de Computadoras] es una actividad de naturaleza matemática. Dijkstra (1974) sostiene que la mayor parte del trabajo matemático tiene las siguientes tres características y que la Programación [de Computadoras] las cumple: (1) Comparado con otros campos de actividad intelectual, las afirmaciones matemáticas tienden a ser inusualmente precisas, (2) Las afirmaciones matemáticas tienden a ser generales en el sentido de ser aplicables a una gran (frecuentemente infinita) cantidad de instancias, (3) Las matemáticas encarnan una disciplina de razonamiento que permite que tales afirmaciones tengan un inusualmente alto grado de confianza.

Lovász (1988) escribió un extenso ensayo en el que presenta su punto de vista sobre la relación entre el lado algorítmico y el lado estructural de la matemática. Aquí aparece una vez más la dualidad algoritmo-concepto en la matemática que Sfard (1991) y Pyzara (2012) describen.

Modeste (2016) también está de acuerdo con que las matemáticas y las ciencias de la computación tienen fuertes nexos y una historia común. Además, (1) comparten fundamentos comunes, estructurados por la lógica, y una relación específica con la demostración formal, (2) hay una cierta continuidad entre ellas con muchos campos desarrollándose en su intermedio, (3) la matemática asistida por computadora cambia la forma en la que algunos matemáticos trabajan, enfatizando la dimensión experimental de las matemáticas⁷, y (4) tienen ambas una relación muy similar con respecto a otras ciencias a través del modelamiento y la simulación.

Knuth (1974) reflexiona sobre la relación entre la Matemática y las, por entonces todavía jóvenes, Ciencias de la Computación. Menciona algunos de los múltiples aportes que se han hecho mutuamente, entre computólogos (*computer scientists*) haciendo matemática y matemáticos haciendo ciencias de la computación, a veces sin darse cuenta. Knuth (1974) reconoce la profunda influencia de la matemática en las Ciencias de la Computación pero también menciona que gracias a esta, aquella se ha enriquecido con nuevas teorías, como la Teoría de Lenguajes, que no existirían de no ser por la disponibilidad de computadoras modernas; y otras que se han potenciado gracias a las computadoras, como la Geometría Fractal, la Investigación de Operaciones y la Teoría de Grafos.

A este respecto, Modeste (2016) considera que es claro que los nuevos objetos matemáticos desarrollados en el intermedio de las matemáticas y las ciencias de la computación en combinatoria, matemática discreta, algorítmica, investigación operativa, criptografía, etc., deben ser analizados epistemológicamente con precisión.

⁷Véase por ejemplo el libro de Stavely (2014), que es un curso de Matemática Discreta utilizando (y aprendiendo) el lenguaje de Programación Python.

Esta creación de nuevas teorías matemáticas (o al menos teoremas nuevos) y enriquecimiento de otras gracias a las computadoras no se ha detenido. Tenemos por ejemplo los descubrimientos recientes de Chmielewski, Janowicz, Gawdzik, y Orłowski (2017) y de Orłowski y Chmielewski (2018) sobre ciertos patrones de números primos que aparecen al analizar numéricamente los resultados gráficos del algoritmo de construcción de la Espiral de Ulam. Estas dos publicaciones no incluyen teoremas, sino conjeturas a partir de los patrones identificados. Los teoremas correspondientes están pendientes de ser demostrados, pero estos patrones jamás habrían sido encontrados de no ser porque hay computólogos (*computer scientists*) especializados en Graficación por Computadora que utilizan computadoras para realizar grandes cantidades de cálculos numéricos.

Otro ejemplo es presentado por d’Azevedo Breda y Dos Santos (2018) que busca una forma sistemática de generar teselaciones esféricas y busca nuevas teselaciones usando el software GeoGebra. De las herramientas que provee el software, emergió la construcción de un algoritmo para obtener la órbita de un conjunto de puntos esféricos bajo la acción de un (sub)grupo de isometrías esféricas. Estos resultados condujeron a dos nuevos teoremas, a partir de las conjeturas desarrolladas gracias al software.

Breda y Dos Santos (2019) creen que GeoGebra puede hacer una contribución sustancial a la descripción y construcción de teselaciones esféricas que no han sido exploradas todavía, además de ser un recurso de gran utilidad para el estudio de la geometría esférica en general.

El mismo Knuth (1985) reflexiona sobre la relación de lo que él llama (sin definirlo explícitamente) el *Pensamiento Algorítmico* y el *Pensamiento Matemático* partiendo de la pregunta ¿Cuál es el rol actual [en 1985] de la noción de un *Algoritmo* en las Ciencias Matemáticas? En el resto del artículo continúa con reflexiones sobre cuál es la relación de los algoritmos en la Matemática, y que si la mayoría de los matemáticos tienen un proceso de pensamiento esencialmente diferente al de la mayoría de los computólogos (*computer scientists*).

Modeste (2012b) nos dice que dado que la actividad matemática se centra en la resolución de problemas, el *Pensamiento Algorítmico*, como el *Pensamiento Matemático*, entre otros, sería un enfoque particular para la resolución de algunos tipos de problemas matemáticos.

Como se citó en Knuth (1985), Aleksandrov, Kolmogorov, y Lavrent’ev (1963) listan las siguientes características distintivas de las ciencias matemáticas: (a) Abstracción, con varios niveles de abstracción; (b) Precisión y rigor lógico; (c) Relaciones cuantitativas; y (d) Amplio rango de aplicaciones. Knuth (1985) está convencido de que estas características también son distintivas de las Ciencias de la Computación, por lo que considera necesario un examen más profundo para dilucidar las diferencias entre *Pensamiento Algorítmico* y *Pensamiento Matemático*.

Después de hacer un análisis del contenido de libros de matemáticas que él usó cuando estudiante, Knuth (1985) concluye que no hay un sólo tipo de *Pensamiento Matemático*, sino que los matemáticos usan una variedad de modos de pensamiento⁸ y que algunos

⁸ver la sección 2.3.1

de estos modos se solapan con los que usan los computólogos (*computer scientists*). En particular, aquí interesa destacar los que él identifica como característicos de los últimos pero que comparten con los matemáticos:

- Manipulación de fórmulas,
- Construir representaciones de la realidad,
- Reducción [descomposición] de los problemas a problemas más simples,
- Razonamiento abstracto,
- Estructuras de información,
- Uso de Algoritmos.

Así también, Knuth (1985) destaca el tratamiento del infinito como algo particularmente ausente del modo de pensar típico de los computólogos (*computer scientists*). Y al contrario, también menciona que estos últimos incluyen en su modo de pensar algunas cosas que los matemáticos usualmente no consideran: (a) La noción de complejidad, relacionada con el «costo» del cómputo (en tiempo de cómputo y espacio de almacenamiento) y con la optimización de los algoritmos; (b) La noción de asignar, es decir la dinámica de Estado de Programa, que se refiere a que las variables van cambiando de valor a medida que los algoritmos se ejecutan; y (c) La necesidad de considerar múltiples pequeños diferentes casos (conocidos como Condiciones Especiales o Condiciones Extremas).

Está claro entonces que existen modos de pensamiento (o razonamiento) que son comunes entre los matemáticos y los computólogos (*computer scientists*). Además, para Modeste (2012b) el *Pensamiento Algorítmico* puede verse como parte del pensamiento matemático y asegura que se ha demostrado cómo este pensamiento algorítmico influye en el pensamiento matemático.

2.3.3. Caracterización y componentes operacionales del Pensamiento Algorítmico

Algunas publicaciones, como las de Ziatdinov y Musa (2012), Douadi, Tahar, y Hamid (2012) y la de Geda y Biró (2020) reportan resultados del desarrollo del *Pensamiento Algorítmico* sin realizar una operacionalización explícita y no van más allá del rendimiento en la resolución efectiva de problemas algorítmicos. Sin embargo, la revisión de la literatura muestra que el *Pensamiento Algorítmico* es más que sólo ser capaz de resolver problemas por medio de algoritmos.

Para Futschek (2006, p. 160) el *Pensamiento Algorítmico* es el siguiente conjunto de habilidades que están conectadas con la construcción y comprensión de algoritmos:

- La habilidad de analizar problemas dados.
- La habilidad de especificar un problema con precisión.
- La habilidad de encontrar acciones⁹ básicas [o elementales o atómicas] que son adecuadas para el problema dado.

⁹se refiere a las operaciones

- La habilidad de construir un algoritmo correcto para un problema dado usando las acciones básicas.
- La habilidad de pensar sobre todos los posibles casos, normales y especiales, de un problema.
- La habilidad de mejorar la eficiencia de un algoritmo.

Y agrega Futschek (2006, p. 160) que para él, el *Pensamiento Algorítmico* tiene un fuerte aspecto de creatividad: La construcción de nuevos algoritmos que resuelven problemas dados.

Para Grozdev y Terzieva (2015, p. 471) los componentes del *Pensamiento Algorítmico* son:

- Análisis: Determinar la condición inicial, objetivo, hipótesis y limitaciones;
- Descomposición: Dividir el problema en sub-problemas y determinar las operaciones básicas de solución;
- Formalización para crear el modelo: re-formular el problema en términos computacionales, creando un algoritmo y definir la relación entre las sub-tareas;
- Comprensión y aplicación de formalismos para representar los algoritmos;
- Ejecución de un determinado algoritmo mediante la ejecución formal y precisa de las principales actividades;
- Analizar algoritmos para determinar la solución óptima;
- Modificación de algoritmos ya familiares para su aplicación en nuevas condiciones;
- Creación de un algoritmo nuevo (desconocido) para un problema dado.

Grozdev y Terzieva (2015, p. 471) también identifican propiedades generales y específicas del *Pensamiento Algorítmico*. Las generales son: *integridad* y *rendimiento*, que ayudan a ver un problema de manera integral e involucra la imagen mental preliminar de la decisión. Las propiedades específicas se relacionan con la *discretización*, la *abstracción*, la *generalización* y consisten en resolver todos los problemas de una clase (o familia), otra propiedad específica es la *formalización* y la de dividir un problema potencialmente difícil y complejo en subproblemas. Estas propiedades sugieren que un algoritmo es un proceso paso a paso, y que permiten: (a) abstraerse de una entrada específica, (b) hacer la transición de una tarea a su forma general, y (c) representar el algoritmo utilizando un lenguaje formal.

En la investigación de Grover (2017, p. 276) se utilizan diferentes instrumentos cuantitativos y cualitativos para evaluar simultáneamente el *pensamiento algorítmico* y el *pensamiento computacional* en alumnos de educación básica (séptimo y octavo grado) utilizando principalmente el software Scratch, pero sin definir los conceptos operacionalmente. Esta evaluación hace énfasis en el control del flujo de ejecución, condicionales, el uso de variables dentro de los ciclos, y la representación gráfica y textual de los programas.

Parra y Delgado (2017), en su tesis en la que busca promover el *Pensamiento Algorítmico* en un grupo de niños de quinto grado, define operacionalmente el *Pensamiento Algorítmico* con tres dimensiones:

1. Abstracción y Modelamiento de la Información
2. Comprensión de Estructuras Condicionales
3. Búsqueda Sistemática de Información

Similar a la publicación de Grover (2017), la interpretación operacional de *Pensamiento Algorítmico* se enfoca en el flujo de ejecución.

Las características específicas del *Pensamiento Algorítmico* según Sadykova y Usolzev (2018, p. 4) son:

1. Discretización¹⁰. Ejecución paso a paso de un algoritmo, instanciación.
2. Abstracción. Identificación de las características esenciales de un objeto y abstracción [diferenciación/independización] de aquellas características que son irrelevantes para el problema.
3. Definición¹¹. Esto significa que un estudiante aplica un algoritmo general para resolver un cierto problema con la información de entrada ya definida.
4. Formalismo. Esto significa, la construcción de un modelo del problema en algún lenguaje formal.
5. Integridad de comprensión de un sistema complejo, y su representación como una unidad de objetos correlacionados.

Mientras que los componentes operacionales son, según Sadykova y Usolzev (2018, p. 4):

- Del Problema:
 - Análisis del problema, su descomposición en los niveles de procesos.
 - Formalización (abstracción) del problema.
- De los Datos:
 - Provisión del contenido.
 - Identificación del flujo de datos.
- Del Procesamiento de información:
 - Procesamiento de datos.
- Del Dominio del problema:
 - Análisis del dominio del problema e identificación de los objetos, reales y abstractos.
 - Ordenamiento jerárquico de los objetos.
- De los Objetos:
 - Identificación de los objetos.
 - Clasificación de los objetos.

¹⁰ *Discreetness*, es decir, lo opuesto a «continuidad»

¹¹ Los autores usan la palabra *Definiteness*

- Determinación de las propiedades de los objetos.
- Identificación de interconexiones entre los objetos.
- Formulación de conclusiones inductivas.
- Formulación de conclusiones deductivas.
- De los Eventos:
 - Predicción de eventos.
 - Procesamiento de eventos.
- Del Modelo:
 - Secuenciación de las operaciones y desarrollo del modelo de solución.
 - Análisis de la reacción del sistema.
 - Corrección (ajuste) del modelo y procesamiento de los algoritmos en caso de que el resultado final sea diferente de la salida esperada (depuración).
- Del Algoritmo:
 - Selección de un algoritmo de un conjunto de algoritmos listos para usar.
 - Ejecución de operaciones algorítmicas básicas.
 - Análisis de los datos de entrada del problema y su interpretación de acuerdo a los datos de entrada del algoritmo dado.
- Del Ejecutor:
 - La resolución de problemas por medio de la ejecución formal y precisa de las operaciones que constituyen el algoritmo.
 - Selección de un ejecutor capaz de realizar esas operaciones.
- Del Programa:
 - Evaluar el rendimiento de las operaciones algorítmicas básicas.
 - Identificación de las operaciones en lenguaje de alto nivel necesarias para resolver el problema.
- De la Experimentación (validación):
 - Solución del problema y comparación de su resultado con el resultado esperado.
 - Análisis del resultado obtenido y ajustes de los datos de entrada en caso de que difiera de la salida esperada.

Wong y Jiang (2018) hicieron una investigación similar a la de Grover (2017), utilizando también Scratch pero con niños de quinto grado de primaria con problemas secuenciales, condicionales y cíclicos. Ellos conceptualizaron operacionalmente la dimensión de *Pensamiento Algorítmico* como:

1. Especificar un problema y encontrar las instrucciones esenciales.

2. Construir un algoritmo correcto [para resolver ese problema].
3. Transformar el algoritmo en programa.

Como se puede ver en esta sección y en la sección 2.1.2, y como confirman Sadykova y Il'bahtin (2019) existe una gran diversidad de definiciones de *Pensamiento Algorítmico* en nuestro tiempo, tanto conceptuales como operacionales. De hecho, en la revisión de literatura de Sadykova y Il'bahtin (2019, p.421) se identifican ocho definiciones conceptuales diferentes de *Pensamiento Algorítmico* además de las que se han incluido en este trabajo.

Las habilidades que componen el *Pensamiento Algorítmico* según Futschek (2006, p. 160) constituyen una referencia importante sobre los conceptos de *Pensamiento Algorítmico* y *Pensamiento Computacional* (Doleck y cols., 2017; Douadi y cols., 2012; Futschek y Moschitz, 2010; Hsu y Wang, 2018; Jun y cols., 2014; Kátai, 2015; Kilhamn y Bråting, 2019; Parra y Delgado, 2017; Plerou y Vlamos, 2016; Wong y Jiang, 2018; Ziatdinov y Musa, 2012). Pero la última habilidad que menciona, la habilidad de mejorar la eficiencia de un algoritmo, no es una habilidad que un estudiante principiante común realice con eficacia, como lo descubrieron Kurland y cols. (2002) utilizando el lenguaje Logo, en cuya investigación sólo el 14 % y 21 % de los alumnos lograron encontrar la solución más eficiente a dos problemas distintos respectivamente.

Los componentes de Grozdev y Terzieva (2015, p. 471) complementan bien la lista de habilidades de Futschek (2006), especialmente en lo relativo a la descomposición del problema, la relación entre las sub-tareas, y la formalización de la representación de los algoritmos.

La lista de componentes operacionales de Sadykova y Usolzev (2018, p. 4) es bastante extensa e incluye la noción de depuración, cosa que Futschek (2006) y Grozdev y Terzieva (2015) pasan por alto, pero incluye nociones no sólo de la programación estructurada sino también de otros paradigmas como la Programación Orientada a Objetos y Programación dirigida por Eventos, así como la selección del ejecutor, humano o máquina, que se puede interpretar como la selección de la arquitectura, del lenguaje o del paradigma. Por ello esta lista de componentes se considera, en esta investigación, demasiado extensa para evaluar estudiantes novatos.

Las investigaciones de Grover (2017); Parra y Delgado (2017); Wong y Jiang (2018) son muy escuetas en la operacionalización y además se enfocan en niños, por lo que no se consideran en la definición operacional de esta investigación.

Por otro lado, aunque no tienen directamente un sustento empírico directo, las publicaciones de Futschek (2006), Grozdev y Terzieva (2015) y Sadykova y Usolzev (2018) son más detalladas en la operacionalización, por lo que se retomará la mayoría de sus componente para esta investigación.

Así que para efectos de esta investigación, y en consonancia con la definición conceptual de *Pensamiento Algorítmico* seleccionada y las definiciones operacionales analizadas, se desarrollará una definición operacional propia del *Pensamiento Algorítmico*. Ver sección 3.3 en la página 58.

2.3.4. Diversas investigaciones sobre Pensamiento Algorítmico

Hay varias investigaciones que buscan encontrar formas para propiciar y desarrollar el *Pensamiento Algorítmico* en los estudiantes en diferentes niveles educativos.

En la investigación de Futschek (2006) se afirma que el *Pensamiento Algorítmico* es una habilidad que puede ser desarrollada independientemente de aprender a programar computadoras. Esta hipótesis es explorada más recientemente por Ziatdinov y Musa (2012), que analizan estrategias de cómputo mental rápido para propiciar la construcción de algoritmos más eficientes para realizar ciertos tipos de multiplicaciones de números enteros; y explorada también por Kátai (2015), quien revela que ciertas herramientas de aprendizaje apropiadamente calibradas tienen potencial para promover efectivamente el *Pensamiento Algorítmico* tanto en estudiantes de ciencias como de humanidades. De hecho, para Ziatdinov y Musa (2012) el desarrollo del *Pensamiento Algorítmico* puede ser iniciado en las clases de matemática de la escuela primaria. Y según Douadi y cols. (2012), puede ser desarrollado independientemente de las tecnologías, implementación o lenguajes de programación específicos. Incluso podemos retomar la publicación de Klein y Hoffmann (2004) en la que se demuestra que el uso irreflexivo de calculadoras graficadoras pueden conducir a la ejecución de los algoritmos incorrectos para resolver los problemas.

Futschek (2007), propone actividades infantiles en el lenguaje LOGO para el desarrollo temprano del *Pensamiento Algorítmico*. Posteriormente, Futschek y Moschitz (2010) proponen una estrategia lúdica con este mismo fin.

Douadi y cols. (2012) utilizan juegos de micro-mundo similares a LOGO para propiciar de forma divertida el desarrollo del *Pensamiento Algorítmico* en niños.

Kátai (2015) realizó una investigación para promover el *Pensamiento Algorítmico* en estudiantes universitarios de carreras de ciencias y de humanidades, y llegó a la conclusión de que no hay barreras infranqueables en la manera en que estos dos grupos pueden desarrollar el *Pensamiento Algorítmico*. Él identifica tres principios para poder lograr esto: (a) utilizar desafíos moderados y progresivos, (b) hacer una transición gradual de lo concreto a lo abstracto, y (c) propiciar un involucramiento activo genuino.

La exploración de Lockwood y cols. (2016) intenta encontrar la importancia que Matemáticos profesionales le atribuyen al *Pensamiento Algorítmico* dentro de su profesión. Ellos entrevistaron matemáticos que se dedican a la investigación en ciencias matemáticas y de estas entrevistas semi-estructuradas extrajeron la importancia que implícitamente le dan al *Pensamiento Algorítmico* y al *Pensamiento Computacional*. Uno de los resultados fue la definición de *Pensamiento Algorítmico* (presentada en la sub-sección de definiciones de *Pensamiento Algorítmico* y *Pensamiento Computacional*, en la página 9).

La investigación de Grover (2017) busca incrementar el nivel de desarrollo del *Pensamiento Algorítmico* y también del *Pensamiento Computacional* en alumnos de educación básica (séptimo y octavo grado) utilizando principalmente el software Scratch.

La tesis de Parra y Delgado (2017) presenta una propuesta didáctica usando el software Scratch y los recursos del sitio web code.org para fomentar el desarrollo del *Pensamiento Algorítmico* en niños de quinto grado de primaria.

Porto Solano y cols. (2017) desarrollan Objetos Virtuales de Aprendizaje para el aprendizaje de los algoritmos como método de solución de problemas en estudiantes de un curso de introducción a la programación de computadoras en las áreas de ingeniería.

Sadykova y Il'bahtin (2019) realizan una revisión de la literatura sobre la definición conceptual de *Pensamiento Algorítmico* (principalmente en el mundo ruso-hablante) y aseguran que este tipo de pensamiento es un sistema que consta de dos tipos de componentes: tareas intermedias y finales (meta), y una comprensión de las operaciones que deberían conducir a la secuencia y ejecución de estas tareas.

Geda y Biró (2020) presenta una propuesta didáctica de desarrollo del *Pensamiento Algorítmico* en la que los niños aprenden las reglas de reproducción de autómatas celulares¹² en diferentes configuraciones (cuadrados, triangulares y hexagonales) y realizan los dibujos a mano con diferentes colores, para enfatizar las generaciones de células vivas.

Doğan (2020) realiza una investigación cualitativa en 36 profesores en un programa de entrenamiento pedagógico sobre sus opiniones sobre cuáles son las características del *Pensamiento Algorítmico*, los beneficios en los alumnos de que sus profesores tengan bien desarrollado el *Pensamiento Algorítmico*, cómo enseñarle a sus alumnos a desarrollarlo, cómo evaluarlo, y qué factores creen que impide su desarrollo. Las principales características que sus sujetos identificaron fueron:

1. Pensamiento sistemático
2. Clasificación / Ordenamiento
3. Planificación
4. Resolución de problemas
5. Pulcritud (relacionado con el orden y la planificación)

Juškevičienė (2020) realiza un análisis de la relación entre la programación de computadoras, el *Pensamiento Algorítmico*, el *Pensamiento Computacional* y las tecnologías para desarrollar el *Pensamiento Computacional*.

2.3.5. Beneficios o aportes del Pensamiento Algorítmico en la formación de matemáticos

El primer claro beneficio del *Pensamiento Algorítmico* en la formación de estudiantes de Ciencias Matemáticas (y de Ingeniería, Economía, etc.) es la posibilidad de la aceleración de los cálculos numéricos que de realizarse a mano, requerirían de mucho tiempo.

En la investigación de Lockwood y cols. (2016), sin excepción, todos los matemáticos entrevistados vieron el *Pensamiento Algorítmico* como útil en su profesión.

Estos matemáticos indicaron que el cómputo que realizan en sus trabajos es útil porque les permite lograr algunas cosas que no pueden ser hechas eficientemente a mano. Algunos de ellos enmarcan tal cómputo como una herramienta que les permite hacer matemática. Es decir, que el cómputo está enmarcado como un medio para un fin, algo que

¹²Ver https://es.wikipedia.org/wiki/Autómata_celular y https://es.wikipedia.org/wiki/Juego_de_la_vida

facilita la exploración de una conjetura o la generación de ejemplos que en última instancia sirven al propósito de proveer información para una exploración matemática teórica. (Lockwood y cols., 2016)

Wilf (2008) explica con ejemplos que «Un matemático puede actuar junto con una computadora para explorar un mundo dentro de las matemáticas. A partir de estas exploraciones puede crecer la comprensión, las conjeturas, los caminos hacia las pruebas y los fenómenos que no hubieran sido imaginables en la era anterior a las computadoras.»

Lovász (2008) asegura que programas como Maple¹³, Mathematica¹⁴, Matlab¹⁵ o programas propios, abren para los matemáticos un rango de observaciones y experimentos que habían sido inaccesibles antes de la era de las computadoras, y que proveen nuevos datos y revelan nuevos fenómenos. De nuevo se puede hablar de ejemplos sobre los Fractales, como presenta Navas-López (2017) con los Conjuntos de Julia; o con la espiral de Ulam, en el ejemplo mencionado en la sub-sección 2.3.2 (ver Chmielewski y cols., 2017; Orłowski y Chmielewski, 2018), o el ejemplo sobre las teselaciones en geometría esférica (ver Breda y Dos Santos, 2019; d’Azevedo Breda y Dos Santos, 2018).

Esta idea del enriquecimiento de la matemática gracias a las computadoras también la encontramos en Modeste (2016) y Navas-López (2017).

En la experiencia profesional de Knuth (1974), muchas personas han creado modelos computacionales de cosas y han descubierto que aprendieron más mientras configuraban el modelo que mientras observaban el resultado del programa final. Esto se basa en la ya mencionada frase, que «Se suele decir que una persona no entiende algo realmente hasta que se lo enseña a alguien más. Realmente una persona no entiende algo realmente hasta que puede enseñárselo a una computadora, lo que se hace expresándolo como un algoritmo.» (Knuth, 1974)

Lockwood y cols. (2016) también hablan de otras prestaciones implícitas del *Pensamiento Algorítmico* además de las aplicaciones prácticas en las cuales puede ser útil. Los matemáticos entrevistados por ellos sugieren un número de otros beneficios potenciales del *Pensamiento Algorítmico*:

Primero, el *Pensamiento Algorítmico* parece estar relacionado con prácticas matemáticas como la demostración y la resolución de problemas¹⁶. El tipo de pensamiento involucrado en el diseño e implementación de un algoritmo está estrechamente alineado con la resolución de problemas matemáticos y, por lo tanto, el desarrollo del *pensamiento algorítmico* podría ayudar a desarrollar la habilidad de resolución de problemas.

En segundo lugar, uno de los matemáticos entrevistados por Lockwood y cols. (2016) hizo comentarios interesantes sobre la relación entre el *Pensamiento Algorítmico* y la comunicación de ideas matemáticas con otros. Tal como se dijo arriba, si una persona es capaz de explicar algo con precisión a alguien más, es porque lo entiende. Para los estudiantes de ese matemático, entonces, él vio que más allá de cualquier conocimiento matemático que

¹³<https://www.maplesoft.com/>

¹⁴<https://www.wolfram.com/mathematica/index.html.es>

¹⁵<https://la.mathworks.com/products/matlab.html>

¹⁶De esto ya se habló en la subsección 2.2.3

estuvieran adquiriendo, tal tipo de pensamiento podría ayudar a mejorar sus habilidades de comunicación matemática.

En tercer lugar, varios matemáticos entrevistados por Lockwood y cols. (2016) también hablaron sobre la importancia de tener una cierta disposición a equivocarse y corregir errores, y sugirieron una relación entre esta idea y el *pensamiento algorítmico*. Esto está asociado al proceso de «depuración» durante la construcción de los algoritmos.

Doğan (2020) indica que los individuos con habilidades de *Pensamiento Algorítmico* son personas de mente abierta, orientadas a la investigación, buenas en matemática, creativas y seguras, con la capacidad de producir soluciones a problemas importantes, modelar datos y pensar sistemáticamente.

Los sujetos entrevistados por Doğan (2020) indican múltiples beneficios para los niños al tener sus profesores bien desarrollado lo que ellos conciben como *Pensamiento Algorítmico*. El principal de estos beneficios es que los vuelve planificados y organizados.

En resumen, se puede decir que los beneficios del desarrollo del *Pensamiento Algorítmico* en los estudiantes de ciencias matemáticas son los siguientes:

- Aceleración de cálculos, que al hacerlos a mano llevaría mucho tiempo o simplemente sería inviable hacerlos.
- Permite hacer Matemática Experimental, ya que las computadoras, al ser muy rápidas con respecto al ser humano, pueden hacer cálculos en grandes cantidades que permiten identificar patrones y explorar conjeturas.
- Fortalece las habilidades de demostración y de resolución de problemas.
- Como herramienta metacognitiva, con el propósito de aprender mejor aquello que se está estudiando, y por consiguiente, también hace posible sistematizarlo mejor y así poder comunicarlo con mayor precisión.
- Fortalecer la habilidad de depuración de errores en diferentes actividades.
- Propiciar conductas deseables en cualquier ciudadano moderno, como la creatividad, la planificación, la organización, el modelamiento de datos y el pensamiento sistemático.

2.3.6. Dificultades del Desarrollo del Pensamiento Algorítmico

Como se mencionó en la fundamentación del problema (sección 1.1), los porcentajes de reprobación de cursos introductorios de programación de computadoras son de entre 15% y 50% en todo el mundo. Arévalo-Mercado y cols. (2019) nos hablan de las posibles razones de estos malos resultados en términos de aprobación en cursos introductorios de programación de computadoras (programación estructurada básica). Estas son: (a) la necesidad de desarrollar *Pensamiento Algorítmico*, (b) resolver problemas usando este tipo de pensamiento, (c) tener que aprender conceptos propios de la programación de computadoras y sus modelos mentales, (d) dominar la sintaxis específica (de los diagramas de flujo, de algún pseudocódigo o de un lenguaje de programación concreto), y (e) aprender

a usar un entorno particular de desarrollo (IDE – Integrated Development Environment). Puesto que usualmente todo esto se imparte de forma simultánea tiende a provocar sobrecarga mental en estudiantes novatos.

Hace ya varias décadas Dijkstra (1974) identificó algunos aspectos que parecen ser únicos en la profesión de la programación [de computadoras]. Lo primero es la Noción de Jerarquía en las herramientas de los programadores (Flujogramas, Diagramas UML, Diagramas de Flujo de Datos, Lenguajes y Paradigmas de programación, IDEs, Estructuras de Dato, RDBMs, el modelo OSI, el modelo de diseño de software MVC, manejo de excepciones, programación concurrente, programación multi-capas, etc.). La Noción de Jerarquía implica que lo que en un nivel [semántico] es considerado como una unidad no analizada [es decir, simple o incluso atómica], es considerado como un objeto compuesto en el nivel [semántico] inmediato siguiente de mayor detalle.

Por ejemplo, en un programa orientado a objetos en el que se usa alguna Clase, la Clase es vista como una unidad no analizada (es decir que no se conoce necesariamente cómo funciona), pero dentro de la Clase, esta es un objeto compuesto de métodos y atributos coordinados que realizan tareas diversas.

Por ejemplo, en un programa estructurado en el que se usa una función, la función es vista como una unidad que realiza una tarea pero que no se conoce necesariamente cómo trabaja, mientras que dentro de la función, hay un algoritmo que recibe los parámetros, los procesa y genera una salida.

Por ejemplo al lanzar manualmente una excepción en un determinado ámbito, se detiene la ejecución, pero en el ámbito superior la excepción debería ser «atrapada» para evitar que el programa se detenga realmente.

Dijkstra (1974) identifica una consecuencia de esta naturaleza jerárquica de las herramientas de los programadores, y que le da a la Programación [de Computadoras] como actividad intelectual un sabor único, es que los programadores deben tener una agilidad [mental] con la cual se cambien [y se muevan] atrás y adelante, [arriba y abajo], entre varios de estos niveles semánticos, entre consideraciones globales y locales, entre asuntos macroscópicos y microscópicos, una agilidad como «lentes de aumento mentales». Y se entiende que esta agilidad es desconcertante para aquellos no acostumbrados a ella. Y esto incluye a los matemáticos.

Dijkstra (1974) también hace un comentario sobre que en los «currículo de matemática estándares» los alumnos sólo ven [estudian] problemas tan pequeños que pueden ser abordados en un sólo nivel semántico. El autor de esta tesis considera que esto claramente es un obstáculo en el aprendizaje de la programación de computadoras por parte de estudiantes de ciencias matemáticas.

Lovász (1988) insiste en que se debe abordar la creación de algoritmos antes de aprender a ejecutarlos. En segundo lugar, considera que el estudio de los algoritmos y su análisis debe abordarse de la misma forma que el de los teoremas y sus demostraciones. También advierte contra el uso demasiado rápido de la computadora en el estudio de algoritmos. En esto coinciden posteriormente Hershkowitz y Kieran (2001), Association for Computing Machinery y cols. (2016) y Wing (2017).

El experimento de Madeo y Bird (1990) sugiere que cuando los estudiantes novatos tienen que especificar partes del lenguaje algorítmico (pseudocódigo) a usar en sus soluciones, estos tienen menor rendimiento que aquellos que usan un lenguaje algorítmico completamente especificado (cerrado). Esto podría conducir a que cuando no se le especifica a los sujetos evaluados qué juego de instrucciones pueden usar, tiendan a tener menor rendimiento que si se les indicaran las instrucciones básicas disponibles.

Kurland y cols. (2002) realizaron una exploración de novatos en el aprendizaje de la programación con Logo y encontraron que las demandas cognitivas son muy altas, especialmente en lo que se refiere al control de flujo y la depuración, a pesar de haber estado en un programa intensivo de 6 semanas diseñado para mejorar las habilidades matemáticas e introducción a la programación.

En esta misma línea de ideas, Modeste (2016) reporta que en el currículum matemático de secundaria en Francia hay una ausencia de objetos matemáticos (en el área de la aritmética y la matemática discreta) que puedan ser rica fuente de problemas algorítmicos, y que esto puede explicar las dificultades identificadas por los profesores de matemática.

Modeste (2012b, 2016) enfatiza las problemáticas asociadas con la noción de asignación de variables, la operación de darle un nuevo valor a una variable, que puede ser representado con símbolos como «=», «:=» o «<-». Esta operación que es no simétrica no debe ser confundida con la igualdad «=» en matemática, que es una relación simétrica. Estas diferencias semánticas de estos símbolos en la programación, la aritmética y el álgebra generan obstáculos epistemológicos¹⁷ que pueden persistir al comienzo de los estudios universitarios. Kilhamn y Bråting (2019) también identifica esta diferencia semántica como una dificultad al aprender álgebra después de haber realizado actividades introductorias a la programación de computadoras.

Plerou y Vlamos (2016) realizan un recuento de las Dificultades en el *Pensamiento Algorítmico* (DAT - *Difficulties in Algorithmic Thinking*), que son diferentes disfunciones y trastornos psicológicos asociados al *Pensamiento Algorítmico*:

- Disfunción de la memoria de largo plazo
- Dificultades para evocar y realizar pasos sucesivos
- Dificultades en el Pensamiento Estratégico
- Ineficiencia en la percepción del tiempo
- Dificultades en la percepción espacial (asociado a la orientación y a dar direcciones)
- Dificultades en la concepción de problemas matemáticos
- Problemas en la memoria de procesamiento
- Problemas en la memoria serial
- Dificultades en el procesamiento, memorización y organización de la información

¹⁷El autor dice: «These different status of symbols in algorithmics, arithmetic and algebra generate language difficulties that can persist at the beginning of the University», pero en esta investigación se ha preferido referir a estas dificultades como «Obstáculos Epistemológicos» en el sentido que explica Artigue (2003, p. 121-122).

- Discalculia

Los algoritmos [...] incorporan varios factores invisibles y abstracciones, y los estudiantes generalmente no tienen referencias de la vida real con respecto a ellos (Ramadhan, 2000). Además, la programación de computadoras implica con frecuencia algoritmos secuenciales diseñados para que sean ejecutados por máquinas, basados en un conjunto limitado de instrucciones elementales, pero para los humanos es natural cooperar, actuar en paralelo y procesar un gran conjunto de instrucciones de alto nivel (Futschek y Moschitz, 2010).

Blanco Hamad (2016) reporta que los estudiantes presentan dificultades al momento de analizar, interpretar y abstraerse para dar solución a una situación problémica, pues ellos comienzan a resolver dicha situación sin tener totalmente claro lo que se les exige, sin haber analizado y determinado las condiciones y exigencias de la misma. Esto conlleva, en muchas ocasiones, a algoritmizar un problema distinto al que se plantea, o a que la solución algorítmica que se obtenga no los resuelva totalmente. Y también reporta que la mayoría de los estudiantes presenta dificultades asociadas al insuficiente tiempo que dedican al análisis e interpretación de los problemas.

Blanco Hamad y cols. (2016) encontraron en su investigación que los estudiantes no reconocen a la *algoritmización* como parte esencial del proceso de programación de computadoras, pues la mayoría no realiza una modelación previa, usando pseudocódigos o diagramas de flujo, sino que consideran que se debe resolver la situación problémica directamente en un lenguaje de programación. Esto claramente dificulta el desarrollo del *Pensamiento Algorítmico* ya que los estudiantes no le dan la importancia necesaria. Y al aplicar la misma metodología en otra población, Salgado Castillo y Alonso Berenguer (2017) llegaron a los mismos resultados.

Entre los principales factores que identifica Doğan (2020) que impiden el desarrollo de este tipo de pensamiento están:

1. Pasividad de los estudiantes
2. Uso de técnicas de enseñanza rutinarias por parte de los profesores
3. Enseñanza no gradual (lo que puede ser un problema curricular o bien didáctico)
4. Instrucción directa por parte de los profesores

Todos estos factores hacen que el *Pensamiento Algorítmico* sea difícil de desarrollar, y según Kátai (2015) difícil de incorporar en los enfoques didácticos.

2.4. Pensamiento Algorítmico y Pensamiento Computacional

2.4.1. Posibles confusiones de los términos

Se debe tener cuidado de no confundir el *Pensamiento Computacional* con el *Pensamiento Algorítmico* como hace Artopoulos (2013, p.73), refiriéndose a ambos como «una habilidad genérica que se aplica para resolver problemas complejos de procesamiento de

información mediante el método formal de explicitar los pasos en forma ordenada». Esta definición se asemeja más a la de *Pensamiento Algorítmico* de Lockwood y cols. (2016, p. 1591), la de Kátai (2015, p.287), la de Hoyles y Noss (2015) (como se citó en Sinclair y Patterson, 2018, p. 63), o la de Grozdev y Terzieva (2011) (como se citó en Grozdev y Terzieva, 2015, p.471) que a las definiciones de *Pensamiento Computacional* como la de Aho (2012, p. 832), la de Wing (2017, p. 8), la de Waterman y cols. (2018, p. 283) o las que se presentarán más adelante.

Otro ejemplo de confusión aparece en la tesis de maestría de López García (2014), cuyo título es «Actividades de aula con *Scratch*¹⁸ que favorecen el uso del pensamiento algorítmico», pero que utiliza en su desarrollo un instrumento de medición de *Pensamiento Computacional*, y en su sección de Resultados y Conclusiones de nuevo se habla en términos de *Pensamiento Algorítmico*.

Esta distinción está más clara en la actualidad, a pesar de que hace pocas décadas, computólogos de gran influencia como Donald Knuth (1985, p.170) se refirieran a las Ciencias de la Computación como la ciencia que estudia los algoritmos, con lo que se podría pensar que el modo de pensar en Ciencias de la Computación es el *Pensamiento Algorítmico* exclusivamente. En cualquier caso, en la actualidad, se considera que «el *Pensamiento Algorítmico* es una parte necesaria de la cosmovisión científica» (Ziatdinov y Musa, 2012, p. 1105), y que el *Pensamiento Algorítmico* es el principal componente del *Pensamiento Computacional* (Juškevičienė, 2020).

El *Pensamiento Algorítmico* puede verse como parte del pensamiento matemático y ya se han mencionado algunas de sus características, además, se ha demostrado cómo este *Pensamiento Algorítmico* influye en el pensamiento matemático (véase de nuevo a Dijkstra, 1974; Knuth, 1974, 1985; Modeste, 2012a, 2012b, 2016; Modeste y cols., 2010; y Lockwood y cols., 2016). Sin embargo, ver el *Pensamiento Algorítmico* como un pensamiento importante en las ciencias de la computación permite un enriquecimiento real de su análisis. Por tanto, para abordar el *Pensamiento Algorítmico*, es imprescindible captar simultáneamente los dos puntos de vista, el intra-matemático pero también extra-matemático. (Modeste, 2012b)

Existe también una marcada tendencia por asociar *Pensamiento Computacional* con programación de computadoras, como indican Wing (2006) y Voogt y cols. (2015), lo que no es correcto. Tang y cols. (2020) muestran en su revisión sistemática sobre la evaluación del *Pensamiento Computacional* que el 44%¹⁹ de las publicaciones evalúan o califican el *Pensamiento Computacional* con conceptos relacionados a la Programación de Computadoras.

Sin embargo sí es cierto es que ambas «constituyen competencias clave que deben ser adquiridas por los jóvenes estudiantes, y cada vez más por los trabajadores, en una amplia gama de actividades industriales y profesionales» (Bordignon y García-Marín, 2018, p.180, como se citó en Téllez Ramírez, 2019, p. 30). También es cierto que encontrar e inventar algoritmos apropiados para un problema dado es un requisito previo necesario para la programación de computadoras (Futschek y Moschitz, 2010; Juškevičienė, 2020), y que la

¹⁸<https://scratch.mit.edu/>

¹⁹Ver figura 3 de Tang y cols. (2020, p. 6) y calcular la media ponderada correspondiente.

programación de computadoras es una herramienta importante para ayudar a desarrollar habilidades de *Pensamiento Computacional* (Voogt y cols., 2015).

Es importante también resaltar que el *Pensamiento Computacional* no se refiere a «pensar como una computadora», ya que las computadoras no piensan, y el pensamiento es un rasgo inherentemente humano. El término se refiere en realidad a pensar como un Computólogo²⁰. (Grover y Pea, 2018)

Por otro lado, «la programación [de computadoras] es un tipo específico de actividad humana y su realización exitosa requiere no sólo la aplicación práctica de conocimientos y habilidades, sino también un tipo específico de pensamiento [el *Pensamiento Algorítmico*]». (Grozdev y Terzieva, 2015)

Esta idea es compartida por Sadykova y Usolzev (2018, p. 2), que afirman que «el *Pensamiento Algorítmico* bien desarrollado es muy importante para el desarrollo de programas informáticos. Si un estudiante no tiene las habilidades del pensamiento algorítmico, incluso el conocimiento de uno o diferentes lenguajes de programación no será de utilidad.»

Para Wing (2006) las ciencias de la computación se basan de manera inherente en el Pensamiento Matemático, dado que, como todas las ciencias, sus fundamentos formales se basan en las matemáticas. Pero también se basan de forma inherente en el pensamiento de la ingeniería, dado que se construyen sistemas que interactúan con el mundo real. Las restricciones de los dispositivos de cómputo subyacentes obligan a los computólogos a pensar de forma computacional, no sólo matemáticamente.

En el Marco de Trabajo de Ciencias de la Computación para la Educación de Secundaria²¹, la Association for Computing Machinery y cols. (2016, p.70) alertan sobre que el *Pensamiento Computacional* puede tener lugar sin una computadora, y también que el uso de una computadora en clase no necesariamente construye o promueve *Pensamiento Computacional*. Esta idea es compartida por Wing (2017, p. 8) que afirma que las personas pueden desarrollar *Pensamiento Computacional* sin una máquina. Voogt y cols. (2015) van más allá al afirmar que el *Pensamiento Computacional* no necesariamente requiere del uso de la programación de computadoras.

2.4.2. Orígenes y definición operacional del término Pensamiento Computacional

Freiman y cols. (2018) afirman que el *Pensamiento Computacional* es un constructo relativamente nuevo, que se está volviendo el centro de los debates educacionales y se está incrementando el número de publicaciones de investigación.

Investigadores como Grover y Pea (2013); Voogt y cols. (2015); Czerkawski y Lyman III (2015); Weintrop y cols. (2016); Baranauskas y Carbajal (2017); Corradini, Lodi, y Nardelli (2017); Denning (2017); Broley y cols. (2017); Freiman y cols. (2018); Israel y Lash (2019); etc., identifican los inicios de la noción de *Pensamiento Computacional* en el trabajo de

²⁰ *Computer Scientist*

²¹ K-12 Computer Science Framework, <http://www.k12cs.org>

Seymour Papert (1980), especialmente en su libro «Mindstorms: Children, computers, and powerful ideas».

Sin embargo, fue el ensayo de Wing (2006) el que inició las discusiones científicas recientes sobre el *Pensamiento Computacional*. En ese ensayo, Wing (2006) describe extensamente su visión del *Pensamiento Computacional*. Ella dice que el *Pensamiento Computacional* implica la resolución de problemas, el diseño de sistemas y la comprensión del comportamiento humano, basándose en los conceptos fundamentales de las ciencias de la computación. Y agrega que el *Pensamiento Computacional* incluye una variedad de herramientas mentales que reflejan la amplitud del campo de las Ciencias de la Computación:

- **Reformular un problema aparentemente difícil en uno que sí se sabe resolver, tal vez mediante reducción, transformación o simulación.**
- **Pensar de forma recursiva.**
- Procesar en paralelo.
- Interpretar el código como datos y los datos como código.
- Verificar tipos como generalización del análisis dimensional.
- Reconocer tanto las virtudes como los peligros de usar un alias, o dar a alguien o algo más de un nombre.
- Reconocer tanto el costo como el poder del direccionamiento indirecto y las llamadas a procedimiento.
- Juzgar un programa no sólo por su correctitud y eficiencia, sino también por su estética, y el diseño de un sistema por su simplicidad y elegancia.
- **Utilizar la abstracción y la descomposición cuando se ataca una tarea grande y compleja o se diseña un sistema grande y complejo.**
- **Elegir una representación adecuada para un problema o modelar los aspectos relevantes de un problema para hacerlo manejable.**
- Utilizar invariantes para describir el comportamiento de un sistema de forma sucinta y declarativa.
- Tener la confianza de que se puede utilizar, modificar e influir de forma segura en un sistema grande y complejo sin comprender todos sus detalles.
- Modularizar algo en anticipación a múltiples usuarios o pre-cargar y almacenar en caché en anticipación del uso futuro.
- Pensar en términos de prevención, protección y recuperación de los peores escenarios a través de la redundancia, la contención de daños y la corrección de errores.
- Aprender a evitar los bloqueos mutuos²² y las condiciones de carrera²³ al sincronizar reuniones entre sí.
- **Utilizar el razonamiento heurístico para descubrir una solución.**

²² *deadlocks*

²³ *race conditions*

- Planificar, aprender y calendarizar en presencia de incertidumbre.
- Utilizar cantidades masivas de datos para acelerar la computación.
- Hacer concesiones y negociaciones entre tiempo y espacio, y entre potencia de procesamiento y capacidad de almacenamiento.

Los elementos de esta lista marcados en negrita son análogos a algunos de los componentes identificados del *Pensamiento Algorítmico* (ver sección 2.3.3).

Posteriormente, la *International Society for Technology in Education* y la *Computer Science Teachers Association* (ISTE y CSTA, 2011) construyeron por consenso una definición operacional del **Pensamiento Computacional**: Es un proceso de resolución de problemas que incluye (pero no se limita a) las siguientes características:

- Formular problemas de una manera que permita usar una computadora y otras herramientas para ayudar a resolverlos.
- Organizar y analizar datos de forma lógica.
- Representar datos a través de abstracciones como modelos y simulaciones.
- Automatizar soluciones a través del **Pensamiento Algorítmico** (una serie de pasos ordenados).
- Identificar, analizar e implementar posibles soluciones con el objetivo de lograr la combinación más eficiente y efectiva de pasos y recursos.
- Generalizar y transferir este proceso de resolución de problemas a una amplia variedad de problemas.

Estas habilidades, según la ISTE y la CSTA (2011), están respaldadas y mejoradas por una serie de disposiciones o actitudes que son dimensiones esenciales del *Pensamiento Computacional*. Estas disposiciones o actitudes incluyen:

- Confianza para afrontar la complejidad.
- Persistencia en trabajar con problemas difíciles.
- Tolerancia a la ambigüedad.
- La capacidad de lidiar con problemas abiertos.
- La capacidad de comunicarse y trabajar con otros para lograr un objetivo o solución común.

De esta definición operacional y descripción del *Pensamiento Computacional* queda muy claro que el *Pensamiento Algorítmico* es una de sus características principales. Esta afirmación también se sustenta en la revisión sistemática de literatura que realizan Selby y Woollard (2013) sobre el concepto de *Pensamiento Computación* cuyos objetivos son (a) definir de manera más estrecha, no más amplia el concepto; (b) poner orden en los criterios y no necesariamente incorporar todos los puntos de vista; (c) refinar la definición para facilitar la evaluación; (d) mantener la vigencia de trabajos realizados anteriormente, como la elaboración de currículos; (e) separar una definición de aquellas actividades que podrían promover la adquisición de habilidades de *Pensamiento Computacional*; y

(f) separar la definición de aquellas actividades que evidencian el uso de habilidades de pensamiento computacional.

Así, para Selby y Woollard (2013) el *Pensamiento Computacional* es una actividad, con frecuencia orientada a producto, asociada con, pero no limitada a la resolución de problemas. Es un proceso cognitivo o de pensamiento que utiliza:

- la abstracción,
- la descomposición [de problemas en subproblemas],
- el ***Pensamiento Algorítmico***,
- la evaluación [de algoritmos y sistemas computacionales], y
- la generalización [entendida como la reutilización de algoritmos en contextos diferentes a los originales].

Para Grover y Pea (2013) el *Pensamiento Computacional* está compuesto por los siguientes elementos:

- Abstracciones y generalizaciones de patrones (incluidos modelos y simulaciones)
- Procesamiento sistemático de información
- Sistemas y representaciones de símbolos
- **Nociones algorítmicas de flujo de ejecución**
- **Descomposición estructurada de problemas (modularización)**
- Pensamiento iterativo, recursivo y paralelo
- **Lógica condicional**
- Limitaciones de eficiencia y desempeño
- **Depuración y detección sistemática de errores**

Similar a como se hizo con la lista de Wing (2006), los elementos de esta lista marcados en negrita son análogos a algunos de los componentes identificados del *Pensamiento Algorítmico* (ver sección 2.3.3).

Marais y Bradshaw (2015) construyeron un instrumento para medir el *Pensamiento Computacional* siguiendo las ideas de la ISTE y CSTA (2011). Su instrumento tiene cuatro dimensiones:

1. ***Pensamiento Algorítmico***
2. Modelos y Simulaciones
3. Eficiencia de las Soluciones
4. Otras características (confianza, inventiva, organización, rigor, etc.)

Weintrop y cols. (2016) presentan una taxonomía de prácticas y habilidades que componen el Pensamiento Computacional pensando en su incorporación en las aulas de clase de ciencias y matemática de educación básica y secundaria:

1. Datos

- a) Recolección de datos
 - b) Creación de datos
 - c) Manipulación de datos
 - d) Análisis de datos
 - e) Visualización de datos
2. Modelación y Simulación
- a) Usar modelos computacionales para entender un concepto
 - b) Usan modelos computacionales para encontrar y verificar soluciones
 - c) Evaluar modelos computacionales
 - d) Diseñar modelos computacionales
 - e) Construir modelos computacionales
3. Resolución de Problemas Computacionales
- a) **Preparar problemas para obtener soluciones computacionales**
 - b) **Programación [de computadoras]**
 - c) Selección efectiva de herramientas computacionales
 - d) Evaluar y valorar diferentes aproximaciones o soluciones a un problema
 - e) **Desarrollar soluciones computacionales modulares**
 - f) **Crear abstracciones computacionales**
 - g) **Solución de problemas y depuración**
4. Pensamiento en Sistemas
- a) Investigar un sistema complejo como un todo
 - b) Entender las relaciones dentro de un sistema
 - c) Pensamiento en niveles
 - d) Comunicar información sobre un sistema
 - e) Definir sistemas y administrar la complejidad

Aunque esta taxonomía no incluye explícitamente el *Pensamiento Algorítmico*, sí hace énfasis en la habilidad de resolución de problemas por medio de soluciones y abstracciones, tal como indican Futschek (2006) y Grozdev y Terzieva (2015); y también hace énfasis en la comprensión de sistemas complejos como hace Sadykova y Usolzev (2018). Por ello se considera que implícitamente Weintrop y cols. (2016) incorpora el *Pensamiento Algorítmico* en su taxonomía de habilidades de *Pensamiento Computacional*.

En la revisión sistemática de literatura de Kalelioğlu y cols. (2016), se identificó que las características más comunes que forman el *Pensamiento Computacional* son:

1. Abstracción

2. Resolución de problemas
3. **Pensamiento Algorítmico**
4. Reconocimiento de patrones
5. Pensamiento basado en diseño

Para Korkmaz y cols. (2017) y Doleck y cols. (2017), las habilidades del *Pensamiento Computacional* son:

- ***Pensamiento Algorítmico***
- Cooperatividad
- Creatividad
- Pensamiento Crítico
- Resolución de Problemas

En la investigación de Shute, Sun, y Asbell-Clarke (2017), el *Pensamiento Computacional* se compone de las siguientes facetas:

- Descomposición
- Abstracción
- Diseño de Algoritmos (***Algoritmización***)
- Depuración
- Iteración
- Generalización

En la investigación de Israel y Lash (2019) se usan juntos los términos de Ciencias de la Computación y *Pensamiento Computacional*, y se asume que describen ampliamente experiencias en las que los estudiantes se comprometen en la resolución de problemas y en el pensamiento algorítmico, tanto para la programación de computadoras como con otras actividades para engranar con conceptos matemáticos.

Barr y Stephenson (2011) presentan lo complicado, y al mismo tiempo importante, que es lograr una definición más unificada de *Pensamiento Computacional* para poder incorporarlo apropiadamente a los currículos escolares.

Tal como sucede con otros constructos psicológicos, esta definición no está universalmente aceptada y sigue en evolución, como muestran las revisiones sistemáticas de literatura de Selby y Woollard (2013), Voogt y cols. (2015), Shute y cols. (2017), Freiman y cols. (2018) y Tang y cols. (2020), y los meta-análisis de Grover y Pea (2013) y Kalelioğlu y cols. (2016) sobre la literatura científica que trata sobre el *Pensamiento Computacional* en los últimos años.

Sobre este hecho, Denning (2017) investiga que tanto la definición, como los métodos de evaluación y las afirmaciones de beneficio universal son puntos problemáticos en el estudio del *Pensamiento Computacional*. Todavía no existe una única y universalmente aceptada definición operacional de *Pensamiento Computacional*.

Tang y cols. (2020, p. 6) reporta entre otras cosas, que más de un 55% de las investigaciones relacionadas a este constructo están orientadas a la educación infantil y básica, y que para educación media son sólo de poco más del 15%, igual que para educación superior. También ocurre que «la mayoría de los artículos [...] no explican explícitamente por qué evaluar las habilidades de *Pensamiento Computacional* puede ser intercambiable con evaluar conceptos de programación de computadoras y de ciencias de la computación.» (Tang y cols., 2020, p. 8) .

Por otro lado, Voogt y cols. (2015) argumentan «que con el propósito de conceptualizar el *Pensamiento Computacional* e integrarlo en la educación, no deberíamos intentar dar una definición definitiva del *Pensamiento Computacional*, sino más bien tratar de encontrar similitudes y relaciones en las discusiones sobre el *Pensamiento Computacional*. Encontrar estas similitudes y relaciones conducirá a una descripción más concisa de "lo que importa" en el *Pensamiento Computacional* y cómo integrarlo en la educación básica.» (p. 726) Este mismo razonamiento se podría aplicar al *Pensamiento Algorítmico*.

Sin embargo, para efectos de esta investigación, el *Pensamiento Algorítmico* es clara y explícitamente una parte importante del *Pensamiento Computacional*. Esto está sustentado tanto en la definición de ISTE y CSTA (2011), como en las revisiones de literatura que realizan Selby y Woollard (2013), Grover y Pea (2013) y Kalelioğlu y cols. (2016).

2.4.3. Beneficios del Desarrollo del Pensamiento Computacional

«Se puede argumentar que el siglo XXI es el siglo de la computación. La inteligencia artificial finalmente ha alcanzado la mayoría de edad al integrarse en la transformación del trabajo, el comercio y la vida cotidiana. El Big Data, el reconocimiento facial y de voz, la robótica, el internet de las cosas (IoT²⁴), la computación en la nube, los vehículos autónomos y el acceso 24×7 a cualquier persona, en cualquier parte del mundo a través de las redes sociales está cambiando cómo y dónde las personas trabajan, colaboran, se comunican, compran, comen, viajan, ven noticias y obtienen entretenimiento, en resumen cómo viven. La computación también está transformando la industria y la innovación en todas las disciplinas, convirtiéndose en una herramienta integral que está impulsando nuevas formas de hacer y de pensar. En un mundo tan saturado por la computación, el *pensamiento computacional* se reconoce ahora como una competencia fundamental para ser un ciudadano informado y tener éxito en los trabajos STEM, que también tiene el potencial como un medio para la resolución creativa de problemas y la innovación en todas las demás disciplinas.» (Grover y Pea, 2018, p. 20)

Wing (2010) describe así los beneficios generales de desarrollar *Pensamiento Computacional*:

- Entender qué aspectos de un problema son susceptibles de computación,
- Evaluar la correspondencia entre las herramientas y técnicas computacionales y un problema,

²⁴https://es.wikipedia.org/wiki/Internet_de_las_cosas

- Comprender las limitaciones y el poder de las herramientas y técnicas computacionales,
- Aplicar o adaptar una herramienta o técnica computacional a un nuevo uso,
- Reconocer la oportunidad de usar la computación de una manera nueva, y
- Aplicar estrategias computacionales como divide y vencerás en cualquier dominio.

Para Wing (2010), además, el desarrollo del *Pensamiento Computacional* para científicos, ingenieros y otros profesionales significa poder:

- Aplicar nuevos métodos computacionales a los problemas propios de su disciplina,
- Reformular problemas de su disciplina para que sean susceptibles de aplicarles estrategias computacionales,
- Descubrir nueva ciencia a través del análisis de grandes grandes cantidades de datos,
- Hacer nuevas preguntas en las que no se pensó ni se atrevieron a abordar debido a la escala de procesamiento disponible, pero que se abordan fácilmente de forma computacional, y
- Explicar problemas y soluciones en términos computacionales.

El informe de Tucker y cols. (2003) y el artículo de Riesco y cols. (2014) insisten en la necesidad de aumentar el nivel de formación en Ciencias de la Computación (y por lo tanto de *Pensamiento Computacional*) en toda la población a través de su incorporación en la educación básica para prepararlos mejor para una ciudadanía efectiva en el siglo 21.

2.4.4. Diferencias entre el Pensamiento Algorítmico y el Pensamiento Computacional

A partir de las definiciones descritas en la sección 2.4.2, podemos identificar que el *Pensamiento Algorítmico* se enfoca en la creación, análisis, manipulación y ejecución de algoritmos, tal como muchas de las definiciones de las secciones 2.1 y 2.3.3 indican. Sin embargo, de las descripciones y definiciones de *Pensamiento Computacional* en la sección 2.4.2 se puede apreciar que este va más allá de la creación, análisis, manipulación y ejecución de algoritmos. Por ejemplo, este incluye la habilidad de «Interpretar el código como datos y los datos como código», la habilidad de «Pensar en términos de prevención, protección y recuperación de los peores escenarios a través de la redundancia, la contención de daños y la corrección de errores» (Wing, 2006), y también la habilidad de «Representar datos a través de abstracciones como modelos y simulaciones» (ISTE y CSTA, 2011).

Recuérdese que para Selby y Woollard (2013) en su revisión sistemática de literatura sobre el *Pensamiento Computacional*, concluyen que es una actividad, asociada con, pero no limitada a la resolución de problemas. Los estándares educativos para la enseñanza de las Ciencias de la Computación que plantea la *Computer Science Teachers Association* (CSTA, 2017), además de algoritmos, programación, abstracción y modelamiento, incluye muchos aspectos como la ciber-seguridad, los impactos de la computación en la cultura,

los impactos de la computación en las leyes y la ética; lo cual es coincidente con la noción de *Technoracy* que plantea D'Ambrosio (1999).

Por lo tanto, de forma muy sucinta, podemos decir que el *Pensamiento Computacional*, a diferencia del *Pensamiento Algorítmico*, no sólo trata con algoritmos, sino que trata con las características y limitantes de la tecnología subyacente a estos algoritmos (cosas como lenguajes de programación, paradigmas de programación, concurrencia, paralelismo, *hardware*, etc.) y de las implicaciones sociales del uso de esta tecnología (importancia de las contraseñas, respeto y responsabilidad en las relaciones en línea, conciencia sobre las alteraciones en las prácticas culturales debido al uso de las tecnologías computacionales, uso de licencias *creative commons*²⁵, y otras similares, etc.).

2.5. Pensamiento Computacional y Matemática

Para cerrar el ciclo de relaciones entre Matemática, *Pensamiento Algorítmico* y *Pensamiento Computacional* se presenta en esta sección la relación reciente entre el *Pensamiento Computacional* y la Matemática.

Aquí debemos considerar una vez más la utilización de la programación de computadoras, como componente indiscutible del Pensamiento Computacional, en la enseñanza y aprendizaje de la matemática. Esto ya fue abordado en la sección 2.2.6 en la página 20.

Hace varias décadas, simultáneamente a la invención de la computadora personal, Papert (1980) visualizó un mundo donde los niños usaran con fluidez esa herramienta como jóvenes matemáticos. Posteriormente esta noción evolucionó como se describe en la sección 2.4.

Más recientemente, Weintrop y cols. (2016) abordan el desafío de definir el *Pensamiento Computacional* (ver sección 2.4.2) en el contexto de la educación básica de ciencias y matemática, y proveer un fundamento teórico sobre qué forma debería tomar este tipo de pensamiento en las aulas de clase de ciencias y matemáticas. Ellos abogan por la idea de empotrar²⁶ y no sólo incorporar el *Pensamiento Computacional* en los contextos matemáticos y científicos en las aulas de clase. Indican que esto es necesario debido a la naturaleza cada vez más computacional de la ciencia y las matemáticas modernas.

Para Broley y cols. (2017), el poder de la computadora ha tenido un gran impacto en la forma en la que los profesionales STEM pueden hacer y hacen su trabajo. Ellos, conscientes de que faltan ejemplos concretos que validen los modelos de incorporación o empotramiento del *Pensamiento Computacional* en las aulas de clase de matemática, realizan pruebas reales con el modelo de Weintrop y cols. (2016) (ver sección 2.4.2). Y muestran que se valida su correspondencia con un diverso conjunto de prácticas profesionales en el contexto de la matemática.

Shute y cols. (2017) examinan el campo del *Pensamiento Computacional* en la educación. Entre otras cosas, comparan el *Pensamiento Computacional* con el Pensamiento Matemá-

²⁵https://es.wikipedia.org/wiki/Creative_Commons

²⁶ *embedding*

tico, el Pensamiento de Ingeniería, el Pensamiento de Diseño y el Pensamiento de Sistemas²⁷. En la figura 2.6 se presenta la traducción del gráfico de intersección entre *Pensamiento Computacional* y Pensamiento Matemático según Shute y cols. (2017, p. 146).

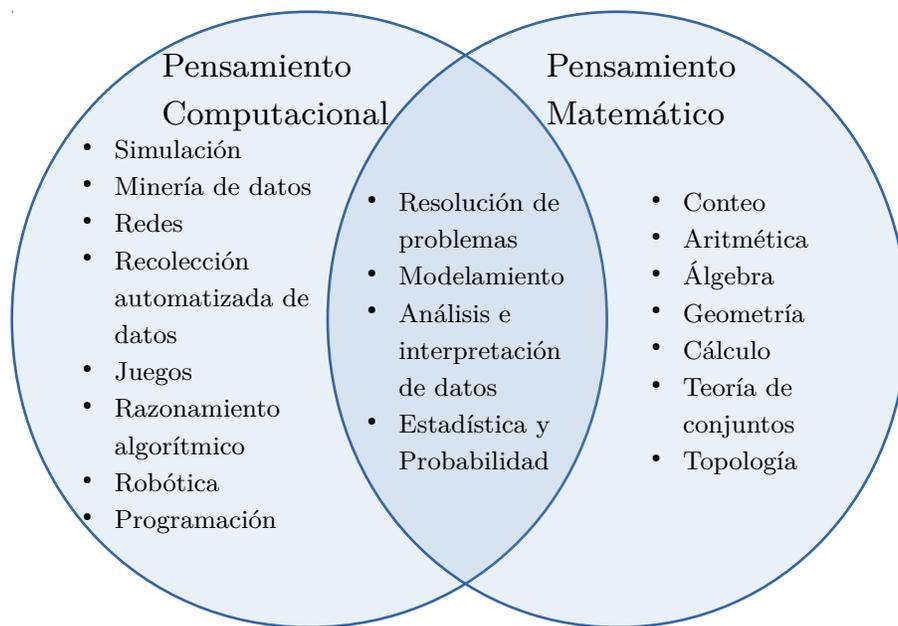


Figura 2.6.: Intersección del Pensamiento Computacional y del Pensamiento Matemático. Traducción de la figura 1 de Shute y cols. (2017, p. 146). Todos los derechos reservados 2017 por Elsevier Inc.

Freiman y cols. (2018) realizaron una discusión al rededor de la pregunta ¿Cómo podríamos conceptualizar el *Pensamiento Computacional* en la investigación de la enseñanza de la matemática? Ellos analizan que existe una multiplicidad de perspectivas sobre lo que es el *Pensamiento Computacional* que pueden ser útiles dependiendo de los objetivos de sus usuarios, por ejemplo dependiendo de los propósitos específicos de un investigador, y que estas diferentes perspectivas pueden corresponderse a diferentes tipos de tareas.

Incluso en un pequeño grupo de investigadores especialistas en el tema, como reportan Freiman y cols. (2018), cada uno define el *Pensamiento Computacional* de una forma ligeramente diferente: Para unos se trata de las praxeologías de una persona (Broley y cols., 2018), para otros se trata de un artefacto o herramienta (Buteau, Muller, Marshall, Sacristán, y Mgombelo, 2016), mientras que para otros se trata de una de muchas modalidades de aprendizaje (Shute y cols., 2017).

Israel y Lash (2019) realizaron unas actividades educativas desarrolladas para enseñanza básica de matemática para niños usando el software Scratch²⁸, y algunos materiales del sitio code.org²⁹. Aplicaron teorías de integración entre el *Pensamiento Computacional* y

²⁷mathematical, engineering, design, and systems thinking

²⁸<https://scratch.mit.edu/>

²⁹<https://code.org/>

la matemática. El estudio proporcionó información sobre el nivel de interconexión entre las disciplinas y la cantidad relativa de tiempo de instrucción dedicado a cada disciplina. Su conclusión es que en cada actividad, las dos disciplinas pueden estar: No integradas, parcialmente integradas, y completamente integradas.

Los contextos matemáticos pueden ser usados para incorporar el *Pensamiento Computacional* a la enseñanza y aprendizaje, proporcionando así oportunidades fértiles para realizar investigaciones significativas; esto puede profundizar la comprensión de los estudiantes de las matemáticas, profundizar la comprensión del papel del *Pensamiento Computacional* y de las posibilidades de las herramientas computacionales. (Freiman y cols., 2018)

Finalmente, Freiman y cols. (2018) reconocen que no está claro cómo evaluar las actividades de *Pensamiento Computacional*. Ni siquiera está claro qué evaluar exactamente: ¿matemática, *Pensamiento Computacional*, *Pensamiento Computacional* y matemática, *Pensamiento Computacional* para hacer matemática? Lo mismo sucede con la metodología, ¿Evaluación cuantitativa, evaluación cualitativa, o ambas?

A manera de resumen esquemático, se presenta en la figura 2.7 las relaciones entre los principales conceptos de esta investigación, que son «Algoritmo», «Algoritmización», «Matemática», «*Pensamiento Algorítmico*» y «*Pensamiento Computacional*».

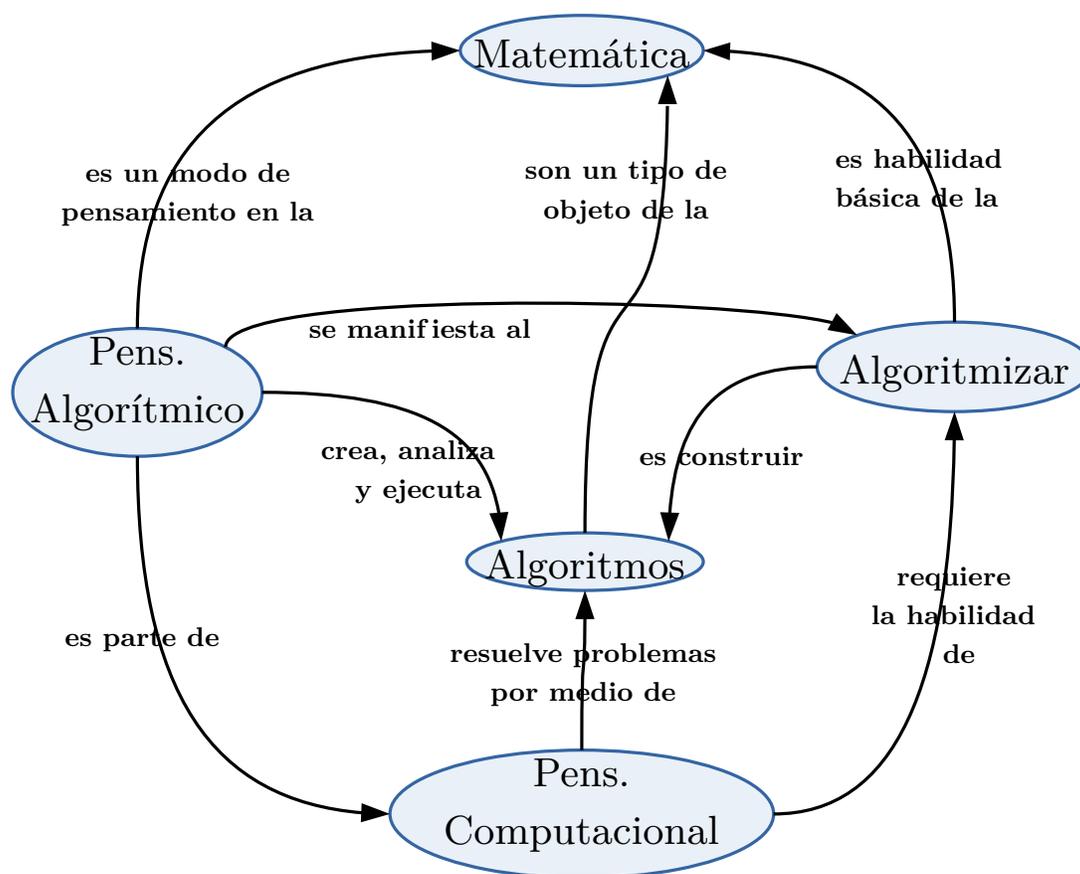


Figura 2.7.: Resumen de la relación entre los principales conceptos de esta investigación. Fuente: Elaboración propia.

3. Metodología

En este capítulo se describe todo lo relacionado con los métodos y técnicas aplicados en esta investigación.

3.1. Tipo, diseño y alcance de la investigación

Para esta sección se han utilizado las categorías de enfoques, tipos, alcances y diseños de investigación descritos por Hernández Sampieri, Fernández Collado, y Baptista Lucio (2014, cap. 5, 7, 15 y 17).

Con el objetivo de enriquecer el entendimiento del *Pensamiento Algorítmico* entre los estudiantes y otros constructos relacionados, esta investigación es de **tipo mixta**, con una parte cuantitativa y una parte cualitativa, siguiendo un **Diseño de Triangulación Concurrente** entre ellas. Se espera que esto provea una perspectiva más amplia y profunda de los conceptos estudiados.

La primera parte es **cuantitativa** ya que se analiza de manera numérica (cuantitativamente) el nivel de desarrollo del *Pensamiento Algorítmico* de los alumnos que conforman la muestra. Es de **diseño no experimental** ya que no se controla intencionalmente ninguna variable de los sujetos que componen la muestra. Es de tipo **transversal** ya que se analizan a todos los sujetos en una única ocasión y no se les da seguimiento en el tiempo. Finalmente, el alcance de esta investigación es **correlacional**, ya que se realiza una correlación múltiple entre las variables investigadas de los sujetos de la muestra (ver la sección 1.5).

La parte **cualitativa** sigue un **Diseño de Teoría Fundamentada** de tipo **Sistemática**, y realiza el procedimiento de **codificación axial** basándose en las categorías de procesos cognitivos, reconocidas en la literatura, implicados en el proceso de resolución de problemas, realizados por parte de los individuos de una muestra de una población equivalente a la de la muestra de la parte cuantitativa. En este caso aplicadas estas categorías a problemas cuya solución es un algoritmo. También identifica la secuencia de acciones que realizan los participantes durante ese proceso, con el objetivo de desarrollar **una teoría sustantiva (o teoría de rango medio)** que tenga validez local. Para poder llegar a tal teoría se desarrolla una **representación o modelo visual** de ese proceso.

Este diseño mixto de triangulación se justifica también por la necesidad de confirmar o corroborar los resultados de la parte cuantitativa con la exploración más profunda de la parte cualitativa. Se compensa además la limitante de usar sólo dos problemas para medir

el *Pensamiento Algorítmico* en la parte cuantitativa por medio de la diversidad de problemas usados en la parte cualitativa, como se verá más adelante. Logrando así una complementación de ambos enfoques y la corroboración mutua entre ambos.

Por medio de la figura 3.1 se presenta un resumen esquemático del proceso de investigación de la parte cuantitativa.

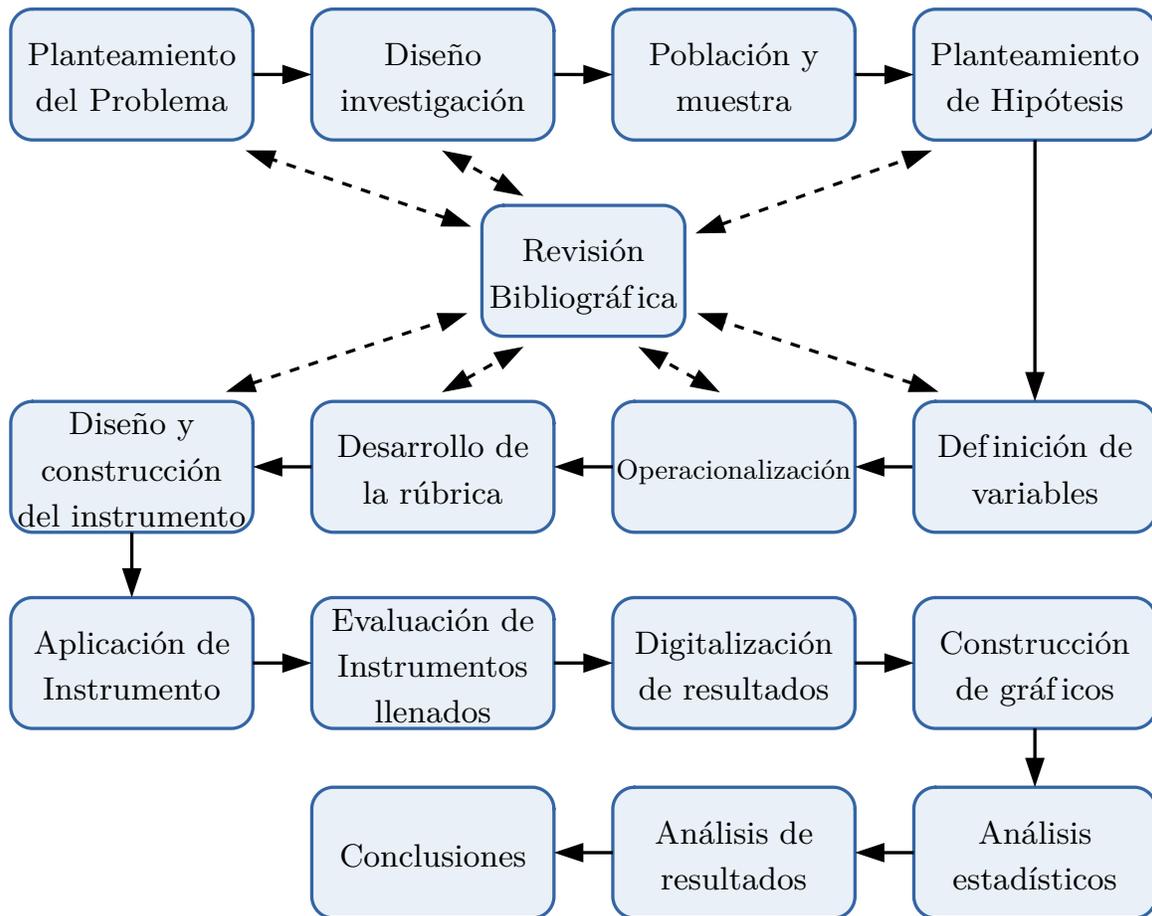


Figura 3.1.: Pasos de la parte cuantitativa de esta investigación.

Fuente: Elaboración propia.

En la figura 3.2 en la página siguiente se presenta un resumen esquemático del proceso de investigación de la parte cualitativa.

Cuando se llegó a la fase de Análisis de Resultados en la parte cualitativa, se detectaron ciertas carencias en la representación gráfica inicialmente desarrollada y por ello se procedió a una fase de rediseño y ajuste de la misma, lo que implicó modificación del programa de computadora que genera estas representaciones gráficas de forma iterativa hasta lograr el resultado actual.

Una vez obtenidos los resultados de las partes cuantitativa y cualitativa, se realiza un proceso de compración de resultados siguiendo el diseño de triangulación concurrente. Ver figura 3.3 en la página siguiente.

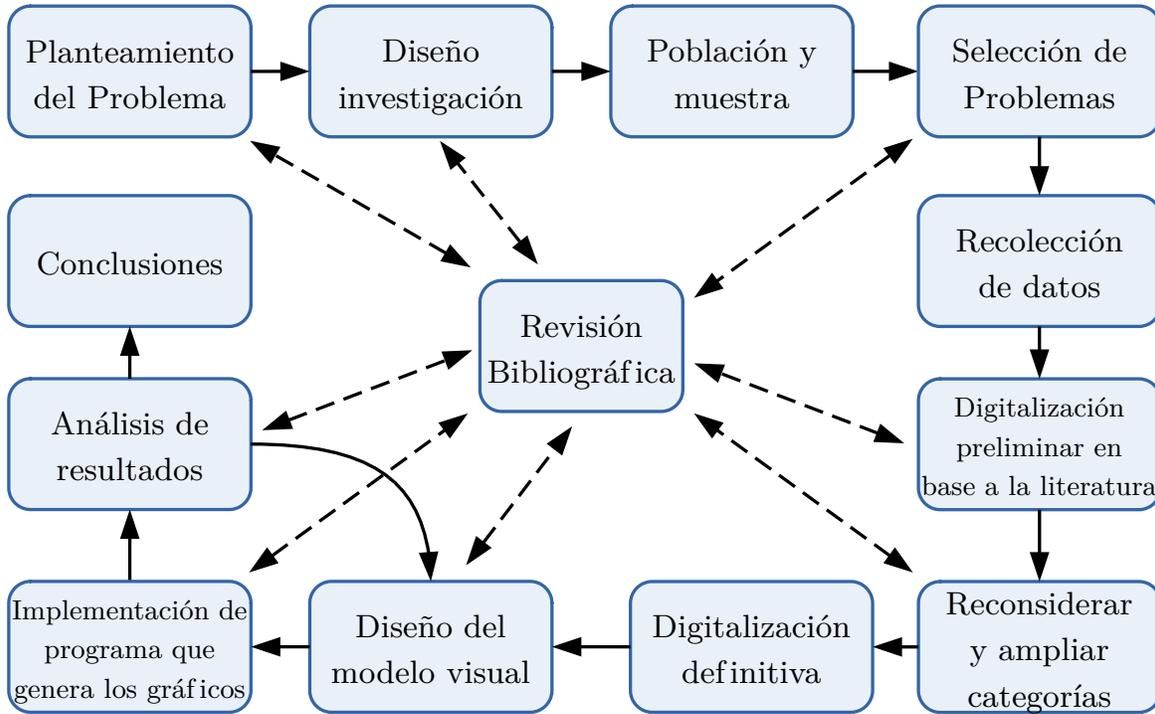


Figura 3.2.: Pasos de la parte cualitativa de esta investigación.
Fuente: Elaboración propia.

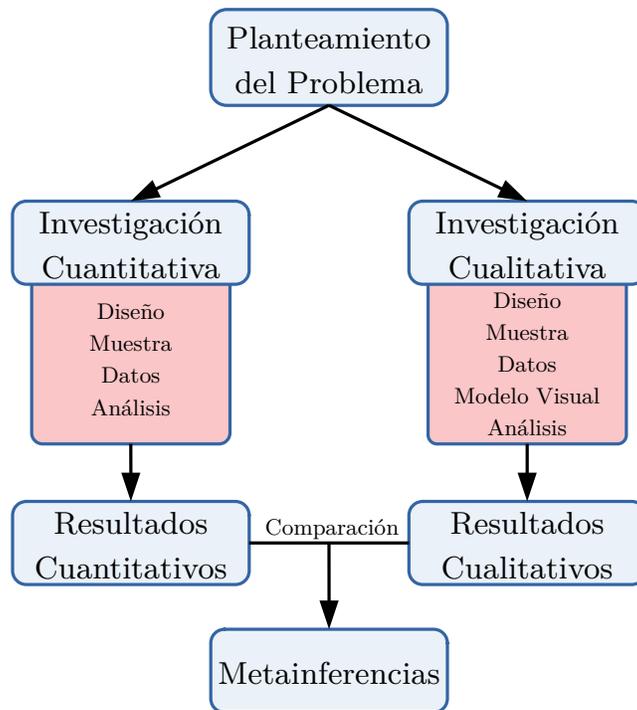


Figura 3.3.: Diseño mixto concurrente de esta investigación.
Fuente: Elaboración propia.

3.2. Población objeto de estudio y muestra

3.2.1. Población y muestra para la parte cuantitativa

Se analiza el desarrollo del *Pensamiento Algorítmico* de diferentes grupos de estudiantes que estén cursando las carreras de Licenciatura en Matemática y Licenciatura en Estadística de la Facultad de Ciencias Naturales y Matemática de la Universidad de El Salvador. Particularmente se analiza un grupo de alumnos que no haya cursado ninguna materia universitaria sobre informática o programación de computadoras (Grupo P0, constituido por 47 estudiantes), otro grupo que haya aprobado una materia de este tipo (Grupo P1, constituido por 9 estudiantes), otro grupo que haya aprobado dos materias de este tipo (Grupo P2, constituido por 13 estudiantes), y otro grupo que haya aprobado tres materias de este tipo (Grupo P3, constituido por 11 estudiantes), que son todas las posibilidades dados los planes de estudio de las carreras analizadas (ver Apéndice C en la página 225). Por lo que el tipo de muestra es **No Probabilística**, seleccionada así **por conveniencia** para resaltar las etapas académicas que deberían propiciar el desarrollo del *Pensamiento Algorítmico* más que otras etapas.

En total, se analizan **80 alumnos** de una población de 252 alumnos activos en ambas carreras durante el ciclo 02/2018. El tamaño de esta población fue determinado por la información proporcionada por la oficina de Administración Académica de la Facultad de Ciencias Naturales y Matemática. Ver cuadro 3.1.

Cuadro 3.1.: Población de para la parte cuantitativa

Carrera	Nuevo ingreso		Antiguo ingreso		Total
	Femenino	Masculino	Femenino	Masculino	
Lic. en Matemática	39	35	59	69	202
Lic. en Estadística	19	9	7	15	50

Fuente: Oficina de Administración Académica de la Facultad de Ciencias Naturales y Matemática de la Universidad de El Salvador.

3.2.2. Población y muestra para la parte cualitativa

Se analiza la grabación de una sesión de resolución de un problema de los 14 alumnos que durante el ciclo 02/2020 cursan la materia «Programación 2» (ver Apéndice C en la página 225), que es la tercera y última materia en las que se estudia programación de computadoras, de las carreras de Licenciatura en Matemática y Licenciatura en Estadística de la Facultad de Ciencias Naturales y Matemática de la Universidad de El Salvador, para explorar los procesos cognitivos que estos alumnos realizan durante la resolución de problemas cuya solución es un algoritmo.

Para contrastar un poco y comparar, también se analiza la grabación de resolución de un problema por parte de un profesor universitario, con varios años de experiencia, que dirige cursos de programación de computadoras, introducción a la programación, lógica para computación, y similares. Se considera a este sujeto un experto en resolución de problemas lógico-matemáticos de la escala de complejidad de los asignados a la muestra de 14 de alumnos.

Por lo que el tipo de muestra es **No Probabilística**, seleccionada así **por conveniencia** debido a dificultades y limitantes derivadas de la Pandemia de COVID-19 durante el año 2020. En total, se analizan **14 alumnos** de una población de 230 alumnos activos en ambas carreras durante el ciclo 02/2020, más el sujeto considerado experto. El tamaño de esta población fue determinado por la información proporcionada por la oficina de Administración Académica de la Facultad de Ciencias Naturales y Matemática. Ver cuadro 3.2.

Cuadro 3.2.: Población de para la parte cualitativa

Carrera	Nuevo ingreso		Antiguo ingreso		Total
	Femenino	Masculino	Femenino	Masculino	
Lic. en Matemática	21	17	75	79	192
Lic. en Estadística	7	2	10	19	38

Fuente: Oficina de Administración Académica de la Facultad de Ciencias Naturales y Matemática de la Universidad de El Salvador.

3.3. Definición operacional de Pensamiento Algorítmico

Tal como se explicó en la sección 2.3.3 se decidió realizar una definición operacional propia de *Pensamiento Algorítmico*. A continuación se procede a describir una definición operacional propia del *Pensamiento Algorítmico de un Principiante* como el siguiente conjunto de habilidades:

- Analizar el problema, es decir, identificar las entradas y las salidas.
- Descomponer el problema en sub-problemas hasta encontrar las operaciones básicas adecuadas al problema. Esto no incluye descomposición recursiva de los problemas, sino sólo descomposición iterativa.
- Construir un algoritmo correcto que resuelve el problema usando las operaciones básicas.
- Construir una representación formal del algoritmo que resuelve el problema, ya sea en diagramas de flujo, en pseudocódigo, un lenguaje de programación concreto, u otro tipo de representación.

- Identificar los casos normales y especiales de un problema.
- Ejecutar un algoritmo, generalmente con el propósito de evaluar su correctitud, y buscar las causas de errores ya identificados (depuración básica).

Ahora se procede a describir una definición operacional propia el *Pensamiento Algorítmico de un Experto*, siguiendo la definición conceptual de Mingus y Grassl (1998) (ver sub-sección 2.1.2) y considerando todos los aspectos considerados por Futschek (2006) y Grozdev y Terzieva (2015), y algunos de Sadykova y Usolzev (2018).

El *Pensamiento Algorítmico de un Experto*, incluye las siguientes habilidades:

- El conjunto de habilidades del *Pensamiento Algorítmico de un Principiante* (es decir, la lista anterior).
- (Re)plantear el enunciado de un problema en términos algorítmicos, es decir, en términos de las entradas y las salidas.
- Descomponer el problema en sub-problemas hasta encontrar las operaciones básicas adecuadas al problema, incluyendo descomposición recursiva de los problemas.
- Aplicar estrategias sistemáticas para la localización de las causas de errores ya identificados (depuración avanzada).
- Determinar las limitantes de un algoritmo dado.
- Mejorar la eficiencia de un algoritmo, es decir construir otro algoritmo que resuelva los mismos problemas pero que requiera menos recursos (tiempo y/o memoria).
- Determinar el algoritmo más óptimo para un problema.
- Modificar algoritmos ya conocidos para su aplicación en condiciones diferentes. Esto incluye modificarlos para que sean más generales.

Se considera que esta separación resulta pertinente y está justificada en la taxonomía de objetivos educacionales para actividades cognitivas de Bloom y Krathwohl (1956). Esta taxonomía de objetivos educacionales son: Conocimiento (saber), comprensión, aplicación, análisis, síntesis y evaluación; como se citó en Booker (2007), Forehand (2010) y en Darwazeh y Branch (2015).

En la lista de habilidades que constituyen el *Pensamiento Algorítmico de un Principiante*, se ha incluido los objetivos de Conocimiento (saber), comprensión y aplicación (ejecución) de algoritmos. Y se deja para la lista de habilidades que constituyen el *Pensamiento Algorítmico de un Experto* los objetivos de análisis¹, síntesis y evaluación de algoritmos.

En base a estas definiciones operacionales de *Pensamiento Algorítmico*, se presenta en la página siguiente una propuesta propia de operacionalización genérica del constructo o variable *Pensamiento Algorítmico de un Principiante*.

En el cuadro 3.4 en la página 61 se presenta una posible rúbrica para calcular el nivel de *Pensamiento Algorítmico de un Principiante* a partir de sus sub-variables para un problema dado, para lo cual se han seguido los pasos del diseño de rúbricas de puntuación de Mertler (2000) y se ha utilizado el estilo de la plantilla presentada por Cebrián de la Serna y Monedero Moya (2014, p.85).

¹Aquí nos referimos a analizar un algoritmo dado, no a analizar un problema para construir un algoritmo

Cuadro 3.3.: Propuesta propia de Operacionalización genérica del *Pensamiento Algorítmico de un Principiante*.

Dimensiones	Indicadores	Sub-variable
Análisis del problema	Identifica las Entradas	pa-input
	Realiza un esquema gráfico útil	pa-dibu
	Identifica las Salidas	pa-output
Descomposición del problema en sub-problemas	Identifica los subproblemas que componen el problema	pa-idsub
Construcción de un algoritmo correcto que resuelve el problema	Implementa una solución a los subproblemas identificados	pa-solsub
	Integra las diferentes soluciones parciales para resolver el problema completo	pa-intsub
Identificación de los casos normales y especiales	Identifica los casos normales	pa-norm
	Identifica los casos especiales o extremos	pa-extre
Construcción de una representación formal del algoritmo que resuelve el problema	Representa el flujo de ejecución. En el caso de usar Pseudocódigo, esto incluye la indentación y el anidamiento de las estructuras de control como si-entonces-sino, hacer-mientras, hacer-para, etc.). En el caso de Diagramas de Flujo, el flujo debe ser representado de forma no ambigua.	pa-flujo
	Utiliza la simbología y las reglas sintácticas apropiadas (como los rombos para las bifurcaciones condicionales en diagramas de flujo, o «palabras reservadas» en pseudocódigo)	pa-sin
Ejecución de un algoritmo	Ejecuta efectivamente un algoritmo dada su representación	pa-exe
	Ejecuta un algoritmo que se considera potencialmente correcto para verificar casos normales y especiales, y buscar las causas de errores ya identificados	pa-err

Fuente: Elaboración propia.

Cuadro 3.4.: Propuesta de Rúbrica de evaluación genérica de las sub-variables del constructo *Pensamiento Algorítmico de un Principiante*.

Sub-variables	Indicadores				
pa-input	No identifica ninguna variable de entrada [0%]	Define/utiliza otra(s) variable(s) como variables de entrada [10%]	Define/utiliza alguna(s) variable(s) de entrada correcta con otra(s) incorrecta(s) [25%]	Define/utiliza alguna(s) variable(s) de entrada correcta(s) pero no todas [50%]	Define/utiliza correctamente las variables de entrada [100%]
pa-dibu	No realiza ningún esquema o dibujo [0%]	Realiza un esquema o dibujo que carece de utilidad para identificar las variables importantes [10%]	Realiza un esquema o dibujo con potencial para identificar las variables importantes pero sin estar identificadas [30%]	Realiza un esquema o dibujo donde se pueden identificar algunas variables importantes pero no todas las que permite dicha representación [70%]	Realiza un dibujo o esquema donde se pueden identificar las variables importantes [100%]
pa-output	No identifica ninguna variable de salida [0%]	Define/utiliza otra(s) variable(s) como variables de salida [10%]	Define/utiliza alguna(s) variable(s) de salida correcta con otra(s) incorrecta(s) [25%]	Define/utiliza alguna(s) variable(s) de salida correcta(s) pero no todas [50%]	Define/utiliza correctamente las variables de salida [100%]
pa-idsub	No identifica ninguno de los subproblemas [0%]	Identifica sólo algunos de los subproblemas [25%]	Identifica la mitad de los subproblemas [50%]	Identifica casi todos los subproblemas [75%]	Identifica todos los subproblemas [100%]

pa-solsub	No intenta resolver ninguno de los subproblemas [0 %]	Intenta, pero no resuelve correctamente ninguno de los subproblemas [5 %]	Resuelve correctamente algunos de los subproblemas [50 %]	Resuelve correctamente la mayoría de los subproblemas [80 %]	Resuelve correctamente todos los subproblemas [100 %]
pa-intsub	No intenta resolver ninguno de los subproblemas [0 %]	Resuelve o intenta resolver sólo uno de los subproblemas [5 %]	Resuelve o Intenta resolver algunos subproblemas de forma aislada o en orden incorrecto [50 %]	Resuelve o Intenta resolver la mayoría de los subproblemas y uno de ellos no está en el orden correcto [80 %]	Resuelve o Intenta resolver todos los subproblemas en el orden correcto [100 %]
pa-norm	No intenta resolver el problema [0 %]	Resuelve o intenta resolver pero no declara en ningún momento que las variables de entrada deben tener valor y tipo correcto [10 %]	Resuelve o intenta resolver pero no declara para todas las variables de entrada que deben tener valor y tipo correcto [50 %]	Resuelve o intenta resolver y declara explícitamente para todas las variables de entrada que deben tener valor o tipo correcto, pero no ambas condiciones [75 %]	Resuelve o intenta resolver y declara explícitamente para todas las variables de entrada que deben tener valor y tipo correcto [100 %]
pa-extre	No intenta resolver el problema [0 %]	Resuelve o intenta resolver sin considerar los casos extremos derivados de las variables de entrada [5 %]	Resuelve o intenta resolver considerando algunos de los casos extremos derivados de las variables de entrada [60 %]	Resuelve o intenta resolver considerando la mayoría de los posibles casos extremos derivados de las variables de entrada [80 %]	Resuelve o intenta resolver considerando todos los posibles casos extremos derivados de las variables de entrada [100 %]

	o bien, si sólo es un caso extremo:				
	No intenta resolver el problema [0 %]	Resuelve o intenta resolver el problema pero en ningún momento considera el caso extremo [5 %]	Identifica pero no trata de resolver el caso extremo del problema [25 %]	Intenta pero no logra resolver el problema del caso extremo [50 %]	Resuelve o intenta resolver el problema considerando el caso extremo [100 %]
pa-flujo	No intenta resolver el problema [0 %]	Intenta resolver pero no respeta la secuenciación y no utiliza ninguna estructura de selección ni de iteración siendo necesarias [5 %]	Intenta resolver respetando la secuenciación pero no utiliza ninguna estructura de selección ni de iteración siendo necesarias [40 %]	Resuelve o intenta resolver utilizando las estructuras necesarias, pero sin respetar la indentación en caso de usar pseudocódigo, o sin dejar claro el flujo de ejecución en caso de usar un diagrama de flujo [60 %]	Resuelve o intenta resolver respetando la secuenciación y utilizando las estructuras necesarias, con su indentación apropiada en caso de usar pseudocódigo, o con su flujo de ejecución claro en caso de usar un diagrama de flujo [100 %]

pa-sin	No intenta resolver el problema [0 %]	Intenta resolver pero no utiliza simbología propia de los diagramas de flujo, o no utiliza palabras reservadas mínimas de Pseudocódigo, como Si, Entonces, Sino, FinSi, Inicio, Fin, Mostrar, Imprimir, etc. [10 %]	Intenta resolver usando simbología correcta, o pseudocódigo parcialmente claro [40 %]	Resuelve o intenta resolver usando simbología correcta en su mayoría, o pseudocódigo mayormente claro [80 %]	Resuelve o intenta resolver usando la simbología correcta de los diagramas de flujo, o pseudocódigo completamente claro [100 %]
pa-exe	No ejecuta el algoritmo [0 %]	Ejecuta el algoritmo cometiendo muchos errores, principalmente relacionados con el flujo de ejecución [10 %]	Ejecuta el algoritmo cometiendo pocos errores, principalmente de cálculo de operaciones y menos relacionados al flujo de ejecución [50 %]	Ejecuta el algoritmo cometiendo un sólo error relacionado a cálculo de operaciones y no relacionado al flujo de ejecución [80 %]	Ejecuta correctamente el algoritmo a partir de su representación [100 %]

pa-err	No ejecuta el algoritmo [0 %]	Ejecuta a mano el algoritmo sólo para una combinación de valores de las variables de entrada [10 %]	Ejecuta a mano el algoritmo para algunas combinaciones de valores de las variables de entrada [40 %]	Ejecuta a mano el algoritmo para varias combinaciones de valores de las variables de entrada sin identificar cómo cubrir las principales combinaciones [75 %]	Ejecuta a mano el algoritmo para las principales combinaciones de valores de las variables de entrada [100 %]
--------	-------------------------------	---	--	---	---

Fuente: Elaboración propia.

Es de resaltar que la ponderación relativa de cada una de estas sub-variables depende de cada problema, con sus propias condiciones y dificultades, o según los objetivos e intenciones del educador y del evaluador.

Cabe mencionar también que esta rúbrica evalúa asumiendo que se usa el *Paradigma de Programación Estructurada*² porque es la que se utiliza en los cursos impartidos a la población analizada en esta investigación. La *Programación Estructurada*, es decir el *Teorema del Programa Estructurado*, establece que toda función computable puede ser implementada en un lenguaje de programación que combine sólo tres estructuras lógicas (Böhm y Jacopini, 1966). Esas tres estructuras, también llamadas *Estructuras de Control*, específicamente son:

1. Secuencia: ejecución de una instrucción tras otra.
2. Selección: ejecución de una de dos instrucciones (o conjuntos de instrucciones), según el valor de una variable booleana. También conocida como Estructura de Control Condicional.
3. Iteración: ejecución de una instrucción (o conjunto de instrucciones) mientras una variable booleana sea 'verdadera'. Esta estructura lógica también se conoce como Ciclo o Bucle³.

Este paradigma de programación fue difundido inicialmente por Dijkstra (1968) con su influyente ensayo que ha marcado tendencia en el diseño de lenguajes de programación desde los años 70's.

3.4. Variables de la parte cuantitativa

Las variables de la parte cuantitativa y sus niveles de medición (tipos) se presentan en el cuadro 3.5 en la página siguiente.

En la subsección 4.1.1 en la página 119 se presenta un análisis descriptivo de los datos recopilados.

3.5. Instrumento de recolección de datos de la parte cuantitativa

Como se ha estudiado en las secciones 2.1 y 2.3.3, la forma idónea de analizar, medir y/o evaluar el desarrollo del *Pensamiento Algorítmico* es por medio de la resolución de problemas algorítmicos. Por lo tanto, el instrumento a utilizar es un **cuestionario de tipo test** (tipo examen) y consiste de cuatro problemas matemáticos. Dos de estos problemas buscan medir la variable PAP y los otros dos la variable RP.

²https://es.wikipedia.org/wiki/Programación_estructurada

³Loop en inglés

Cuadro 3.5.: Tipología de las variables de la parte cuantitativa.

Variable	Descripción	Nivel/Tipo	Dominio
matProg	Cantidad de materias relacionadas con la programación de computadoras que han aprobado, incluyendo cursos que no son parte de la carrera que estudian	de Razón	{0,1,2,3}
matTot	Cantidad total de materias universitarias que han aprobado	de Razón	Entero positivo
cum	Coefficiente de Unidades de Mérito (CUM)	de Razón	Real no negativo, [0.0,10.0]
edad	Edad	de Razón	Entero positivo
gen	Género	Nominal	{Masculino, Femenino}
alg	Si la indicación del instrumento pide explícitamente un «algoritmo», o si pide «instrucciones»	Nominal	{Verdadero, Falso}
PAP	Nivel de Desarrollo del <i>Pensamiento Algorítmico de Principiante</i>	de Razón	Real no negativo, [0.0,100.0]
RP	Habilidad de modelamiento y resolución de problemas de referencia	de Razón	Real no negativo, [0.0,100.0]

Fuente: Elaboración propia.

Además se recolecta la información de clasificación y agrupamiento como el año de ingreso a la universidad, la edad, el CUM (Coeficiente de Unidades de Mérito), el género, el número de materias aprobadas y el número de materias de programación aprobadas. Al final se incluye un espacio para escribir comentarios.

La apariencia concreta del instrumento puede consultarse en el Apéndice A en la página 217.

3.5.1. Indicaciones generales

Estas son las indicaciones escritas en el instrumento, y que fueron leídas en voz alta al iniciar la ejecución del test:

- Por favor intente resolver en las páginas adicionales todos los problemas que se le presentan a continuación. Cada problema en página separada. Plasme por escrito todas las ideas que le surjan durante su proceso de resolución. Utilice dibujos, diagramas, tablas y todos los recursos que desee.
- No borre nada de lo que escriba, pero si cree que no es el procedimiento correcto, simplemente táchelo y no lo borre. Si necesita más papel, solicítelo.
- Si cree que falta información precisa para resolverlos, siéntase en la libertad de suponer valores para los datos que considere faltantes, y deje por escrito que lo hará y por qué.

- Puede resolverlos en el orden que usted prefiera.

3.5.2. Problemas

El Problema 1 está inspirado en las reflexiones de Knuth (1974, 1985), Bell, Alexander, Freeman, y Grimley (2009, p. 6) y Modeste (2012b, 2016) acerca de la posibilidad y beneficios de utilizar problemas del ámbito de la Matemática Discreta para fomentar el desarrollo del *Pensamiento Algorítmico*. En este caso particular se trata de una instancia del Problema de la Ruta más Corta⁴, con 9 nodos y 13 aristas: Encontrar la ruta más corta del nodo A al nodo I (ver figura 3.4).

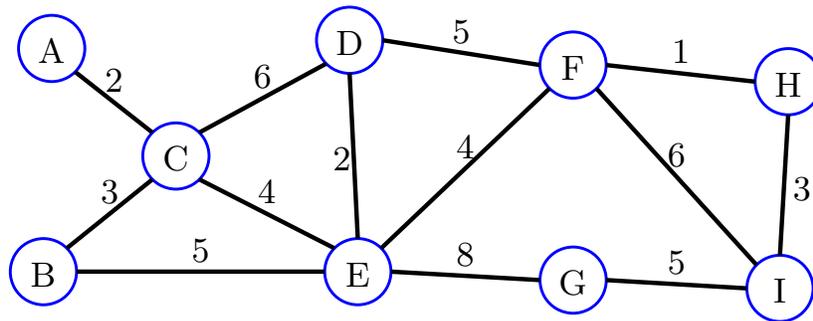


Figura 3.4.: Grafo que representa la información del Problema 1.

Fuente: Elaboración propia.

Los Problemas 2 y 4 están inspirados en el problema del cuasi-experimento de Pyzara (2014) que tiene dos variantes:

Variante 1: «Cree un *esquema* que permita calcular el costo de un viaje escolar de un día. El *esquema* debería ser tan general y claro para que este pueda ser usado por otro organizador mientras planea el mismo viaje el próximo año. Datos de entrada: (a) Es un viaje guiado, (b) hay un acompañante por cada 15 alumnos, (c) al viaje van cuatro grupos de clase, (d) hay programada una visita a un museo y a un teatro en el viaje.» (p. 29).

Variante 2: «Cree un *algoritmo* que permita calcular el costo de un viaje escolar de un día. El *algoritmo* debería ser tan general y claro para que este pueda ser usado por otro organizador mientras planea el mismo viaje el próximo año. Datos de entrada: (a) Es un viaje guiado, (b) hay un acompañante por cada 15 alumnos, (c) al viaje van cuatro grupos de clase, (d) hay programada una visita a un museo y a un teatro en el viaje.» (Pyzara, 2014, p.30).

Estos problemas, 2 y 4, son intencionalmente vagos sobre las variables de salida que se espera del algoritmo solución. Esto con la intención de alejar al resolutor de los problemas académicos clásicos donde todo está claro en el enunciado. Además, los problemas no

⁴https://es.wikipedia.org/wiki/Problema_del_camino_más_corto

mencionan explícitamente cuál es la información de entrada, por lo que el resolutor debe decidir cuáles serán las variables de entrada. Hay varias posibilidades y algunas variables son co-dependientes, por lo que se debe elegir con cuidado cuáles son las variables de entrada y de salida sin que entren en conflicto.

El Problema 3 es de creación propia del autor, incorporando modelamiento matemático geométrico a una situación cotidiana de la vida urbana en el país. El procedimiento de solución es simplemente encontrar un cateto de un triángulo rectángulo. La dificultad de este problema reside en identificar el triángulo (ver figura 3.5).

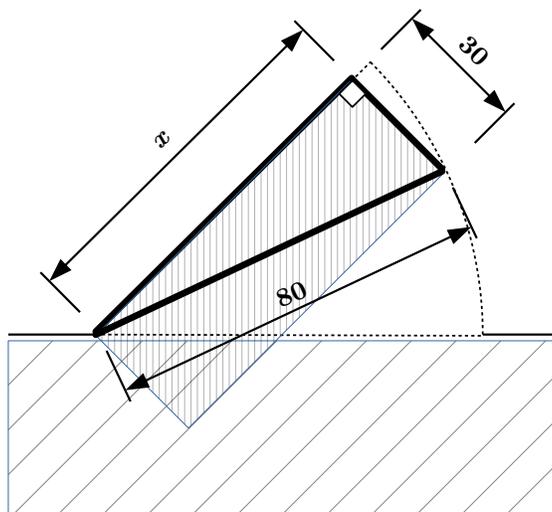


Figura 3.5.: Triángulo rectángulo que hay que identificar en el Problema 3.
Fuente: Elaboración propia.

A continuación se presentan los problemas tal como fueron redactados en el instrumento de recolección de datos de la parte cuantitativa:

PROBLEMA 1

En un país llamado Alicialandia existen nueve ciudades importantes: Almeida, Brunswick, Charlestown, Dendera, Edinburgo, Fukushima, Galium, Hormisdas e Idílicum. Se le pide que encuentre la ruta más corta para ir en carretera desde Almeida hasta Idílicum.

Se sabe que la distancia desde Almeida a Charlestown es de 2km, de Brunswick a Charlestown es de 3km, de Charlestown a Dendera es de 6km, de Brunswick a Edinburgo son 5km, de Charlestown a Edinburgo son 4km, de Dendera a Edinburgo son 2km, de Dendera a Fukushima son 5km, de Edinburgo a Fukushima son 4km, de Edinburgo a Galium son 8km, de Galium a Idílicum son 5km, de Fukushima a Hormisdas es 1km, de Fukushima a Idílicum son 6km, y finalmente de Hormisdas a Idílicum son 3km.

Las demás distancias no se han mencionado porque no hay carretera entre esas parejas de ciudades. Por ejemplo entre Brunswick y Galium no hay carretera.

PROBLEMA 2

Variante 1:

En una escuela que sirve desde séptimo grado hasta noveno grado se va cada dos meses a un viaje que incluye la visita a un museo y a una función de teatro. La directora de la escuela ha establecido como norma que a estos viajes debe ir un adulto responsable por cada 15 alumnos, y que cada viaje sea organizado por un profesor diferente (o sea que se van rotando), por lo que cada profesor pasa varios meses (incluso años) sin organizar un viaje y cuando les toca de nuevo ya no recuerdan qué pasos seguir. Desarrolle un algoritmo simple que le permita a cualquier organizador que lo lea calcular cuánto costará el viaje (esto para poder establecer cuánto deberá aportar cada estudiante).

Variante 2:

En una escuela que sirve desde séptimo grado hasta noveno grado se va cada dos meses a un viaje que incluye la visita a un museo y a una función de teatro. La directora de la escuela ha establecido como norma que a estos viajes debe ir un adulto responsable por cada 15 alumnos, y que cada viaje sea organizado por un profesor diferente (o sea que se van rotando), por lo que cada profesor pasa varios meses (incluso años) sin organizar un viaje y cuando les toca de nuevo ya no recuerdan qué pasos seguir. Desarrolle unas instrucciones simples que le permitan a cualquier organizador que las lea calcular cuánto costará el viaje (esto para poder establecer cuánto deberá aportar cada estudiante).

PROBLEMA 3

Una familia se acaba de mudar a una casa y desean convertirla en tienda. Por lo que están en proceso de acondicionar la casa (la parte de enfrente) para tal fin. Entre otras cosas, necesitan colocar una superficie plana en la puerta-balcón (que mide 80cm de ancho) para colocar y mostrar los productos a los clientes, también debe servir para que los clientes puedan colocar el dinero que pagan. Sin embargo hay un problema: la puerta se abre hacia afuera y la superficie a colocar no se puede poner por el lado de afuera ya que obstaculizaría el paso peatonal por enfrente de la casa (además de tener mayor seguridad frente a clientes desconocidos). Ver el dibujo adjunto (Figura 3.6).

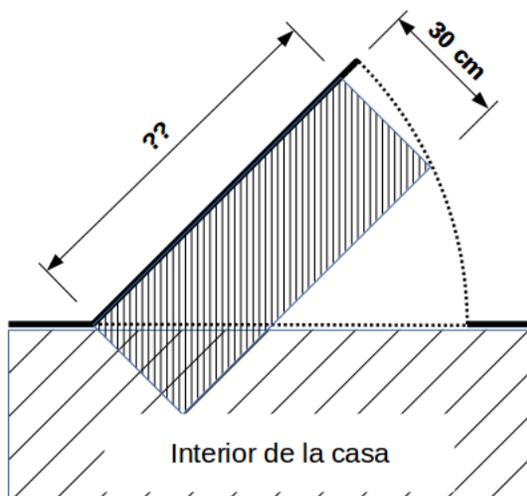


Figura 3.6.: Figura del Problema 3.

Fuente: Elaboración propia.

La familia ha decidido que 30cm son espacio suficiente para colocar los productos que la tienda venderá (botellas de aceite, sobres de café instantáneo, huevos, libras de azúcar, etc.).

Se le pide que encuentre una expresión que calcule el ancho máximo que puede tener la superficie para realizar el intercambio con los clientes y que la puerta se pueda cerrar sin problema.

PROBLEMA 4

Variante 1:

Como usted sabe, a la mayoría de los buses que ingresan a El Salvador para trabajar en el transporte colectivo les quitan los asientos originales y les ponen otros con menos espacio entre sí para aumentar su capacidad y disminuir su confort.

La Empresa «Apretados S.A. de C.V.» se dedica a ofrecer este servicio de modificación a las empresas de transporte colectivo cuando estas «traen un nuevo bus» (que ya sabemos que no sólo son usados sino que también están descartados en otros países).

Escriba *un algoritmo* para que el gerente operativo (el jefe de los obreros) realice el trabajo de calcular cuántos asientos y a qué distancia deben colocarse. Asuma que los asientos originales ya han sido retirados y los «nuevos» están en un almacén cercano ya armados y listos para poner. Puesto que la empresa se dedica a esto, tiene una provisión casi ilimitada de asientos «nuevos».

Variante 2:

Como usted sabe, a la mayoría de los buses que ingresan a El Salvador para trabajar en el transporte colectivo les quitan los asientos originales y les ponen otros con menos espacio entre sí para aumentar su capacidad y disminuir su confort.

La Empresa «Apretados S.A. de C.V.» se dedica a ofrecer este servicio de modificación a las empresas de transporte colectivo cuando estas «traen un nuevo bus» (que ya sabemos que no sólo son usados sino que también están descartados en otros países).

Escriba *unas instrucciones genéricas* para que el gerente operativo (el jefe de los obreros) realice el trabajo de calcular cuántos asientos y a qué distancia deben colocarse. Asuma que los asientos originales ya han sido retirados y los «nuevos» están en un almacén cercano ya armados y listos para poner. Puesto que la empresa se dedica a esto, tiene una provisión casi ilimitada de asientos «nuevos».

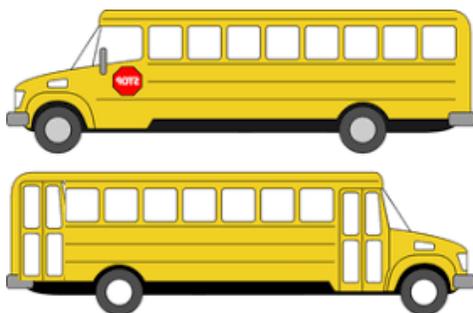


Figura 3.7.: Figura del Problema 4.

Adaptado y modificado de <https://www.educima.com/dibujo-para-colorear-autobus-escolar-i29479.html>

3.5.3. Aplicación del instrumento

La aplicación del instrumento se realizó en aulas de clase estándar durante el transcurso normal del segundo ciclo lectivo del año 2018. La resolución se realizó de forma individual y anónima. A cada alumno se le proporcionó las indicaciones, papel y 110min para completar el instrumento. Además el investigador estuvo presente y brindó consultas a aquellos que lo solicitaron durante la resolución de los problemas.

Se aplicó a los cuatro grupos, Grupo P0, Grupo P1, Grupo P2, Grupo P3, en cuatro días diferentes.

3.5.4. Confiabilidad y validez

La **confiabilidad** (o fiabilidad) del instrumento se verifica por medio del coeficiente **Alfa de Cronbach**.

El coeficiente *Alfa de Cronbach* para los datos de las puntuaciones del instrumento es de **0,88**. El cálculo se muestra en el siguiente fragmento de código en Lenguaje Estadístico R⁵ v3.6.3:

```
1 > cuanti = read.csv("datos-cuanti.csv", header=TRUE)
2 > library(psych) #install.packages("psych")
3 > respuestas = cuanti[c(10:17,19:37)] #Las columnas de la evaluación del instrumento
4 > alpha(respuestas)
5
6 Reliability analysis
7 Call: alpha(x = respuestas)
8
9   raw_alpha std.alpha G6(smc) average_r S/N ase mean sd median_r
10   0.87      0.88      0.97      0.22 7.5 0.02   36 16      0.2
11
12 lower alpha upper      95% confidence boundaries
13 0.83 0.87 0.91
14
15 Reliability if an item is dropped:
16   raw_alpha std.alpha G6(smc) average_r S/N alpha se var.r med.r
17 p1.graf      0.86      0.88      0.96      0.22 7.1   0.022 0.040 0.20
18 p1.ciu       0.87      0.88      0.96      0.22 7.5   0.020 0.041 0.21
19 p1.carr      0.86      0.88      0.96      0.21 7.1   0.022 0.040 0.20
20 p1.sol       0.87      0.88      0.97      0.23 7.6   0.020 0.039 0.21
21 p1.res       0.87      0.89      0.96      0.23 7.9   0.020 0.037 0.21
22 p3.mod       0.87      0.88      0.96      0.22 7.5   0.020 0.039 0.21
23 p3.sol       0.87      0.88      0.96      0.22 7.5   0.020 0.038 0.21
24 p3.res       0.87      0.88      0.97      0.23 7.6   0.020 0.039 0.21
25 p2.input     0.86      0.88      0.96      0.21 7.0   0.022 0.040 0.20
26 p2.output    0.86      0.88      0.96      0.21 7.1   0.022 0.038 0.20
27 p2.idsub     0.86      0.88      0.96      0.21 7.0   0.022 0.038 0.20
28 p2.solsub    0.86      0.87      0.96      0.21 6.9   0.022 0.038 0.20
29 p2.intsub    0.86      0.87      0.96      0.21 6.9   0.022 0.038 0.20
30 p2.norm      0.87      0.88      0.96      0.22 7.4   0.021 0.040 0.21
31 p2.extre     0.86      0.88      0.96      0.21 7.0   0.022 0.040 0.20
32 p2.flujo     0.86      0.88      0.96      0.22 7.2   0.021 0.040 0.20
33 p2.sin       0.87      0.88      0.96      0.22 7.5   0.021 0.039 0.21
34 p4.input     0.86      0.88      0.96      0.22 7.2   0.021 0.039 0.20
35 p4.dibu      0.86      0.88      0.96      0.22 7.3   0.021 0.041 0.20
36 p4.output    0.86      0.87      0.96      0.21 6.9   0.022 0.039 0.20
37 p4.idsub     0.86      0.87      0.96      0.21 6.9   0.022 0.037 0.20
```

⁵<https://www.r-project.org/>

```

38 p4.solsub      0.86      0.87      0.96      0.21 7.0      0.021 0.038 0.20
39 p4.intsub      0.86      0.88      0.96      0.21 7.1      0.021 0.037 0.20
40 p4.norm        0.87      0.88      0.96      0.22 7.3      0.021 0.039 0.21
41 p4.extre       0.87      0.88      0.96      0.22 7.4      0.021 0.039 0.21
42 p4.flujo       0.86      0.88      0.96      0.21 7.1      0.021 0.039 0.20
43 p4.sin         0.86      0.88      0.96      0.21 7.0      0.021 0.038 0.20
44
45 Item statistics
46      n raw.r std.r r.cor r.drop mean sd
47 p1.graf      80 0.57 0.54 0.54 0.51 59.1 45.3
48 p1.ciu       80 0.42 0.34 0.32 0.30 58.3 39.4
49 p1.carr      80 0.63 0.56 0.55 0.56 55.1 43.6
50 p1.sol       80 0.40 0.29 0.26 0.28 60.4 37.0
51 p1.res       80 0.18 0.14 0.11 0.10 65.5 34.3
52 p3.mod       64 0.41 0.33 0.33 0.31 38.0 41.2
53 p3.sol       64 0.41 0.32 0.32 0.29 38.4 39.5
54 p3.res       64 0.34 0.28 0.25 0.23 39.0 41.9
55 p2.input     77 0.63 0.62 0.60 0.56 53.4 38.6
56 p2.output    77 0.61 0.59 0.58 0.55 50.6 36.3
57 p2.idsub     77 0.64 0.61 0.61 0.59 58.2 34.6
58 p2.solsub    77 0.68 0.70 0.70 0.66 43.5 34.6
59 p2.intsub    77 0.69 0.66 0.66 0.62 50.3 40.9
60 p2.norm      77 0.33 0.41 0.40 0.33 9.6 5.5
61 p2.extre     77 0.59 0.62 0.61 0.56 27.6 40.0
62 p2.flujo     77 0.47 0.49 0.48 0.44 33.3 25.3
63 p2.sin       77 0.28 0.33 0.31 0.23 18.4 23.8
64 p4.input     71 0.51 0.53 0.52 0.43 38.0 35.5
65 p4.dibu      71 0.48 0.45 0.43 0.40 32.0 36.9
66 p4.output    71 0.62 0.67 0.66 0.57 28.9 28.8
67 p4.idsub     71 0.62 0.67 0.67 0.57 28.6 28.1
68 p4.solsub    71 0.58 0.62 0.62 0.53 20.8 26.3
69 p4.intsub    71 0.51 0.58 0.58 0.45 14.3 29.5
70 p4.norm      71 0.34 0.47 0.47 0.33 8.0 4.0
71 p4.extre     71 0.30 0.42 0.41 0.28 4.1 1.9
72 p4.flujo     71 0.53 0.57 0.56 0.47 21.0 24.4
73 p4.sin       71 0.53 0.61 0.61 0.49 17.1 23.4
74
75 Non missing response frequency for each item
76      0 5 10 miss
77 p4.norm 0.20 0.00 0.8 0.11
78 p4.extre 0.18 0.82 0.0 0.11

```

Las columnas evaluadas se corresponden con los valores de las sub-variables descritas en los cuadros 3.6 en la página siguiente y 3.7 en la página 76.

La **validez de contenido** del instrumento está sustentada en que permite indagar el constructo «*Pensamiento Algorítmico*» según la definición operacional descrita en la subsección 3.3 en la página 58 basada en las definiciones operacionales descritas en la subsección 2.3.3.

3.6. Operacionalización de las variables de la parte cuantitativa

En base a la Definición Operacional de *Pensamiento Algorítmico de un Principiante* asumida en la subsección 3.3, y la correspondiente rúbrica de evaluación, se muestra en el cuadro 3.6 en la página siguiente la operacionalización concreta de la variable PAP de la parte cuantitativa (ver cuadro 3.5 en la página 67 para mayor información sobre las variables de la parte cuantitativa).

Cuadro 3.6.: Operacionalización de la Variable PAP.

Dimensiones	Indicadores	Sub-variable	Ítem del cuestionario
Análisis del problema 2	Identifica las Entradas.	p2-input	Problema 2
	Identifica las Salidas.	p2-output	
Descomposición del problema 2 en sub-problemas	Identifica los subproblemas que componen el problema.	p2-idsub	
Construcción de un algoritmo correcto que resuelve el problema 2	Implementa una solución a los subproblemas identificados.	p2-solsub	
	Integra las diferentes soluciones parciales para resolver el problema completo.	p2-intsub	
Identificación de los casos normales y especiales del problema 2	Identifica los casos normales.	p2-norm	
	Identifica los casos especiales o extremos.	p2-extre	
Construcción de una representación formal del algoritmo que resuelve el problema 2	Representa el flujo de ejecución (en el caso de usar Pseudocódigo, esto incluye la indentación y el anidamiento de las estructuras de control como si-entonces-sino, hacer-mientras, hacer-para, etc.).	p2-flujo	
	Utiliza la simbología y las reglas sintácticas apropiadas (como los rombos para las bifurcaciones condicionales en diagramas de flujo, o «palabras reservadas» en pseudocódigo).	p2-sin	

Análisis del problema 4	Identifica las Entradas.	p4-input	Problema 4
	Realiza un esquema gráfico.	p4-dibu	
	Identifica las Salidas.	p4-output	
Descomposición del problema 4 en sub-problemas	Identifica los subproblemas que componen el problema.	p4-idsub	
Construcción de un algoritmo correcto que resuelve el problema 4	Implementa una solución a los subproblemas identificados.	p4-solsub	
	Integra las diferentes soluciones parciales para resolver el problema completo.	p4-intsub	
Identificación de los casos normales y especiales del problema 4	Identifica los casos normales.	p4-norm	
	Identifica los casos especiales o extremos.	p4-extre	
Construcción de una representación formal del algoritmo que resuelve el problema 4	Representa el flujo de ejecución (en el caso de usar Pseudocódigo, esto incluye la indentación y el anidamiento de las estructuras de control como si-entonces-sino, hacer-mientras, hacer-para, etc.)	p4-flujo	
	Utiliza la simbología y las reglas sintácticas apropiadas (como los rombos para las bifurcaciones condicionales en diagramas de flujo, o «palabras reservadas» en pseudocódigo).	p4-sin	

Fuente: Elaboración propia.

No se espera que haya corridas a mano, por lo que se han omitido las sub-variables pa-exe y pa-err definidas en el cuadro 3.3 en la página 60.

En el cuadro 3.7 se muestra la operacionalización concreta de la variable RP de la parte cuantitativa.

Cuadro 3.7.: Operacionalización de la Variable RP.

Dimen- siones	Sub-dimensiones	Indicadores	Sub- variable	Ítem del cuestio- nario
Habilidad de Resolver el Problema 1	Representación gráfica del problema 1	Representar gráficamente la situación	p1-graf	Problema 1
	Modelamiento de las Ciudades del problema 1	Representar las ciudades por medio de una sólo letra	p1-ciu	
	Modelamiento de las Carreteras (camino) del problema 1	Representar las aristas en un orden que facilite la resolución del problema	p1-carr	
	Resolución del problema 1	Resolver el problema dada alguna representación gráfica	p1-sol	
	Presentación de la solución del problema 1	Expresar la solución una vez encontrada	p1-res	
Habilidad de Resolver el Problema 3	Modelamiento del problema 3	Identificar el triángulo rectángulo en el dibujo	p3-mod	Problema 3
	Resolución del problema 3	Resolver el problema habiendo identificado el triángulo rectángulo	p3-sol	
	Presentación de la solución del problema 3	Expresar la solución una vez encontrada	p3-res	

Fuente: Elaboración propia.

3.6.1. Detalle de la Rúbrica de medición de la variable Pensamiento Algorítmico de un Principiante

A continuación se presenta el detalle de la rúbrica para la medición de las sub-variables de la variable PAP para esta investigación:

p2-input

A partir de aquí, se describen las sub-variables de la variable PA asociadas al Problema 2 (ver el enunciado en la página 70).

VARIABLES POR DEFECTO DEL PROBLEMA 2:

- Cantidad de alumnos, `cantAlumnos`.
- Costo de viaje por persona (suma del costo de transporte por persona, costo de entrada al museo y costo de entrada al teatro), `costoPorPersona`, suponiendo que el costo sea igual para cualquier persona.
También puede tratarse de tres variables separadas, `costoTransPorPersona`, `costoMuseoPorPersona` y `costoTeatroPorPersona`.

Niveles:

1. No identifica ninguna variable de entrada.
2. Define/utiliza otra(s) variable(s) como variables de entrada.
3. Define/utiliza alguna(s) variable(s) de entrada correcta con otra(s) incorrecta(s).
4. Define/utiliza alguna(s) variable(s) de entrada correcta(s) pero no todas.
5. Define/utiliza correctamente las variables de entrada.

p2-output

VARIABLES POR DEFECTO:

- Cantidad de adultos acompañantes, `adultosAcomp`.
- Costo total del viaje, `costoTotal`.

Otra posibilidad de variables de salida, en caso de que el resolutor asuma que el costo de los adultos acompañantes deberá ser sufragado por los estudiantes:

- Cantidad de adultos acompañantes, `adultosAcomp`.
- Costo total del viaje, `costoTotal`.
- Costo por alumno, `costoPorAlumno`.

Niveles:

1. No identifica ninguna variable de salida.
2. Define/utiliza otra(s) variable(s) como variables de salida.

3. Define/utiliza alguna(s) variable(s) de salida correcta con otra(s) incorrecta(s).
4. Define/utiliza alguna(s) variable(s) de salida correcta(s) pero no todas.
5. Define/utiliza correctamente las variables de salida.

p2-idsub

Subproblemas:

- Calcular la cantidad de adultos acompañantes necesarios para el viaje a partir de la cantidad de alumnos.
- Calcular el costo del alquiler del transporte en caso de asumir que se alquilará transporte privado, e incorporarlo al costo por persona junto con las entradas al museo y al teatro, o bien, incorporarlo al costo total por aparte.
- Calcular el costo total del viaje a partir de la cantidad de alumnos y la cantidad de adultos acompañantes.
- Calcular el costo de viaje por alumno si se asume que el costo de los adultos acompañantes deberá ser sufragado por los estudiantes.

Niveles:

1. No identifica ninguno de los subproblemas.
2. Identifica sólo uno de los subproblemas.
3. Identifica sólo dos de los subproblemas.
4. Identifica tres o cuatro de los subproblemas según la variante asumida.

p2-solsub y p2-intsub

Sean `cantAlumnos` y `costoPorPersona` los valores de las variables de entrada básicas para todas las interpretaciones posibles del problema.

Sean `adultosAcomp`, `costoTotal` y `costoPorAlumno` las variables de salida.

Soluciones esperadas:

- `adultosAcomp` será `cantAlumnos/15` si `cantAlumnos` es divisible por 15, pero sino, `adultosAcomp` será el mayor entero de `cantAlumnos/15` (o lo que es lo mismo, la parte entera de `cantAlumnos` más uno). Este es el problema más inmediato y además necesario para resolver los otros dos (o tres, según la variante que se asuma). Para efectos de esta rúbrica, se considerará correctamente resuelto este sub-problema incluso si se ignora la divisibilidad por 15, dejando dicha evaluación en la variable `p2-extre` (ver página 81).
- Si se asume con más detalle el alquiler del transporte privado, entonces hay algunas variables intermedias:

- La cantidad de vehículos necesarios, que pueden ser varios, y se calcula de la siguiente forma:
 cantVehículos será, en general, el mayor entero de $(\text{cantAlumnos} + \text{adultosAcomp}) / \text{CAPACIDAD_VEHÍCULOS}$, a menos que tal cociente sea entero, en cuyo caso, cantVehículos será simplemente este cociente.
- El costo total de este alquiler de transporte privado, $\text{costoTotalTransporte}$, que es la suma del alquiler de cada vehículo. Si suponemos que son iguales todos, entonces es cantVehículos multiplicado por el costo de alquiler individual (COSTO_VEHÍCULO).
- El costo de transporte por persona, $\text{costoTransPorPersona}$, que se calcula como el costo total del alquiler de todos los vehículos (que pueden ser varios) dividido por $(\text{cantAlumnos} + \text{adultosAcomp})$.
De esta forma, costoPorPersona es $\text{costoTransPorPersona} + \text{costoMuseoPorPersona} + \text{costoTeatroPorPersona}$.

Siguiendo esta vía, deben agregarse $\text{CAPACIDAD_VEHÍCULOS}$ y COSTO_VEHÍCULO como variables de entrada.

- costoTotal es $\text{costoPorPersona} * (\text{cantAlumnos} + \text{adultosAcomp})$.

Si se calcula el costo del transporte total por separado, entonces costoTotal es $(\text{costoMuseoPorPersona} + \text{costoTeatroPorPersona}) * (\text{cantAlumnos} + \text{adultosAcomp}) + \text{costoTotalTransporte}$.

- Si se asume que cada adulto acompañante pagará sus costos, entonces costoPorAlumno es igual a costoPorPersona . y no se realizará ningún cálculo, e incluso podría no considerarse como variable de salida.
Si se asume que los costos de los adultos acompañantes serán sufragados por los alumnos, entonces costoPorAlumno es $\text{costoTotal} / \text{cantAlumnos}$.

Las principales combinaciones previstas se resumen en el cuadro 3.8.

Podemos llamar a estas variantes, A1, A2, A3, A4, B1, B2, B3 y B4 respectivamente.

Surgen además otras variantes, por ejemplo si se asume que el costo de las entradas al museo y al teatro son diferentes para estudiantes (niños) y adultos. Así se harían otras ocho variantes. También podría no asumirse que todos los vehículos de transporte tienen la misma capacidad, por lo que surgiría la necesidad de una lista o arreglo de valores.

Niveles para p2-solsub:

1. No intenta resolver ninguno de los subproblemas.
2. Intenta, pero no resuelve correctamente ninguno de los subproblemas.
3. Resuelve correctamente uno de los subproblemas basado en sus suposiciones.
4. Resuelve correctamente dos de los subproblemas basado en sus suposiciones.
5. Resuelve correctamente al menos tres de los subproblemas basado en sus suposiciones.

Cuadro 3.8.: Variantes básicas del problema 2.

	A	B
1	Omitir el costo del transporte y sólo considerar el costo de las entradas, y asumir que cada adulto pagará sus costos.	Omitir el costo del transporte y sólo considerar el costo de las entradas, y asumir que el costo de los adultos será pagado por los alumnos.
2	Asumir un costo de viaje por persona que incluye el transporte y las entradas, y que cada adulto pagará sus costos.	Asumir un costo de viaje por persona que incluye el transporte y las entradas, y que el costo de los adultos será pagado por los alumnos.
3	Calcular el costo de transporte por persona e incorporarlo al costo por persona, y asumir que cada adulto pagará sus costos.	Calcular el costo de transporte por persona e incorporarlo al costo por persona, y asumir que el costo de los adultos será pagado por los alumnos.
4	Calcular el costo total de transporte e incorporarlo al costo total, y asumir que cada adulto pagará sus costos.	Calcular el costo total de transporte e incorporarlo al costo total, y asumir que el costo de los adultos será pagado por los alumnos.

Fuente: Elaboración propia.

Para este caso, «resolver correctamente» puede ocurrir sin considerar los casos extremos. Esto se justifica porque eso está considerado en la subvariable **p2-extre** (ver página 81).

Niveles para **p2-intsub**:

1. No intenta resolver ninguno de los subproblemas.
2. Resuelve o intenta resolver sólo uno de los subproblemas.
3. Resuelve o intenta resolver dos o más subproblemas de forma aislada o en orden incorrecto.
Por ejemplo la cantidad de adultos acompañantes debe resolverse primero, ya que es variable de entrada para los demás subproblemas.
4. Resuelve o intenta resolver dos subproblemas en el orden correcto.
5. Resuelve o intenta resolver tres subproblemas y uno de ellos no está en el orden correcto.
Por ejemplo, en la variante B2 (ver cuadro 3.8), el orden correcto de solución sería $\text{adultosAcomp} \rightarrow \text{costoTotal} \rightarrow \text{costoPorAlumno}$.
6. Resuelve o intenta resolver al menos tres subproblemas en el orden correcto según la variante asumida.

p2-norm

Para este problema, los casos normales son simplemente que todos los parámetros y las variables de entrada son números positivos. La cantidad de alumnos debe ser un número entero y los costos deben ser números reales, todos positivos.

Niveles:

1. No intenta resolver el problema o no identifica variables de entrada.
2. Resuelve o intenta resolver pero no declara en ningún momento que las variables de entrada deben tener valores positivos y del tipo correcto.
Por ejemplo, la cantidad de alumnos y la capacidad de los vehículos de transporte, `CAPACIDAD_VEHÍCULOS`, en caso de usar, deben ser enteros, mientras que todos los costos deben ser reales.
3. Resuelve o intenta resolver pero no declara para todas las variables de entrada que deben ser positivas y del tipo correcto.
4. Resuelve o intenta resolver y declara explícitamente para todas las variables de entrada que deben ser positivas, o bien, del tipo correcto, pero no ambas condiciones.
5. Resuelve o intenta resolver y declara explícitamente para todas las variables de entrada que deben ser positivas y del tipo correcto.
Esto puede explicitarse en el algoritmo mismo si se validan el tipo y signo de las variables de entrada (con estructuras `if`).

p2-extre

Para este problema, el caso «extremo» radica en que la fórmula para calcular la cantidad de adultos acompañantes, `adultosAcomp`, no es una sola, ya que depende de que si `cantAlumnos` es divisible por 15 (tal como se explicó en la página 78). Lo mismo ocurriría en las variantes A3, B3, A4 y B4 (ver cuadro 3.8) para calcular `cantVehículos` si se considera la necesidad de calcular la cantidad de vehículos, en base a `CAPACIDAD_VEHÍCULOS` (que básicamente es cantidad de asientos por vehículo), para saber el costo del transporte, total y/o por persona (ver página 79).

Niveles para las Variantes A1, B1, A2 y B2:

1. No intenta resolver el problema de calcular el número de adultos acompañantes.
2. Intenta resolver el problema de calcular la cantidad de adultos, `adultosAcomp`, pero sin diferenciar si es divisible por 15 o no.
Por ejemplo si se define simplemente que `adultosAcomp := cantAlumnos/15`, o bien, `adultosAcomp := (cantAlumnos/15) + 1`.
3. Intenta pero no logra resolver el problema de calcular la cantidad de adultos, `adultosAcomp`, aunque considera la divisibilidad por 15.
4. Expresa la solución correcta al problema de calcular la cantidad de adultos sin usar simbología matemática o sintaxis válida.

5. Resuelve correctamente el problema de calcular la cantidad de adultos.

Niveles para las Variantes A3, B3, A4 y B4:

1. No intenta resolver el problema de calcular el número de adultos acompañantes ni de vehículos.
2. Intenta resolver el problema de calcular la cantidad de adultos, `adultosAcomp`, y/o `cantVehículos`, pero sin diferenciar si son divisibles o no (es decir, ignora si el resultado es entero o no).
3. Resuelve correctamente el problema de calcular la cantidad de adultos, pero calcula `cantVehículos` sin considerar la divisibilidad por `CAPACIDAD_VEHÍCULOS`.
4. Expresa la solución correcta al problema de calcular la cantidad de adultos sin usar simbología matemática o sintaxis válida.
5. Resuelve correctamente el problema de calcular la cantidad de adultos, y también calcula correctamente `cantVehículos`.

p2-flujo

En esta investigación se utiliza el paradigma de Programación Estructurada⁶ como parámetro de medición ya que es el que se utiliza en los cursos de los alumnos que componen la población.

Niveles:

1. No intenta resolver el problema.
2. Intenta resolver pero no respeta la secuenciación y no utiliza ninguna estructura de control condicional.
3. Intenta resolver respetando la secuenciación pero no utiliza ninguna estructura de control condicional.
En el caso de este problema sólo es necesaria la secuenciación y la estructura si-entonces-sino.
4. Resuelve o intenta resolver utilizando estructuras de control pero sin respetar la indentación en caso de usar pseudocódigo, o sin dejar claro el flujo de ejecución en caso de usar un diagrama de flujo.
5. Resuelve o intenta resolver respetando la secuenciación y utilizando estructuras de control condicional, con su indentación apropiada en caso de usar pseudocódigo, o con su flujo de ejecución claro en caso de usar un diagrama de flujo.

p2-sin

Los componente sintácticos básicos de todo algoritmo son:

⁶https://es.wikipedia.org/wiki/Programación_estructurada

- Marca del punto de inicio
- Solicitud de los datos de entrada
- Procesamiento de la respuesta
- Impresión de los datos de salida (la o las respuestas)
- Marca del punto de finalización

Para este problema, el procesamiento de la respuesta consiste de una o dos estructuras de control condicional (dependiendo de la variante asumida (ver cuadro 3.8), además de la asignación de algunas variables.

La sintaxis de las asignaciones es bastante simple (`variable := nuevo_valor`), por lo que se omite considerarlas como elemento a evaluar.

Niveles:

1. No intenta resolver el problema.
2. Intenta resolver pero no utiliza simbología propia de los diagramas de flujo, o no utiliza palabras reservadas mínimas, como Si, Entonces, Sino, FinSi, Inicio, Fin, Mostrar, Imprimir, etc.
3. Intenta resolver usando simbología parcialmente correcta para alguna estructura de control condicional.
4. Resuelve o intenta resolver usando simbología correcta para las estructuras de control condicional, pero no incluye completamente cosas como el inicio, el fin, la petición de datos de entrada y la impresión de la respuesta.
5. Resuelve o intenta resolver usando las estructuras de control condicional correspondientes y utilizando la simbología de inicio, fin, petición de datos e impresión de la respuesta.

p4-input

A partir de aquí, se describen las sub-variables de la variable PA asociadas al Problema 4 (vér el enunciado en la página 71).

VARIABLES POR DEFECTO DEL PROBLEMA 4:

- Longitud del espacio para poner asientos «nuevos» del lado derecho (donde van las puertas de entrada y salida), `largoDer`; y la longitud del espacio para poner asientos «nuevos» del lado izquierdo (donde va el motorista), `largoIzq`. Ver figura 3.8 en la página siguiente.
- Profundidad de los asientos, `profundAsiento`, y el espacio mínimo entre los asientos, `espMinEntreAsientos`. Ver figura 3.9 en la página siguiente
Estas dos variables de entrada pueden tener al menos dos alternativas: La Variante A, como en la figura 3.9a, y la Variante B, como en la figura 3.9b.
La variante A es más precisa con el uso del espacio dentro del bus, pero la variante B es más fácil de implementar por un grupo de obreros, ya que se basa en las posiciones de los tornillos o pernos que deben colocarse para fijar los asientos.

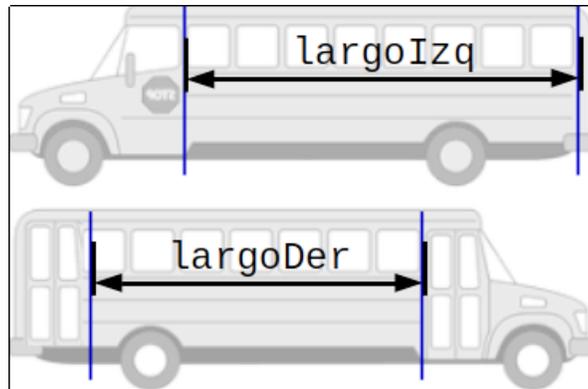
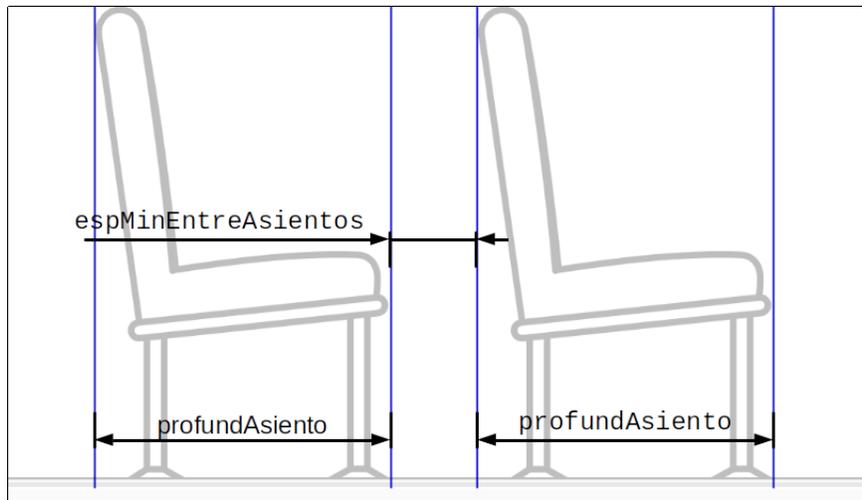
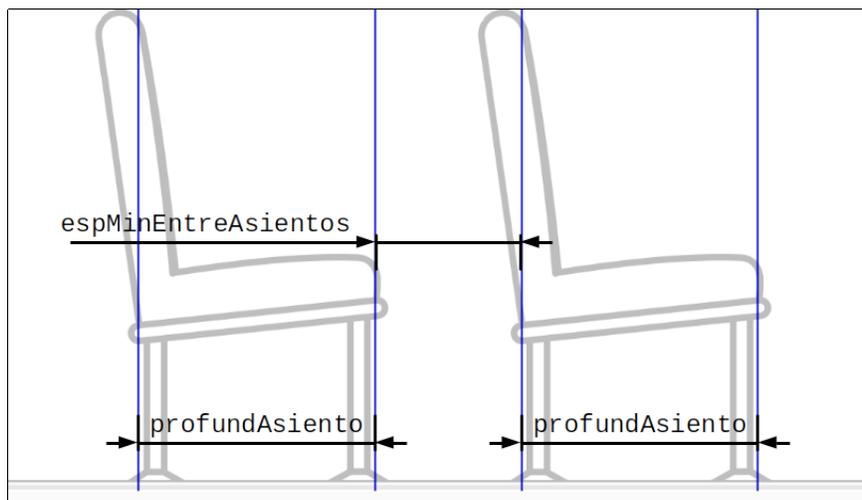


Figura 3.8.: Vista lateral del bus del Problema 4.

Adaptado y modificado de <https://www.educima.com/dibujo-para-colorear-autobus-escolar-i29479.html>



(a) Alternativa A



(b) Alternativa B

Figura 3.9.: Vista lateral de los asientos del Problema 4.

Adaptado y modificado de la figura 1 de la sección S6.4 de <https://www.law.cornell.edu/cfr/text/49/571.217>

Otra posible combinación de variables de entrada está compuesta por:

- Longitud del espacio para poner asientos «nuevos» del lado derecho (donde van las puertas de entrada y salida), `largoDer`; y la longitud del espacio para poner asientos «nuevos» del lado izquierdo (donde va el motorista), `largoIzq`. Ver figura 3.8. Igual que en la combinación anterior.
- La profundidad de los asientos, `profundAsiento`, y la cantidad de asientos que se desean poner de cada lado del bus, `cantAsientosIzq` y `cantAsientosDer`.

Llamamos a esta, Variante C.

Y tal como en la Variante B, en la Variante C la profundidad del asiento puede representar alternativamente la distancia entre los agujeros externos de las patas de los asientos que son para atornillarlos. Se podría llamar a esta, Variante D, sin embargo su tratamiento no representa diferencias significativas respecto del tratamiento de C.

Niveles:

1. No identifica ninguna de las variables de entrada.
2. Define/utiliza otra(s) variable(s) como variables de entrada.
3. Define/utiliza las variables mezclando el lado izquierdo con el derecho.
4. Define/utiliza alguna(s) variable(s) de entrada correcta con otra(s) incorrecta(s).
Para este problema, sería incorrecto asumir `espMinEntreAsientos`, junto con `cantAsientosIzq` y `cantAsientosDer` como variables de entrada simultáneamente.
5. Define/utiliza alguna(s) variable(s) de entrada correcta(s) pero no todas.
6. Define/utiliza todas las variables de entrada junto con otra(s) irrelevante(s) Define/utiliza correctamente las variables de entrada.

p4-dibu

Este problema amerita la realización de un esquema que permita identificar tanto las variables de entrada y salida como los cálculos geométricos que se deben realizar. Se esperan diagramas simples como los de las figuras 3.8 y/o 3.9.

Niveles:

- No realiza ningún esquema o dibujo.
- Realiza un esquema o dibujo que carece de utilidad para identificar las variables importantes.
- Realiza un esquema o dibujo con potencial para identificar las variables importantes pero sin estar identificadas.
- Realiza un esquema o dibujo donde se pueden identificar algunas variables importantes pero no todas las que permite dicha representación.
- Realiza un dibujo o esquema donde se pueden identificar las variables importantes.

p4-output

VARIABLES POR DEFECTO (PARA LAS VARIANTES A Y B):

- La cantidad de asientos que se pueden poner de cada lado del bus, `cantAsientosIzq` y `cantAsientosDer`.
- El espacio ajustado entre los asientos, del lado izquierdo del bus, `espEntreAsientosIzq`, y del lado derecho del bus, `espEntreAsientosDer`.

PARA LA VARIANTE C:

- El espacio entre los asientos, del lado izquierdo del bus, `espEntreAsientosIzq`, y del lado derecho del bus, `espEntreAsientosDer`.

NIVELES:

1. No identifica ninguna de las variables de salida.
2. Define/utiliza otra(s) variable(s) como variables de salida.
3. Define/utiliza las variables mezclando el lado izquierdo con el derecho.
4. Define/utiliza alguna(s) variable(s) de salida correcta con otra(s) incorrecta(s).
5. Define/utiliza alguna(s) variable(s) de salida correcta(s) pero no todas
6. Define/utiliza todas las variables de salida junto con otra(s) irrelevante(s).
7. Define/utiliza correctamente las variables de salida.

p4-idsub

SUBPROBLEMAS DE LAS VARIANTES A Y B:

- Calcular la cantidad de asientos que se pueden poner del lado izquierdo del bus, `cantAsientosIzq`, y
- Calcular la cantidad de asientos que se pueden poner del lado derecho del bus, `cantAsientosDer`.
- Calcular el espacio ajustado entre los asientos del lado izquierdo del bus, `espEntreAsientosIzq`, y
- Calcular el espacio ajustado entre los asientos del lado derecho del bus, `espEntreAsientosDer`.

SUBPROBLEMAS DE LA VARIANTE C:

- Calcular el espacio entre los asientos del lado izquierdo del bus, `espEntreAsientosIzq`.
- Considerar que la cantidad de asientos solicitados del lado izquierdo, `cantAsientosIzq`, no supere el espacio disponible `largoIzq`.
- Calcular el espacio entre los asientos del lado derecho del bus, `espEntreAsientosDer`.
- Considerar que la cantidad de asientos solicitados del lado derecho, `cantAsientosDer`, no supere el espacio disponible `largoDer`.

Niveles:

1. No identifica ninguno de los subproblemas.
2. Intenta resolver sin separar las variables de entrada del lado izquierdo de las del lado derecho.
3. Identifica sólo dos de los subproblemas.
4. Identifica los cuatro subproblemas.

p4-solsub

Soluciones esperadas para las Variantes A y B⁷:

- La cantidad de asientos que se pueden poner del lado izquierdo del bus, `cantAsientosIzq` es el menor entero de $\text{largoIzq} / (\text{espMinEntreAsientos} + \text{profundAsiento})$. Esta cantidad debe ser mayor que cero.
- La cantidad de asientos que se pueden poner del lado derecho del bus, `cantAsientosDer` es el menor entero de $\text{largoDer} / (\text{espMinEntreAsientos} + \text{profundAsiento})$. Esta cantidad debe ser mayor que cero.
- El espacio ajustado de los asientos del lado izquierdo, `espEntreAsientosIzq` se puede calcular así:
 - $\text{cantAsientosIzq} * (\text{espMinEntreAsientos} + \text{profundAsiento})$ es el espacio mínimo a ocupar por los asientos del lado izquierdo.
 - Luego, $\text{largoIzq} - \text{cantAsientosIzq} * (\text{espMinEntreAsientos} + \text{profundAsiento})$ es el espacio «sobrante», sea `espSobranteIzq`.
 - Ahora se debe distribuir el espacio «sobrante» entre los asientos que sí caben completos:
 $\text{espEntreAsientosIzq}$ es $\text{espMinEntreAsientos} + (\text{espSobranteIzq} / \text{cantAsientosIzq})$.
- El cálculo de `espEntreAsientosDer` es análogo:
 - $\text{cantAsientosDer} * (\text{espMinEntreAsientos} + \text{profundAsiento})$ es el espacio mínimo a ocupar por los asientos del lado derecho.
 - Luego, $\text{largoDer} - \text{cantAsientosDer} * (\text{espMinEntreAsientos} + \text{profundAsiento})$ es el espacio «sobrante», sea `espSobranteDer`.
 - Ahora se debe distribuir el espacio «sobrante» entre los asientos que sí caben completos:
 $\text{espEntreAsientosDer}$ es $\text{espMinEntreAsientos} + (\text{espSobranteDer} / \text{cantAsientosDer})$.

Soluciones esperadas para la Variante C:

⁷Para la Variante B, esta solución no es exacta, ya que hace falta considerar el espacio necesario atrás de las patas traseras del último asiento.

- El espacio de los asientos del lado izquierdo, $\text{espEntreAsientosIzq}$ se puede calcular así:
 - $\text{cantAsientosIzq} * \text{profundAsiento}$ es el espacio necesario para los asientos del lado izquierdo.
 - Luego, $\text{largoIzq} - \text{cantAsientosIzq} * \text{profundAsiento}$ es el espacio «sobrante» para que los usuarios puedan entrar a los asientos del lado izquierdo, sobranteIzq .
 - En este punto debe considerarse la posibilidad de que sobranteIzq sea no positivo, lo que significa que la cantidad solicitada de asientos simplemente no cabe en ese lado del bus, y se trata entonces de una condición de error y el algoritmo termina aquí.
 - Ahora se debe distribuir el espacio «sobrante» entre los asientos que se desean: $\text{espEntreAsientosIzq}$ es $\text{sobranteIzq} / \text{cantAsientosIzq}$.
- De forma análoga, el espacio de los asientos del lado derecho, $\text{espEntreAsientosDer}$ se puede calcular así:
 - $\text{cantAsientosDer} * \text{profundAsiento}$ es el espacio necesario para los asientos del lado derecho.
 - Luego, $\text{largoDer} - \text{cantAsientosDer} * \text{profundAsiento}$ es el espacio «sobrante» para que los usuarios puedan entrar a los asientos del lado derecho, sobranteDer .
 - En este punto debe considerarse la posibilidad de que sobranteDer sea no positivo, lo que significa que la cantidad solicitada de asientos simplemente no cabe en ese lado del bus, y se trata entonces de una condición de error y el algoritmo termina aquí.
 - Ahora se debe distribuir el espacio «sobrante» entre los asientos que se desean: $\text{espEntreAsientosDer}$ es $\text{sobranteDer} / \text{cantAsientosDer}$.

Niveles para las Variantes A y B:

1. No intenta resolver ninguno de los subproblemas.
2. Intenta, pero no resuelve correctamente ninguno de los subproblemas.
3. Resuelve correctamente el subproblema de la cantidad de asientos suponiendo que ambos lados son iguales.
4. Resuelve correctamente el subproblema de la cantidad de asientos de cada lado pero no recalcula ni distribuye el espacio sobrante.
5. Resuelve correctamente el subproblema de la cantidad de asientos de cada lado, además recalcula correctamente el espacio sobrante, pero no lo distribuye correctamente.
6. Resuelve correctamente el subproblema de la cantidad de asientos de cada lado y además recalcula y distribuye correctamente el espacio sobrante.

Niveles para la Variante C:

1. No intenta resolver ninguno de los subproblemas.
2. Intenta, pero no resuelve correctamente ninguno de los subproblemas.
3. Intenta resolver pero no calcula ni distribuye el espacio para entrar a los asientos.
4. Resuelve correctamente el cálculo y distribución del espacio para entrar a los asientos.

p4-intsub

Niveles:

1. No intenta resolver ninguno de los subproblemas.
2. Resuelve o intenta resolver sólo dos de los subproblemas.
3. Resuelve o intenta resolver los cuatro subproblemas de forma aislada.
4. Resuelve o intenta resolver los cuatro subproblemas en orden incorrecto.
Por ejemplo, en las Variantes A y B se debe resolver primero la cantidad de asientos de cada lado. En la Variante C se debe resolver primero la cantidad de espacio disponible para entrar a los asientos.
5. Resuelve o intenta resolver los cuatro subproblemas en el orden correcto según la variante asumida.

p4-norm

Para este problema, los casos normales son simplemente que todos los parámetros y las variables de entrada son números positivos. Todos deben ser números reales, a excepción de `cantAsientosIzq` y `cantAsientosDer` en la Variante C que deben ser enteros.

Niveles:

1. No intenta resolver el problema o no identifica variables de entrada.
2. Resuelve o intenta resolver pero no declara en ningún momento que las variables de entrada deben tener valores positivos y del tipo correcto.
3. Resuelve o intenta resolver pero no declara para todas las variables de entrada que deben ser positivas y del tipo correcto.
4. Resuelve o intenta resolver y declara explícitamente para todas las variables de entrada que deben ser positivas, o bien, del tipo correcto, pero no ambas condiciones.
5. Resuelve o intenta resolver y declara explícitamente para todas las variables de entrada que deben ser positivas y del tipo correcto.

p4-extre

Para este problema, el caso «extremo» para las Variantes A y B, es que la cantidad de asientos de cada lado debe ser al menos uno, ya que el espacio entre cada asiento resultaría en el espacio total de ese lado del bus, con cero asientos, lo cual no tiene sentido.

Para la Variante C, el caso «extremo» ocurre cuando el espacio sobrante es no positivo, lo que significa que esa cantidad de asientos no cabe de ese lado del bus y no se puede proseguir con el algoritmo.

Niveles:

1. No intenta resolver el problema.
2. Resuelve o intenta resolver el problema pero en ningún momento considera el caso extremo de la variante asumida.
3. Identifica pero no trata de resolver el caso extremo de la variante asumida.
4. Intenta pero no logra resolver el problema del caso extremo de la variante asumida.
5. Resuelve o intenta resolver el problema considerando el caso extremo de la variante asumida.

p4-flujo

Idéntico a p2-flujo. Ver la página 82.

p4-sin

Idéntico a p2-sin. Ver la página 82.

Resumen de la rúbrica

Tal como antes, se han seguido los pasos del diseño de rúbricas de puntuación de Mertler (2000) y se ha utilizado el estilo de la plantilla presentada por Cebrián de la Serna y Monedero Moya (2014, p.85). Se presenta en la página siguiente la rúbrica concreta usada en esta investigación para la variable PAP (ver cuadro 3.6 en la página 74 para mayor información sobre la variable PAP).

Cuadro 3.9.: Rúbrica de evaluación de las sub-variables de la variable PAP.

Sub-variables	Indicadores					Ponderación
p2-input	No identifica ninguna variable de entrada [0%]	Define/utiliza otra(s) variable(s) como variables de entrada [10%]	Define/utiliza alguna(s) variable(s) de entrada correcta con otra(s) incorrecta(s) [25%]	Define/utiliza alguna(s) variable(s) de entrada correcta(s) pero no todas [50%]	Define/utiliza correctamente las variables de entrada [100%]	5%
p2-output	No identifica ninguna variable de salida [0%]	Define/utiliza otra(s) variable(s) como variables de salida [10%]	Define/utiliza alguna(s) variable(s) de salida correcta con otra(s) incorrecta(s) [25%]	Define/utiliza alguna(s) variable(s) de salida correcta(s) pero no todas [50%]	Define/utiliza correctamente las variables de salida [100%]	5%
p2-idsub	No identifica ninguno de los subproblemas [0%]	Identifica sólo uno de los subproblemas [25%]	Identifica sólo dos de los subproblemas [50%]	Identifica tres o cuatro de los subproblemas según la variante asumida [100%]		5%
p2-solsub	No intenta resolver ninguno de los subproblemas [0%]	Intenta, pero no resuelve correctamente ninguno de los subproblemas [5%]	Resuelve correctamente uno de los subproblemas basado en sus suposiciones [40%]	Resuelve correctamente dos de los subproblemas basado en sus suposiciones [60%]	Resuelve correctamente al menos tres de los subproblemas basado en sus suposiciones [100%]	10%

continúa en las siguientes páginas...

p2-intsub	No intenta resolver ninguno de los subproblemas [0 %]	Resuelve o intenta resolver dos o más subproblemas de forma aislada o en orden incorrecto [25 %]	Resuelve o intenta resolver dos subproblemas en el orden correcto [50 %]	Resuelve o intenta resolver tres subproblemas y uno de ellos no está en el orden correcto [80 %]	Resuelve o intenta resolver al menos tres subproblemas en el orden correcto según la variante asumida [100 %]	5 %
	Resuelve o intenta resolver sólo uno de los subproblemas [5 %]					
p2-norm	No intenta resolver el problema o no identifica variables de entrada [0 %]	Resuelve o intenta resolver pero no declara en ningún momento que las variables de entrada deben tener valores positivos y del tipo correcto [10 %]	Resuelve o intenta resolver pero no declara para todas las variables de entrada que deben ser positivas y del tipo correcto [50 %]	Resuelve o intenta resolver y declara explícitamente para todas las variables de entrada que deben ser positivas, o bien, del tipo correcto, pero no ambas condiciones [75 %]	Resuelve o intenta resolver y declara explícitamente para todas las variables de entrada que deben ser positivas y del tipo correcto [100 %]	5 %

p2-extre	No intenta resolver el problema de calcular el número de adultos acompañantes [0%]	Intenta resolver el problema de calcular la cantidad de adultos, <code>adultosAcomp</code> , pero sin diferenciar si es divisible por 15 o no [5%]	Intenta pero no logra resolver el problema de calcular la cantidad de adultos, <code>adultosAcomp</code> , aunque considera la divisibilidad por 15 [50%]	Expresa la solución correcta al problema de calcular la cantidad de adultos sin usar simbología matemática o sintaxis válida [95%]	Resuelve correctamente el problema de calcular la cantidad de adultos [100%]	5%
	No intenta resolver el problema de calcular el número de adultos acompañantes ni de vehículos [0%]	Intenta resolver el problema de calcular la cantidad de adultos, <code>adultosAcomp</code> , y/o <code>cantVehículos</code> , pero sin diferenciar si son divisibles o no. [5%]	Resuelve correctamente el problema de calcular la cantidad de adultos, pero calcula <code>cantVehículos</code> sin considerar la divisibilidad por <code>CAPACIDAD_VEHÍCULOS</code> [50%]	Expresa la solución correcta al problema de calcular la cantidad de adultos sin usar simbología matemática o sintaxis válida [95%]	Resuelve correctamente el problema de calcular la cantidad de adultos, y también calcula correctamente <code>cantVehículos</code> [100%]	

p2-flujo	No intenta resolver el problema [0 %]	Intenta resolver pero no respeta la secuenciación y no utiliza ninguna estructura de control condicional [5 %]	Intenta resolver respetando la secuenciación pero no utiliza ninguna estructura de control condicional [40 %]	Resuelve o intenta resolver utilizando estructuras de control pero sin respetar la indentación en caso de usar pseudocódigo, o sin dejar claro el flujo de ejecución en caso de usar un diagrama de flujo [60 %]	Resuelve o intenta resolver respetando la secuenciación y utilizando estructuras de control condicional, con su indentación apropiada en caso de usar pseudocódigo, o con su flujo de ejecución claro en caso de usar un diagrama de flujo [100 %]	5 %
p2-sin	No intenta resolver el problema [0 %]	Intenta resolver pero no utiliza simbología propia de los diagramas de flujo, o no utiliza palabras reservadas mínimas, como Si, Entonces, Sino, FinSi, Inicio, Fin, Mostrar, Imprimir, etc. [10 %]	Intenta resolver usando simbología parcialmente correcta para alguna estructura de control condicional [40 %]	Resuelve o intenta resolver usando simbología correcta para las estructuras de control condicional, pero no incluye completamente cosas como el inicio, el fin, la petición de datos de entrada y la impresión de la respuesta [80 %]	Resuelve o intenta resolver usando las estructuras de control condicional correspondientes y utilizando la simbología de inicio, fin, petición de datos e impresión de la respuesta [100 %]	5 %

p4-input	No identifica ninguna de las variables de entrada [0 %]	Define/utiliza otra(s) variable(s) como variables de entrada [5 %]	Define/utiliza alguna(s) variable(s) de entrada correcta con otra(s) incorrecta(s) [25 %]	Define/utiliza alguna(s) variable(s) de entrada correcta(s) pero no todas [50 %]	Define/utiliza correctamente las variables de entrada [100 %]	5 %
		Define/utiliza las variables mezclando el lado izquierdo con el derecho [15 %]		Define/utiliza todas las variables de entrada junto con otra(s) irrelevante(s) [75 %]		
p4-dibu	No realiza ningún esquema o dibujo [0 %]	Realiza un esquema o dibujo que carece de utilidad para identificar las variables importantes [10 %]	Realiza un esquema o dibujo con potencial para identificar las variables importantes pero sin estar identificadas [30 %]	Realiza un esquema o dibujo donde se pueden identificar algunas variables importantes pero no todas las que permite dicha representación [70 %]	Realiza un dibujo o esquema donde se pueden identificar las variables importantes [100 %]	5 %
p4-output	No identifica ninguna de las variables de salida [0 %]	Define/utiliza otra(s) variable(s) como variables de salida [5 %]	Define/utiliza alguna(s) variable(s) de salida correcta con otra(s) incorrecta(s) [25 %]	Define/utiliza alguna(s) variable(s) de salida correcta(s) pero no todas [50 %]	Define/utiliza correctamente las variables de salida [100 %]	5 %
		Define/utiliza las variables mezclando el lado izquierdo con el derecho [15 %]		Define/utiliza todas las variables de salida junto con otra(s) irrelevante(s) [75 %]		
p4-idsub	No identifica ninguno de los subproblemas [0 %]	Intenta resolver sin separar las variables de entrada del lado izquierdo de las del lado derecho [25 %]		Identifica sólo dos de los subproblemas [50 %]	Identifica los cuatro subproblemas [100 %]	5 %

p4-solsub	No intenta resolver ninguno de los subproblemas [0 %]	Intenta, pero no resuelve correctamente ninguno de los subproblemas [5 %]	Resuelve correctamente el subproblema de la cantidad de asientos de cada lado pero no recalcula ni distribuye el espacio sobrante [50 %]	Resuelve correctamente el subproblema de la cantidad de asientos de cada lado, además recalcula correctamente el espacio sobrante, pero no lo distribuye correctamente [70 %]	Resuelve correctamente el subproblema de la cantidad de asientos de cada lado y además recalcula y distribuye correctamente el espacio sobrante [100 %]	10 %
		Resuelve correctamente el subproblema de la cantidad de asientos suponiendo que ambos lados son iguales [40 %]				
	No intenta resolver ninguno de los subproblemas [0 %]	Intenta, pero no resuelve correctamente ninguno de los subproblemas [5 %]	Intenta resolver pero no calcula ni distribuye el espacio para entrar a los asientos [50 %]	Resuelve correctamente el cálculo y distribución del espacio para entrar a los asientos [100 %]		
p4-intsub	No intenta resolver ninguno de los subproblemas [0 %]	Resuelve o intenta resolver sólo dos de los subproblemas [5 %]	Resuelve o intenta resolver los cuatro subproblemas de forma aislada [25 %]	Resuelve o intenta resolver los cuatro subproblemas en orden incorrecto [50 %]	Resuelve o intenta resolver los cuatro subproblemas en el orden correcto según la variante asumida [100 %]	4 %

p4-norm	No intenta resolver el problema o no identifica variables de entrada [0%]	Resuelve o intenta resolver pero no declara en ningún momento que las variables de entrada deben tener valores positivos y del tipo correcto [10%]	Resuelve o intenta resolver pero no declara para todas las variables de entrada que deben ser positivas y del tipo correcto [50%]	Resuelve o intenta resolver y declara explícitamente para todas las variables de entrada que deben ser positivas, o bien, del tipo correcto, pero no ambas condiciones [75%]	Resuelve o intenta resolver y declara explícitamente para todas las variables de entrada que deben ser positivas y del tipo correcto [100%]	4%
p4-extre	No intenta resolver el problema [0%]	Resuelve o intenta resolver el problema pero en ningún momento considera el caso extremo de la variante asumida [5%]	Identifica pero no trata de resolver el caso extremo de la variante asumida [25%]	Intenta pero no logra resolver el problema del caso extremo de la variante asumida [50%]	Resuelve o intenta resolver el problema considerando el caso extremo de la variante asumida [100%]	4%

p4-flujo	No intenta resolver el problema [0 %]	Intenta resolver pero no respeta la secuenciación y no utiliza ninguna estructura de control condicional [5 %]	Intenta resolver respetando la secuenciación pero no utiliza ninguna estructura de control condicional [40 %]	Resuelve o intenta resolver utilizando estructuras de control pero, sin respetar la indentación en caso de usar pseudocódigo, o sin dejar claro el flujo de ejecución en caso de usar un diagrama de flujo [60 %]	Resuelve o intenta resolver respetando la secuenciación y utilizando estructuras de control condicional, con su indentación apropiada en caso de usar pseudocódigo, o con su flujo de ejecución claro en caso de usar un diagrama de flujo [100 %]	4 %
p4-sin	No intenta resolver el problema [0 %]	Intenta resolver pero no utiliza simbología propia de los diagramas de flujo, o no utiliza palabras reservadas mínimas, como Si, Entonces, Sino, FinSi, Inicio, Fin, Mostrar, Imprimir, etc. [10 %]	Intenta resolver usando simbología parcialmente correcta para alguna estructura de control condicional [40 %]	Resuelve o intenta resolver usando simbología correcta para las estructuras de control condicional, pero no incluye completamente cosas como el inicio, el fin, la petición de datos de entrada y la impresión de la respuesta [80 %]	Resuelve o intenta resolver usando las estructuras de control condicional correspondientes y utilizando la simbología de inicio, fin, petición de datos e impresión de la respuesta [100 %]	4 %

Fuente: Elaboración propia.

3.6.2. Rúbrica de medición de la variable Habilidad de modelamiento y resolución de problemas de referencia

Se presenta en la página siguiente la rúbrica concreta usada en esta investigación para la medición de las sub-variables de la variable RP (ver cuadro 3.7 en la página 76 para mayor información sobre la variable RP). Tal como antes, se han seguido los pasos del diseño de rúbricas de puntuación de Mertler (2000) y se ha utilizado el estilo de la plantilla presentada por Cebrián de la Serna y Monedero Moya (2014, p.85).

Cuadro 3.10.: Rúbrica de evaluación de las sub-variables de la variable RP.

Sub-variables	Indicadores					Ponderación
p1-graf	No hace ninguna representación gráfica [0%]	Hace un dibujo pero no es un grafo [10%]	Hace un grafo y luego lo descarta por otro dibujo que no es un grafo [25%]	Hace un grafo, pero no es del todo correcto [80%]	Hace el grafo correspondiente [100%]	20%
p1-ciu	No intenta resolver [0%]	Identifica a las ciudades por su nombre completo [10%]	Utiliza alguna abreviatura para los nombres de las ciudades [50%]	Utiliza las iniciales de las ciudades en algún contexto pero no en todo el problema [75%]	Identifica las ciudades por la letra inicial [100%]	10%
p1-carr	No intenta resolver o no hay dibujo ni grafo donde representar las carreteras [0%]	Presenta el grafo (o dibujo) con las carreteras de forma muy poco legible y muy difícil de trabajar [25%]	Presenta el grafo (o dibujo) con las carreteras de forma poco legible y difícil de trabajar [50%]	Ordena el grafo (o dibujo) de una forma legible y fácil de trabajar después de haberlo hecho una primera vez [90%]	Presenta el grafo (o dibujo) de una forma legible y fácil de trabajar de una sola vez [100%]	20%
p1-sol	No intenta resolver [0%]	Utiliza formulaciones matemáticas no útiles o inapropiadas [5%]	Identifica sólo una ruta [20%] o dos rutas posibles [40%]	Identifica tres rutas posibles [75%]	Evalúa al menos cuatro rutas posibles [100%]	20%

p1-res	No llega a ningún resultado [0%] o bien, enuncia como resultado una distancia y no una ruta [10%]	Llega a un resultado (no necesariamente correcto) pero no lo enuncia ni lo resalta [30%]	Llega a un resultado (no necesariamente correcto) y no lo resalta pero es notorio que es la respuesta [50%]	Llega a un resultado (no necesariamente correcto) y lo resalta [80%]	Llega a un resultado (no necesariamente correcto) y escribe la respuesta con palabras [100%]	5%
p3-mod	No intenta resolver [0%]	Intenta resolver pero el modelo no es apropiado [5%]	Identifica el ancho de la puerta en el dibujo [25%]	Identifica la longitud de la diagonal del rectángulo [50%]	Identifica los tres lados del triángulo rectángulo [100%]	10%
p3-sol	No intenta resolver [0%]	Utiliza formulaciones matemáticas no útiles o inapropiadas [15%]	Formula la necesidad del Teorema de Pitágoras [75%]	Despeja la variable apropiada (el cateto largo) [90%]	Encuentra el valor de solución [100%]	10%
p3-res	No llega a ningún resultado [0%]	Llega a un resultado (no necesariamente correcto) pero no lo enuncia ni lo resalta [10%]	Llega a un resultado (no necesariamente correcto) y no lo resalta pero es notorio que es la respuesta [50%]	Llega a un resultado (no necesariamente correcto) y lo resalta [80%]	Llega a un resultado (no necesariamente correcto) y escribe la respuesta con palabras [100%]	5%

Fuente: Elaboración propia.

3.7. Extracción de datos de la parte cuantitativa

Una vez que los instrumentos fueron completados por los alumnos, el investigador los analizó y les aplicó las ponderaciones indicadas en la rúbrica del cuadro 3.9. Para hacer esto se utilizó el programa Xournal v0.4.8.2016⁸ anotando sobre la versión PDF de los instrumentos escaneados.

Por ejemplo, en la figura 3.10 en la página siguiente se presenta a la izquierda la hoja de resolución del problema 1 del sujeto I-30 y a la derecha la evaluación y ponderación de las sub-variables correspondientes.

En la figura 3.11 en la página 104 se presenta a la izquierda la hoja de resolución del problema 3 del sujeto I-32 y a la derecha la evaluación y ponderación de las sub-variables correspondientes.

En la figura 3.12 en la página 105 se presenta a la izquierda la hoja de resolución del problema 4 del sujeto D-05 y a la derecha la evaluación y ponderación de las sub-variables correspondientes.

Este proceso se realizó para los 4 problemas de los 80 sujetos de la muestra de la parte cuantitativa.

⁸<http://xournal.sourceforge.net>



Problema 1

• A → I

1) A²→C⁶→D²→E⁸→G⁵→I = 23 Km

2) A²→C⁴→E⁴→F¹→H³→I = 14 Km

3) A²→C⁶→D²→F⁶→I = 19 Km

4) A²→C⁶→D²→E⁴→F¹→H³→I = 19 Km

5) A²→C⁴→E⁸→G⁵→I = 19 Km

6) A²→C⁴→E⁴→F⁶→I = 16 Km

7) A²→C⁶→D⁵→F⁴→H³→I = 17 Km

8) A²→C⁶→D²→E⁴→F⁶→I = 20 Km

R// La ruta más corta es de 14 Km que va desde Almeida a Charlestown, de Charlestown a Edinburgo, de Edinburgo a Fukushima, de Fukushima a Hormisdas, de Hormisdas a Idilicom.

I-30-1



Problema 1

• A → I

1) A²→C⁶→D²→E⁸→G⁵→I = 23 Km

2) A²→C⁴→E⁴→F¹→H³→I = 14 Km

3) A²→C⁶→D²→F⁶→I = 19 Km

4) A²→C⁶→D²→E⁴→F¹→H³→I = 19 Km

5) A²→C⁴→E⁸→G⁵→I = 19 Km

6) A²→C⁴→E⁴→F⁶→I = 16 Km

7) A²→C⁶→D⁵→F⁴→H³→I = 17 Km

8) A²→C⁶→D²→E⁴→F⁶→I = 20 Km

R// La ruta más corta es de 14 Km que va desde Almeida a Charlestown, de Charlestown a Edinburgo, de Edinburgo a Fukushima, de Fukushima a Hormisdas, de Hormisdas a Idilicom.

p1-graf: 0

p1-ciu: 100

p1-carr: 0

p1-sol: 100

p1-sol: 100

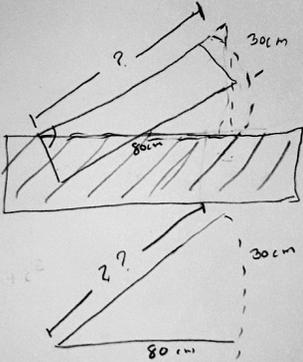
Figura 3.10.: Solución realizada por el sujeto I-30 del problema 1 (a la izquierda), y evaluación y ponderación de esa solución por parte del investigador (a la derecha).

Problema 3

 Datos

$\theta = ?$
 $h = 30 \text{ cm}$
 $d = 80$

$c^2 = a^2 + b^2$
 $a^2 = b^2 - c^2$



$a^2 = b^2 - c^2$

$a^2 = b^2 - c^2$
 $a = \sqrt{b^2 - c^2}$
 $a = \sqrt{80^2 - 30^2}$
 $a = \sqrt{5,500}$
 $a = 74.16$

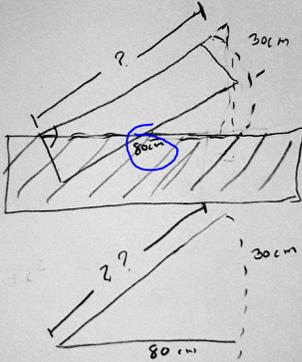
I-32-3

Problema 3

 Datos

$\theta = ?$
 $h = 30 \text{ cm}$
 $d = 80$

$c^2 = a^2 + b^2$
 $a^2 = b^2 - c^2$



$a^2 = b^2 - c^2$

$a^2 = b^2 - c^2$
 $a = \sqrt{b^2 - c^2}$
 $a = \sqrt{80^2 - 30^2}$
 $a = \sqrt{5,500}$
 $a = 74.16$

p3-mod: 5

p3-sol: 15

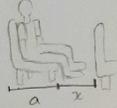
p3-res: 80

Figura 3.11.: Solución realizada por el sujeto I-32 del problema 3 (a la izquierda), y evaluación y ponderación de esa solución por parte del investigador (a la derecha).



Problema 4

- Conocer la longitud del autobús de ambos lados, ya que en uno hay que restarle la anchura de la puerta trasera. $C_1 = b$ $C_2 = c$
- Conocer el espacio que una persona promedio utiliza, cuando se sienta en ellos: \hookrightarrow que a ellos les convenga.



- Conociendo el espacio que ocupa cada asiento más la que ocupa la persona. $a+x$

Se tendría para $C_1 = \frac{b}{a+x} = Y_1 \rightarrow$ la cantidad de asientos de un lado.

$$\wedge C_2 = \frac{c}{a+x} = Y_2 \rightarrow \text{la cantidad de asientos del otro lado.}$$

$$Y_1 + Y_2 = \text{Cantidad Total.}$$

D-05-4



Problema 4

- Conocer la longitud del autobús de ambos lados, ya que en uno hay que restarle la anchura de la puerta trasera. $C_1 = b$ $C_2 = c$?
- Conocer el espacio que una persona promedio utiliza, cuando se sienta en ellos: \hookrightarrow que a ellos les convenga.



- Conociendo el espacio que ocupa cada asiento más la que ocupa la persona. $a+x$

Se tendría para $C_1 = \frac{b}{a+x} = Y_1 \rightarrow$ la cantidad de asientos de un lado.

$$\wedge C_2 = \frac{c}{a+x} = Y_2 \rightarrow \text{la cantidad de asientos del otro lado.}$$

$$Y_1 + Y_2 = \text{Cantidad Total.}$$

p4-input:	95	100	p4-output:	50
p4-idsub:	50		p4-solsub:	50
p4-intsub:	5		p4-norm:	10
p4-extre:	5		p4-flujo:	5
p4-sin:	10			

Figura 3.12.: Solución realizada por el sujeto D-05 del problema 4 (a la izquierda), y evaluación y ponderación de esa solución por parte del investigador (a la derecha).

3.8. Archivo de datos de la parte cuantitativa

Las 80 respuestas obtenidas del instrumento de la parte cuantitativa fueron volcadas sobre un archivo de hoja de cálculo y luego exportado hacia un archivo CSV (`datos - cuant i . csv`) para poder ser procesado digitalmente por medio de software estadístico.

Las columnas del archivo CSV son las presentadas en el cuadro 3.11 (véase también el listado de código D.1 en la página 227).

Cuadro 3.11.: Columnas del archivo de datos cuantitativos.

Índice	Nombre de Columna en el Archivo	Variable equivalente en la investigación (ver cuadro 3.5)	Descripción
0	id		Identificador único del instrumento
1	ingreso		Año de ingreso a la Universidad
2	edad	edad	
3	cum	cum	
4	matTot	matTot	
5	matProg	matProg	
6	grupo	equivalente a matProg pero como texto {'P0', 'P1', 'P2', 'P3'}	
7	gen	gen	
8	alg	alg	
9...16	p1-graf...p3-res	Detalle de la operacionalización de RP	Ver cuadro 3.7 en la página 76
17	RP	RP	Promedio ponderado de las columnas 9 a 16 según el cuadro 3.10 en la página 100
18...36	p2-input...p4-sin	Detalle de la operacionalización de PAP	Ver cuadro 3.6 en la página 74
37	PAP	PAP	Promedio ponderado de las columnas 18 a 36 según el cuadro 3.9 en la página 91

Fuente: Elaboración propia.

3.9. Análisis estadísticos aplicados a los datos de la parte cuantitativa

Primero se realiza un análisis descriptivo de las variables (ver sub-sección 4.1.1 en la página 119), por medio de sus medidas de tendencia central y de sus gráficos de dispersión, histogramas y de pastel según el tipo de cada variable.

Luego se verifica si las variables son susceptibles de ser analizadas por medio de técnicas paramétricas o no, por medio de pruebas de normalidad (ver subsección 4.1.2 en la página 131).

Finalmente se realizan los cálculos de los coeficientes de correlación correspondientes entre todas las variables y algunas comparaciones de grupos (ver subsección 4.1.3 en la página 133).

3.10. Recolección de datos de la parte cualitativa

Se siguió para esta parte una metodología similar a la aplicada por Schoenfeld (1985) y Granberg (2016), pero adaptada a las circunstancias de aislamiento y confinamiento impuestas por la pandemia de COVID-19:

Además de la sesión del Sujeto Experto, se realizaron 14 sesiones de videollamada de aproximadamente 25 minutos cada una, en las que un alumno individualmente recibe el enunciado de un problema corto de construcción de un algoritmo y debe realizar su correspondiente implementación en el lenguaje de programación Python⁹, que ya conoce. Las sesiones fueron grabadas por medio del sistema de videoconferencias BigBlueButton¹⁰ y la codificación fue realizada por medio del servicio de codificación colaborativa REPLit¹¹ que se había estado utilizando todo el semestre.

Esta actividad era parte de la evaluación sumativa de la materia «Programación 2» que cursaban los alumnos en ese momento, con un 10% de ponderación de la nota final. Esta evaluación tenía sus propias ponderaciones, conocidas por los estudiantes previamente a su realización:

- Identificar las entradas y las salidas del problema 20%
- Identificar casos especiales 20%
- Identificar el algoritmo correcto 20%
- Construir un programa correcto sintácticamente 20%
- Verificar que la solución sea correcta 20%

Pero para efectos de esta investigación, no hay ponderaciones para esta actividad, ya que el objetivo aquí es analizar cronológicamente la secuencia de tareas cognitivas.

⁹<https://www.python.org/>

¹⁰<https://bigbluebutton.org/>

¹¹<https://replit.com/>

Los alumnos fueron informados sobre las características de esta actividad semanas antes de realizarse. Recibieron la indicación de que dado que era una actividad sincrónica a distancia, debían expresar sus pensamientos en voz alta para que estos fueran considerados en la evaluación.

El profesor e investigador estuvo disponible para consultas en cada sesión todo el tiempo, minimizando su participación pero respondiendo preguntas y dando sugerencias cuando se detectaba un bloqueo significativo en los alumnos.

Luego de la realización de todas las sesiones, se revisó minuciosa e iterativamente cada grabación, identificando cada una de las tareas cognitivas realizadas por los sujetos (según el detalle de Granberg, 2016) y en qué momento relativo ocurrieron. Esto se hizo observando detenida y repetidamente sus gestos, sus vocalizaciones, sus preguntas, sus comentarios, el código digitado, el código borrado, etc., y también en qué momento de la grabación ocurrió. Ver figura 3.13 en la página siguiente.

Estas tareas cognitivas de Granberg (2016) son las que se usan principalmente como las categorías de este estudio cualitativo para realizar la codificación axial correspondiente. Ver Hernández Sampieri y cols. (2014, cap. 14 y 15). Sin embargo, a medida que se analizaban las primeras sesiones, fue evidente que las categorías de Schoenfeld (2016) y Granberg (2016) no eran suficientes para describir lo observado en las grabaciones. Por ello, y siendo común en las investigaciones cualitativas, hubo que definir nuevas categorías para describir mejor a los procesos cognitivos de los sujetos analizados. Estas nuevas categorías se encuentran descritas en la siguiente sección.

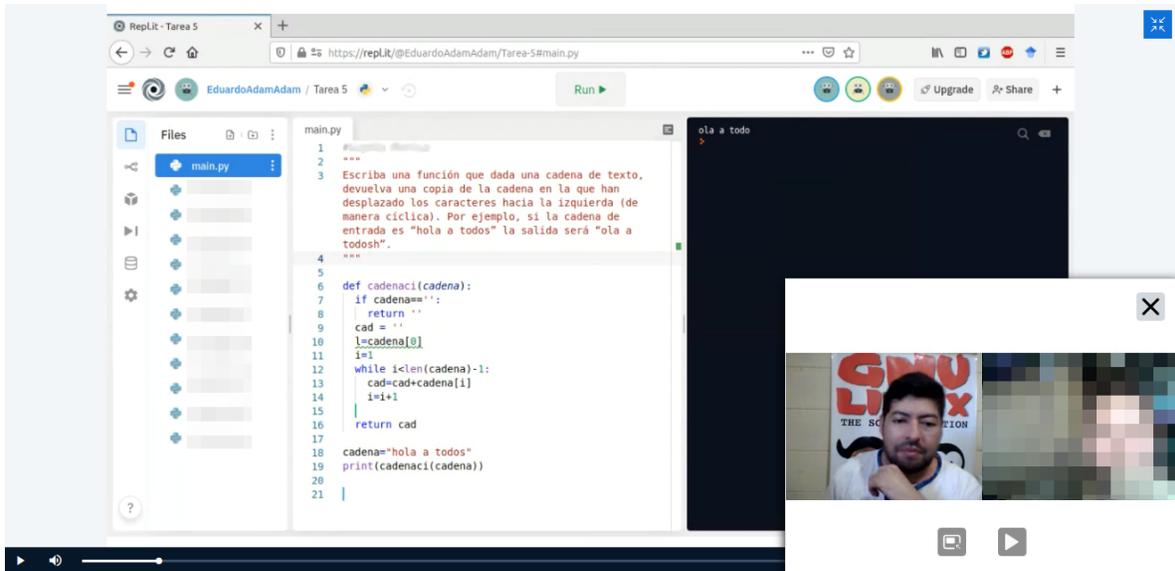
Todos los problemas se pueden resolver en un sólo nivel semántico, evadiendo la dificultad de la Noción de Jerarquía de la que habla Dijkstra (1974) (ver sección 2.3.6).

Cada sujeto recibió un problema diferente, tomado al azar de una lista de problemas, que está presentada en el Apéndice B en la página 221 en el orden en que fueron asignados los problemas. El primer sujeto recibió el problema S01, el segundo sujeto recibió el problema S02, y así sucesivamente. Estos mismos códigos se utilizan para nombrar a los sujetos.

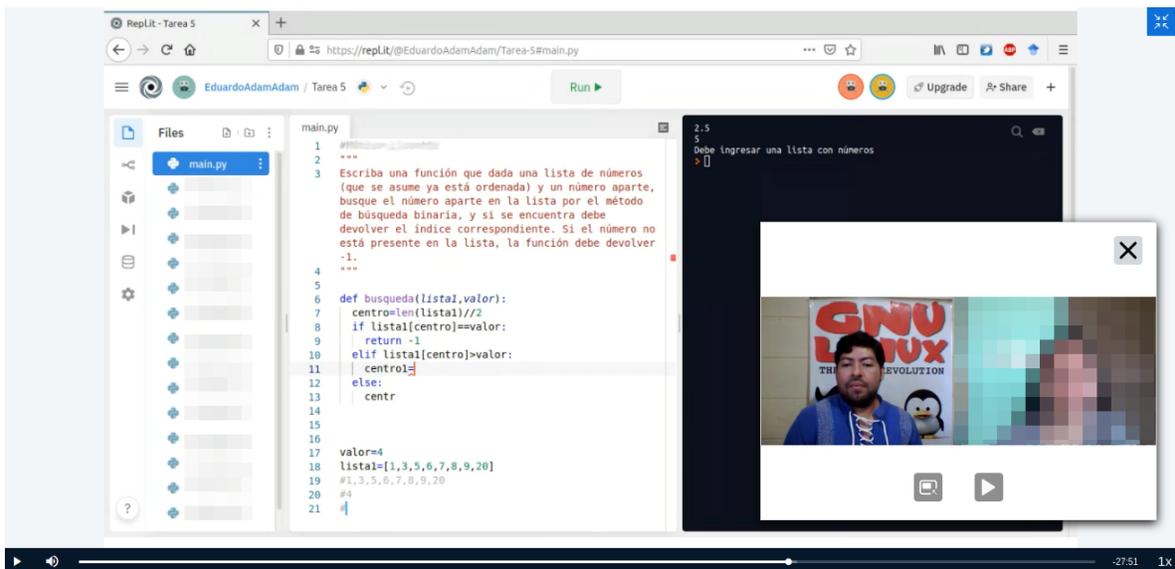
Uno de los problemas, el S11, no era tal, según la interpretación de Ferrer Vicente (2000), ya que casi explícitamente indicaba los pasos que se debían realizar. Esto para compararlo con lo que indican Schoenfeld (2016) y Granberg (2016) sobre lo que ocurre cuando el resolutor conoce el procedimiento de solución. Ver línea A de la figura 2.3 en la página 15.

Dado que la actividad era evaluada desde la perspectiva de los alumnos, que el tiempo era limitado, y que serían el foco de atención total del profesor, era previsible que el nivel de estrés y ansiedad sería alto, por lo que se seleccionaron problemas de relativamente baja complejidad, que sólo requirieran de una o dos estructuras iterativas y arreglos. Estos problemas fueron tomados de las guías de ejercicios de la materia «Introducción a la Informática», que es la primera de las tres materias de programación de computadoras que incluyen los programas de estudio de los alumnos (Ver apéndice C). Como se mencionó en la sub-sección de población y muestra (en la página 57) los alumnos analizados en esta parte están terminando de cursar «Programación 2» que es la tercera de estas materias.

Para poder contrastar con los sujetos estudiantes, y hacer eco de los resultados de Schoenfeld (1985), se examinó también a un sujeto considerado experto. Este sujeto es un profe-



(a)



(b)

Figura 3.13.: Capturas de pantalla de dos de las grabaciones de las sesiones de recolección de datos de la parte cualitativa.

Fuente: Elaboración propia.

sor universitario de cursos de programación y de informática, con varios años de experiencia. Se le asignó el último problema de la lista y es el sujeto Exp. El problema que se le asignó es de la misma escala de complejidad que los asignados a los alumnos, tomado de las mismas guías de ejercicios.

3.11. Categorías para la modelación de los procesos cognitivos durante la resolución de problemas de la parte cualitativa

Las tareas cognitivas identificadas son básicamente las mismas planteadas por Granberg (2016) (ver figura 2.3 en la página 15) numeradas del cero al ocho, junto con algunas anotaciones adicionales particulares que después de hacer una análisis preliminar de las grabaciones se consideraron pertinentes dadas las circunstancias en las que se realizó esta actividad. Ver cuadro 3.12.

Cuadro 3.12.: Tareas cognitivas y anotaciones consideradas en esta investigación.

Correlativo	Código	Descripción
0	le	Leer el problema
1	an	Analizar el problema
2	co	Explorar – Corregir conocimiento previo
3	re	Explorar – Recordar conocimiento previo
4	nu	Explorar – Crear nuevo conocimiento
5	so	Planear – Construir estrategia de solución
6	aj	Planear – Ajustar estrategia de solución
7	im	Implementar la solución
8	ve	Verificar la solución
	ay	Recibir/Pedir ayuda
	no	Expresa que no lo puede hacer
	te	Interrupción o Falla técnica
	pr	Problema resuelto
	pn	Problema no resuelto

Fuente: Elaboración propia.

Los códigos adicionales agregados después de un análisis preliminar de las grabaciones son:

- ay Es una marca que indica que el sujeto pidió ayuda por no entender algo, o bien, el profesor observó algún bloqueo y decidió intervenir aclarando la aparente duda.
- no Es una marca para indicar el momento en que el sujeto se da por vencido y declara explícitamente que no puede resolver el problema. Este es un fenómeno del que no hablan ni Schoenfeld (2015, 2016) ni Granberg (2016), pero observado en varios de los sujetos aquí analizados.

te Esta es una marca de falla técnica o falla de comunicación. En esta investigación la que más ocurrió fue la desconexión momentánea de los sujetos durante las videollamadas, pero también ocurrieron algunos episodios de desconexión o falla momentánea del sistema de REPLit.

pr Esta marca indica el momento en que el sujeto logró resolver el problema, dando por finalizada la sesión.

pn Esta marca indica el momento que se dio por terminada la sesión sin que el sujeto lograra resolver el problema satisfactoriamente.

Cabe aclarar que en para este tipo de problemas, «Analizar el problema» consistió en identificar las entradas y las salidas únicamente. Esto para ser consistente con lo supuesto en la rúbrica de la parte cuantitativa (ver cuadros 3.4 en la página 61 y 3.9 en la página 91).

Por otro lado, «Implementar» consiste en escribir código fuente.

3.12. Un diseño ampliado de Gráficos tipo Schoenfeld para representación de procesos cognitivos durante la resolución de problemas

Se diseñó una nueva versión de gráfico tipo Schoenfeld basado en los gráficos de Granberg (2016) para representar el desarrollo y transición de las tareas cognitivas de los sujetos analizados en la parte cualitativa de este trabajo. Estos gráficos constituyen las representaciones o modelos visuales que explica Hernández Sampieri y cols. (2014, p. 472) sobre los diseños cualitativos de Teoría Fundamentada. Ver como ejemplo la figura 3.17 en la página 116.

Se agregan líneas blancas para indicar las fallas técnicas o de comunicación (te) y también líneas rojas para indicar la duración de las intervenciones de ayuda (ay).

Además se agrega un segundo diagrama con la misma información sobre las transiciones entre las etapas cognitivas pero sin considerar la duración de estas transiciones. Este segundo diagrama permite analizar propiamente la secuencia de las transiciones a pesar de lo corta o extensa de la duración de estas, lo que es especialmente útil cuando hay muchas transiciones de corta duración, y que pueden no apreciarse con suficiente detalle en el diagrama principal. Las unidades del eje horizontal de este diagrama secundario indica la cantidad de actividades realizadas por el alumno. Los únicos «pasos» que no se cuentan son las fallas técnicas y/o de comunicación.

El diagrama principal es parecido a los de Schoenfeld (1985, 2016) al representar el tiempo real, pero con más detalle en la escala. Mientras que el diagrama secundario es más parecido a los de Granberg (2016) que hace énfasis las transiciones y no en el tiempo real que cada una de estas lleva.

Se presenta a continuación cuatro diagramas de sesiones ficticias de resolución de un problema de 6:15 min cada una para explicar su interpretación. Serán llamados sujetos E1, E2,

E3 y E4. Los primeros tres, representan los episodios exitosos presentados en la figura 2.3 en la página 15.

En la figura 3.14 se presenta el episodio A de la figura 2.3, que muestra una resolución de un ejercicio cuyo procedimiento es conocido por el alumno.

En la figura 3.15 se presenta el episodio B de la figura 2.3, donde se muestra una resolución ideal de un problema, sin errores.

En la figura 3.16 se presenta el episodio C de la figura 2.3, donde se muestra una resolución, con dos errores, de un problema.

La figura 3.17 en la página 116 muestra una resolución similar al de la figura 3.16, pero con las marcas adicionales explicadas en la sección 3.11: Después de un minuto (1:00 min después de iniciada la grabación) ocurrió una falla técnica o de comunicación (τ_e) durante la etapa de Análisis (an). A la mitad del tercer minuto (2:30 min después de iniciada la grabación) ocurrió una petición de ayuda (ay) de parte del alumno durante la etapa de Construcción de la Estrategia de Solución (so). Al final del cuarto minuto (4:00 min después de iniciada la grabación) el alumno expresó de viva voz que no podía resolver el problema (no). Y finalmente, después del sexto minuto (6:15 min después de iniciada la grabación) el estudiante proporcionó una solución errónea del problema (pn).

Es de aclarar que en estos cuatro ejemplos, el diagrama principal y el diagrama secundario parecen idénticos, debido a que el espacio de tiempo entre cada transición se ha definido uniforme. Sin embargo, esto no suele ocurrir así en la realidad. Véase los gráficos de las páginas 168 a la 182, correspondientes a los sujetos analizados en la parte cualitativa.

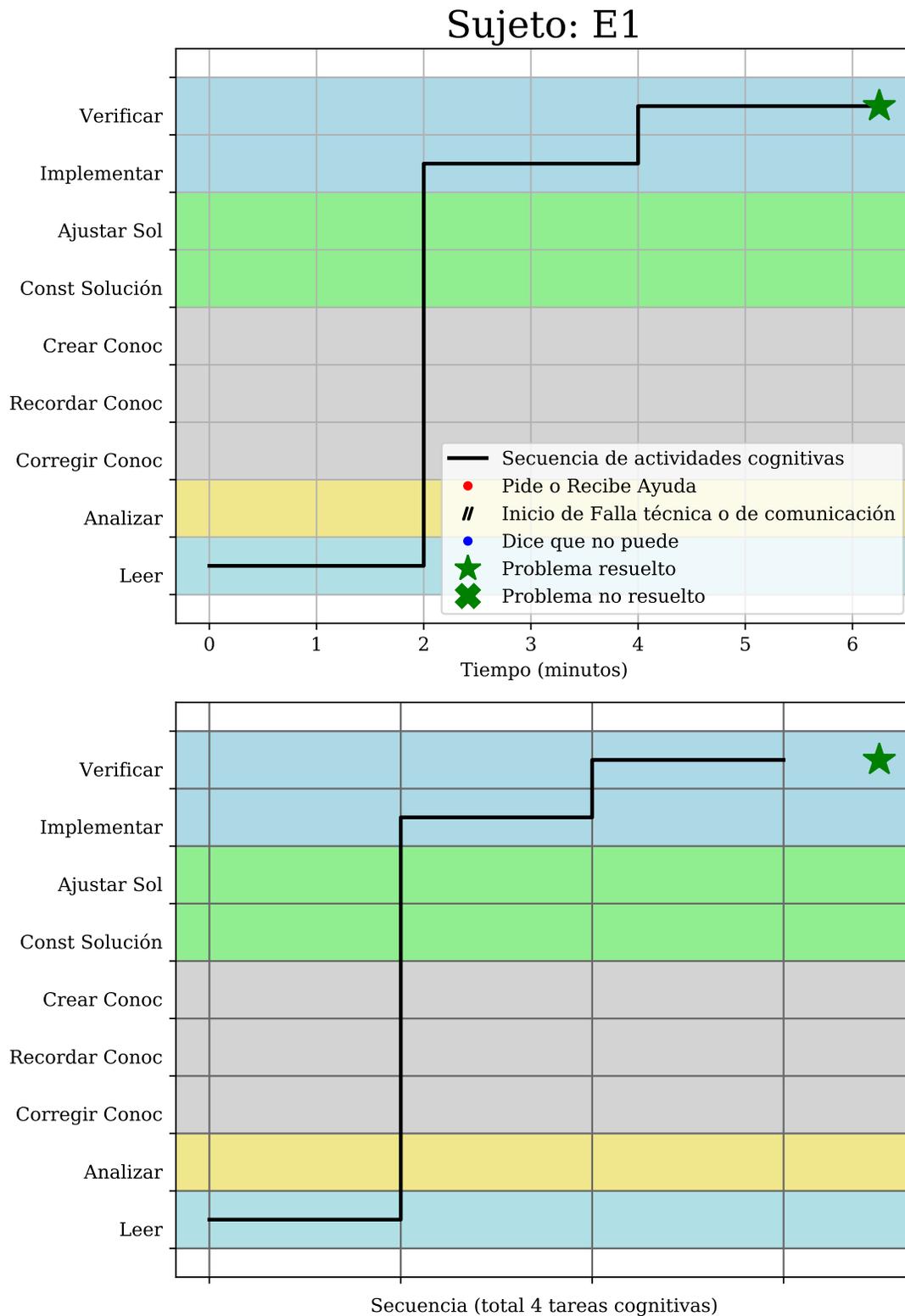


Figura 3.14.: Representación de una resolución de un ejercicio cuyo procedimiento es conocido. Equivalente al episodio A de la Figura 2.3 en la página 15.
Fuente: Elaboración propia.

Sujeto: E2

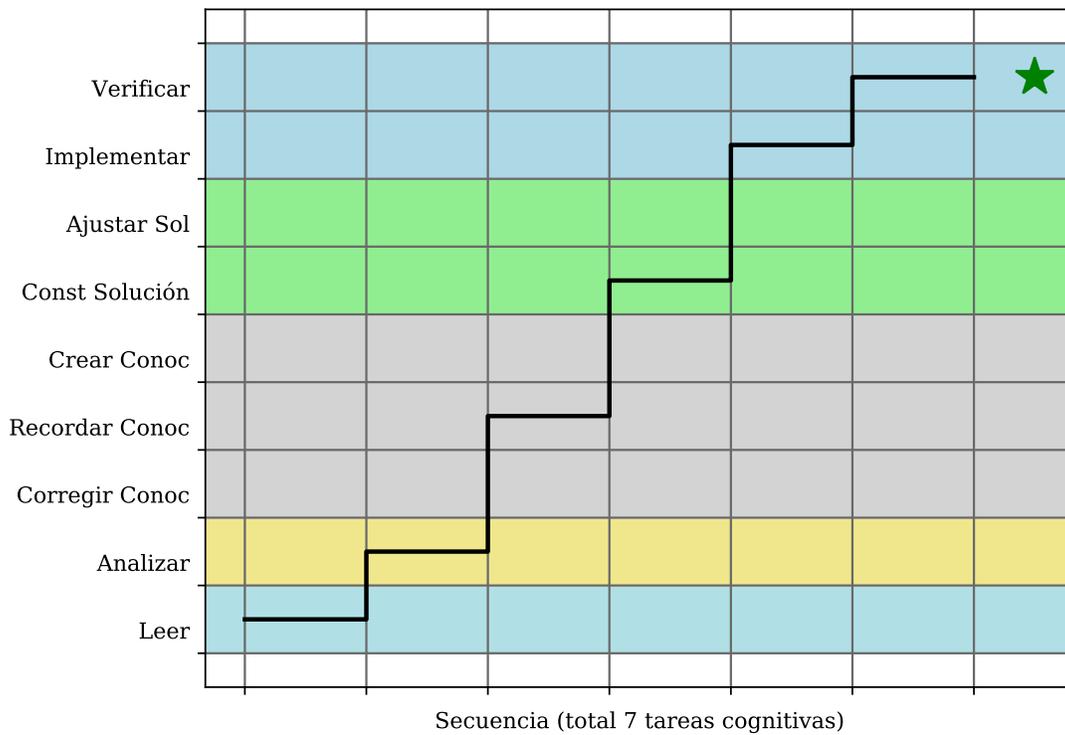
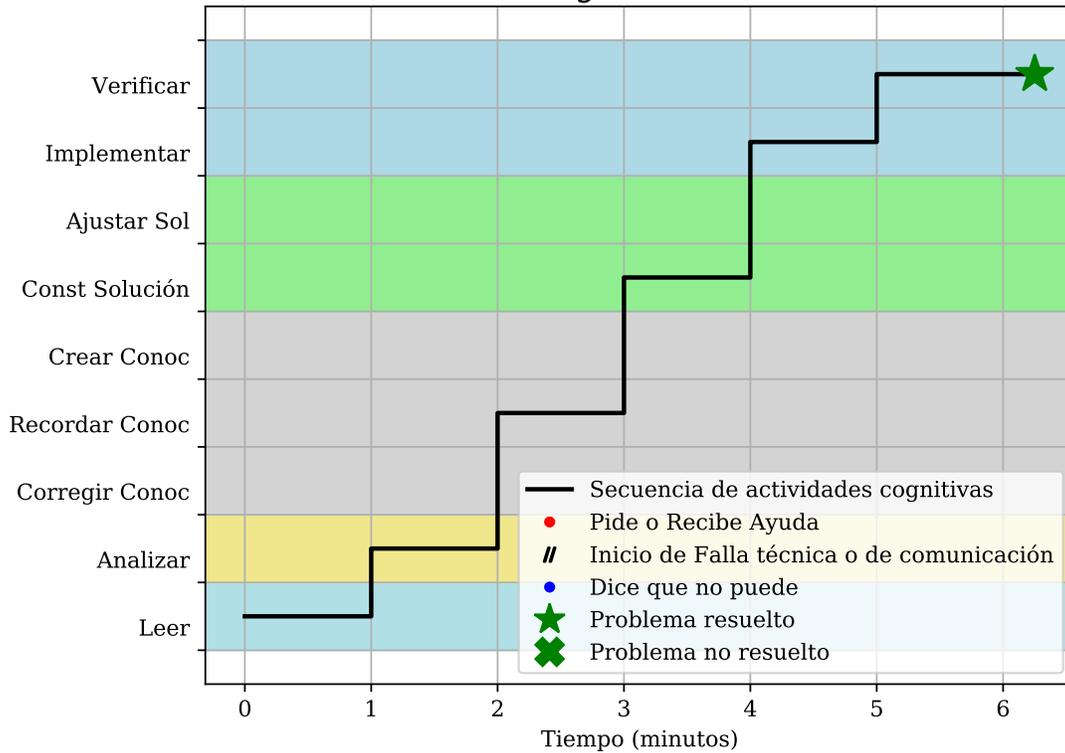


Figura 3.15.: Representación de un proceso creativo de resolución de problema sin errores. Equivalente al episodio B de la Figura 2.3 en la página 15.
Fuente: Elaboración propia.

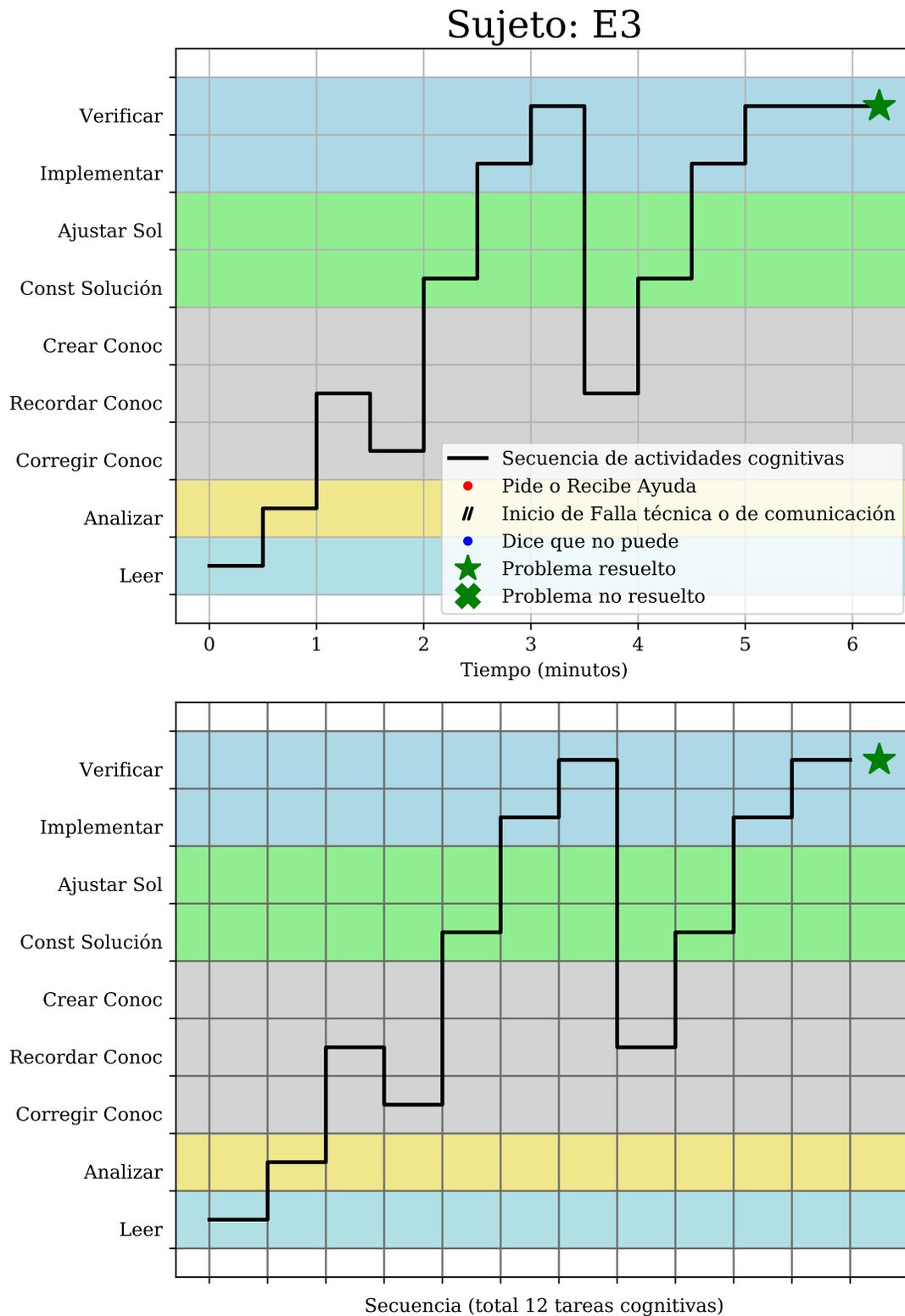


Figura 3.16.: Representación de una resolución creativa de problema que incluye dos errores. Equivalente al episodio C de la Figura 2.3 en la página 15.
Fuente: Elaboración propia.

Sujeto: E4

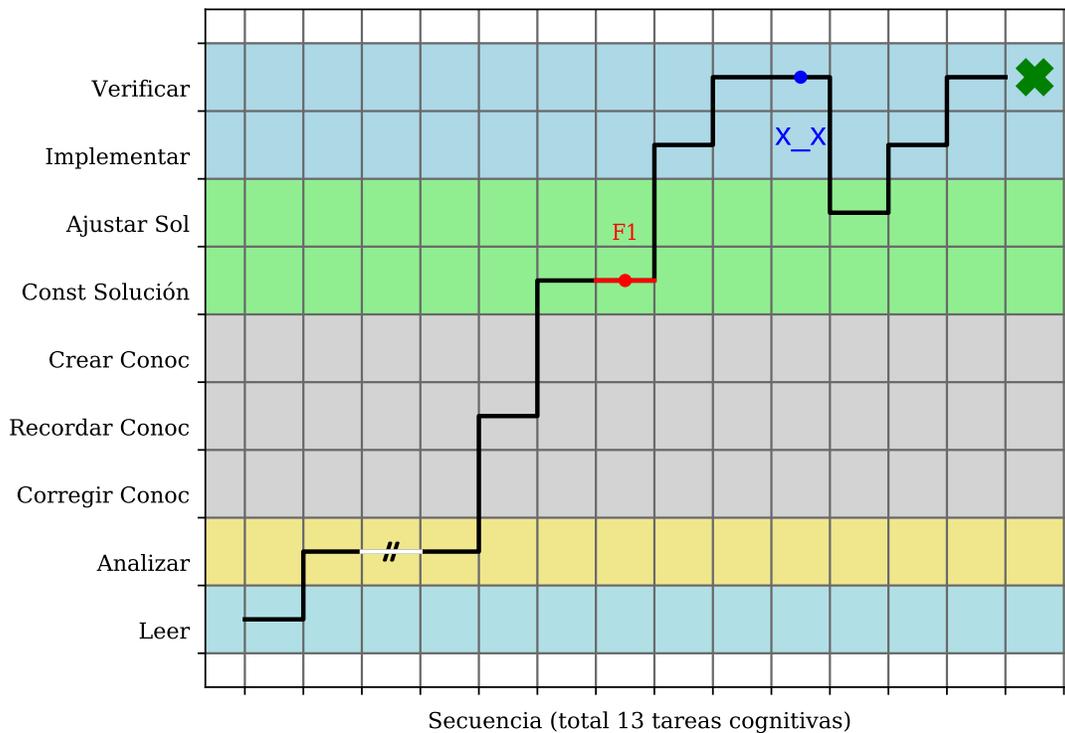
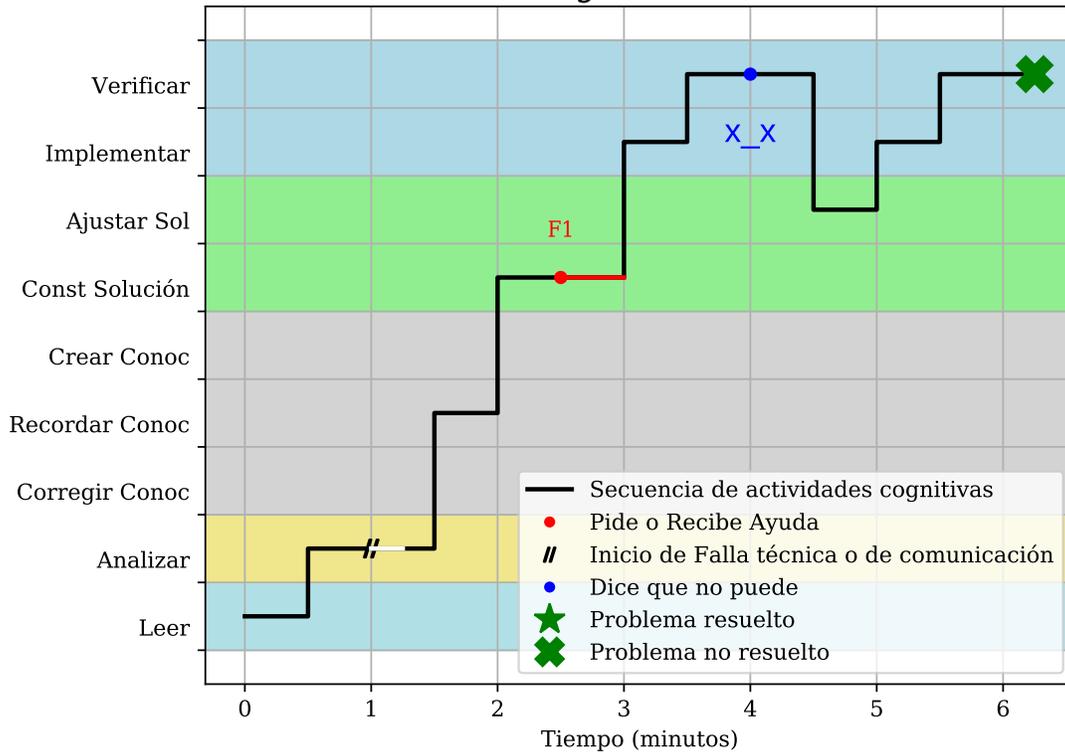


Figura 3.17.: Representación de una resolución fallida de un problema, con una falla técnica, una petición de ayuda y una expresión del alumno de incapacidad para resolver el problema.

Fuente: Elaboración propia.

3.13. Archivo de datos de la parte cualitativa

Se creó una hoja de cálculo (*datos-cuali.ods*) en la que para cada sujeto, se apuntó el momento de inicio de su lectura del problema, acompañado por el código *le*. Luego, para cada tarea cognitiva diferente se anotó el momento de la grabación en que se detecta su transición y el código correspondiente (ver cuadro 3.12).

Por ejemplo, para el episodio ficticio del sujeto ficticio E4 representado en la figura 3.17, los primeros registros se muestran en el cuadro 3.13. Los datos del cuadro indican que a los 30 minutos de iniciada la grabación el sujeto inició la lectura del problema (*le*). Luego, a los 30 minutos con 30 segundos de iniciada la grabación procedió a analizar el problema (*an*), es decir a identificar las entradas y las salidas del problema. Luego, exactamente al minuto 31 se dio una falla técnica o de comunicación (*te*). Esta falla terminó en el minuto 31 con 15 segundos de iniciada la grabación y el estudiante continuó con el análisis, hasta los 30 segundos del minuto 31 donde procedió a recodar conocimiento previo (*re*). Y así sucesivamente se registran las transiciones a las diferentes tareas cognitivas.

Cuadro 3.13.: Ejemplo de registros en el archivo de datos cualitativos del ejemplo de la figura 3.17 en la página anterior.

Sujeto (columna A)	Tiempo (columna B)	Actividad (columna G)
E4	30.00	<i>le</i>
	30.30	<i>an</i>
	31.00	<i>te</i>
	31.15	<i>an</i>
	31.30	<i>re</i>
	⋮	⋮

Fuente: Elaboración propia.

Otro ejemplo, para el primer sujeto real del estudio de la parte cualitativa, S01, los primeros registros se muestran en el cuadro 3.14. Los datos del cuadro indican que a los 12 segundos de iniciada la grabación el sujeto inició la lectura del problema (*le*). Luego, a los 35 segundos de iniciada la grabación comenzó la implementación (*im*) o *codificación* en este caso, es decir que comenzó a escribir código fuente. Después, a los 55 segundos el sujeto procedió a analizar el problema (*an*), es decir a identificar las entradas y las salidas del problema. Luego, exactamente a un minuto de iniciada la grabación, el sujeto da muestras de estar construyendo una estrategia de solución (*so*). Posteriormente, a un minuto con trece segundos de iniciada la grabación, el sujeto retoma la implementación o codificación (*im*). Y así sucesivamente para cada sujeto se registran las transiciones a las diferentes tareas cognitivas que pudieron ser identificadas mediante la grabación de video.

Cuadro 3.14.: Ejemplo de registros en el archivo de datos cualitativos.

Sujeto (columna A)	Tiempo (columna B)	Actividad (columna G)
S01	0.12	le
	0.35	im
	0.55	an
	1.00	so
	1.13	im
	⋮	⋮

Fuente: Elaboración propia.

El registro de cada sujeto termina cuando resuelve el problema (**pr**), o dice haberlo resuelto aún incorrecta o insatisfactoriamente (**pn**), o bien cuando ya no hay muestras claras de poder terminar en el tiempo previsto (**pn**) y el investigador da por terminada la sesión.

Posteriormente, mediante fórmulas de hoja de cálculo, se construyó una columna con la conversión a decimal del tiempo relativo de cada transición calculado a partir del inicio de la lectura del problema (columna F). Esto para poder crear una secuencia de números para construir con ella, por cada sujeto, un gráfico en escala de tiempo decimal en minutos tomando como cero el inicio de la lectura del problema (ver sección 4.2 en la página 167).

A continuación la hoja de cálculo se exportó a formato CSV (`datos-cuali.csv`) sólo con las tres columnas necesarias para poder construir los gráficos (columnas A, F y G). Y finalmente se implementó un programa de computadora que genera los gráficos correspondientes a partir de este archivo. El código fuente de este programa para generar los diagramas está en el Apéndice E en la página 271.

4. Resultados y Análisis de resultados

En este capítulo se presentan los principales resultados de las partes cuantitativa y cualitativa de este trabajo.

4.1. Resultados y Análisis de la Parte Cuantitativa

Además de realizar análisis descriptivos básicos, se realiza un análisis de correlación estadística múltiple entre las variables `matProg`, `matTot`, `cum`, `edad`, `gen`, `PAP` y `RP` definidas en la sección 3.6. Ver figura 4.1.

El proceso general está brevemente descrito en la sección 3.9. El código fuente de los programas para la generación de los gráficos de esta sección se pueden consultar en el Apéndice D en la página 227.

Todos los análisis estadísticos de esta sección se realizaron por medio de instrucciones en el Lenguaje Estadístico R¹ v3.6.3 desarrollado por el R Core Team (2020), aplicado al archivo de datos descrito en la sección 3.8.

Los gráficos de caja, además de incluir la media aritmética muestral (puntos negros) y la mediana de los datos (con una línea horizontal naranja), se han enriquecido con los correspondientes *Gráficos de Violín* de Wickham y Stryjewski (2011) que aportan una idea más clara de la forma de la distribución de los datos, además de la información tradicional que aportan los gráficos de caja.

4.1.1. Análisis descriptivo de las variables

A continuación se presentan las estadísticas descriptivas de las variables indicadas en el cuadro 3.5 en la página 67 y en la figura 4.1:

```
1 > cuanti = read.csv("datos-cuanti.csv", header=TRUE)
2 > #...
3 > summary(cuanti)
4      id      ingreso      edad      cum      matTot
5 ANO1   : 1   Min.    :2002   Min.    :17.00   Min.    :6.700   Min.    : 2.00
6 ANO2   : 1   1st Qu.:2014   1st Qu.:19.00   1st Qu.:7.100   1st Qu.: 4.00
7 ANO3   : 1   Median :2016   Median :21.00   Median :7.475   Median :12.00
8 ANO4   : 1   Mean    :2016   Mean    :20.81   Mean    :7.628   Mean    :12.99
```

¹<https://www.r-project.org/>

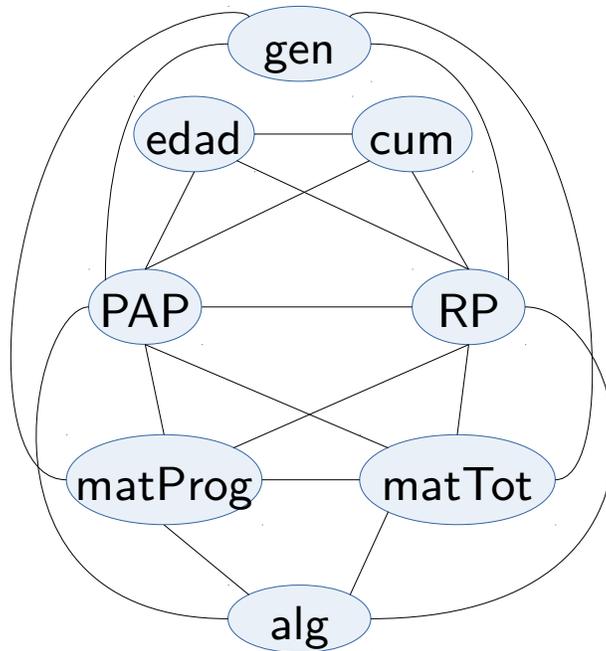


Figura 4.1.: Esquema de las relaciones examinadas entre las variables de la parte cuantitativa.

Fuente: Elaboración propia.

```

9 | AN05 : 1 3rd Qu.:2018 3rd Qu.:22.00 3rd Qu.:8.005 3rd Qu.:20.00
10 | AN06 : 1 Max. :2018 Max. :33.00 Max. :9.430 Max. :33.00
11 | (Other):74
12 | matProg grupo gen alg p1.graf p1.ciu
13 | Min. :0.00 P0:47 f:36 Min. :0.0 Min. : 0.00 Min. : 10.00
14 | 1st Qu.:0.00 P1: 9 m:44 1st Qu.:0.0 1st Qu.: 7.50 1st Qu.: 10.00
15 | Median :0.00 P2:13 Median :0.5 Median : 85.00 Median : 50.00
16 | Mean :0.85 P3:11 Mean :0.5 Mean : 59.06 Mean : 58.31
17 | 3rd Qu.:2.00 3rd Qu.:1.0 3rd Qu.:100.00 3rd Qu.:100.00
18 | Max. :3.00 Max. :1.0 Max. :100.00 Max. :100.00
19 |
20 | p1.carr p1.sol p1.res p3.mod
21 | Min. : 0.00 Min. : 0.00 Min. : 10.0 Min. : 0.00
22 | 1st Qu.: 0.00 1st Qu.: 20.00 1st Qu.: 30.0 1st Qu.: 5.00
23 | Median : 50.00 Median : 75.00 Median : 80.0 Median : 25.00
24 | Mean : 55.06 Mean : 60.38 Mean : 65.5 Mean : 38.05
25 | 3rd Qu.:100.00 3rd Qu.:100.00 3rd Qu.:100.0 3rd Qu.:100.00
26 | Max. :100.00 Max. :100.00 Max. :100.0 Max. :100.00
27 | NA's :16
28 | p3.sol p3.res RP p2.input
29 | Min. : 0.00 Min. : 0.00 Min. : 5.50 Min. : 0.00
30 | 1st Qu.: 15.00 1st Qu.: 0.00 1st Qu.:29.25 1st Qu.: 25.00
31 | Median : 15.00 Median : 10.00 Median :54.50 Median : 50.00
32 | Mean : 38.36 Mean : 38.98 Mean :51.68 Mean : 53.44
33 | 3rd Qu.: 76.25 3rd Qu.: 80.00 3rd Qu.:74.12 3rd Qu.:100.00
34 | Max. :100.00 Max. :100.00 Max. :99.00 Max. :100.00
35 | NA's :16 NA's :16 NA's :3
36 | p2.output p2.idsub p2.solsub p2.intsub
37 | Min. : 0.00 Min. : 0.00 Min. : 0.00 Min. : 0.00
38 | 1st Qu.: 25.00 1st Qu.: 25.00 1st Qu.: 5.00 1st Qu.: 5.00
39 | Median : 50.00 Median : 50.00 Median : 40.00 Median : 50.00
40 | Mean : 50.65 Mean : 58.18 Mean : 43.51 Mean : 50.32
41 | 3rd Qu.:100.00 3rd Qu.:100.00 3rd Qu.: 60.00 3rd Qu.:100.00

```

```

42 Max. :100.00 Max. :100.00 Max. :100.00 Max. :100.00
43 NA's :3 NA's :3 NA's :3 NA's :3
44 p2.norm p2.extre p2.flujo p2.sin
45 Min. : 0.00 Min. : 0.0 Min. : 0.00 Min. : 0.00
46 1st Qu.:10.00 1st Qu.: 0.0 1st Qu.: 5.00 1st Qu.: 10.00
47 Median :10.00 Median : 5.0 Median : 40.00 Median : 10.00
48 Mean : 9.61 Mean : 27.6 Mean : 33.31 Mean : 18.38
49 3rd Qu.:10.00 3rd Qu.: 50.0 3rd Qu.: 40.00 3rd Qu.: 10.00
50 Max. :50.00 Max. :100.0 Max. :100.00 Max. :100.00
51 NA's :3 NA's :3 NA's :3 NA's :3
52 p4.input p4.dibu p4.output p4.idsub
53 Min. : 0.00 Min. : 0.00 Min. : 0.00 Min. : 0.00
54 1st Qu.: 5.00 1st Qu.: 0.00 1st Qu.: 0.00 1st Qu.: 0.00
55 Median : 25.00 Median : 30.00 Median : 25.00 Median : 25.00
56 Mean : 38.03 Mean : 31.97 Mean : 28.87 Mean : 28.59
57 3rd Qu.: 50.00 3rd Qu.: 50.00 3rd Qu.: 50.00 3rd Qu.: 50.00
58 Max. :100.00 Max. :100.00 Max. :100.00 Max. :100.00
59 NA's :9 NA's :9 NA's :9 NA's :9
60 p4.solsub p4.intsub p4.norm p4.extre
61 Min. : 0.00 Min. : 0.0 Min. : 0.000 Min. :0.000
62 1st Qu.: 0.00 1st Qu.: 0.0 1st Qu.:10.000 1st Qu.:5.000
63 Median : 5.00 Median : 5.0 Median :10.000 Median :5.000
64 Mean : 20.77 Mean : 14.3 Mean : 8.028 Mean :4.085
65 3rd Qu.: 50.00 3rd Qu.: 5.0 3rd Qu.:10.000 3rd Qu.:5.000
66 Max. :100.00 Max. :100.0 Max. :10.000 Max. :5.000
67 NA's :9 NA's :9 NA's :9 NA's :9
68 p4.flujo p4.sin PAP
69 Min. : 0.00 Min. : 0.00 Min. : 0.50
70 1st Qu.: 5.00 1st Qu.: 10.00 1st Qu.:15.29
71 Median : 5.00 Median : 10.00 Median :27.38
72 Mean : 20.99 Mean : 17.11 Mean :28.65
73 3rd Qu.: 40.00 3rd Qu.: 10.00 3rd Qu.:40.04
74 Max. :100.00 Max. :100.00 Max. :69.25
75 NA's :9 NA's :9

```

En las siguientes páginas se presentan los gráficos descriptivos básicos de las variables indicadas en el cuadro 3.5 en la página 67. Estos gráficos fueron generados con los programas listados en el Apéndice D en la página 227.

Para las variables edad, cum, año de ingreso, matTot, PAP y RP se presentan histogramas con diferentes anchos de intervalo. Para las variables gen, alg y matProg se presentan los correspondientes gráficos de pastel mostrando el porcentaje y frecuencia absoluta de cada categoría.

En el caso de la edad, figura 4.2 en la página 123, la mayoría de los alumnos están entre los 20 y 22 años con media de 20.81 años y asimetría hacia la derecha.

En el caso del CUM, figura 4.3 en la página 124, la mayoría de los alumnos tiene notas de entre 7.0 y 7.5 con media de 7.628 puntos (en escala de 0.0 a 10.0) y asimetría hacia la derecha.

La variable ingreso no figura como tal entre las variables de interés para la investigación pero se decidió presentarla al menos en el análisis descriptivo. El correspondiente gráfico en la figura 4.4 en la página 125 muestra que la mayoría de los alumnos que completaron el instrumento cuantitativo ingresaron a la universidad entre los años 2014 y 2018, con asimetría muy fuerte hacia la izquierda. Esto debido a la alta desersión que sufren las carreras de la Escuela de Matemática.

El gráfico de la figura 4.5 en la página 126 presenta la proporción de hombres y mujeres analizados en la parte cuantitativa.

Mientras tanto, la figura 4.6 en la página 126 muestra la proporción 1:1 de los instrumentos que solicitaban un algoritmo para los problemas 2 y 4 frente a los que solicitaban una instrucción (y no explícitamente un algoritmo). Esto es así porque los instrumentos fueron hechos en esa misma proporción y fueron entregados a los alumnos de forma alternada. Aunque el código de identificación (columna *id* en el cuadro 3.11 en la página 106) fue asignado hasta que los alumnos entregaron el instrumento ya resuelto y no antes.

En el gráfico de la figura 4.7 en la página 127 se observa una mayoría significativa de estudiantes que todavía no han cursado ninguna materia de programación de computadoras. Esto es debido a la alta tasa de deserción de las carreras analizadas. En el segundo año de las carreras, la cantidad de alumnos inscritos disminuye drásticamente (ver Mallas curriculares en el Apéndice C en la página 225).

Con respecto a la variable *matTot*, figura 4.8 en la página 128, se puede apreciar claramente los cuatro grupos de alumnos por medio de sus modas, 3 y 4 materias para el grupo P0, 12 materias para el grupo P1, 20 materias para el grupo P2, y 27 materias para el grupo P3.

Para la variable PAP, se presentan tres histogramas en la figura 4.9 en la página 129. Esta variable se obtuvo al aplicar la rúbrica presentada en el cuadro 3.9 en la página 91, y se obtuvo un puntaje medio de 28.65 (en escala de 0 a 100, ver cuadro 3.5 en la página 67) y un puntaje máximo de 69.25, con asimetría hacia la derecha.

La figura 4.10 en la página 130 muestra tres histogramas de la variable RP. Esta se obtuvo luego de aplicar la rúbrica presentada en el cuadro 3.10 en la página 100, y se obtuvo un puntaje medio de 54.50 (también en escala de 0 a 100) y un puntaje máximo de 99.0. La tendencia es bastante más platicúrtica, casi uniforme como puede apreciarse al agrandar a 15 puntos el ancho de los intervalos en la figura 4.10 en la página 130, que en el caso de la variable PAP, figura 4.9 en la página 129.

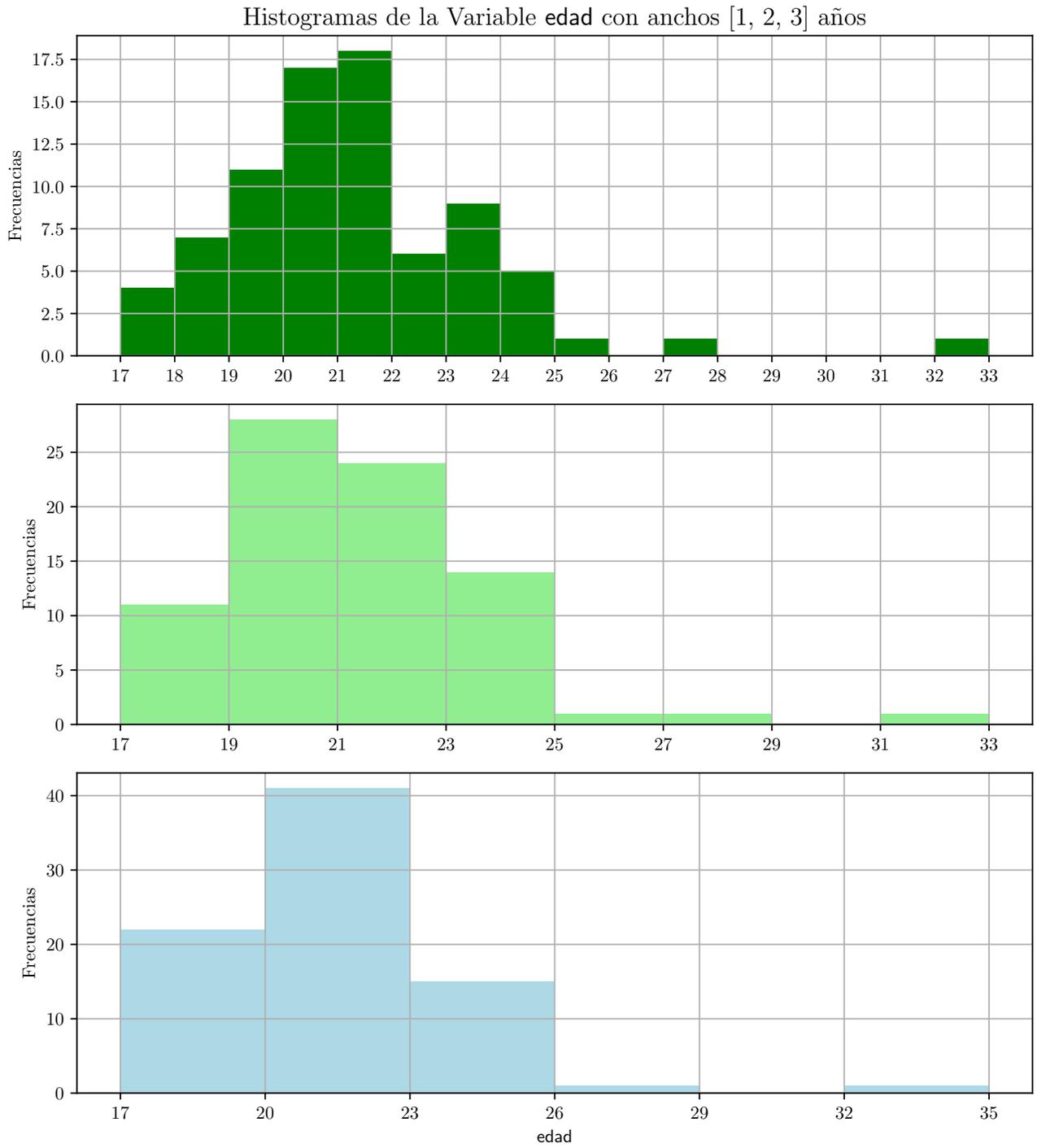


Figura 4.2.: Histogramas de la variable edad.
Fuente: Elaboración propia.

Histogramas de la Variable cum con anchos [0.2, 0.25, 0.5] puntos

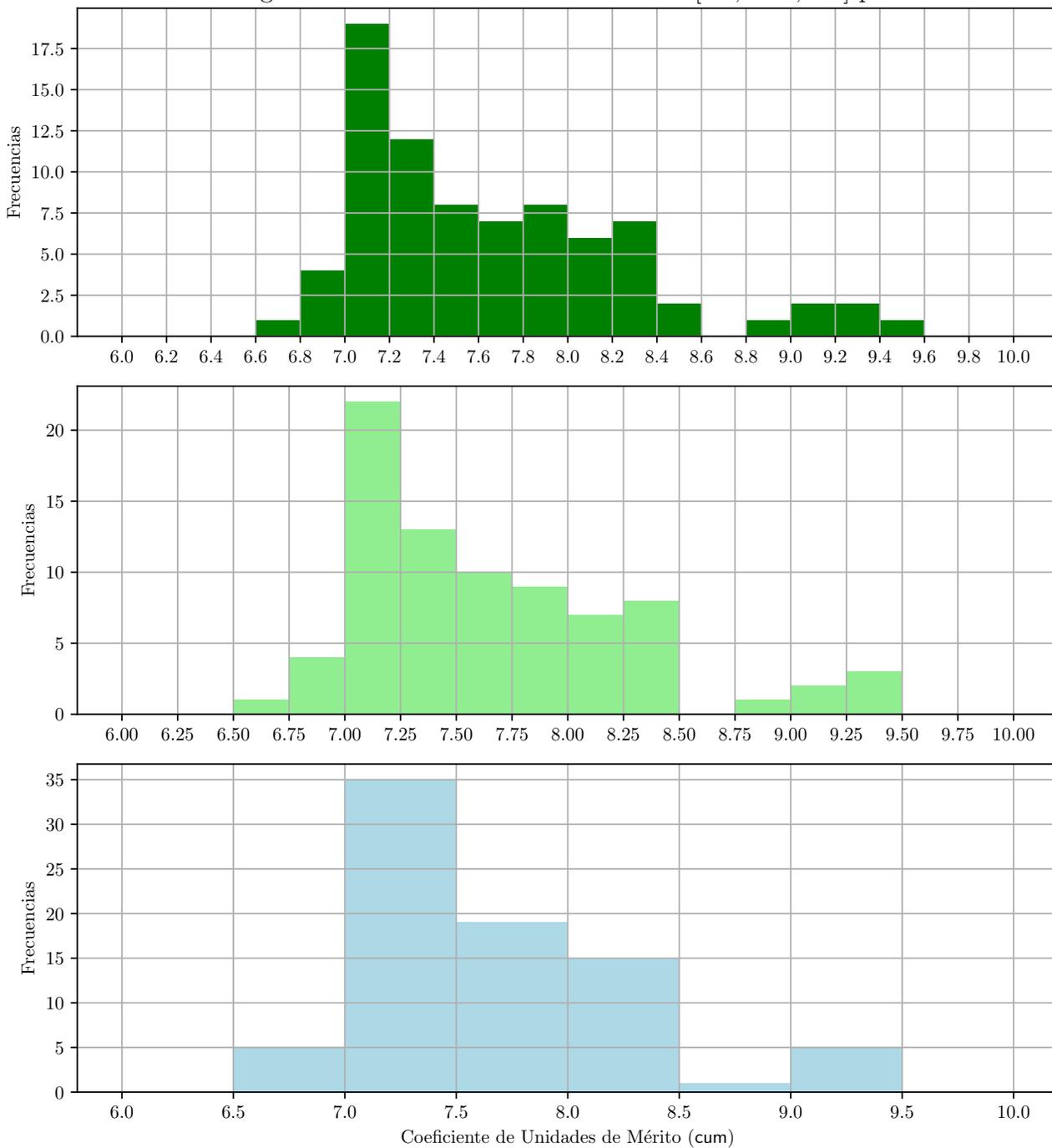


Figura 4.3.: Histogramas de la variable cum.

Fuente: Elaboración propia.

Histogramas de la Variable Año de ingreso con anchos [1, 2, 3] años

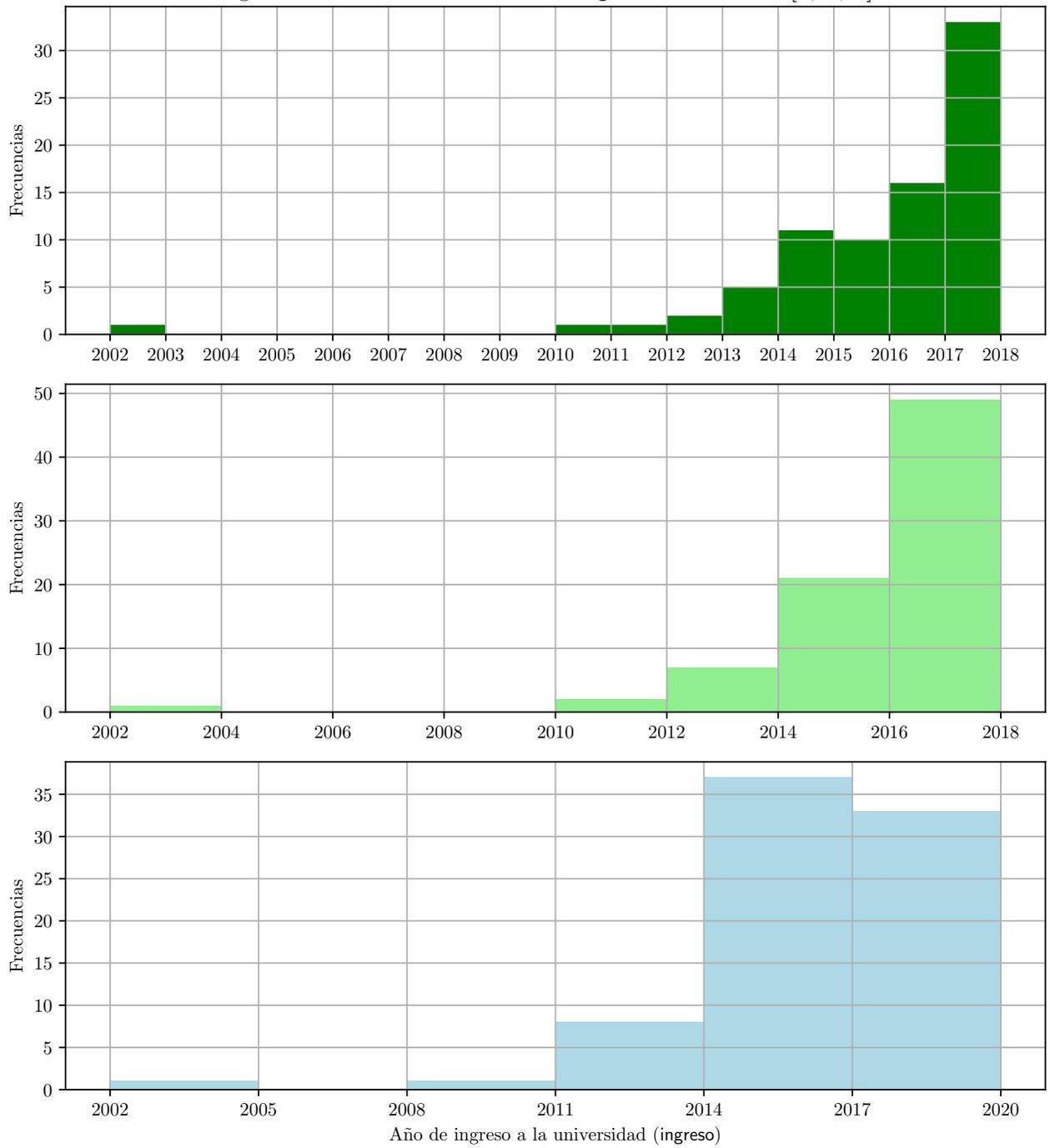


Figura 4.4.: Histogramas de la variable año de ingreso.

Fuente: Elaboración propia.

Proporciones de la Variable gen

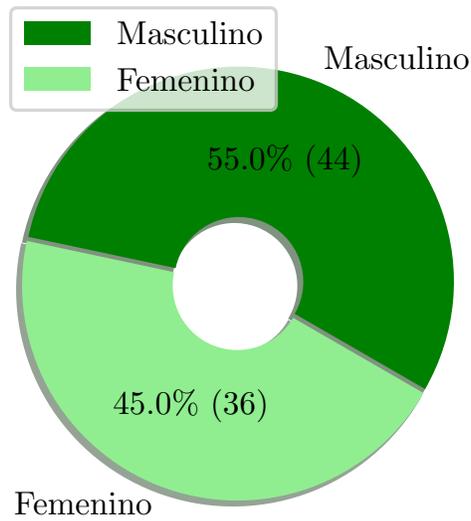


Figura 4.5.: Gráfico de proporciones de la variable gen (Género).
Fuente: Elaboración propia.

Proporciones de la Variable alg

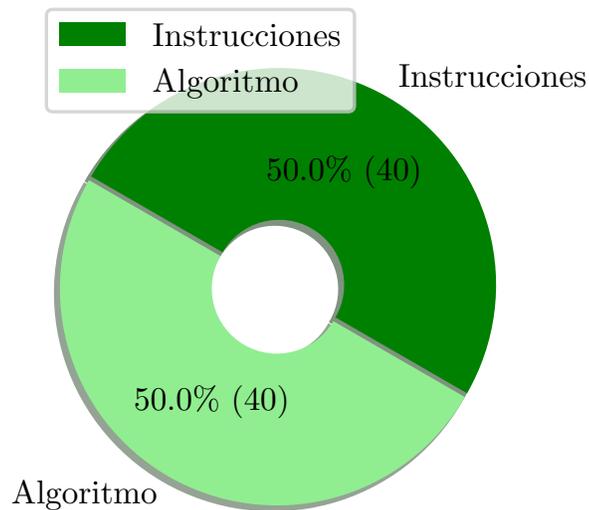


Figura 4.6.: Gráfico de proporciones de la variable alg.
Fuente: Elaboración propia.

Proporciones de la Variable matProg

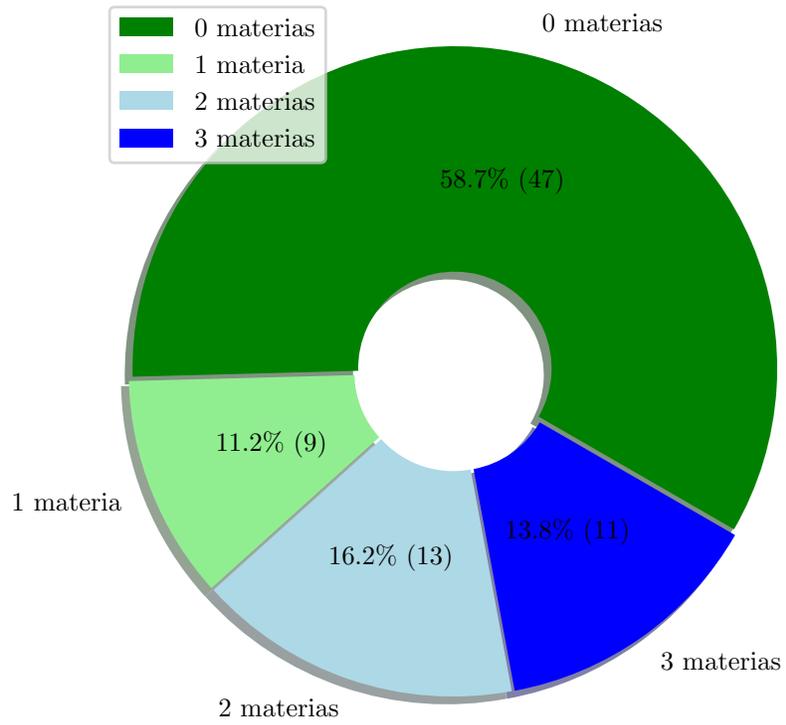


Figura 4.7.: Gráfico de proporciones de la variable matProg.
Fuente: Elaboración propia.

Histogramas de la Variable matTot con anchos [1, 2, 3] materias

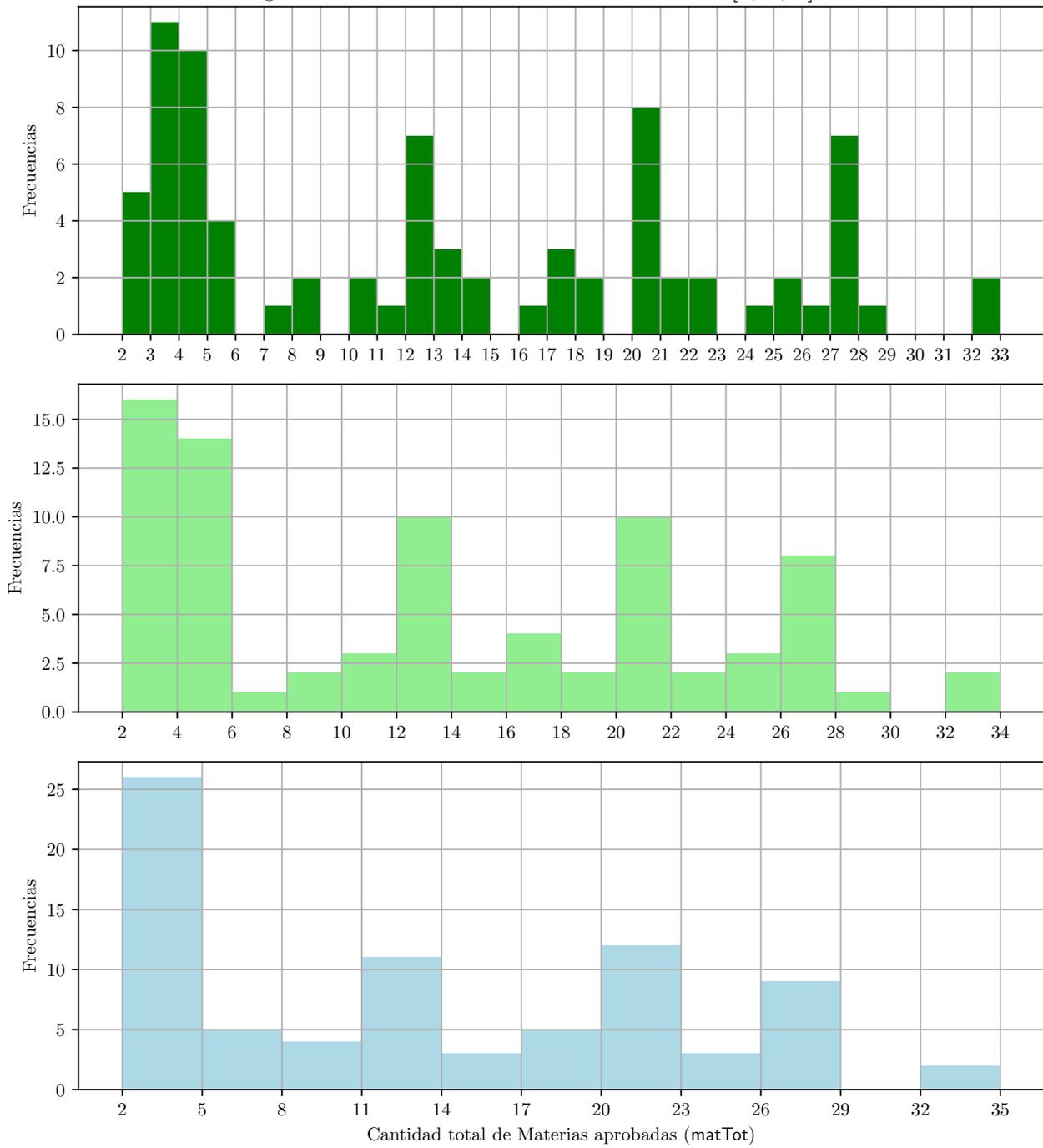


Figura 4.8.: Histogramas de la variable matTot.

Fuente: Elaboración propia.

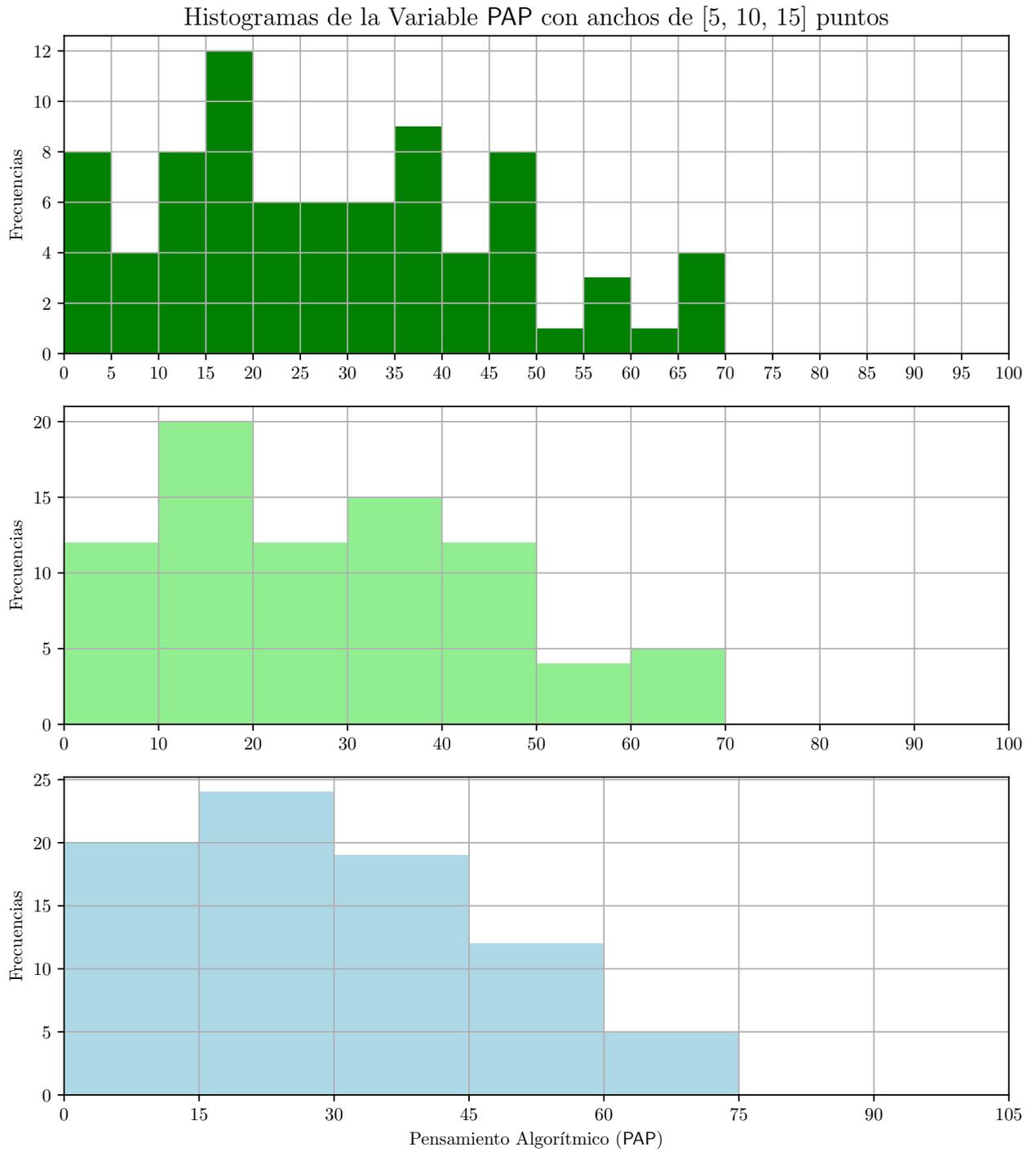


Figura 4.9.: Histograma de la variable PAP.
Fuente: Elaboración propia.

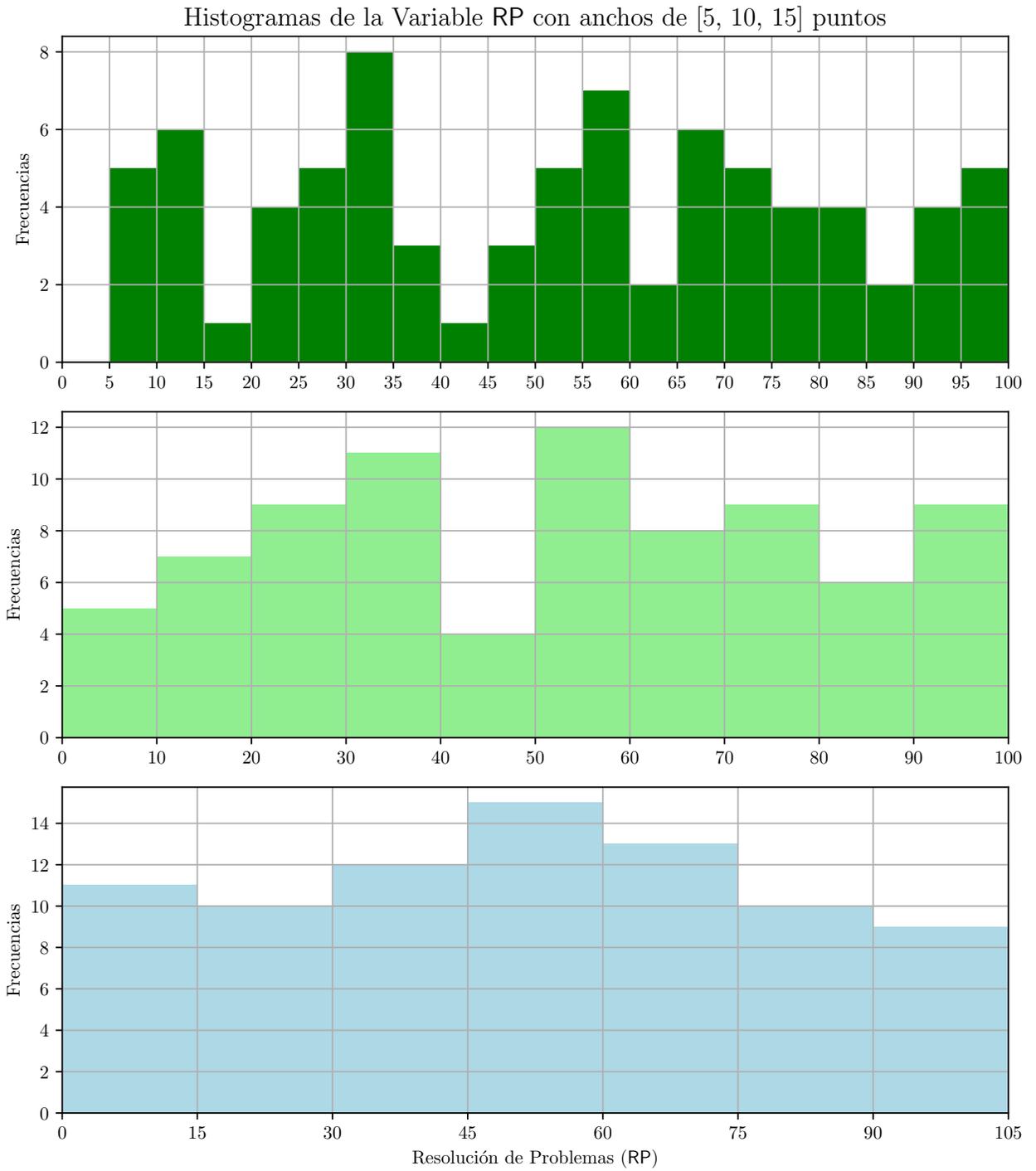


Figura 4.10.: Histograma de la Variable RP.
Fuente: Elaboración propia.

4.1.2. Análisis de normalidad

Siguiendo con el análisis de los datos, se aplicó la prueba de *Lilliefors* (*Kolmogorov-Smirnov* modificado) y también la prueba de *Anderson-Darling* para determinar la «normalidad» de los datos, por medio de las siguientes instrucciones y resumidas en el cuadro 4.1 en la página 133:

```
1 > cuanti = read.csv("datos-cuanti.csv", header=TRUE)
2 > #...
3 > library(nortest) #install.packages("nortest")
4 > lillie.test(cuanti$edad)
5
6     Lilliefors (Kolmogorov-Smirnov) normality test
7
8 data:  cuanti$edad
9 D = 0.1818, p-value = 6.649e-07
10
11 > lillie.test(cuanti$cum)
12
13     Lilliefors (Kolmogorov-Smirnov) normality test
14
15 data:  cuanti$cum
16 D = 0.1111, p-value = 0.01609
17
18 > lillie.test(cuanti$matTot)
19
20     Lilliefors (Kolmogorov-Smirnov) normality test
21
22 data:  cuanti$matTot
23 D = 0.18367, p-value = 4.704e-07
24
25 > lillie.test(cuanti$matProg)
26
27     Lilliefors (Kolmogorov-Smirnov) normality test
28
29 data:  cuanti$matProg
30 D = 0.36007, p-value < 2.2e-16
31
32 > lillie.test(cuanti$RP)
33
34     Lilliefors (Kolmogorov-Smirnov) normality test
35
36 data:  cuanti$RP
37 D = 0.10153, p-value = 0.04042
38
39 > lillie.test(cuanti$PAP)
40
41     Lilliefors (Kolmogorov-Smirnov) normality test
42
43 data:  cuanti$PAP
44 D = 0.10393, p-value = 0.03242
45
46
47 > ad.test(cuanti$edad)
```

```
48
49 Anderson-Darling normality test
50
51 data: cuanti$edad
52 A = 1.8124, p-value = 0.0001126
53
54 > ad.test(cuanti$cum)
55
56 Anderson-Darling normality test
57
58 data: cuanti$cum
59 A = 2.1213, p-value = 1.941e-05
60
61 > ad.test(cuanti$matTot)
62
63 Anderson-Darling normality test
64
65 data: cuanti$matTot
66 A = 2.7276, p-value = 6.232e-07
67
68 > ad.test(cuanti$matProg)
69
70 Anderson-Darling normality test
71
72 data: cuanti$matProg
73 A = 10.012, p-value < 2.2e-16
74
75 > ad.test(cuanti$RP)
76
77 Anderson-Darling normality test
78
79 data: cuanti$RP
80 A = 0.94272, p-value = 0.01618
81
82 > ad.test(cuanti$PAP)
83
84 Anderson-Darling normality test
85
86 data: cuanti$PAP
87 A = 0.80374, p-value = 0.03589
```

Cuadro 4.1.: p-valores de las Pruebas de normalidad para los datos de las variables cuantitativas no dicotómicas.

	Lilliefors	Anderson-Darling
edad	6.649e-07	0.0001126
cum	0.01609	1.941e-05
matTot	4.704e-07	6.232e-07
matProg	< 2.2e-16	< 2.2e-16
RP	0.04042	0.01618
PAP	0.03242	0.03589

Fuente: Elaboración propia.

De todas las pruebas se puede concluir que ninguna de las variables se puede considerar normal, por lo que independientemente de su homocedasticidad y del tamaño de las muestras, todas las pruebas de correlación que se apliquen deberán ser No Paramétricas.

4.1.3. Análisis de correlación

Se realizó inicialmente un gráfico de pares, figura 4.11, por medio de las siguientes instrucciones:

```

1 > cuanti = read.csv("datos-cuanti.csv", header=TRUE)
2 > #...
3 > correlacionables = cuanti[c(3,4,5,6,18,38)]
4 > pdf("pares.eps")
5 > pairs(correlacionables)
6 > dev.off()

```

El resultado muestra que no hay correlaciones lineales muy fuertes y claras. Por ello, y siguiendo las recomendaciones de Khamis (2008) y Hernández Sampieri y cols. (2014, p. 323) sobre los tipos de pruebas estadísticas de correlación para diferentes tipos de variables, se calcularon los coeficientes de correlación de Spearman y Kendall para las mismas variables de la figura 4.11 por medio de las siguientes instrucciones:

```

1 > cuanti = read.csv("datos-cuanti.csv", header=TRUE)
2 > #...
3 > correlacionables = cuanti[c(3,4,5,6,18,38)]
4 > cor(correlacionables, method="spearman")
5          edad      cum      matTot      matProg      RP      PAP
6 edad      1.00000000 0.1671209 0.5077665 0.4727794 0.04084003 0.3369046
7 cum      0.16712094 1.0000000 0.2676265 0.2263455 0.26283694 0.3499065
8 matTot   0.50776651 0.2676265 1.0000000 0.7581221 0.20260838 0.5046972
9 matProg  0.47277936 0.2263455 0.7581221 1.0000000 0.15451139 0.5325480
10 RP      0.04084003 0.2628369 0.2026084 0.1545114 1.00000000 0.3838873
11 PAP     0.33690459 0.3499065 0.5046972 0.5325480 0.38388728 1.0000000
12 > cor(correlacionables, method="kendall")
13          edad      cum      matTot      matProg      RP      PAP
14 edad      1.00000000 0.1198375 0.3930174 0.3977837 0.03105781 0.2548434

```

15	cum	0.11983754	1.0000000	0.1809434	0.1760477	0.18350722	0.2390371
16	matTot	0.39301741	0.1809434	1.0000000	0.6307473	0.13300209	0.3587413
17	matProg	0.39778370	0.1760477	0.6307473	1.0000000	0.11869377	0.4189794
18	RP	0.03105781	0.1835072	0.1330021	0.1186938	1.00000000	0.2597942
19	PAP	0.25484338	0.2390371	0.3587413	0.4189794	0.25979421	1.0000000

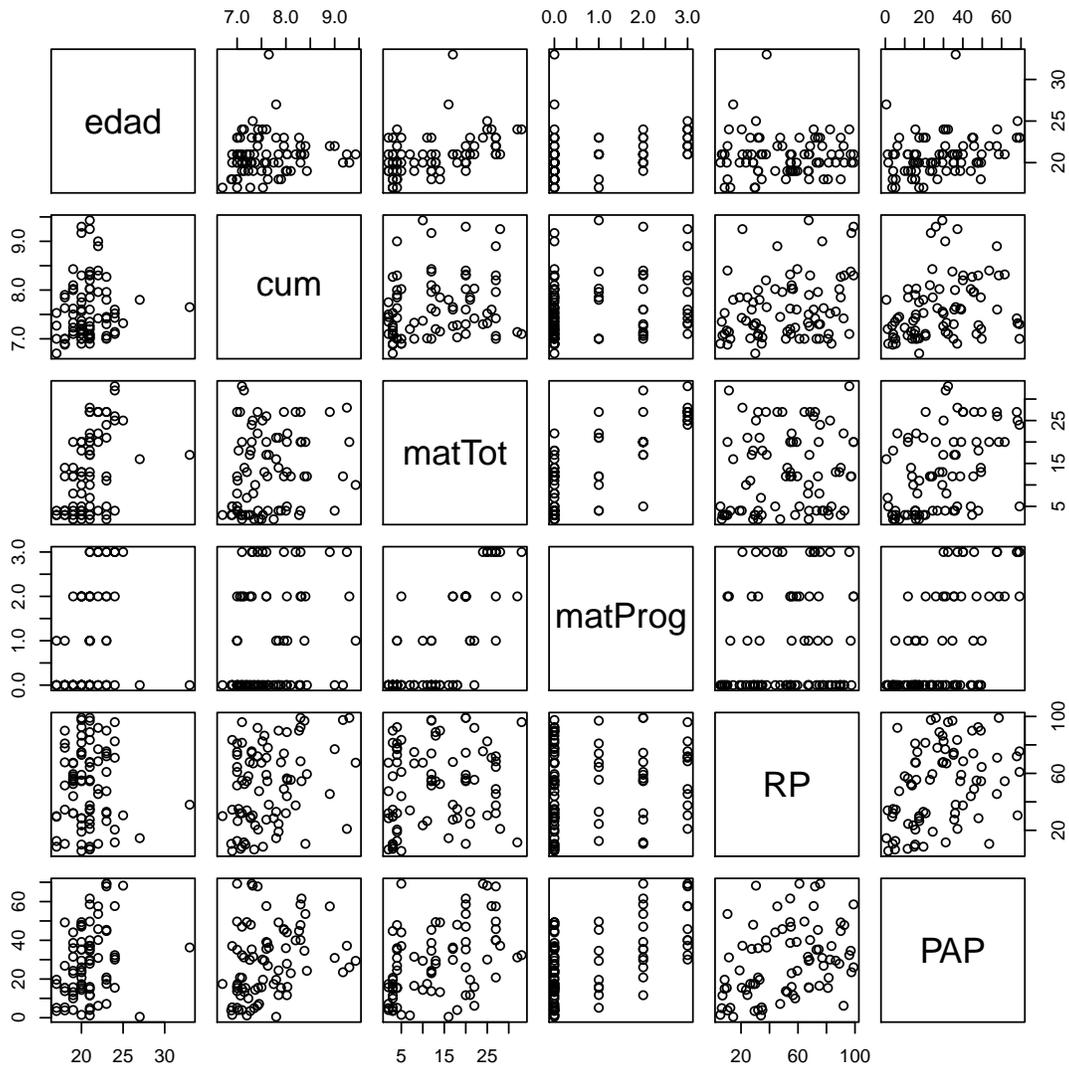


Figura 4.11.: Correlación entre las principales variables continuas y ordinales.
Fuente: Elaboración propia.

A partir de estos resultados y con el código [D.4 en la página 229](#), se construyó un mapa de calor, figura [4.12](#), con los coeficientes de correlación de Spearman, y los coeficientes de correlación de Kendall. Abajo de la diagonal principal, en tonos rojos, los coeficientes de Spearman. Arriba de la diagonal principal, en tonos azules, los coeficientes de Kendall.

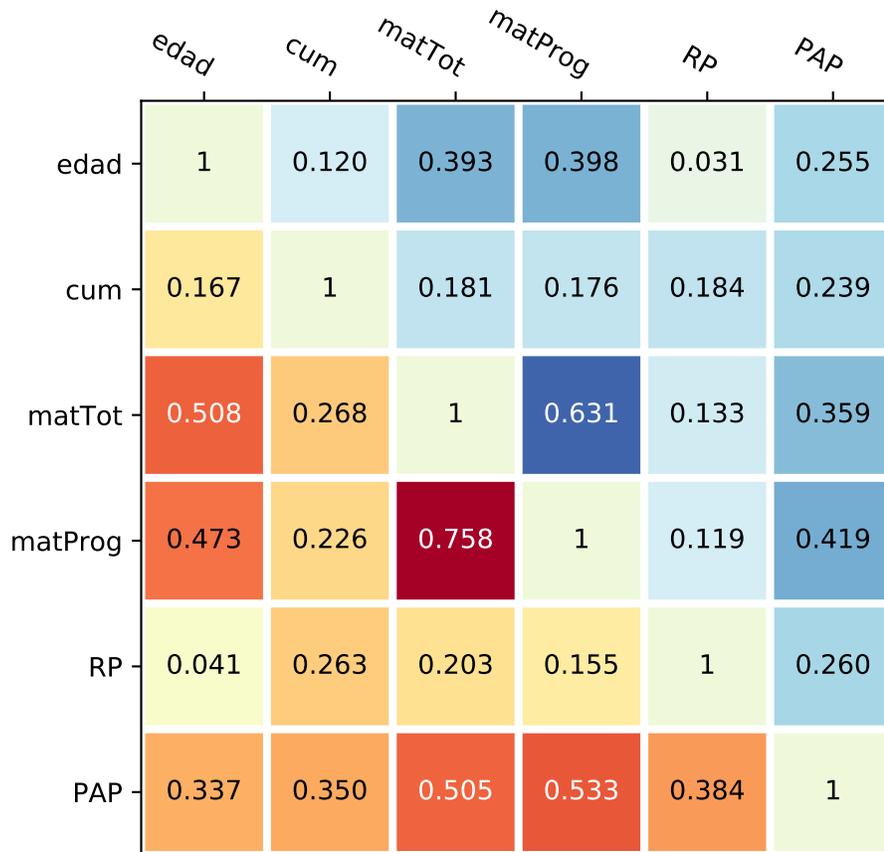


Figura 4.12.: Mapa de Calor de los Coeficientes de Correlación de Spearman y Kendall entre las principales variables continuas y ordinales.

Fuente: Elaboración propia.

Es digno de mencionar que todas estas correlaciones en la figura 4.12, aunque algunas muy débiles, son todas positivas, es decir, que cuando una aumenta, las demás también aumentan, al menos un poco.

Cantidad de Materias cursadas

El resultado de las correlaciones entre las variables de razón, en la figura 4.12, muestra que la correlación más fuerte es entre las variables matTot y matProg (ver figura 4.14 en la página 139). Y le siguen en fuerza, la correlación entre la variable PAP con matProg (4.17 en la página 142) tanto como PAP con matTot (ver figura 4.32 en la página 155). Esto confirma *moderadamente* las hipótesis 1 y 2 de esta investigación (ver sección 1.5 en la página 7).

Aunque sí se puede observar en la figura 4.12 que la correlación entre PAP y matProg (ver figura 4.17 en la página 142) es ligeramente más fuerte que entre PAP y matTot (ver figura 4.32 en la página 155), lo cual tiene sentido porque son las materias de programación de

computadoras las que deberían aumentar el nivel de *Pensamiento Algorítmico*. Similarmente la correlación entre RP y matTot (ver figura 4.43 en la página 165) es ligeramente más fuerte que entre RP y matProg (ver figura 4.40 en la página 162), lo cual también es natural.

CUM

Por otro lado, destacan en la figura 4.12 lo débiles que son las correlaciones entre el cum y todas las demás variables (ver figuras 4.15 en la página 140, 4.26 en la página 150 y 4.35 en la página 158); y también entre la variable RP con todas las demás, incluyendo con matTot y matProg (ver figuras 4.43 en la página 165 y 4.40 en la página 162).

Estas bajas correlaciones entre el cum y todas las variables, particularmente la baja correlación con PAP (ver figura 4.26 en la página 150) refuta la hipótesis 3 de esta investigación (ver sección 1.5 en la página 7).

Edad

Además se observa en la figura 4.12 en la página anterior una moderada correlación natural entre la edad y las variables matTot y matProg, que lo único que indican es que los alumnos avanzan entre las materias a un ritmo más o menos uniforme.

Sin embargo, en la figura 4.12 se observa que la edad está muy debilmente correlacionada con PAP (ver figura 4.27 en la página 151), por lo que se refuta la hipótesis 4 de esta investigación (ver sección 1.5 en la página 7).

VARIABLES DICOTÓMICAS

Así mismo, siguiendo las recomendaciones de Khamis (2008) sobre los tipos de pruebas estadísticas de correlación para diferentes tipos de variables, se calculó el coeficiente de correlación biserial puntual para las variables dicotómicas (gen y alg) con las demás, por medio de las siguientes instrucciones y resumidos en el cuadro 4.2:

```
1 > cuanti = read.csv("datos-cuanti.csv", header=TRUE)
2 > #...
3 > library(ltm) #install.packages("ltm")
4 Loading required package: MASS
5 Loading required package: msm
6 Loading required package: polycor
7 > biserial.cor(cuanti$edad, cuanti$alg)
8 [1] -0.01550124
9 > biserial.cor(cuanti$edad, cuanti$gen)
10 [1] -0.06491387
11 > biserial.cor(cuanti$cum, cuanti$alg)
12 [1] 0.06255053
13 > biserial.cor(cuanti$cum, cuanti$gen)
14 [1] 0.1098843
15 > biserial.cor(cuanti$matTot, cuanti$alg)
```

```

16 [1] -0.04811984
17 > biserial.cor(cuanti$matTot, cuanti$gen)
18 [1] 0.1670571
19 > biserial.cor(cuanti$matProg, cuanti$alg)
20 [1] -0.0221187
21 > biserial.cor(cuanti$matProg, cuanti$gen)
22 [1] 0.1200427
23 > biserial.cor(cuanti$RP, cuanti$alg)
24 [1] 0.006927953
25 > biserial.cor(cuanti$RP, cuanti$gen)
26 [1] 0.08591706
27 > biserial.cor(cuanti$PAP, cuanti$alg)
28 [1] 0.0266372
29 > biserial.cor(cuanti$PAP, cuanti$gen)
30 [1] 0.1352629

```

Cuadro 4.2.: Coeficientes de correlación biserial puntual.

	alg	gen
edad	-0.01550124	-0.06491387
cum	0.06255053	0.1098843
matTot	-0.04811984	0.1670571
matProg	-0.0221187	0.1200427
RP	0.006927953	0.08591706
PAP	0.0266372	0.1352629

Fuente: Elaboración propia.

Todos estos coeficientes muestran correlaciones muy débiles o prácticamente nulas.

Género

Es muy baja la correlación entre PAP y gen (ver figura 4.31 en la página 154), además, para verificar la diferencia entre los géneros, se aplicó la prueba U de Mann-Whitney entre estas variables por medio de las siguientes instrucciones:

```

1 > cuanti = read.csv("datos-cuanti.csv", header=TRUE)
2 > #...
3 > wilcox.test(cuanti$PAP ~ cuanti$gen, paired=F, alternative="two.
  sided")
4
5     Wilcoxon rank sum test with continuity correction
6
7 data:  cuanti$PAP by cuanti$gen
8 W = 934, p-value = 0.1712
9 alternative hypothesis: true location shift is not equal to 0

```

Y como el p-valor es mayor que 0.05, se confirma la hipótesis 5 de esta investigación (ver sección 1.5 en la página 7). Aunque los gráficos de las figuras 4.28 en la página 152 y 4.30 en

la página 153 parecieran indicar que las mujeres obtuvieron mejor desempeño, este es un efecto resultado de que la cantidad de hombres y de mujeres es diferente. La corrección correspondiente se visualiza en la figura 4.31 en la página 154 donde se confirma lo que los cálculos estadísticos indican, que no hay diferencias significativas entre el grupo de hombres y el de mujeres.

En el cuadro 4.2 se observa que es aún más baja la correlación entre RP y gen (ver figura 4.37 en la página 160 y 4.39 en la página 161). Aunque en la gráfica de la figura 4.38 en la página 160 también pareciera, igual que entre PAP y gen, que las mujeres tienen mayor desempeño, esto es un efecto causado por la diferente cantidad de hombre y mujeres. Esto se corrige en la figura 4.39 en la página 161.

Esto confirma la independencia del género con los resultados de las variables PAP y RP.

¿Algoritmo o Instrucciones?

Particularmente destacable en el cuadro 4.2 es la baja correlación entre PAP y alg (ver figura 4.24 en la página 149), además, se aplicó la prueba U de Mann-Whitney entre estas variables por medio de las siguientes instrucciones para saber si había diferencia entre los dos grupos:

```
1 > cuanti = read.csv("datos-cuanti.csv", header=TRUE)
2 > #...
3 > wilcox.test(cuanti$PAP ~ cuanti$alg, paired=F, alternative="two.
   sided")
4
5     Wilcoxon rank sum test with continuity correction
6
7 data:  cuanti$PAP by cuanti$alg
8 W = 831.5, p-value = 0.7655
9 alternative hypothesis: true location shift is not equal to 0
```

Y como el p-valor es mucho mayor que 0.05, se refuta la hipótesis 6 de esta investigación (ver sección 1.5 en la página 7).

También es casi nula la correlación entre RP y alg (ver figura 4.34 en la página 157), como era de esperarse. No tiene nada que ver una indicación de los problemas 2 y 4 con la resolución de los problemas 1 y 3.

Resumen de correlaciones

En la figura 4.13 se resumen los principales valores de correlación entre las variables, con los correspondientes coeficientes de correlación de Spearman y Biseriales según la figura 4.12 y el cuadro 4.2 respectivamente.

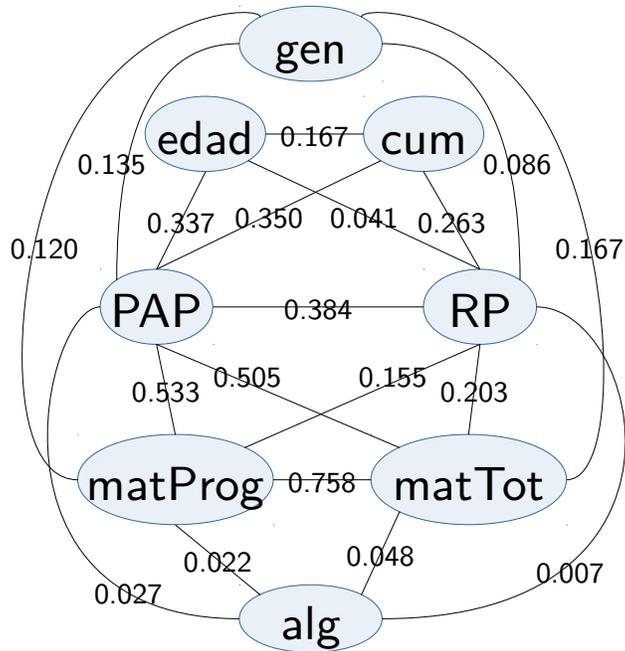


Figura 4.13.: Esquema de las correlaciones entre las variables de la parte cuantitativa. Fuente: Elaboración propia.

4.1.4. Gráficos de correlación entre las diferentes variables de la parte cuantitativa

En las siguientes páginas se presentan los gráficos que muestran las correlaciones de las variables indicadas en los cuadros 3.5 en la página 67, 3.6 en la página 74 y 3.7 en la página 76. Estos gráficos fueron generados con los programas listados en el Apéndice D en la página 227.

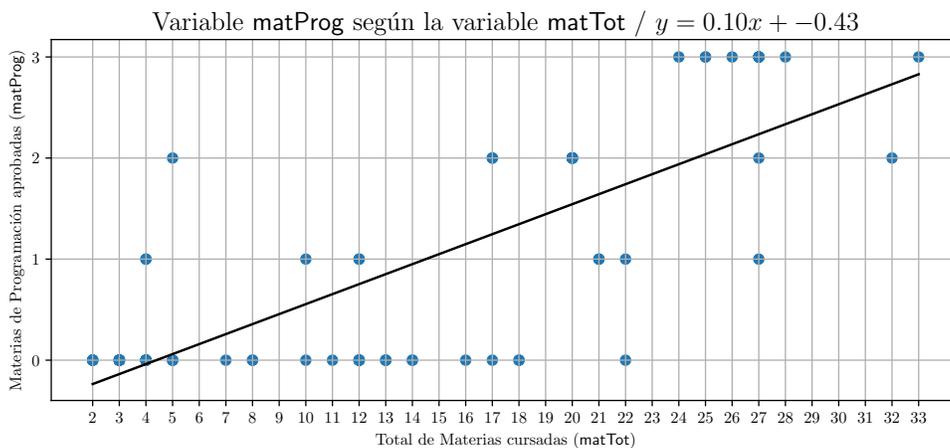


Figura 4.14.: Correlación entre matProg y matTot. Fuente: Elaboración propia.

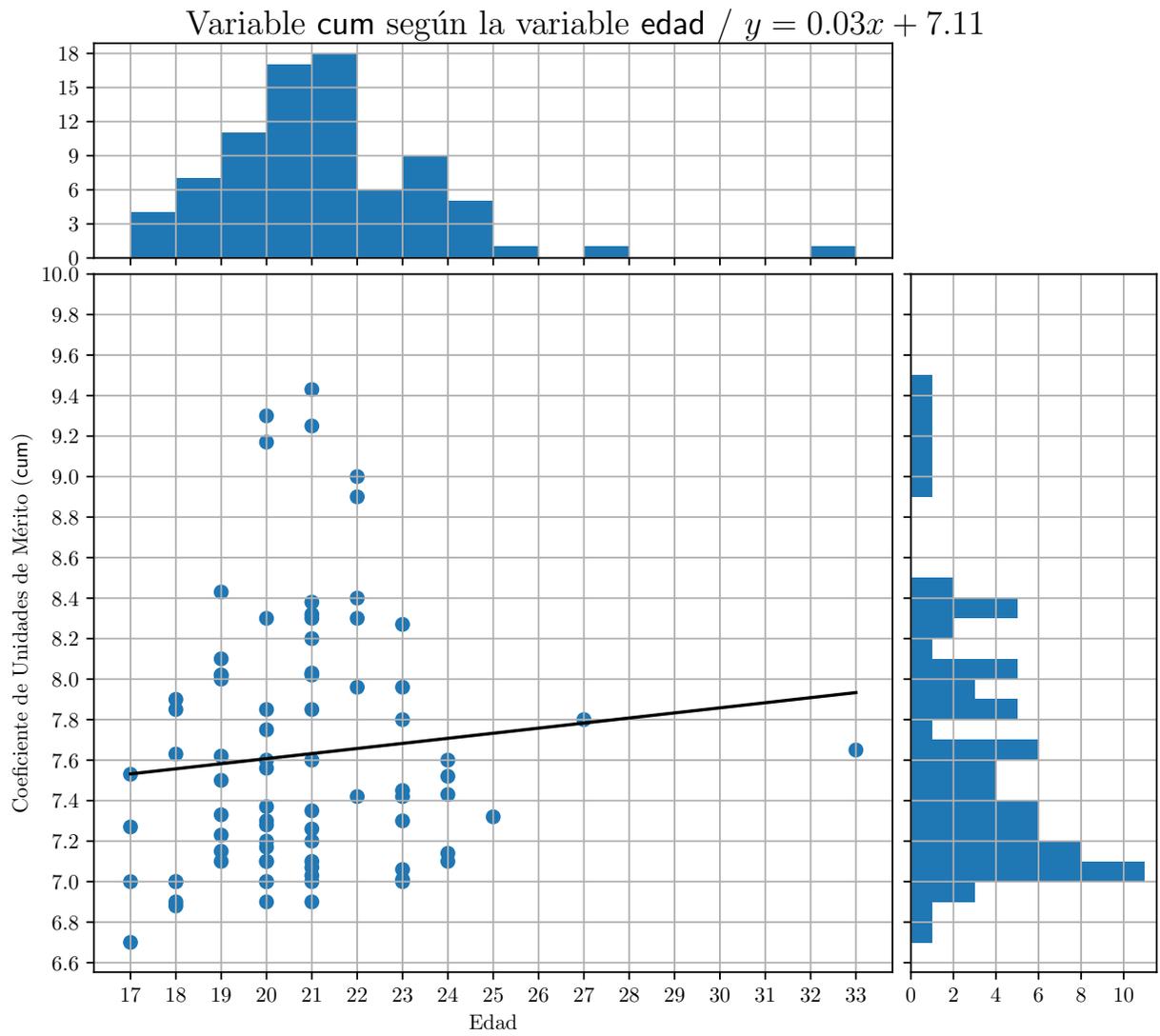


Figura 4.15.: Correlación entre las variables cum y edad.
Fuente: Elaboración propia.

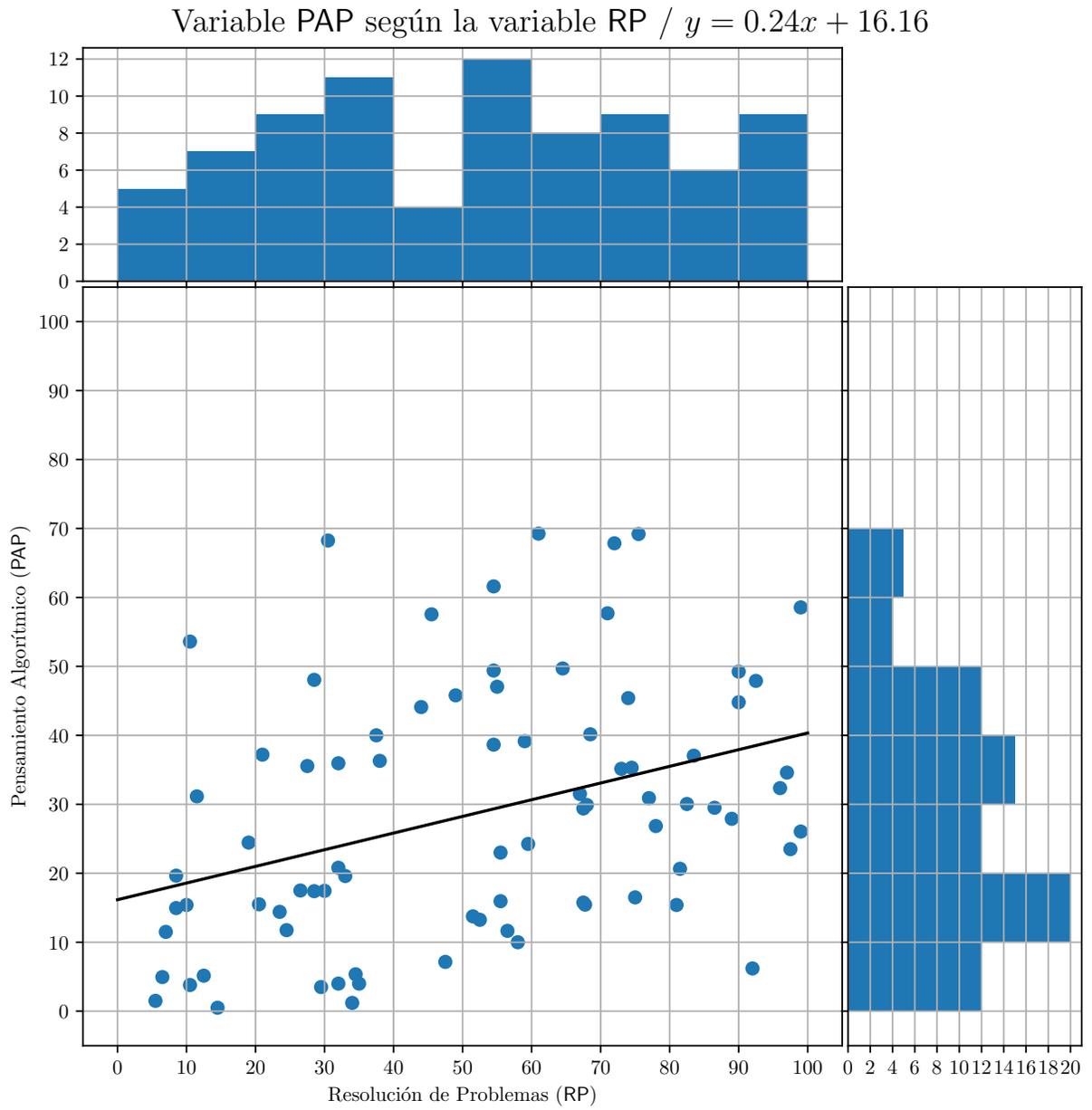


Figura 4.16.: Correlación entre PAP y RP.

Fuente: Elaboración propia.

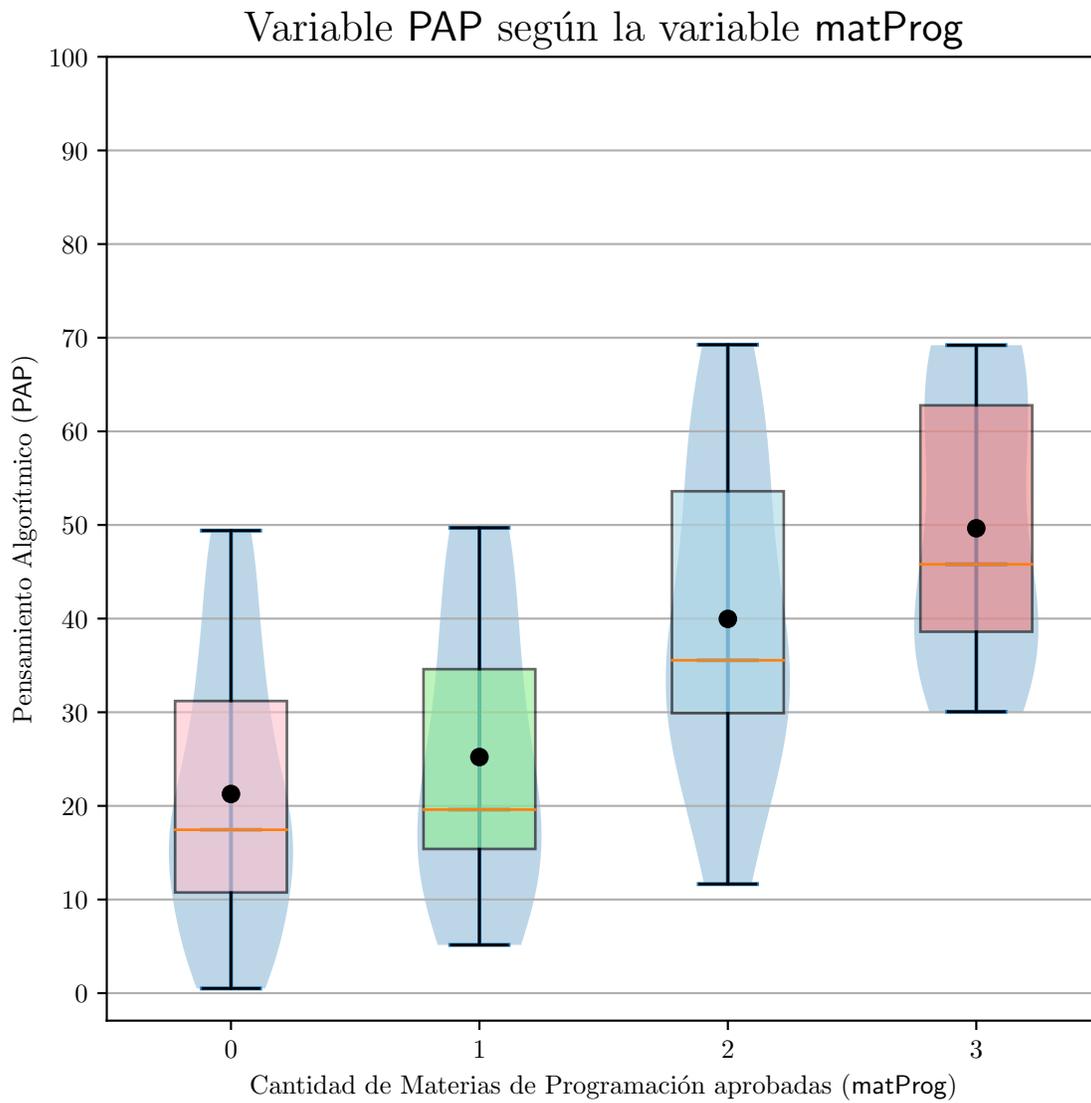


Figura 4.17.: Variable PAP según matProg.
Fuente: Elaboración propia.

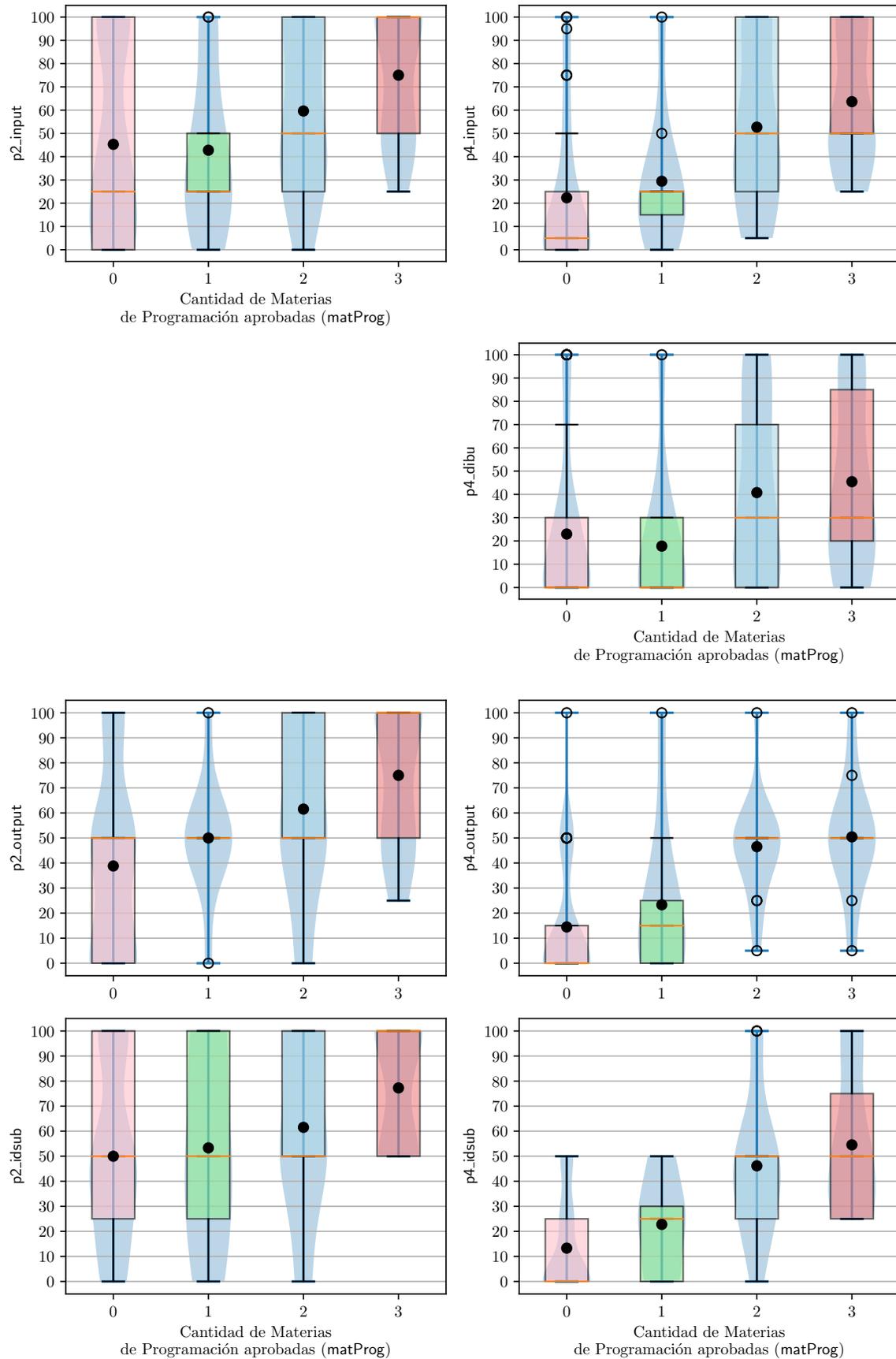


Figura 4.18.: Detalle de las sub-variables de PAP según matProg — Parte 1.

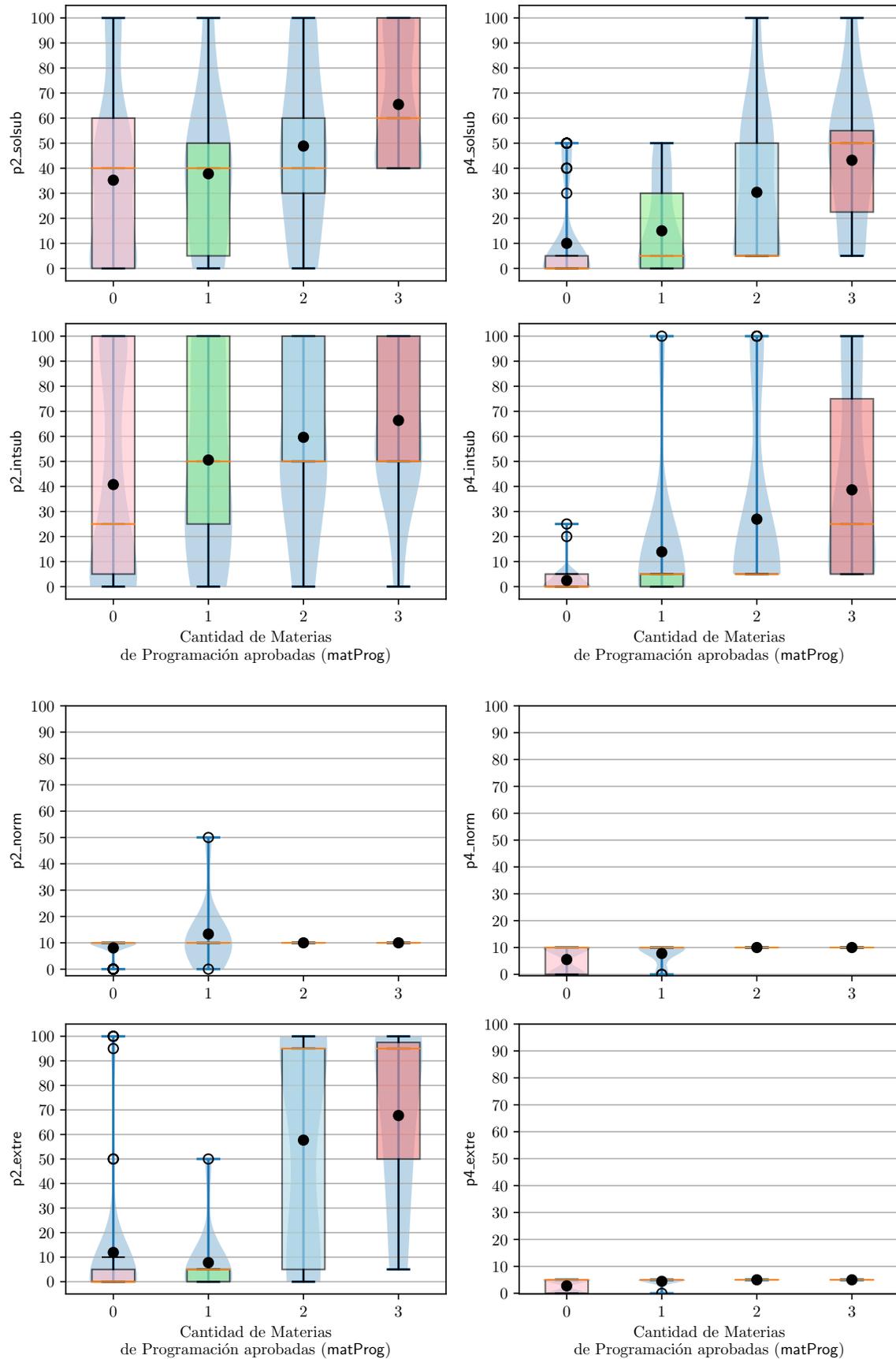


Figura 4.19.: Detalle de las sub-variables de PAP según matProg — Parte 2.

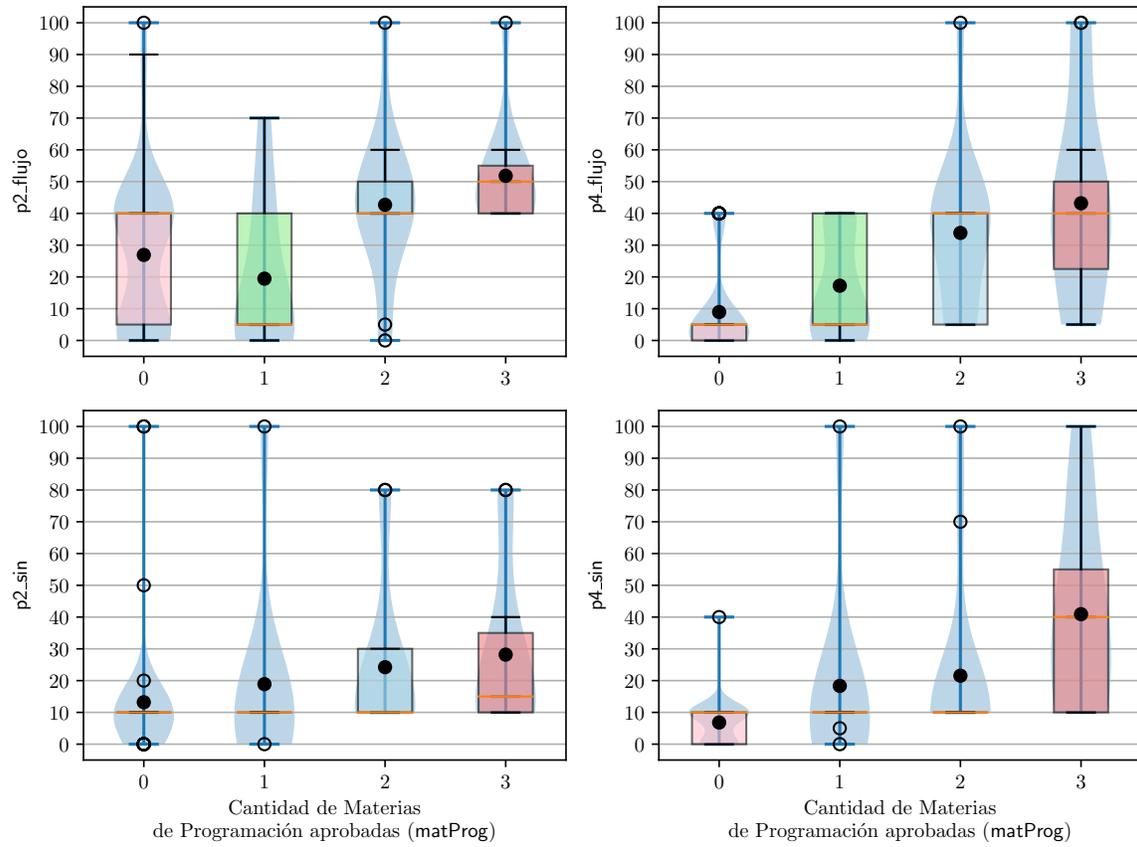


Figura 4.20.: Detalle de las sub-variables de PAP según matProg — Parte 3.
Fuente: Elaboración propia.

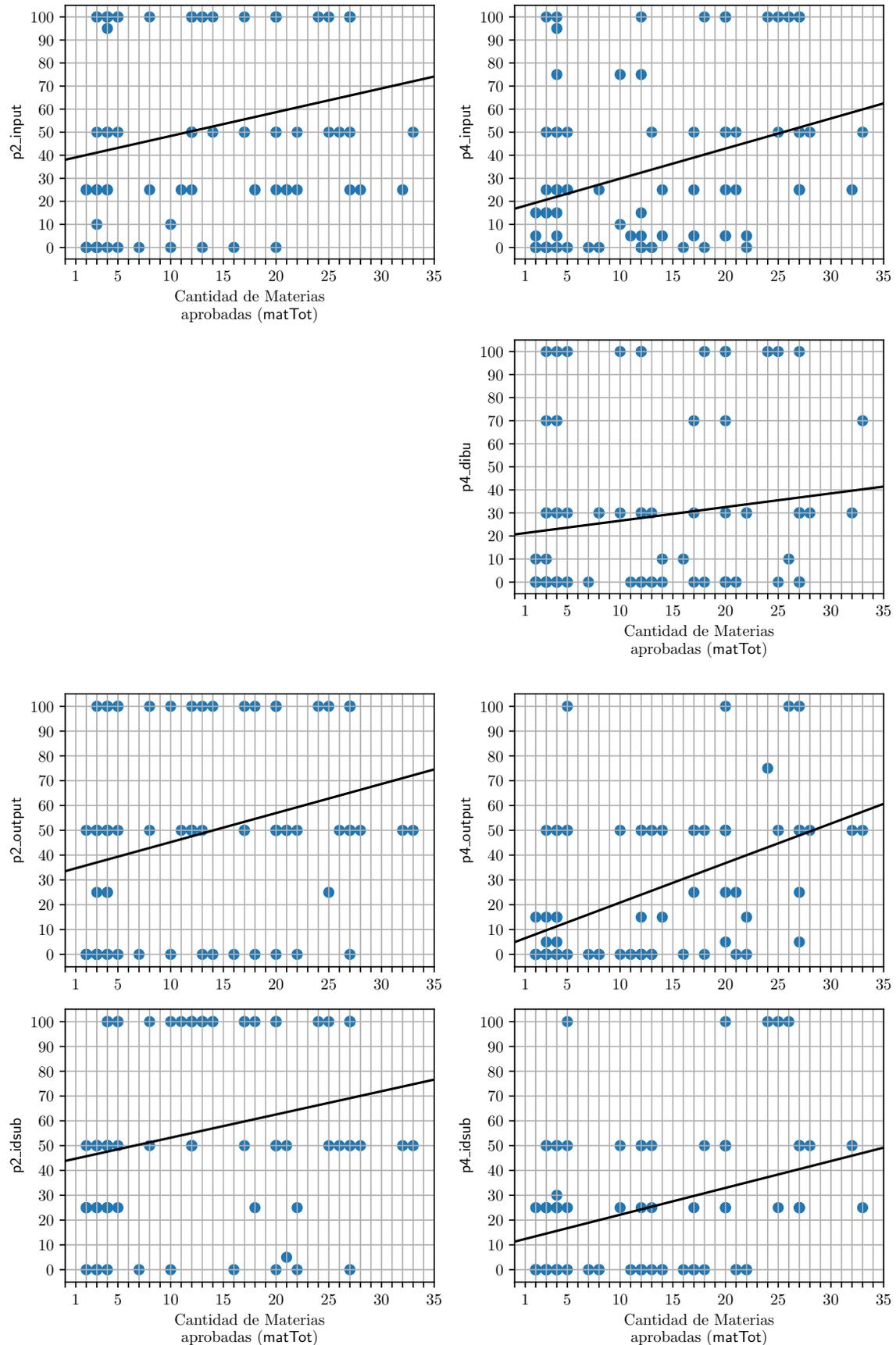


Figura 4.21.: Detalle de las sub-variables de PAP según matTot — Parte 1.

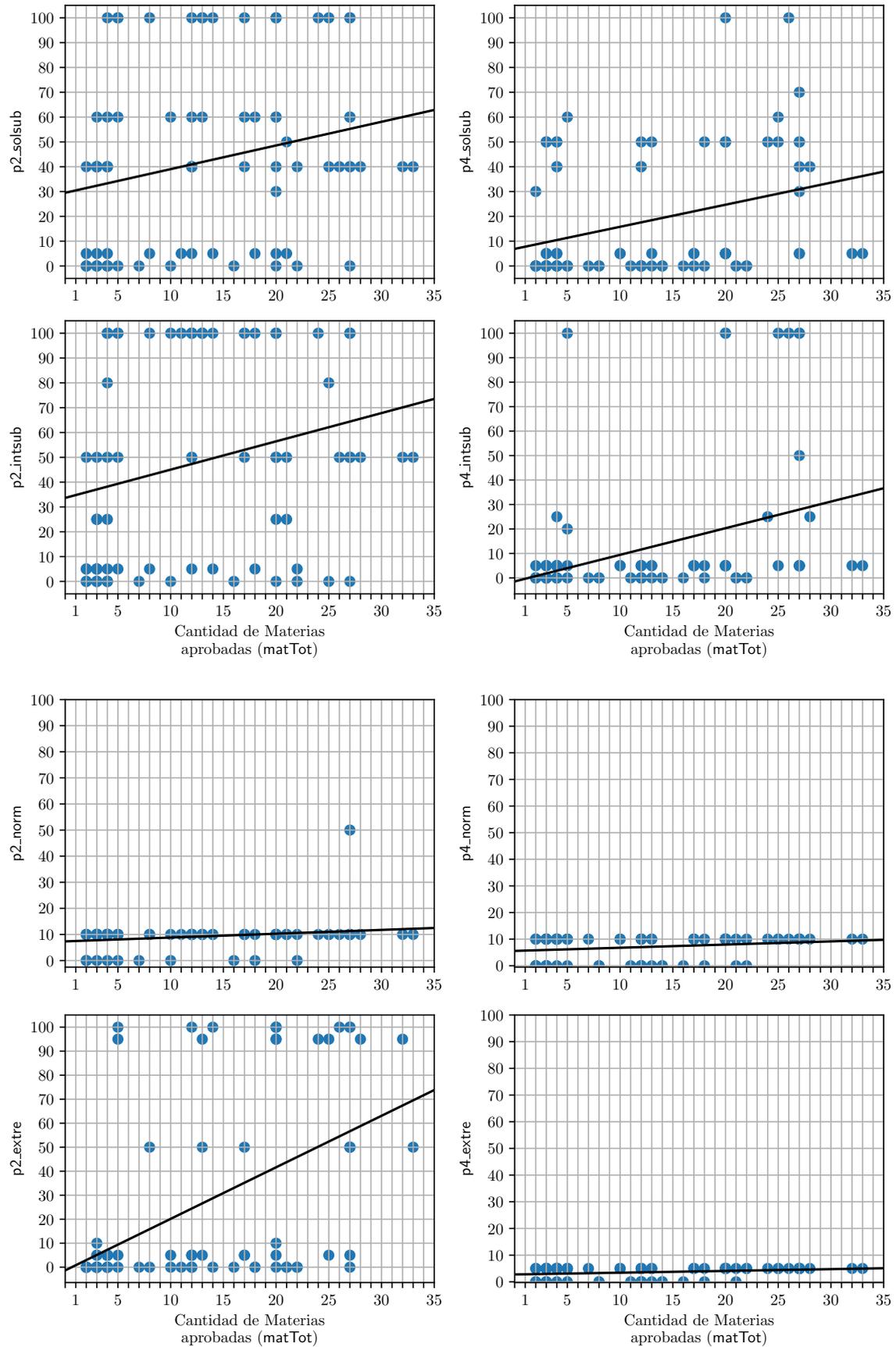


Figura 4.22.: Detalle de las sub-variables de PAP según matTot — Parte 2.

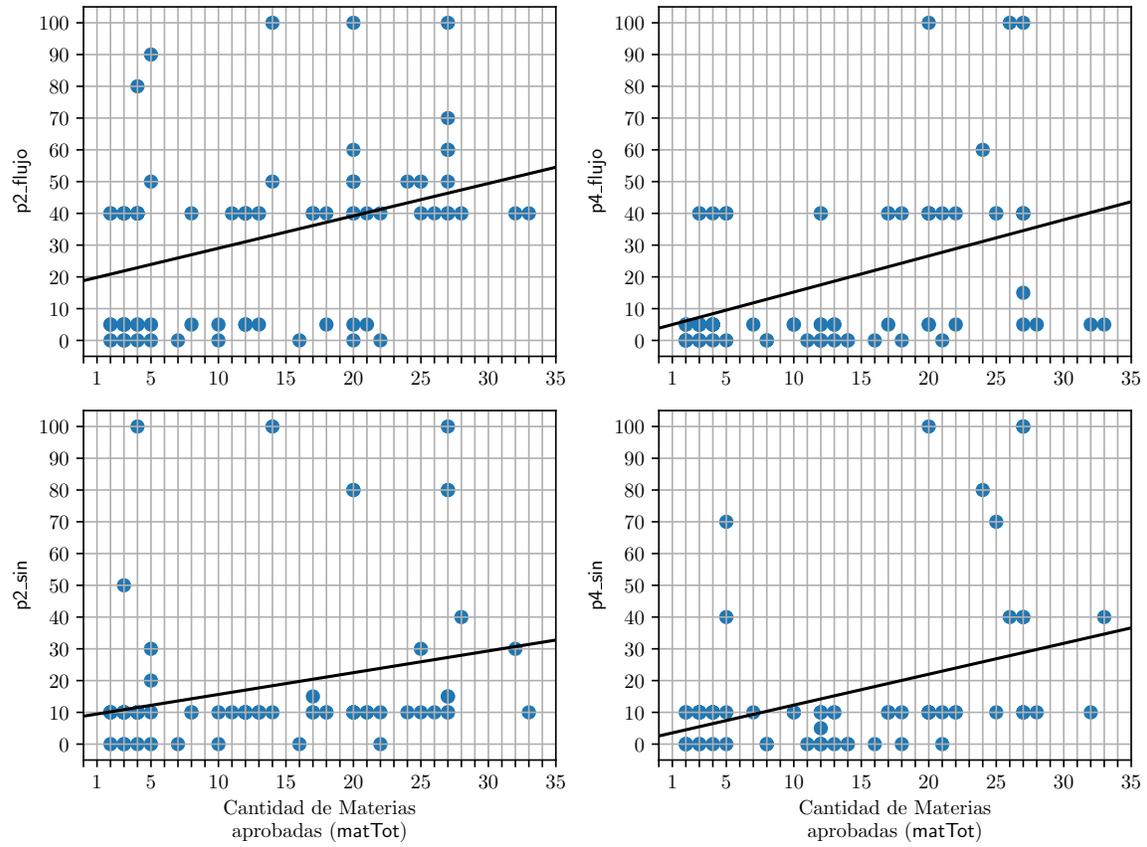


Figura 4.23.: Detalle de las sub-variables de PAP según matTot — Parte 3.
Fuente: Elaboración propia.

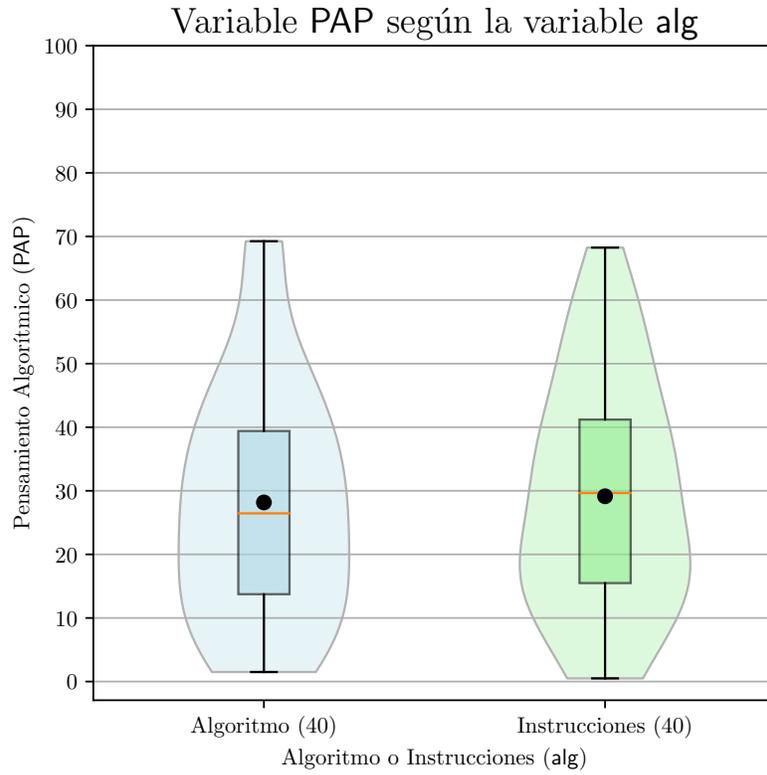


Figura 4.24.: Variable PAP según la variable alg.
Fuente: Elaboración propia.

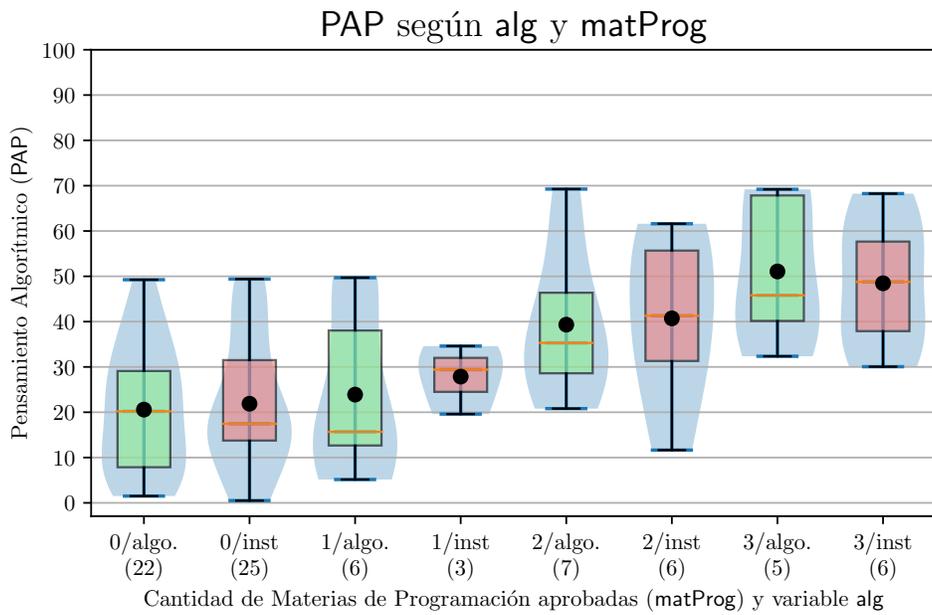


Figura 4.25.: Variable PAP según matProg y alg.
Fuente: Elaboración propia.

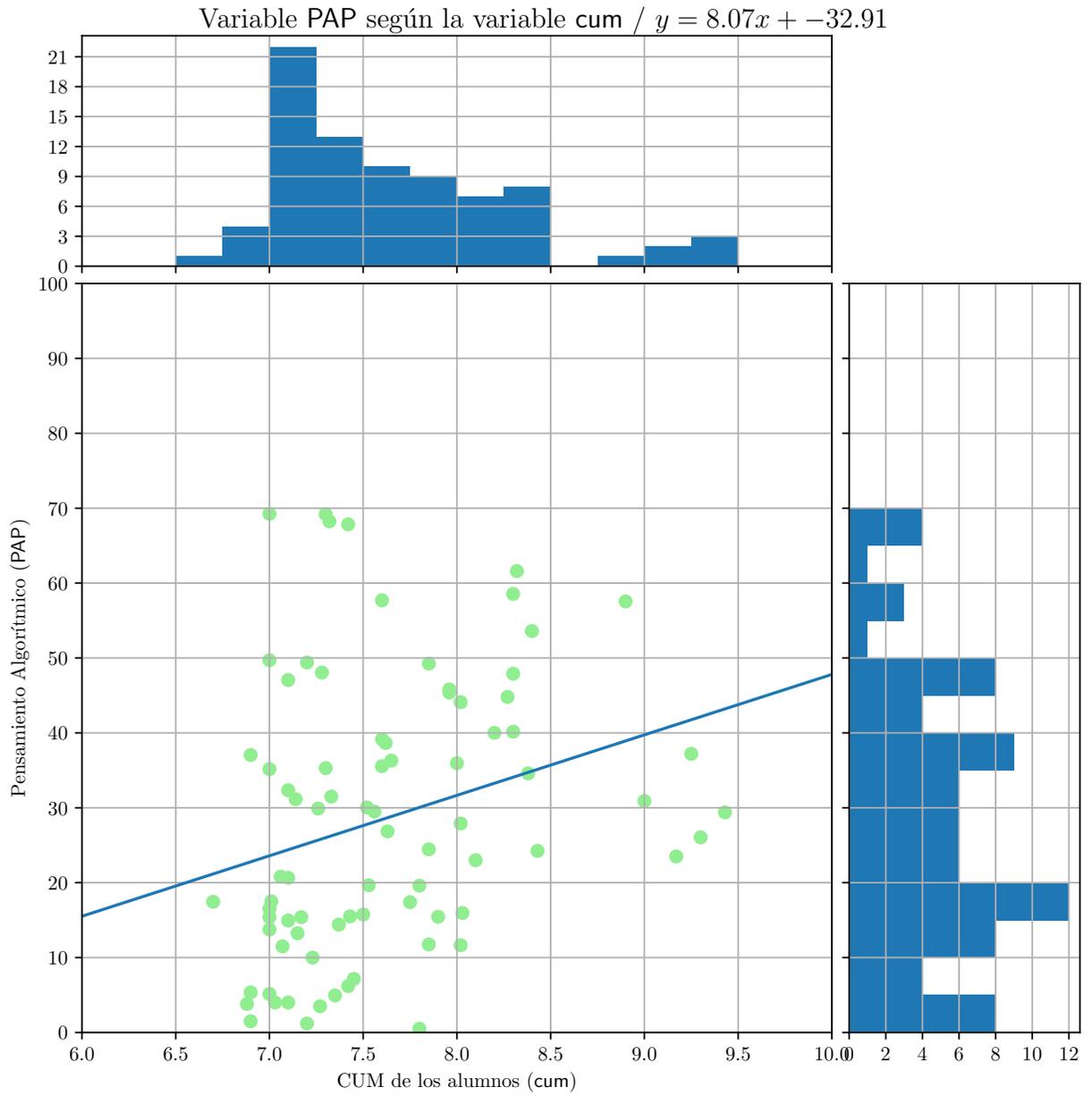


Figura 4.26.: Variable PAP según cum.

Fuente: Elaboración propia.

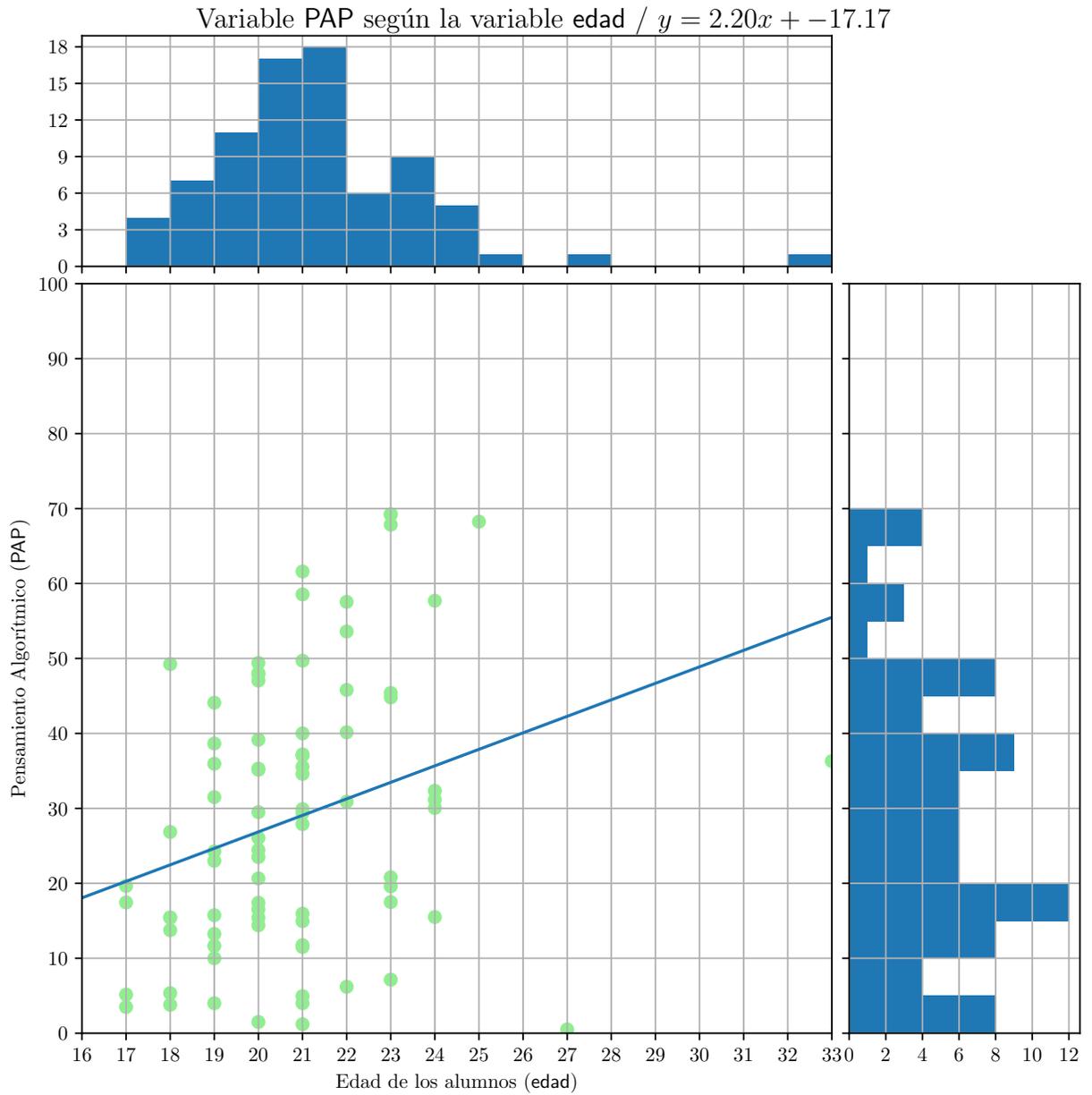


Figura 4.27.: Variable PAP según edad.

Fuente: Elaboración propia.

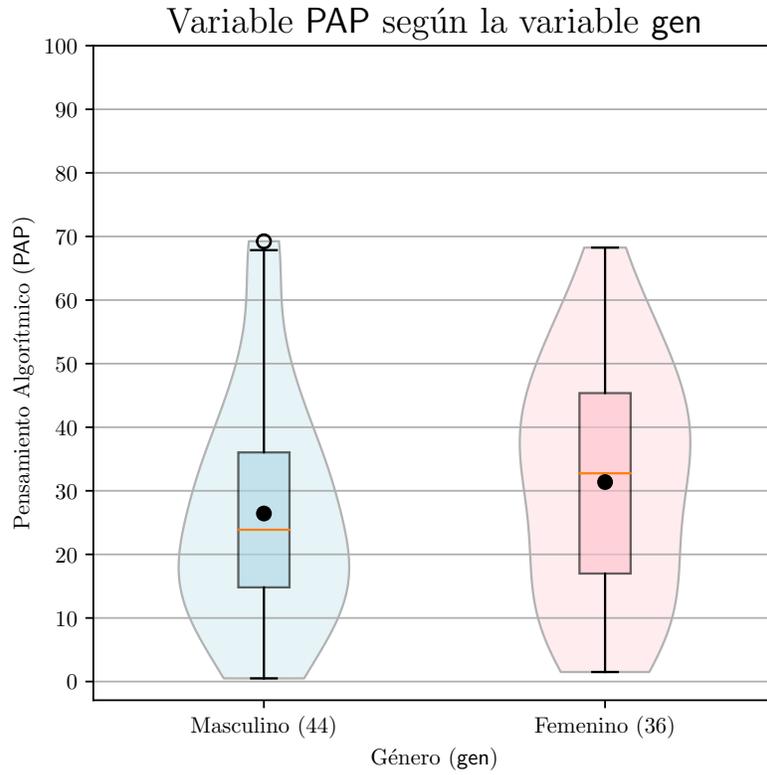


Figura 4.28.: Variable PAP según gen.

Fuente: Elaboración propia.

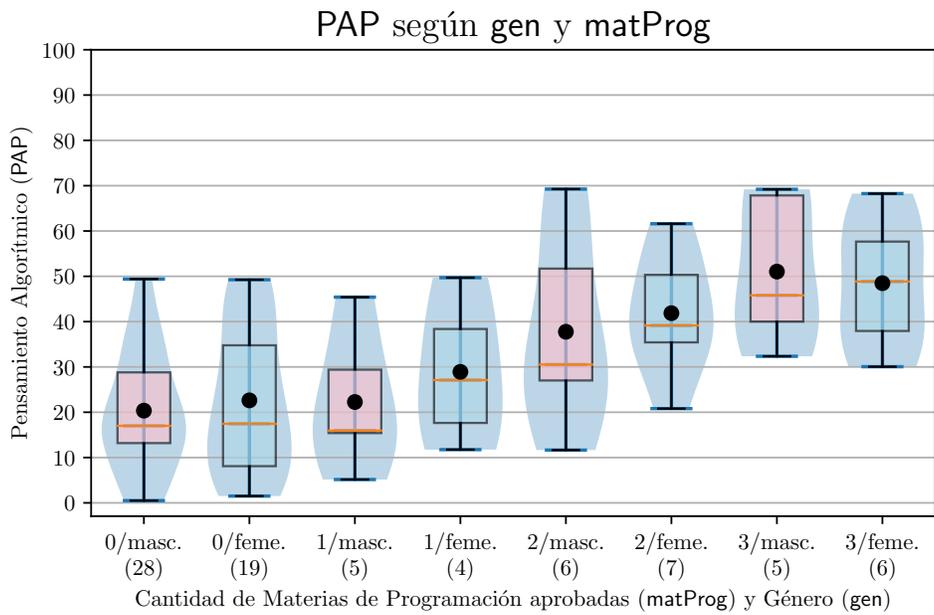


Figura 4.29.: Variable PAP según matProg y gen.

Fuente: Elaboración propia.

Variable PAP según la variable gen
en orden ascendente

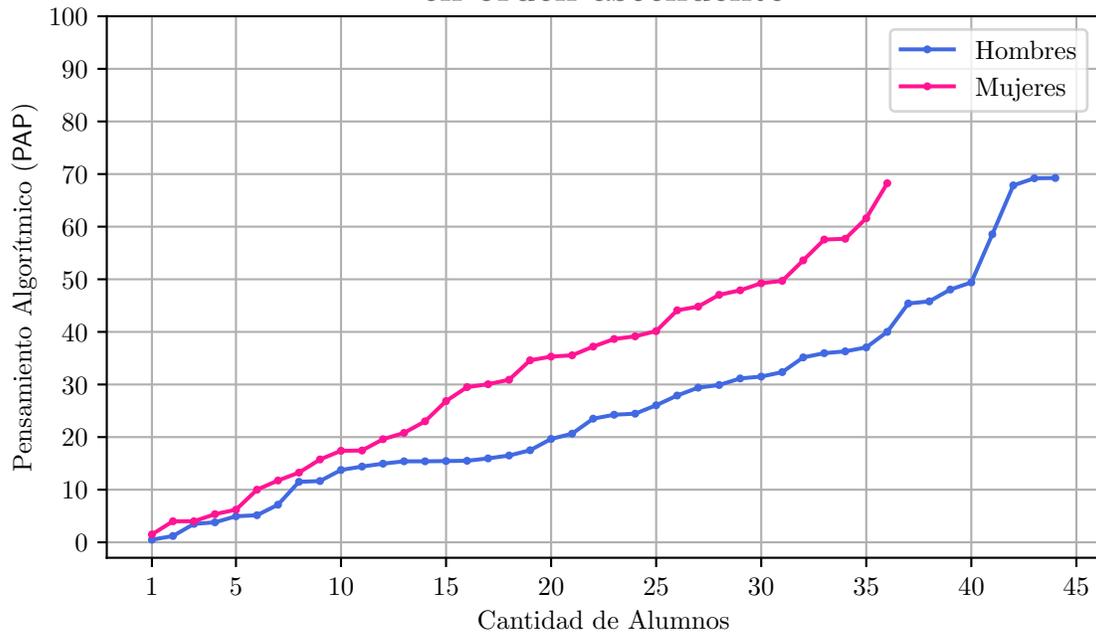


Figura 4.30.: Variable PAP según gen en orden ascendente.

Fuente: Elaboración propia.

Variable PAP según la variable gen
en orden ascendente
con corrección de escala

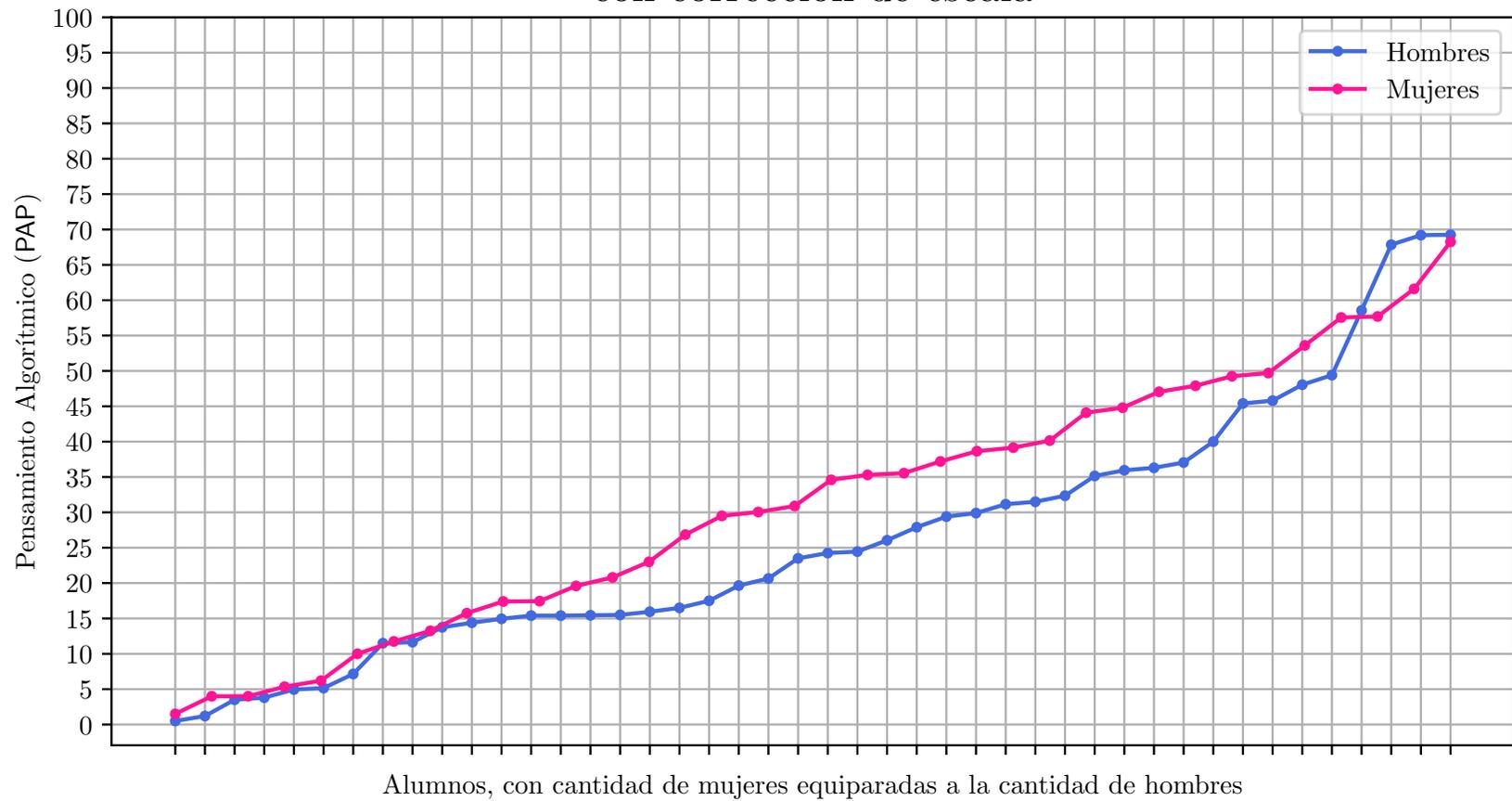


Figura 4.31.: Variable PAP según gen en orden ascendente con corrección en la escala horizontal de mujeres.

Fuente: Elaboración propia.

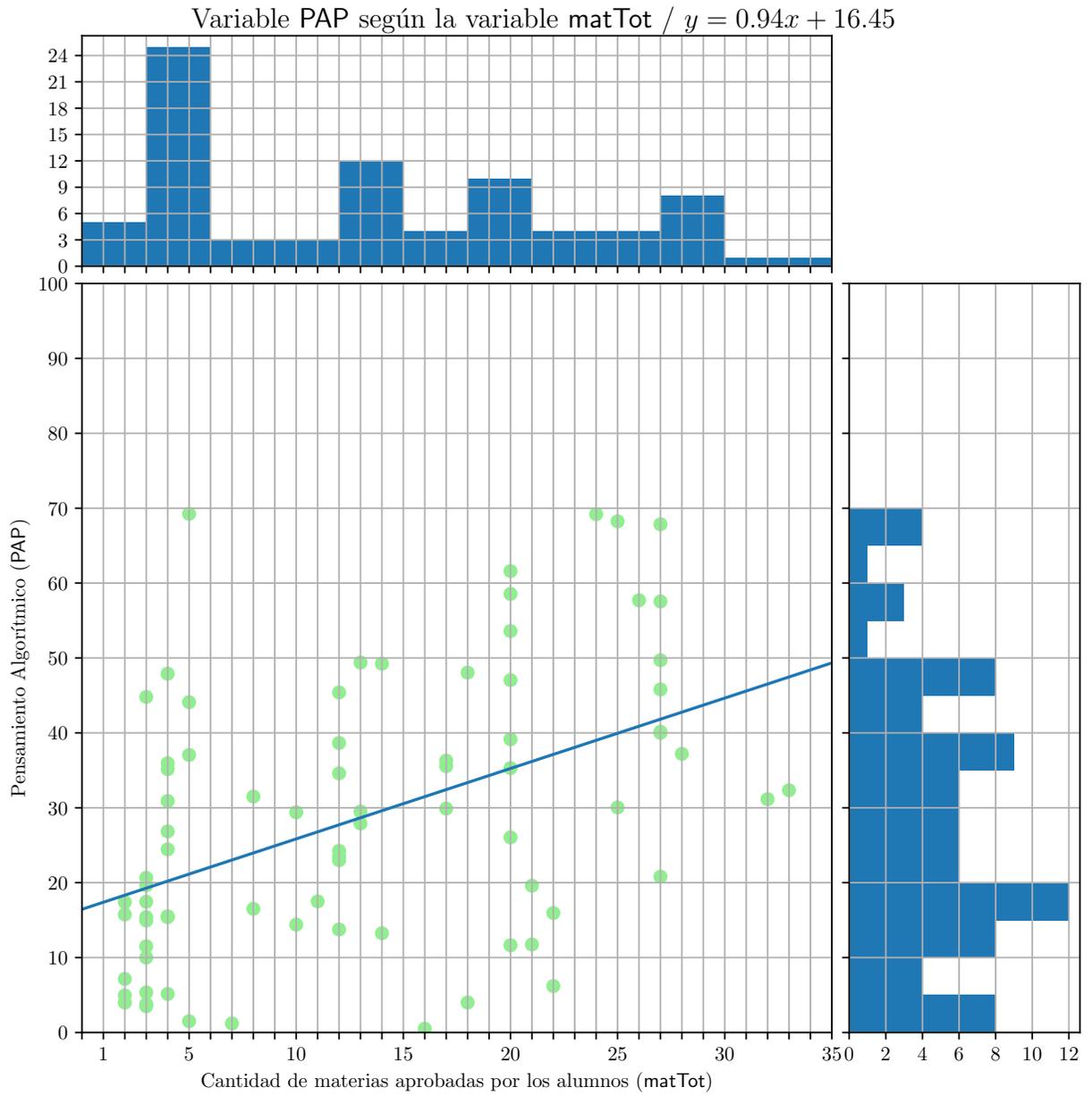


Figura 4.32.: Variable PAP según matTot.

Fuente: Elaboración propia.

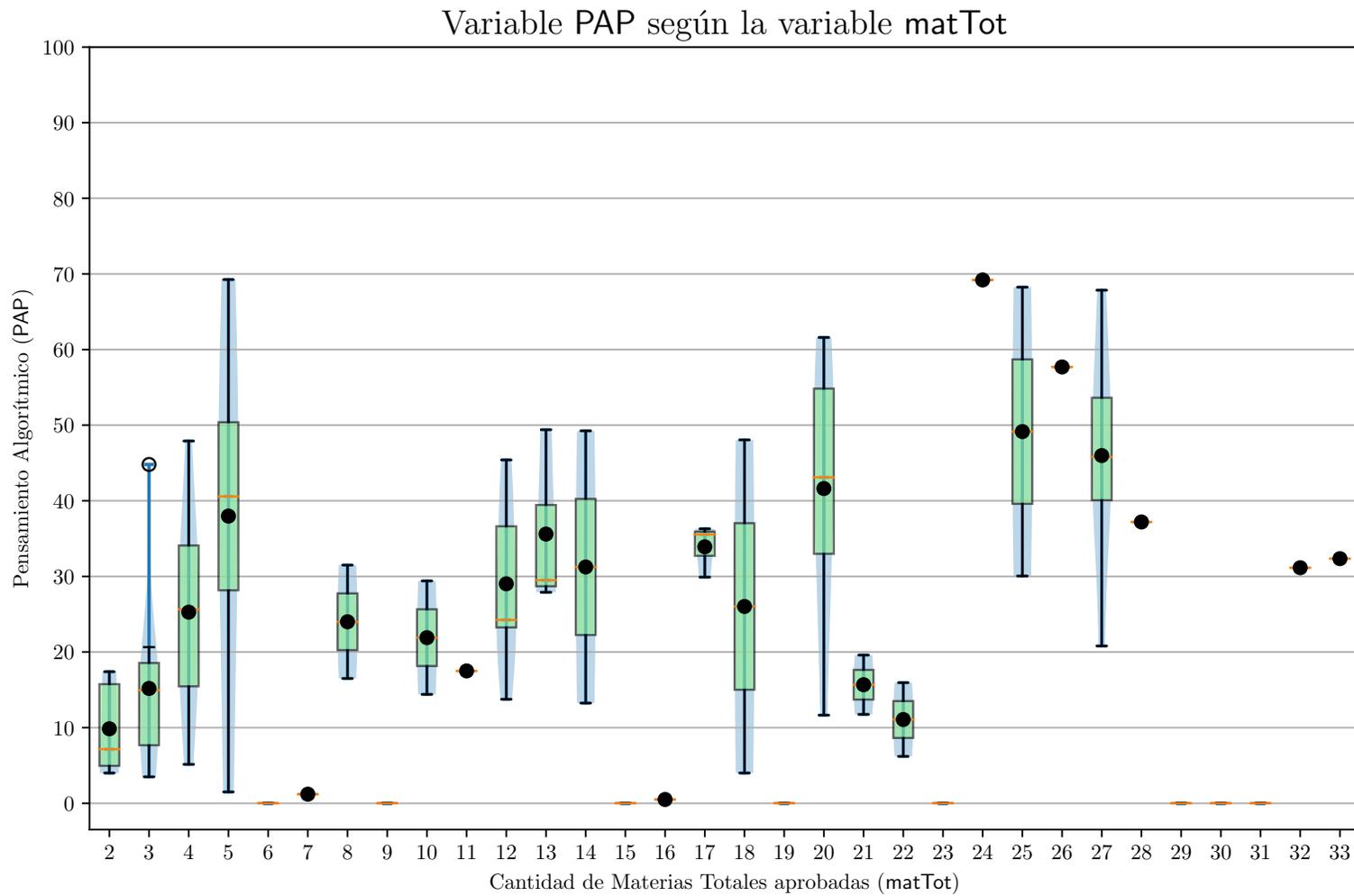


Figura 4.33.: Dispersión de la variable PAP según matTot.

Fuente: Elaboración propia.

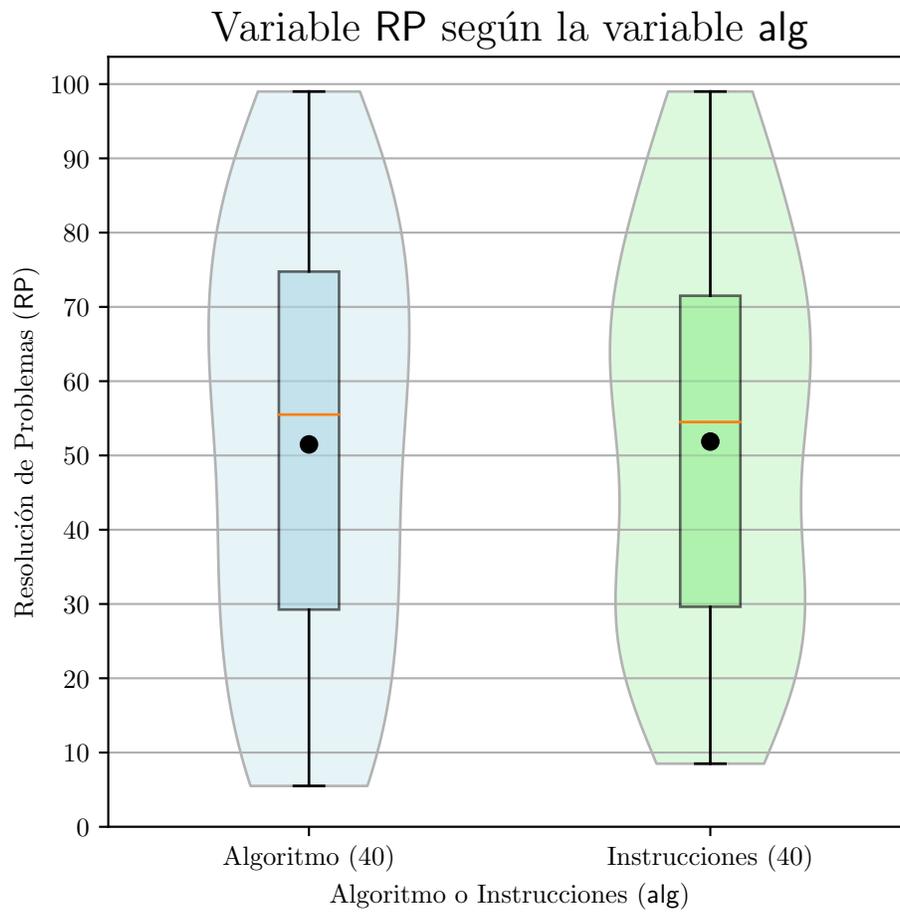


Figura 4.34.: Variable RP según alg.

Fuente: Elaboración propia.

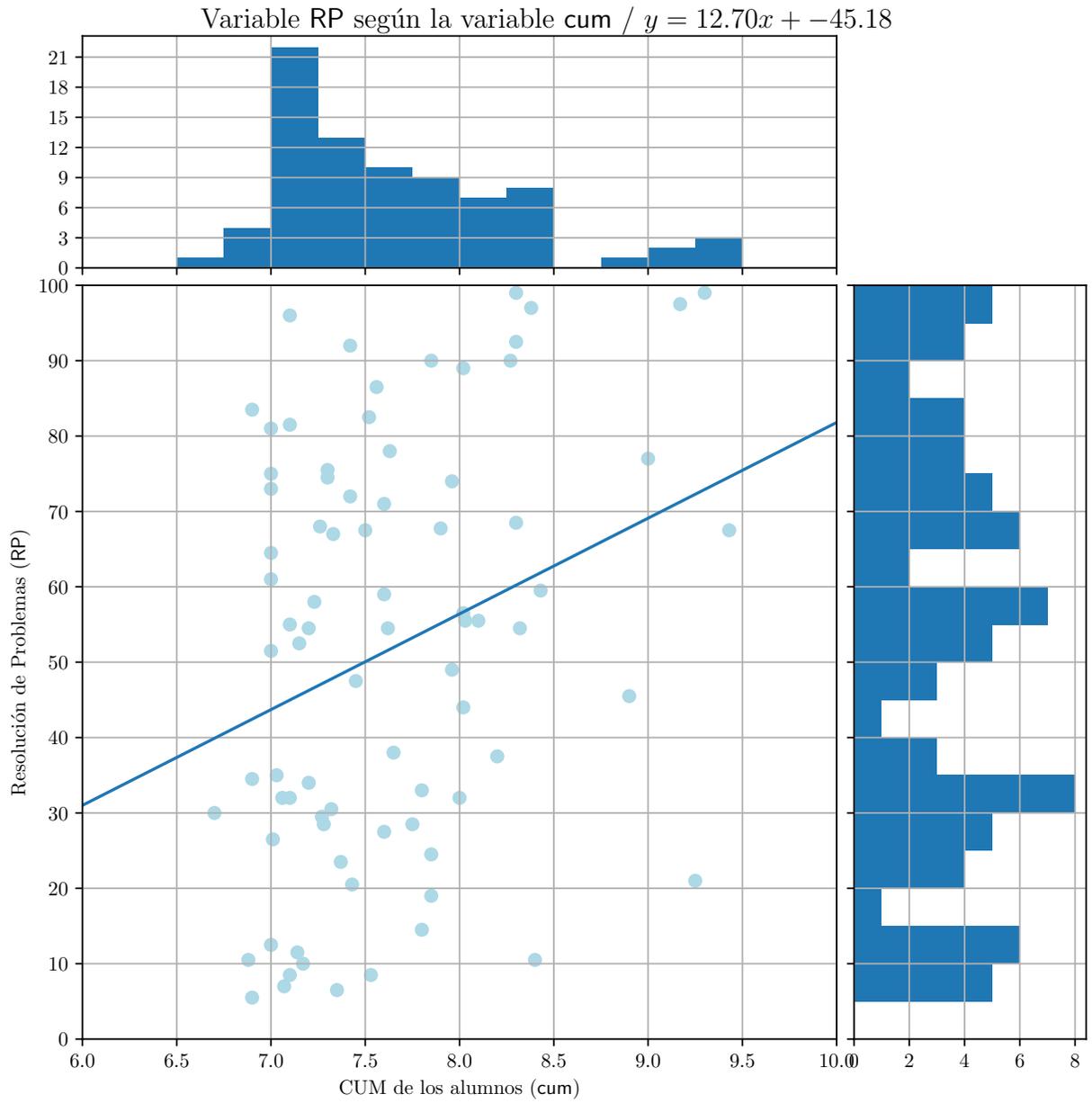


Figura 4.35.: Variable RP según cum.

Fuente: Elaboración propia.

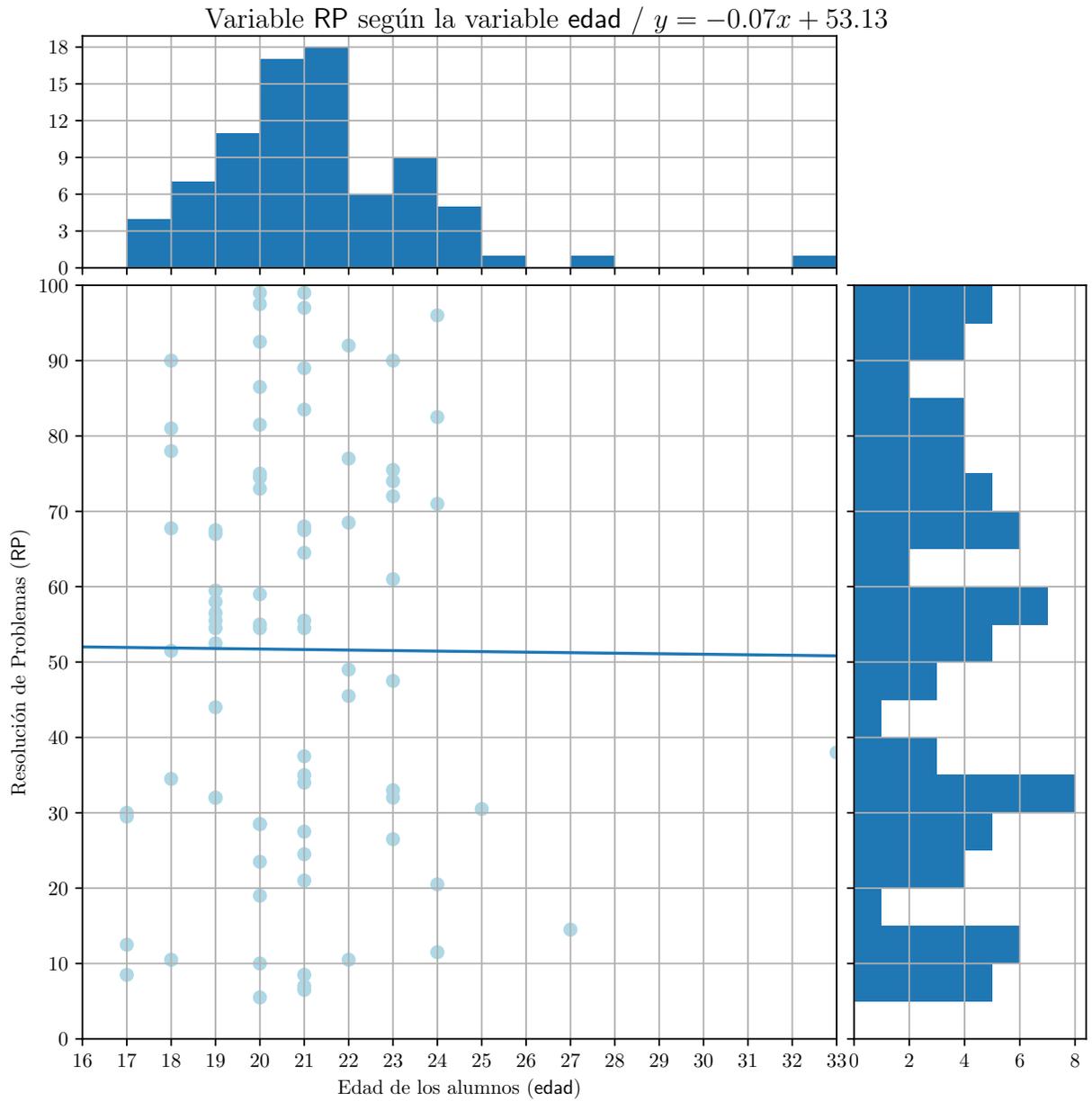


Figura 4.36.: Variable RP según edad.

Fuente: Elaboración propia.

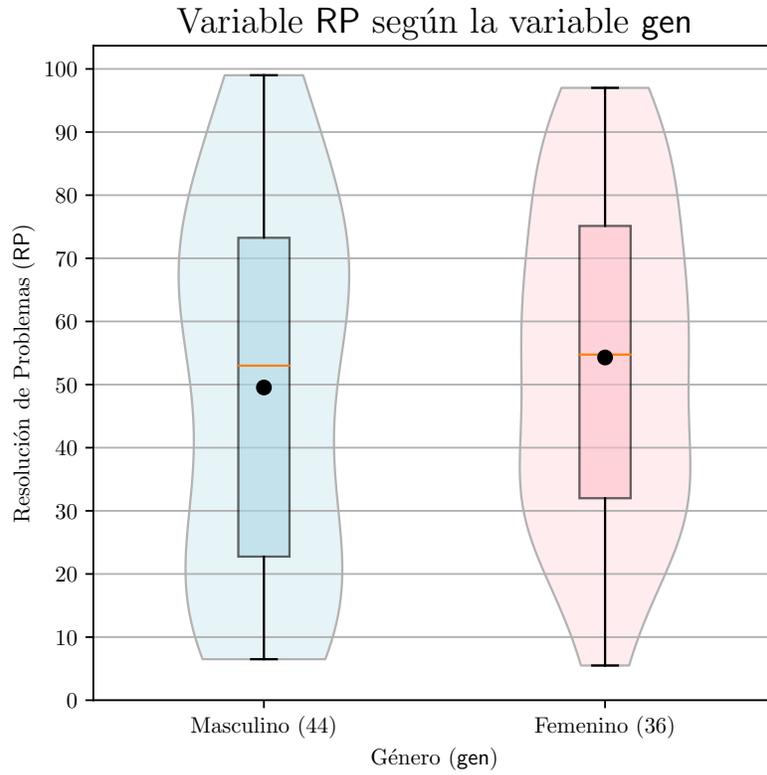


Figura 4.37.: Variable RP según gen.

Fuente: Elaboración propia.

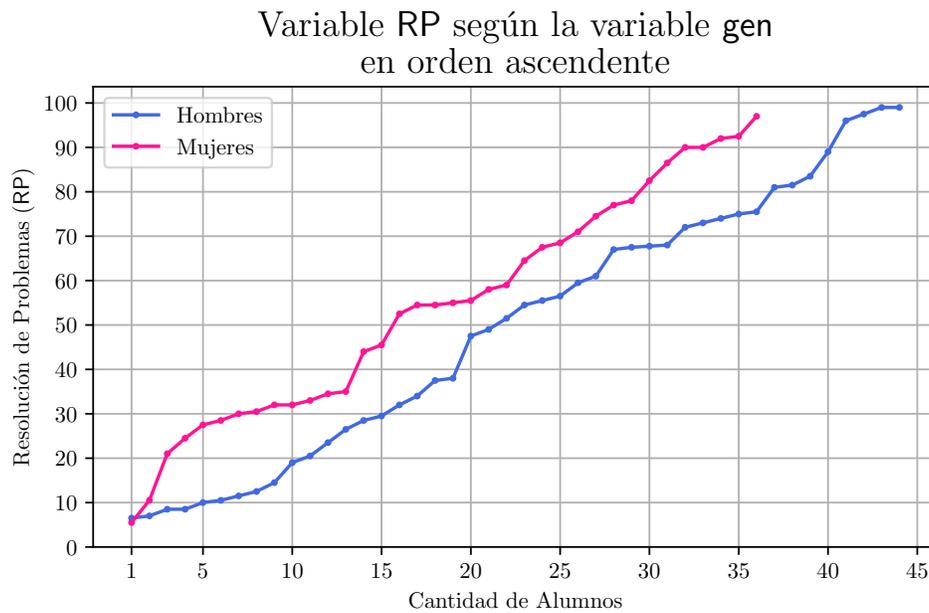


Figura 4.38.: Variable RP según gen en orden ascendente.

Fuente: Elaboración propia.

Variable RP según la variable gen
en orden ascendente
con corrección de escala

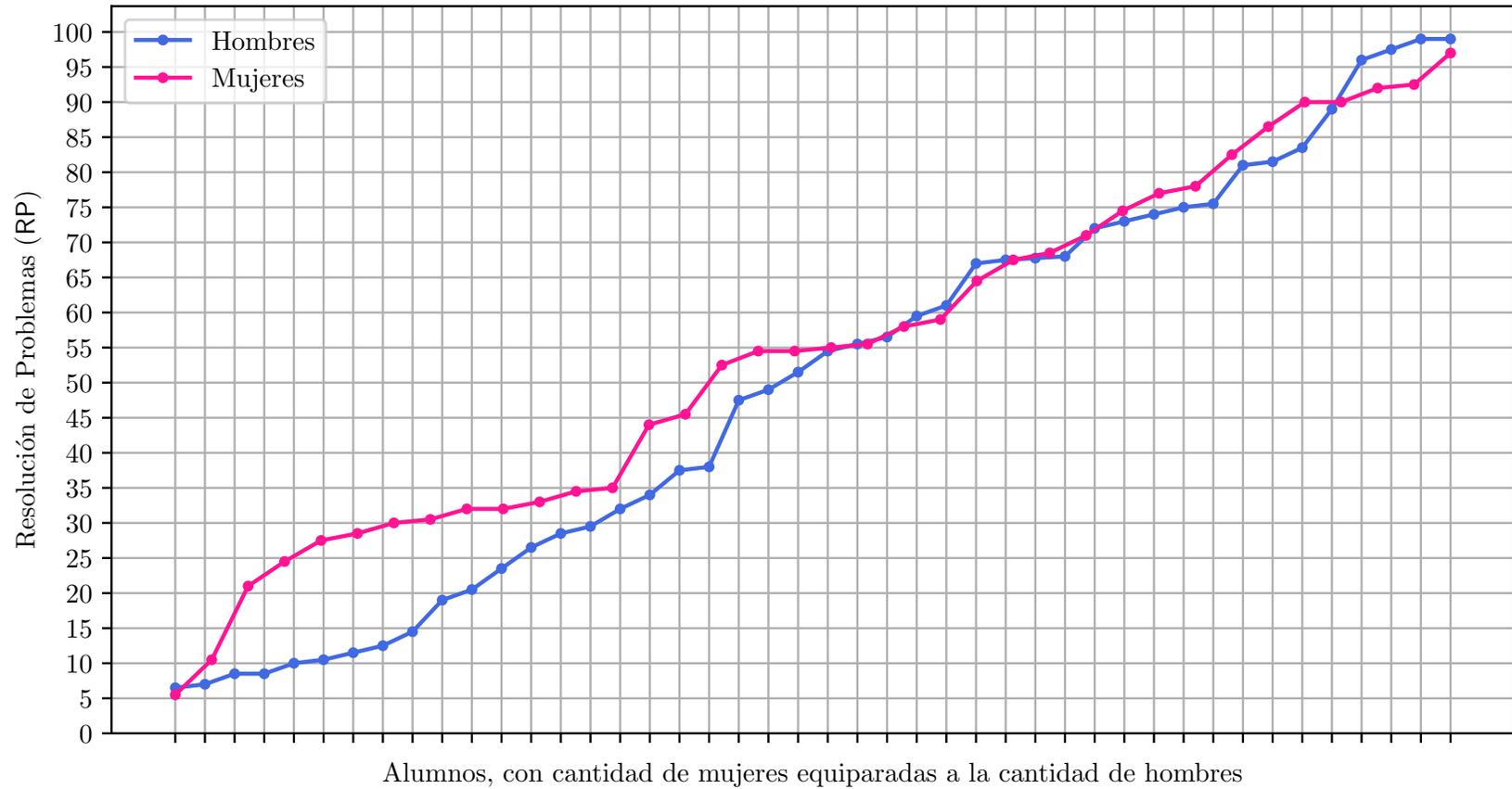


Figura 4.39.: Variable RP según gen en orden ascendente con corrección en la escala horizontal de mujeres.

Fuente: Elaboración propia.

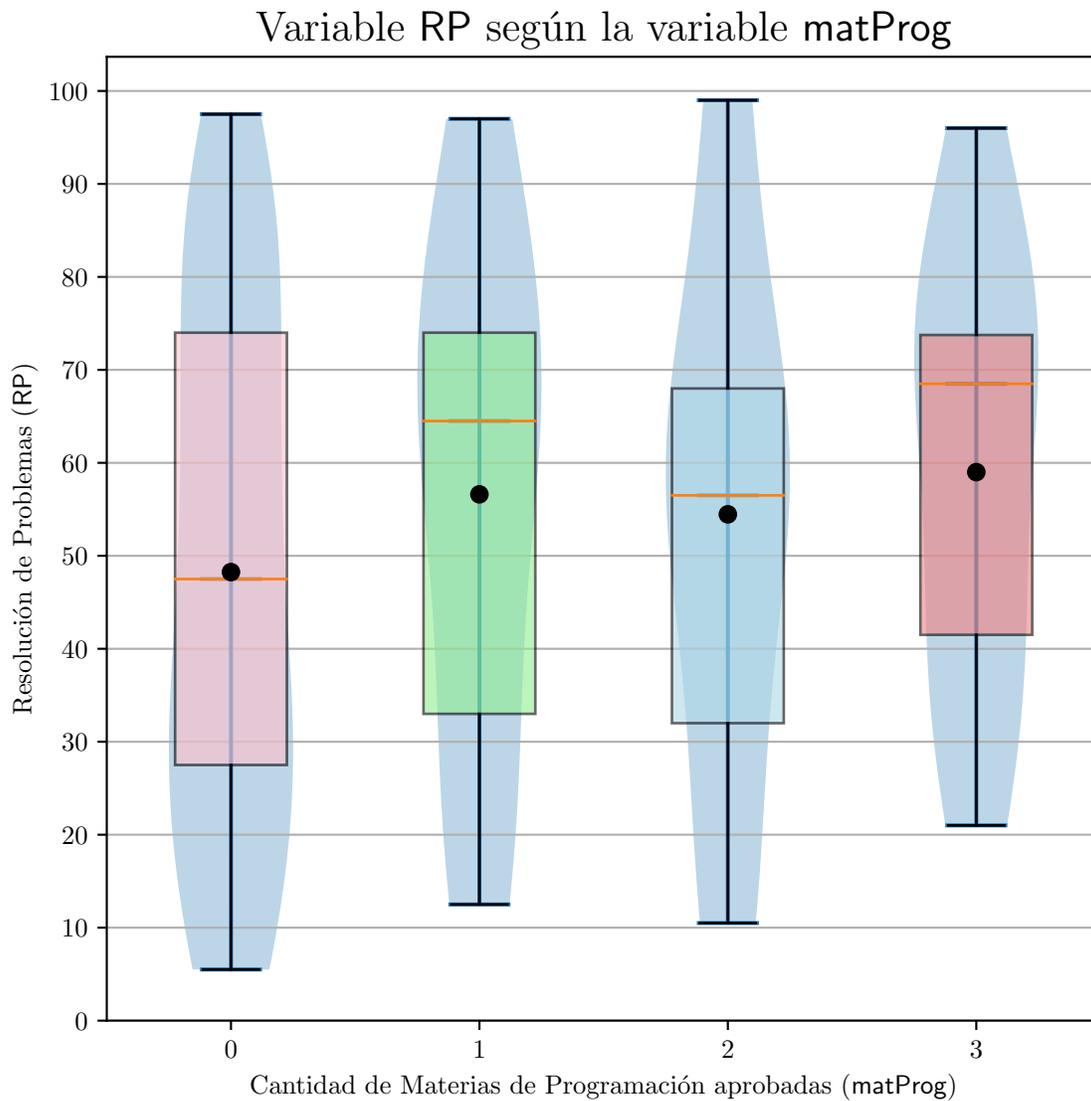


Figura 4.40.: Variable RP según matProg.
Fuente: Elaboración propia.

Detalle de RP según matProg

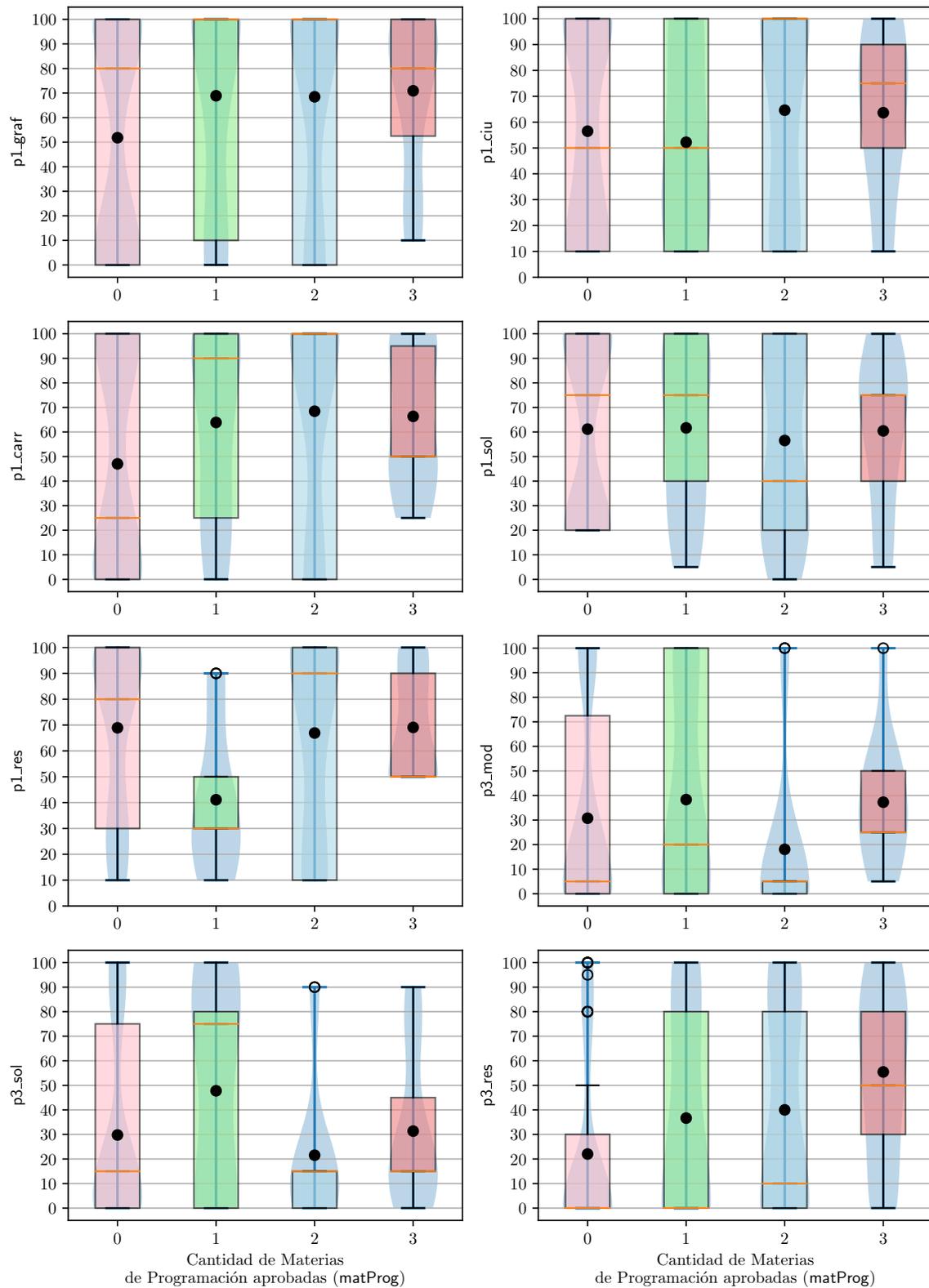


Figura 4.41.: Detalle de las sub-variables de RP según matProg.

Fuente: Elaboración propia.

Detalle de RP según matTot

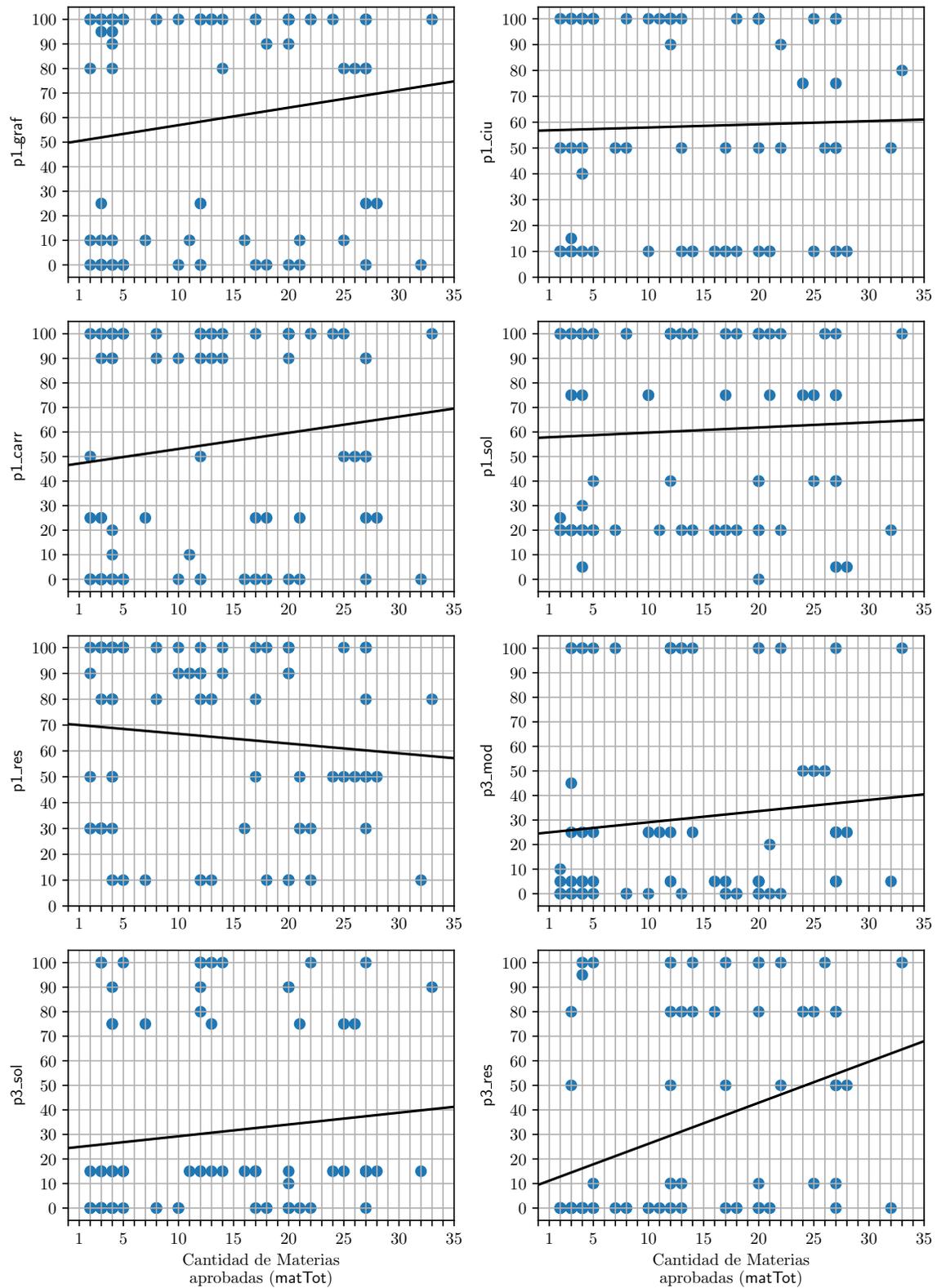


Figura 4.42.: Detalle de las sub-variables de RP según matTot.

Fuente: Elaboración propia.

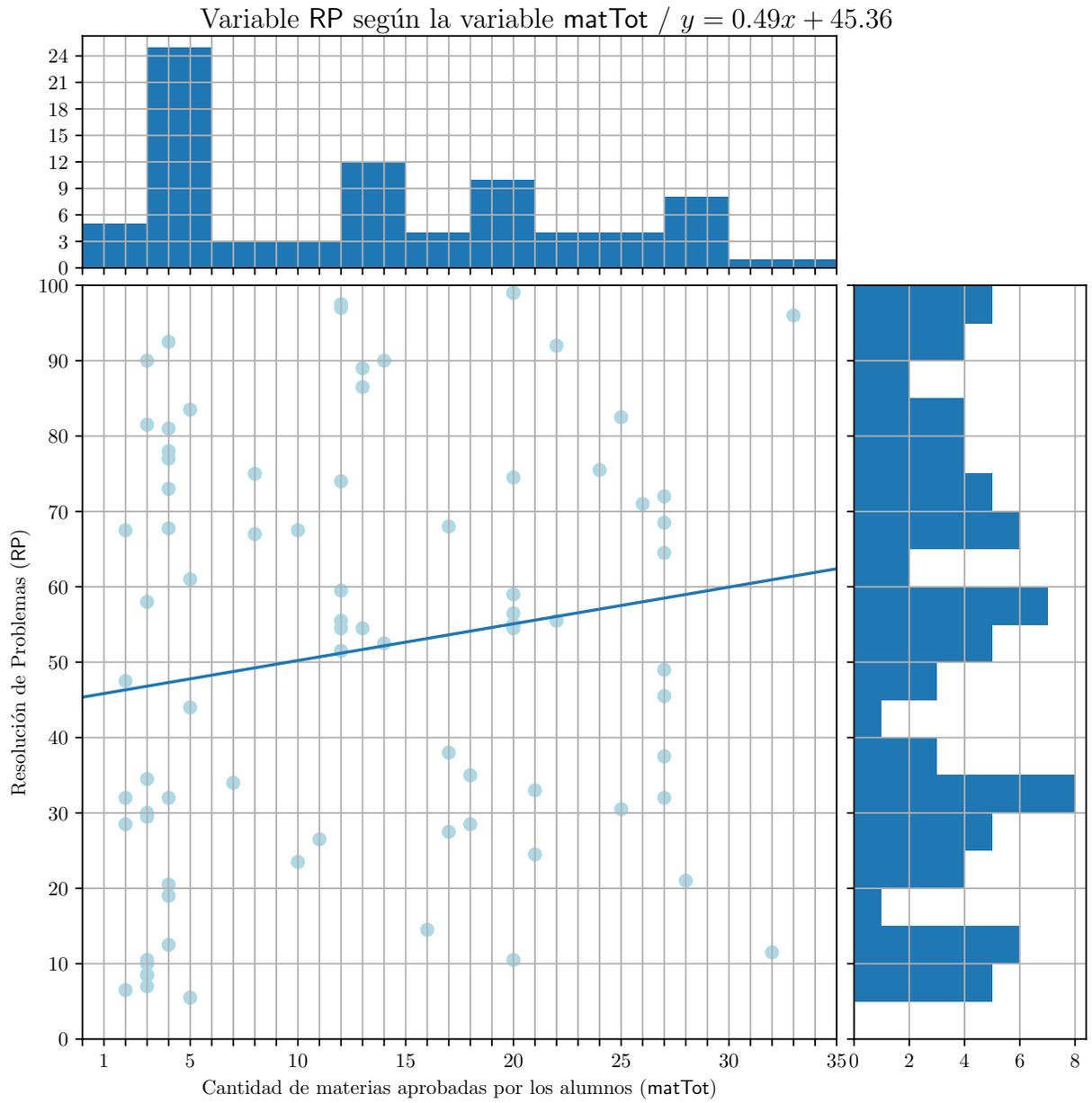


Figura 4.43.: Variable RP según matTot.
Fuente: Elaboración propia.

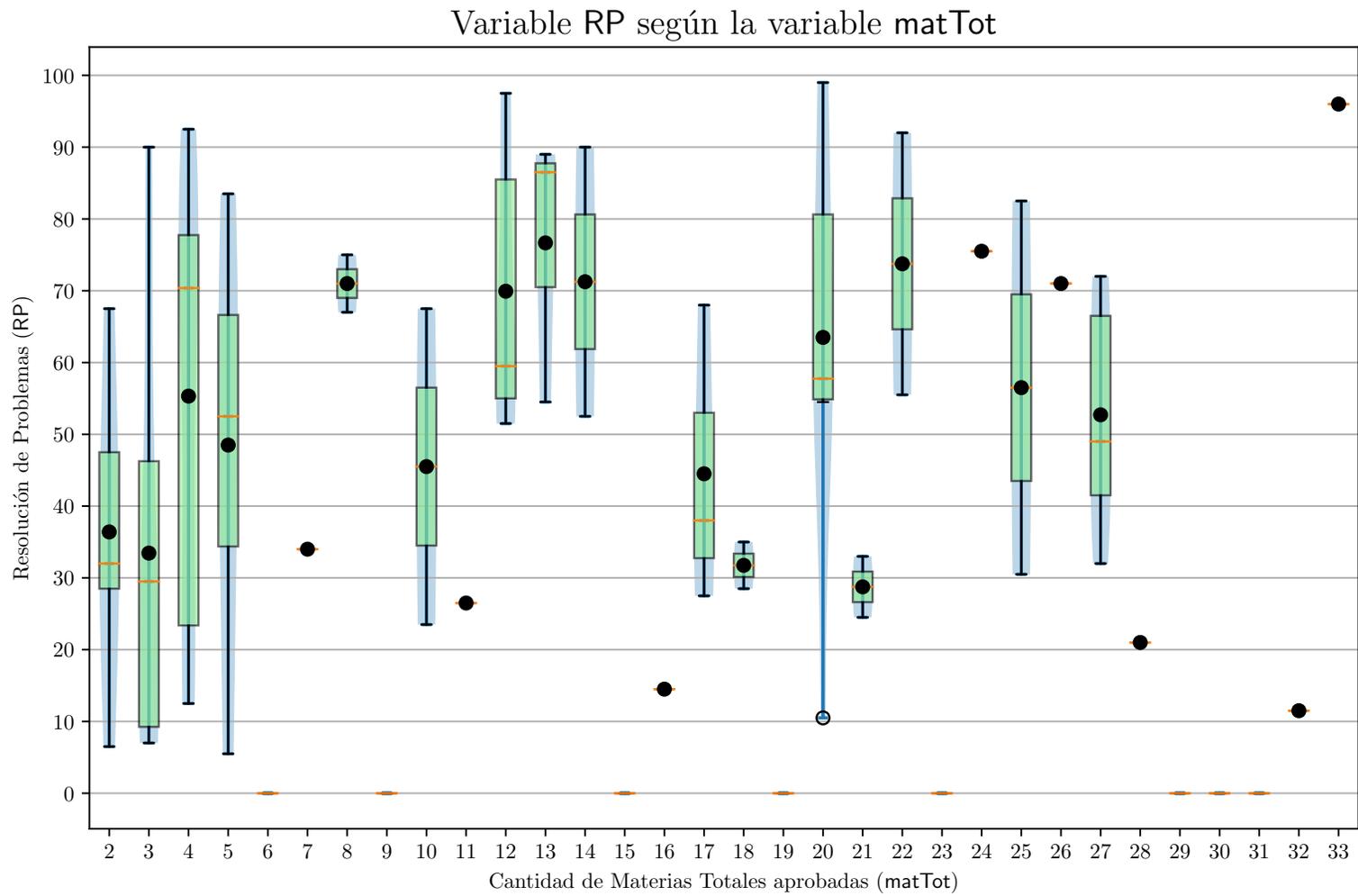


Figura 4.44.: Dispersión de la variable RP según mat Tot.

Fuente: Elaboración propia.

4.2. Resultados y Análisis de la Parte Cualitativa

Tal como se explicó en la sección 3.12 en la página 111, se desarrolló una serie de gráficos tipo Schoenfeld para representar el intento de resolución de un problema por parte de cada sujeto particular. Los sujetos analizados son los descritos en la sección 3.2.2, la metodología de recolección de datos está explicada en la sección 3.10 y la metodología de representación de los datos está explicada en las secciones 3.11 y 3.12.

4.2.1. Características generales de los sujetos

Las características generales de los sujetos de la parte cualitativa se resumen en el cuadro 4.3. Estas son las mismas características registradas de los sujetos de la parte cuantitativa (ver cuadro 3.5).

Cuadro 4.3.: Características de los sujetos de la parte cualitativa.

Sujeto	edad	cum	matTot	matProg	gen	Resultado
S01	22	7.91	20	2	Mujer	Fracaso
S02	27	7.88	19	2	Hombre	Fracaso
S03	24	8.04	17	2	Mujer	Fracaso
S04	26	7.4	25	2	Hombre	Fracaso
S05	22	7.46	34	2	Mujer	Éxito
S06	23	8.28	19	2	Hombre	Fracaso
S07	20	7.8	20	2	Hombre	Éxito
S08	22	7.11	16	2	Hombre	Éxito
S09	21	7.05	17	2	Mujer	Éxito
S10	21	7.31	18	2	Hombre	Fracaso
S11	25	7.88	21	2	Hombre	Éxito
S12	19	7.36	18	2	Hombre	Fracaso
S13	22	7.33	26	2	Hombre	Fracaso
S14	20	7.47	20	2	Hombre	Fracaso

Fuente: Elaboración propia.

4.2.2. Resultados Gráficos

Los siguientes gráficos tipo Schoenfeld, cuya lectura se explica en la sección 3.12, modelan los procesos cognitivos identificados en los 15 sujetos de esta parte de la investigación al resolver cada uno de los problemas de la lista en el Apéndice B en la página 221, y son analizados en la siguiente sub-sección (sub-sección 4.2.3 en la página 183):

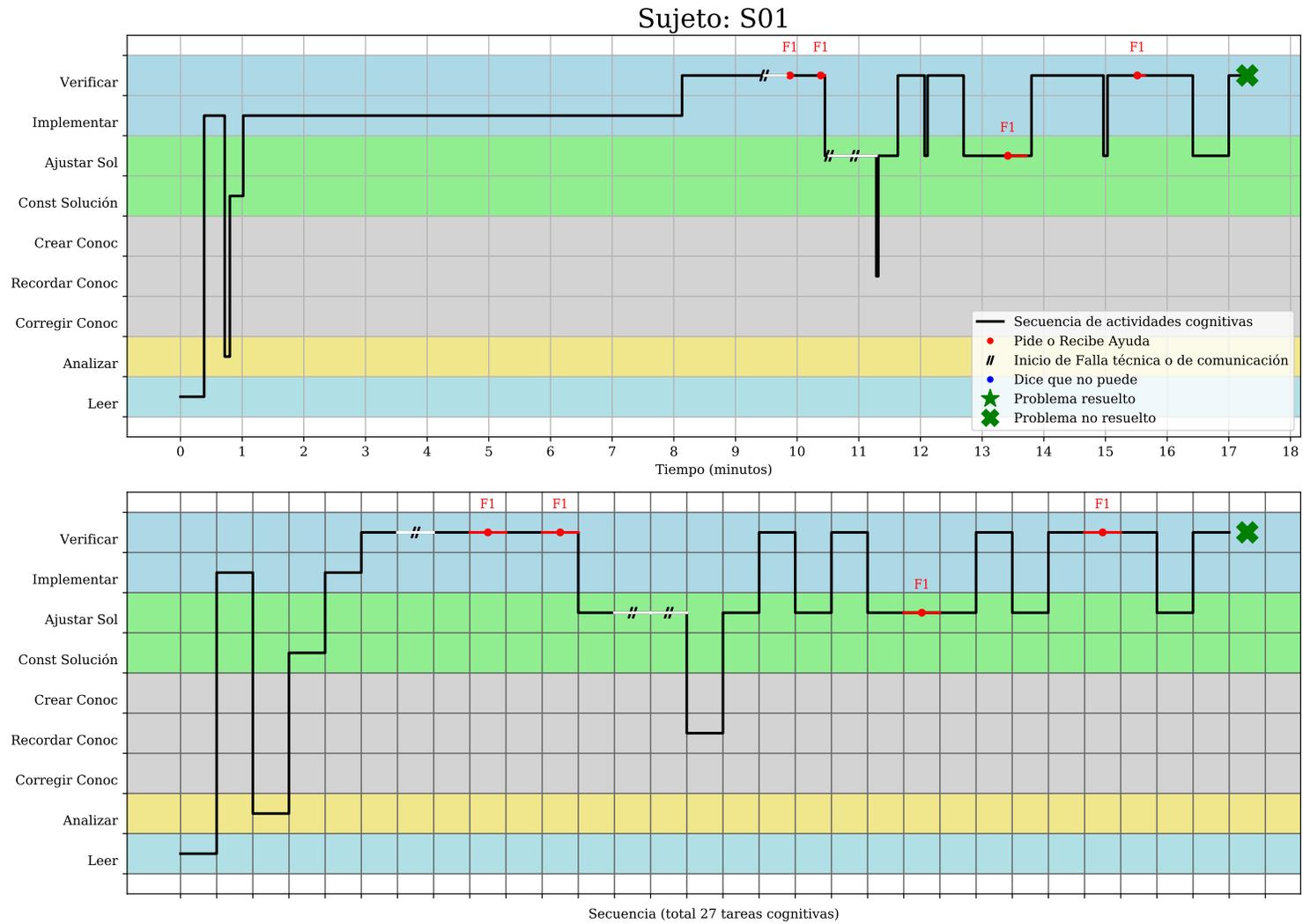


Figura 4.45.: Representación gráfica de la resolución del Sujeto S01.

Fuente: Elaboración propia.

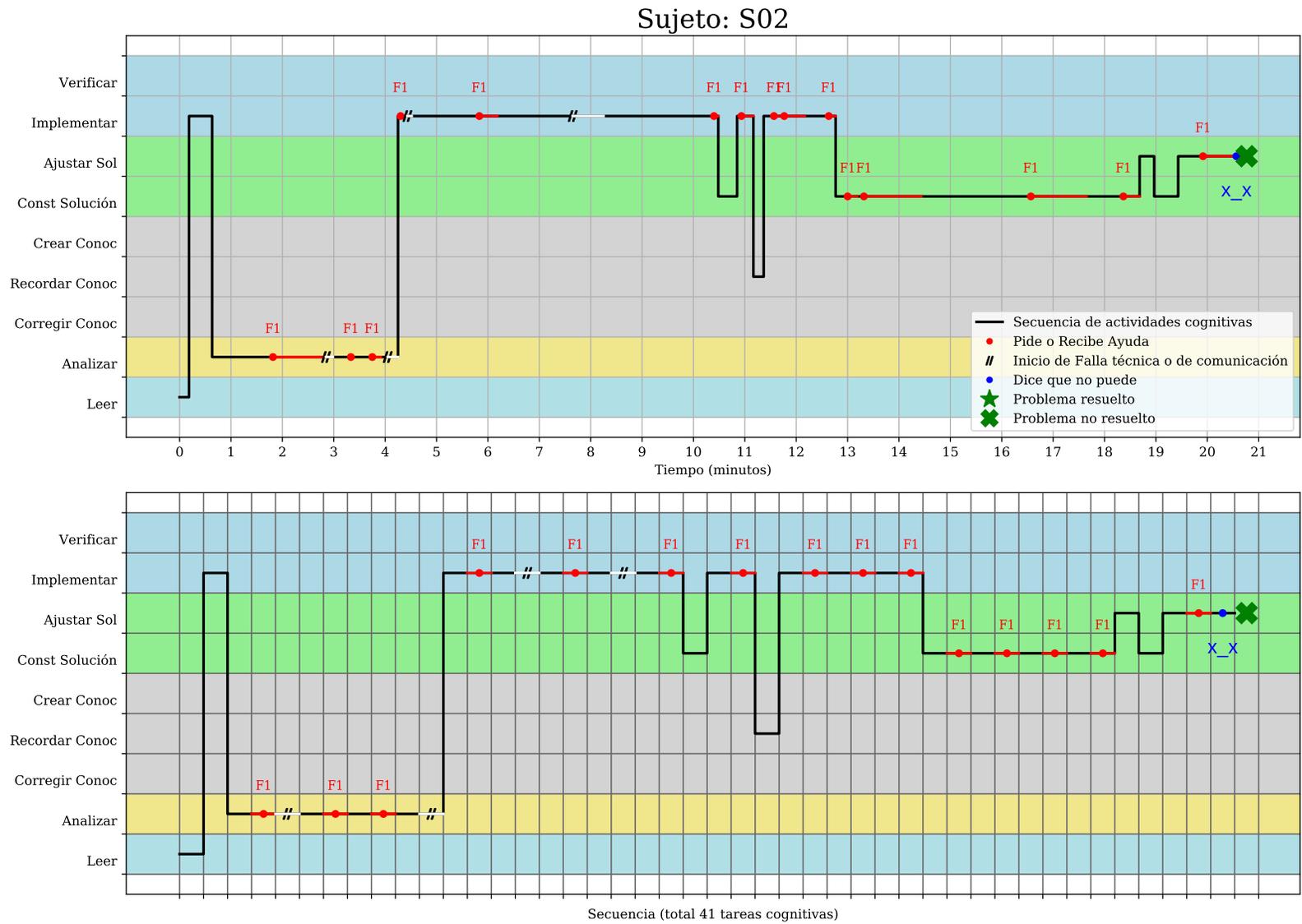


Figura 4.46.: Representación gráfica de la resolución del Sujeto S02.

Fuente: Elaboración propia.

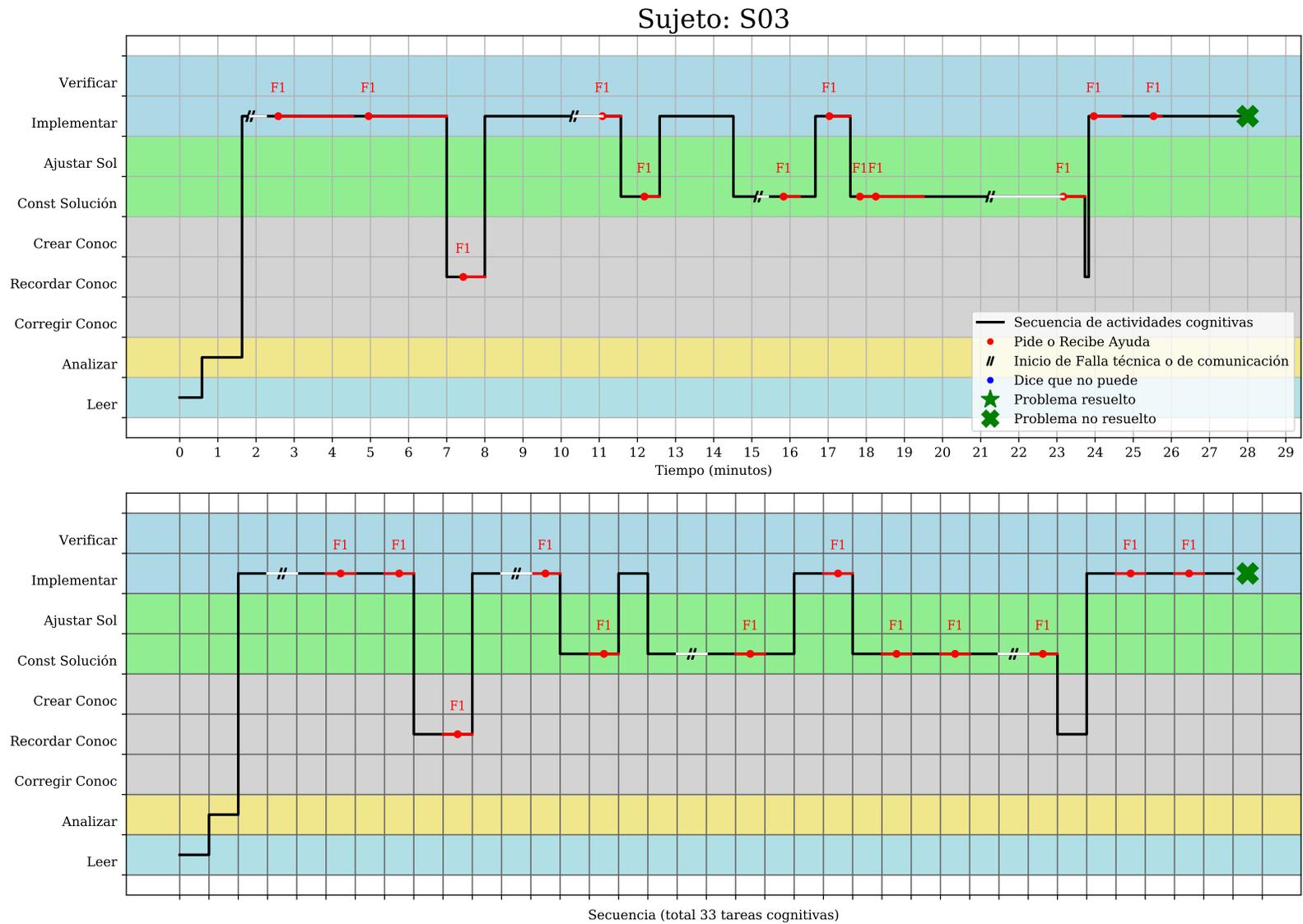


Figura 4.47.: Representación gráfica de la resolución del Sujeto S03.

Fuente: Elaboración propia.

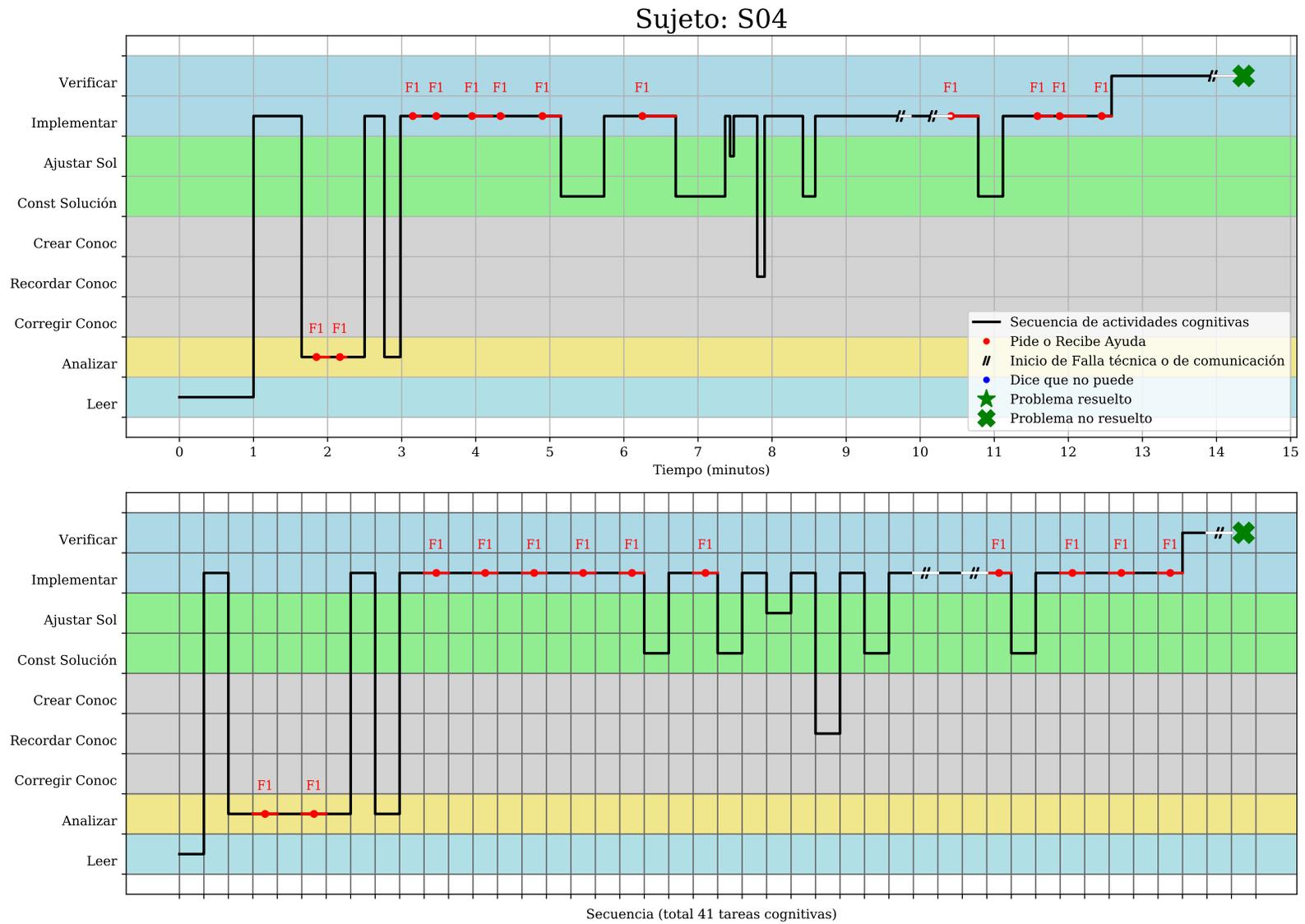


Figura 4.48.: Representación gráfica de la resolución del Sujeto S04.

Fuente: Elaboración propia.

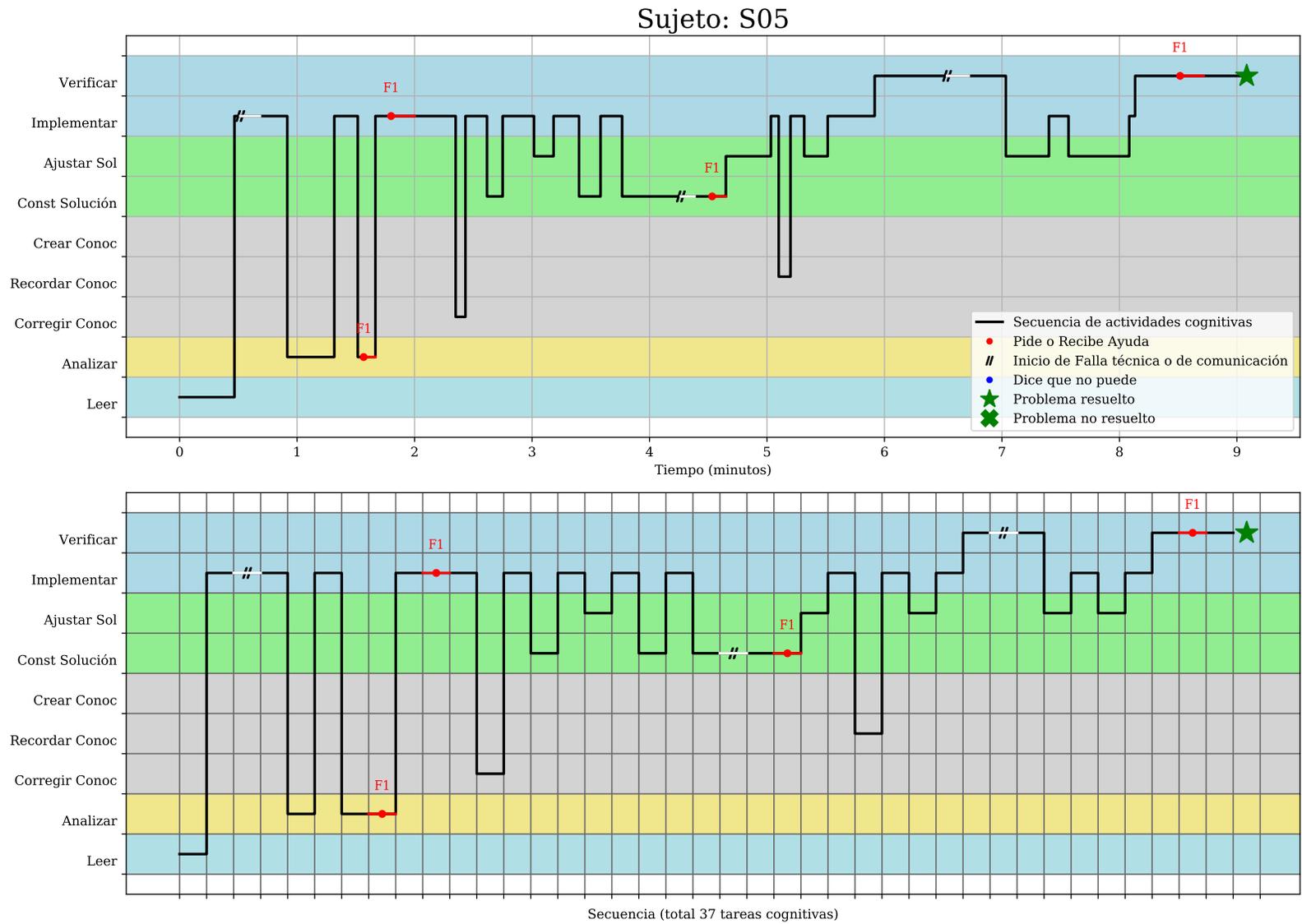


Figura 4.49.: Representación gráfica de la resolución del Sujeto S05.

Fuente: Elaboración propia.

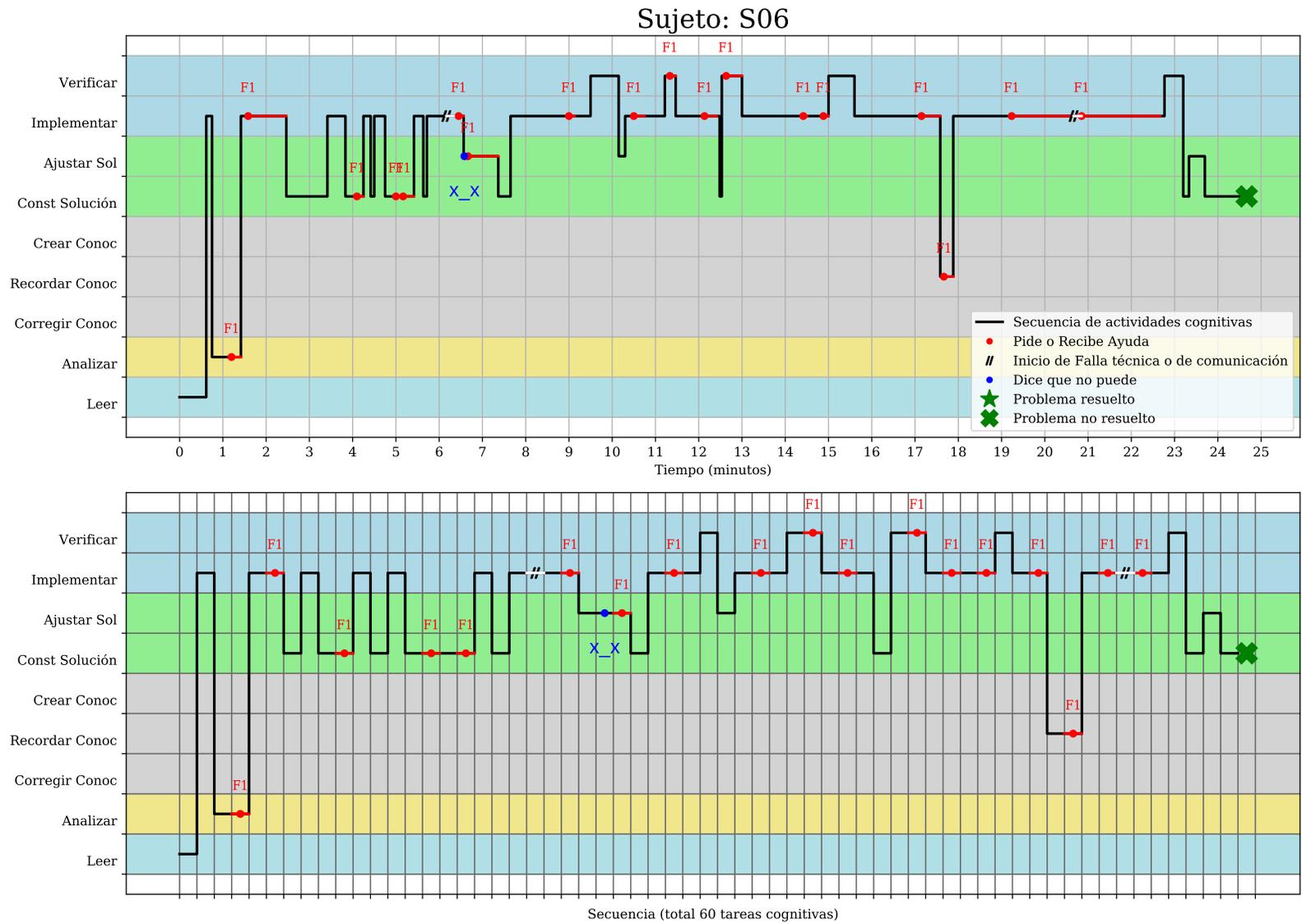


Figura 4.50.: Representación gráfica de la resolución del Sujeto S06.

Fuente: Elaboración propia.

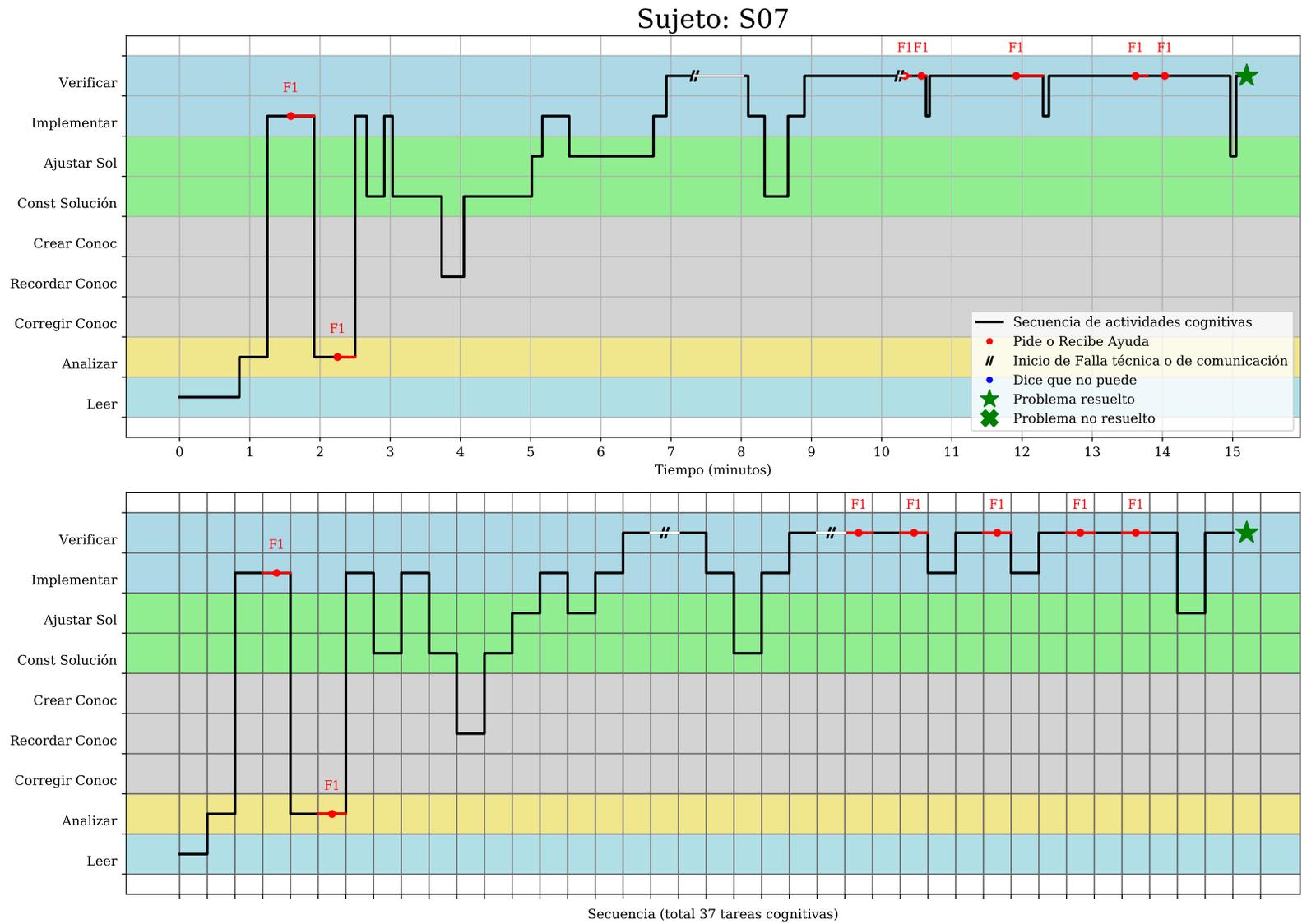


Figura 4.51.: Representación gráfica de la resolución del Sujeto S07.

Fuente: Elaboración propia.

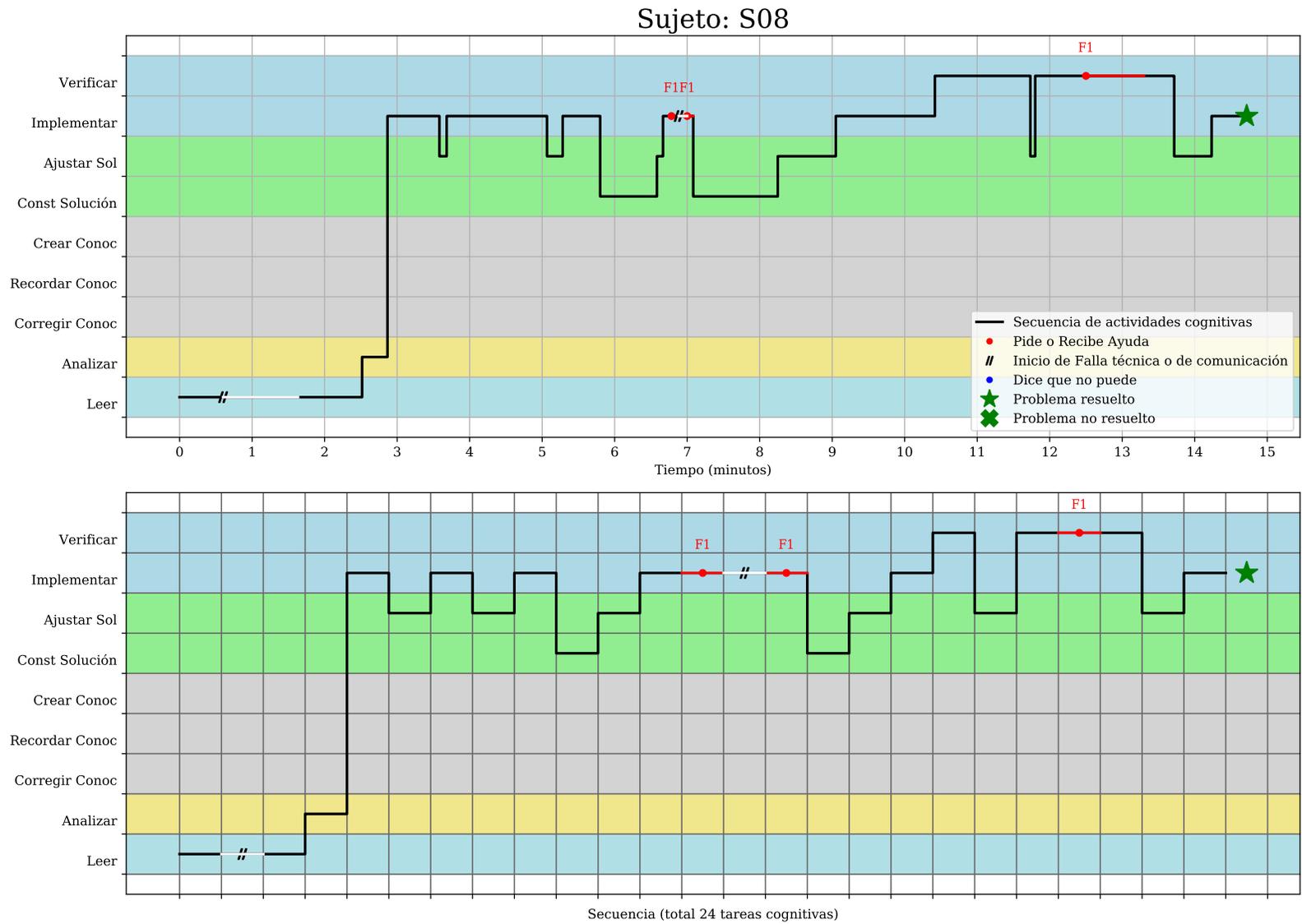


Figura 4.52.: Representación gráfica de la resolución del Sujeto S08.

Fuente: Elaboración propia.

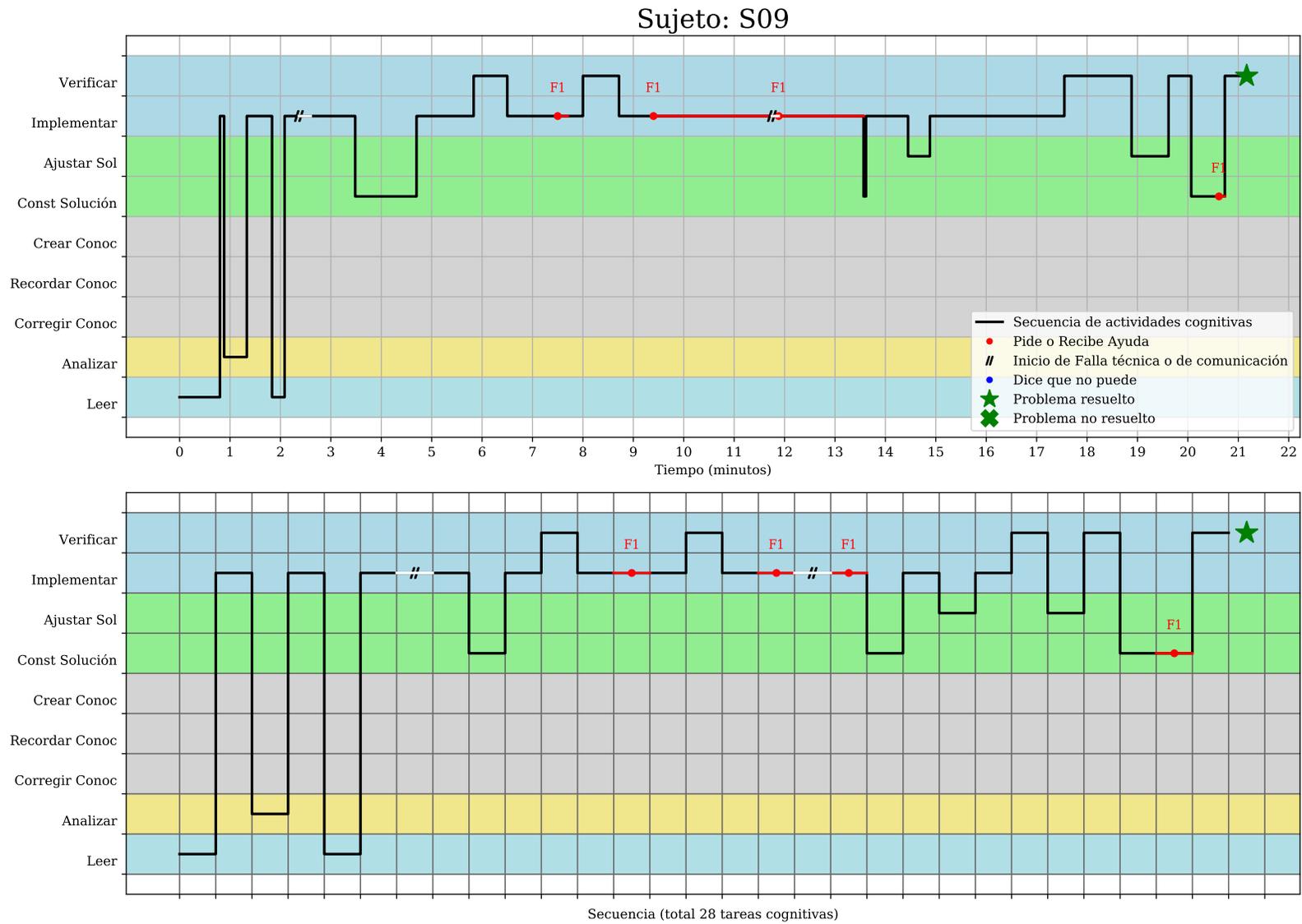


Figura 4.53.: Representación gráfica de la resolución del Sujeto S09.

Fuente: Elaboración propia.

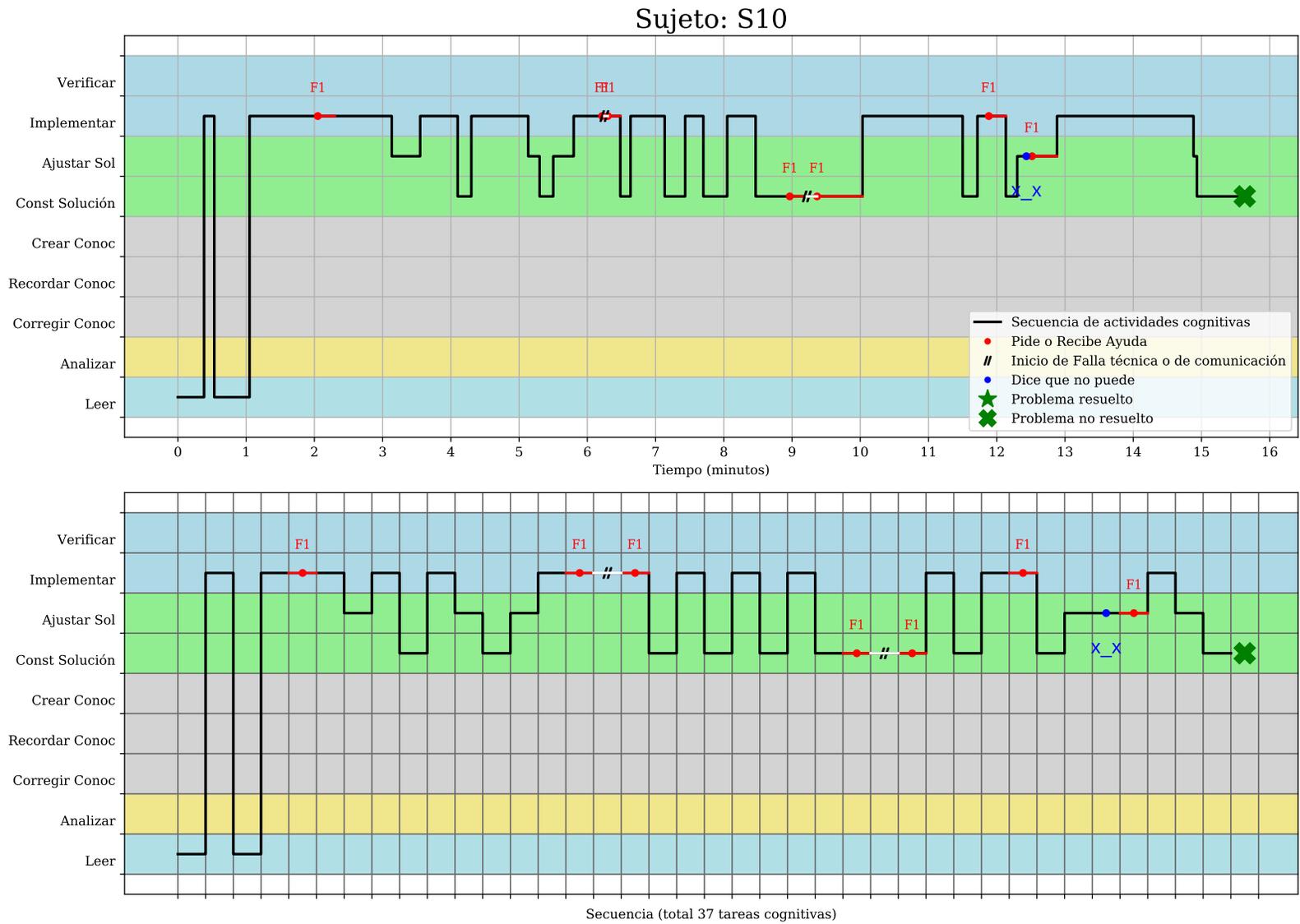


Figura 4.54.: Representación gráfica de la resolución del Sujeto S10.

Fuente: Elaboración propia.

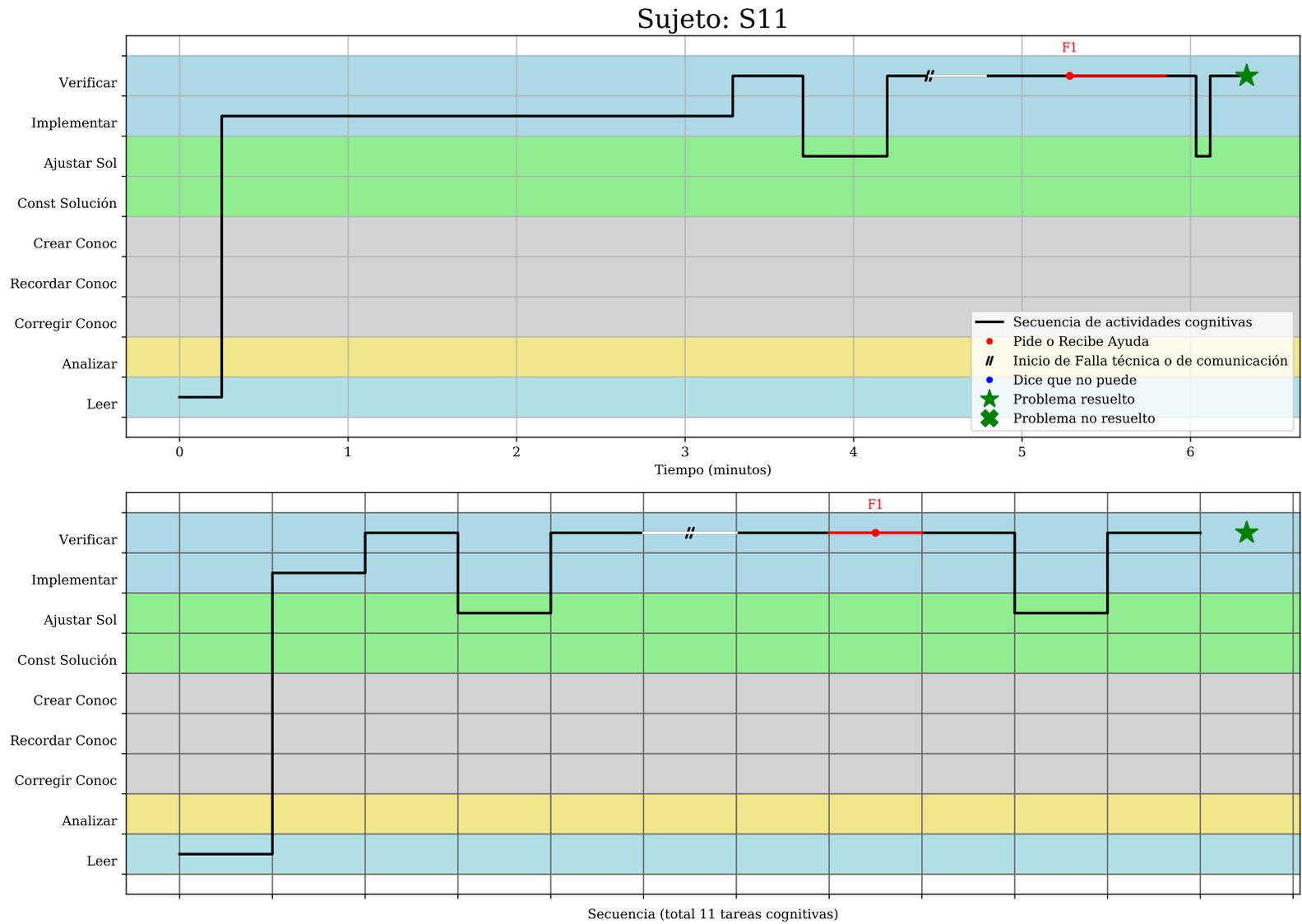


Figura 4.55.: Representación gráfica de la resolución del Sujeto S11.

Fuente: Elaboración propia.

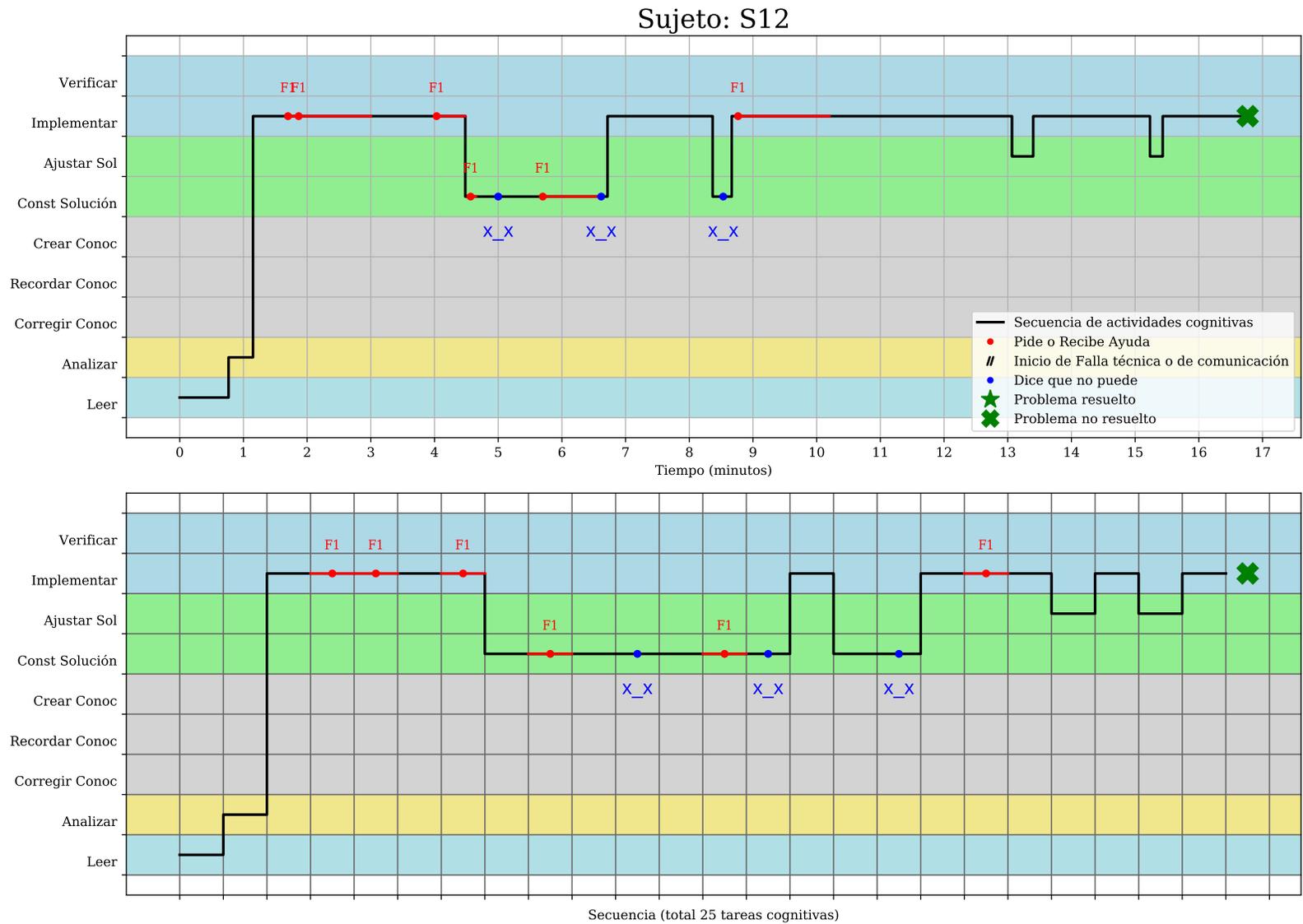


Figura 4.56.: Representación gráfica de la resolución del Sujeto S12.

Fuente: Elaboración propia.

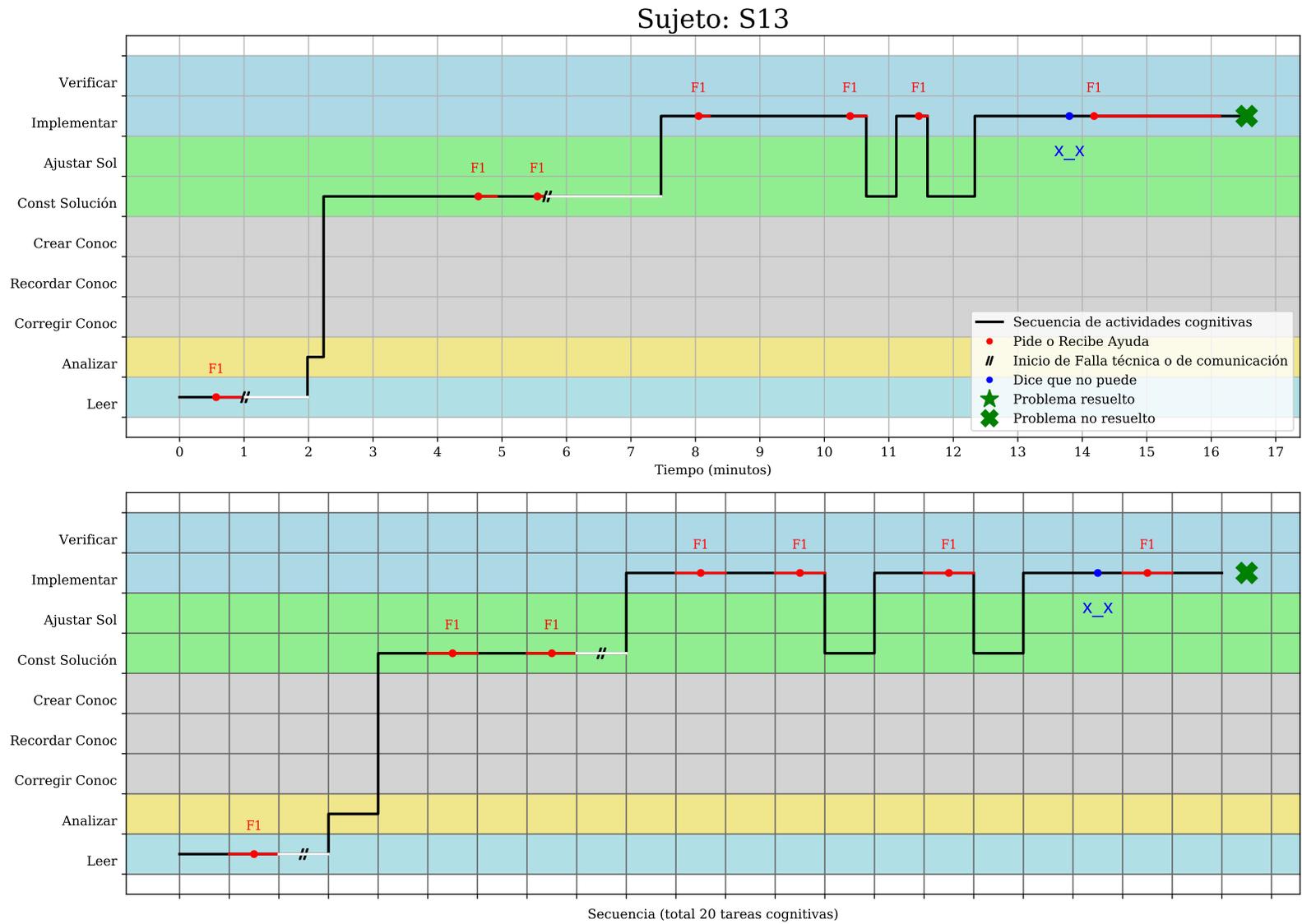


Figura 4.57.: Representación gráfica de la resolución del Sujeto S13.

Fuente: Elaboración propia.

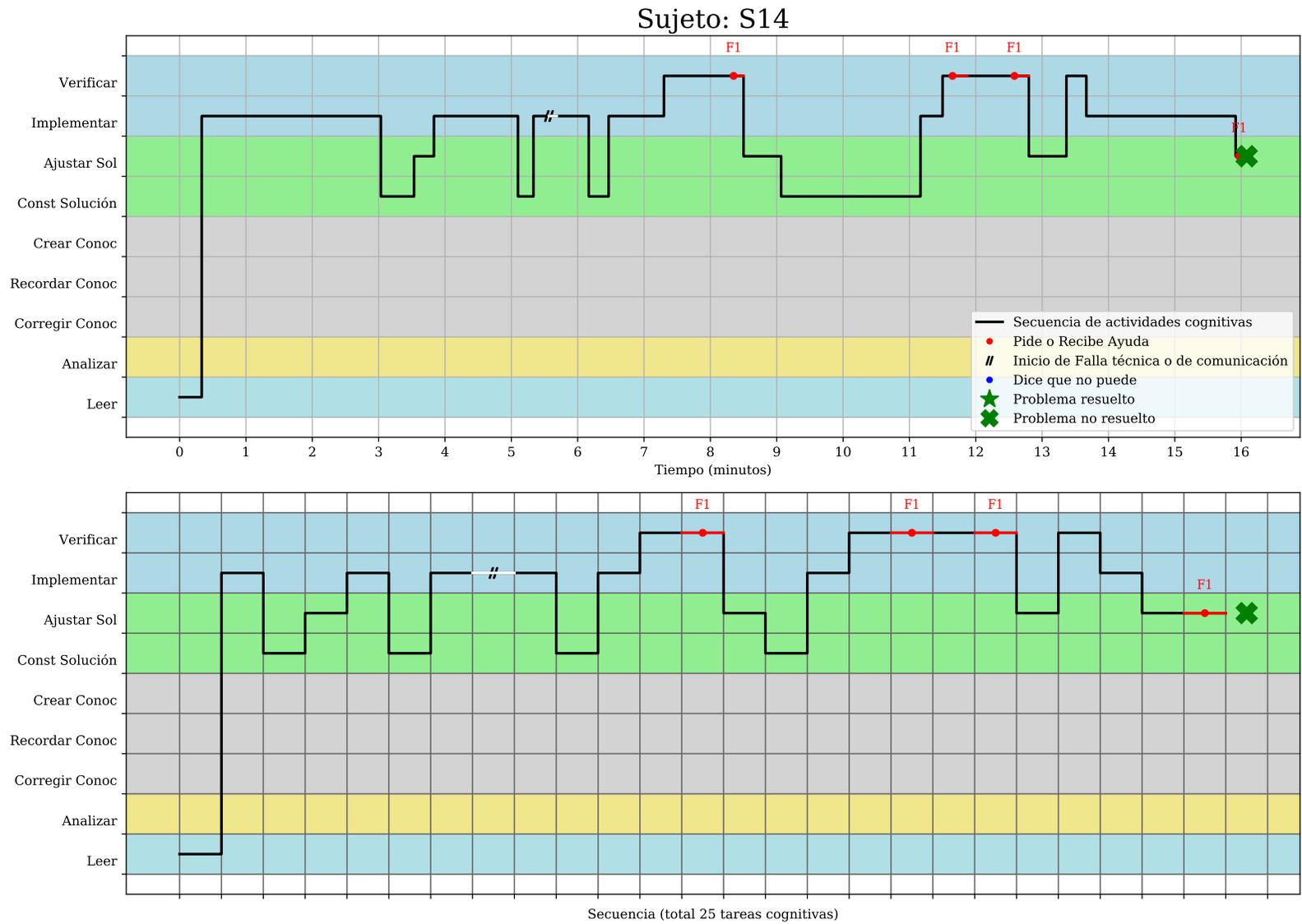


Figura 4.58.: Representación gráfica de la resolución del Sujeto S14.

Fuente: Elaboración propia.

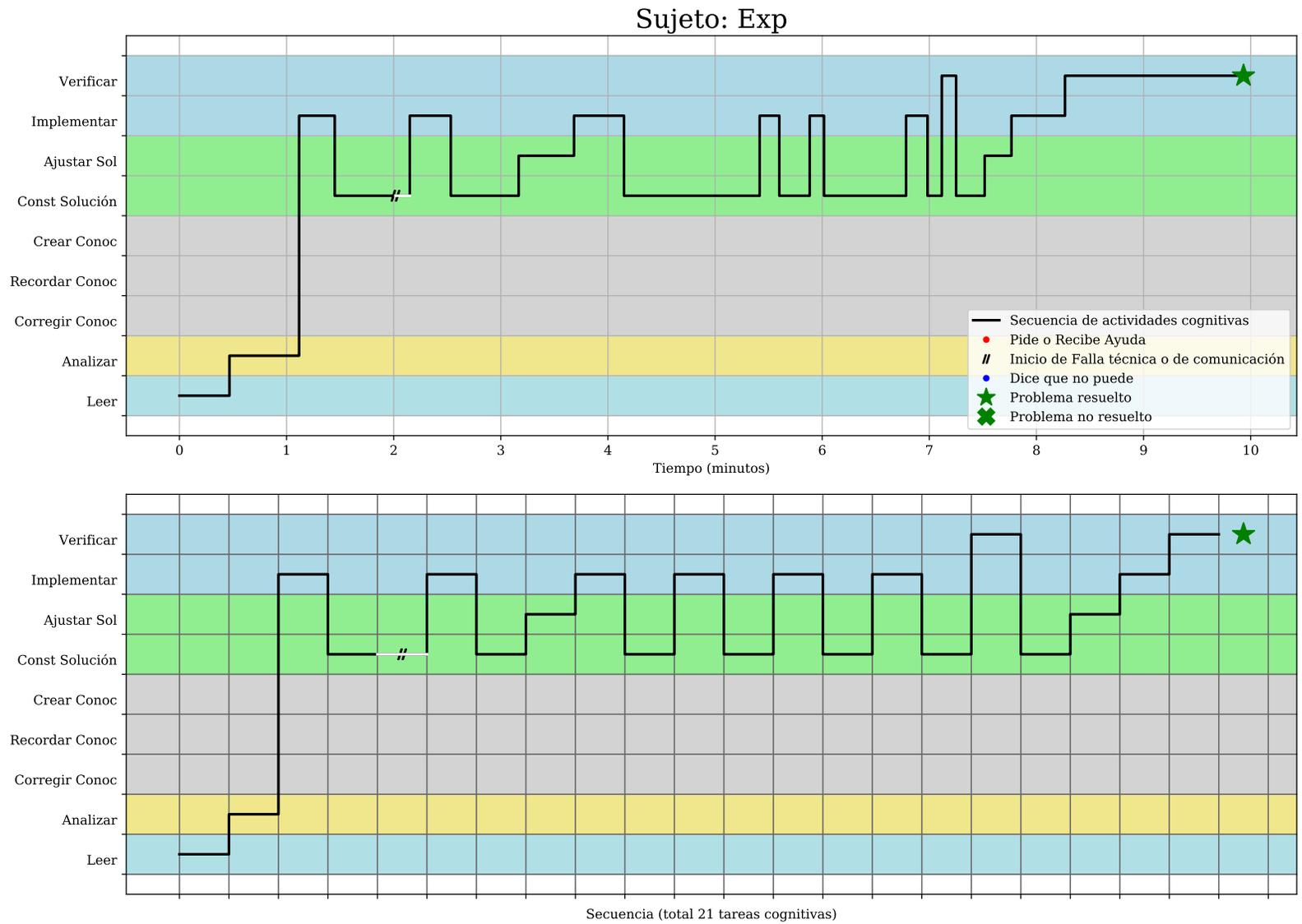


Figura 4.59.: Representación gráfica de la resolución del Sujeto Exp (experto).

Fuente: Elaboración propia.

4.2.3. Análisis de los gráficos de la parte cualitativa

Los gráficos tipo Schoenfeld que representan las tareas cognitivas de los sujetos analizados durante la resolución de un problema, de los que se habla en esta sub-sección aparecen desde la página 168 hasta la página 182 en la sub-sección anterior.

A continuación se presenta un análisis cualitativo de esos resultados gráficos:

Poco éxito

De los 14 sujetos alumnos, tuvieron éxito al resolver el problema sólo los sujetos S05, S07, S08, S09 y S11, 5 de los 14. Mientras que 9 de los 14 alumnos no tuvieron éxito en resolver el problema que se les asignó.

Comenzar a Implementar sin Analizar

El primer detalle que salta a la vista es que la mayoría de los sujetos, todos excepto S03, S07, S08, S12, S13 y Exp, saltan a la implementación inmediatamente después de terminar la lectura sin realizar un proceso de análisis del problema antes de comenzar a escribir. Esto se corresponde con lo planteado por Schoenfeld (2016) y Blanco Hamad (2016) acerca de que los estudiantes novatos no dedican suficiente tiempo al análisis y construcción de la solución. Sin embargo, la mayoría de los sujetos vuelve a la fase de análisis después de unos segundos de implementación (equivalente aproximadamente a una línea o media línea de código), cuando se dan cuenta que no saben realmente qué deben escribir.

Mucha ayuda

Todos los sujetos alumnos requirieron al menos una intervención de ayuda (14 de los 15). Tuvieron éxito al resolver el problema los sujetos S05, S07, S08, S09, S11 y Exp, 6 de los 15. El resto de los sujetos no logró resolver su problema. Y esto se corresponde negativamente con la cantidad de intervenciones de ayuda proporcionada, siendo el sujeto S06 un caso extremo. Ver cuadro 4.4 en la página siguiente.

Cuadro 4.4.: Relación entre cantidad de intervenciones de ayuda y la resolución exitosa.

Sujeto	Resultado	Cantidad de Ayudas
S01	Fracaso	4
S02	Fracaso	15
S03	Fracaso	12
S04	Fracaso	12
S05	Éxito	4
S06	Fracaso	18
S07	Éxito	7
S08	Éxito	3
S09	Éxito	4
S10	Fracaso	7
S11	Éxito	1
S12	Fracaso	6
S13	Fracaso	7
S14	Fracaso	4
Exp	Éxito	0

Fuente: Elaboración propia.

En varios casos, la intervención de ayuda no fue de utilidad para los sujetos alumnos porque presentaban lagunas de conocimiento básico, ya sea del lenguaje de programación usado y su sintaxis, o bien de la forma de implementar un cálculo iterativo, etc.

Se aplicó la prueba U de Mann-Whitney entre estas variables, para verificar si hay diferencia entre la cantidad de intervenciones de ayuda entre los sujetos que tuvieron éxito y los que no, por medio de las siguientes instrucciones:

```

1 > datosAyuda = read.csv("ayuda.csv", header=TRUE)
2 > datosAyuda$CantAyuda
3 [1] 4 15 12 12 4 18 7 3 4 7 1 6 7 4 0
4 > datosAyuda$Resultado
5 [1] Fracaso Fracaso Fracaso Fracaso Éxito Fracaso Éxito Éxito
6 [9] Éxito Fracaso Éxito Fracaso Fracaso Fracaso Éxito
7 Levels: Éxito Fracaso
8 > wilcox.test(datosAyuda$CantAyuda ~ datosAyuda$Resultado, paired=F,
9 alternative="less", conf.level = 0.95)
10
11 Wilcoxon rank sum test with continuity correction
12 data: datosAyuda$CantAyuda by datosAyuda$Resultado
13 W = 6, p-value = 0.007163
14 alternative hypothesis: true location shift is less than 0

```

Y como el p-valor es menor que 0.05, se confirma que los sujetos que no tuvieron éxito en resolver su problema sí requirieron de más intervenciones de ayuda que los que sí lo lograron.

Mientras más ayuda requiere un alumno, más probable parece ser que no logrará resolver el problema. Al menos eso ocurrió con los sujetos investigados.

Esto es consistente con los resultados de las recientes investigaciones de Arens, Frenzel, y Goetz (2020) y de Li, Liu, Zhang, y Liu (2020) que informan sobre una relación recíproca positiva a lo largo del tiempo entre la auto-eficacia en matemática y las notas de exámenes de matemática, entendiendo **auto-eficacia** como la confianza auto-percibida de los estudiantes de realizar exitosamente ciertas tareas dadas en un dominio específico (Bandura, 2001). El único sujeto que recibió varias veces ayuda y sí logró resolver su problema fue el S07. Sin embargo hay que aclarar que se trata de un problema muy sencillo (ver Apéndice B en la página 221).

También se ha identificado en las investigaciones de Avsec y Szewczyk-Zakrzewska (2018) y en Kisti y Aini (2012, como se citó en Hidayat y cols., 2018) la relación positiva y significativa entre la auto-eficacia y la creatividad, siendo la creatividad un constructo fuertemente asociado a la resolución de problemas (Hidayat y cols., 2018)² y con la resolución de problemas y el *Pensamiento Computacional* (Doleck y cols., 2017; Voskoglou y Buckley, 2012).

El Sujeto S11

El sujeto S11, figura 4.55 en la página 178, ejemplifica el episodio A de la Figura 1 de Granberg (2016, p. 35), copiada en este trabajo en la figura 2.3 en la página 15. Esto debido a que el enunciado del problema S11 menciona casi explícitamente cómo se debe implementar el algoritmo solución, por lo que no se trata propiamente de un problema según las definiciones de Ferrer Vicente (2000) y Companioni Madeu (2005) asumidas en esta investigación. Además puede comprobarse que este sujeto fue el que menor cantidad de tiempo requirió para alcanzar la solución, debido a la misma razón.

Sin embargo, aún con esta «facilidad» este alumno requirió de una intervención de ayuda de parte del profesor de más de medio minuto.

Tiempo de análisis

Los sujetos S01, S03, S06, S08, S09, S10, S11, S12, S13 y S14, 10 de los 15, realizaron muy poco tiempo o nada de tiempo la actividad de Análisis, y de ellos, sólo los sujetos S08, S09 y S11 lograron tener éxito en la resolución, pero hay que considerar la situación descrita en el párrafo anterior con respecto al sujeto S11. El resto, 7 de 10, realizaron poco o nada de análisis y no tuvieron éxito al resolver su problema. De nuevo, esto corrobora lo planteado por Schoenfeld (1985, 2016), Granberg (2016) y Blanco Hamad (2016) acerca de que los novatos proceden a explorar o a implementar una solución sin haberla pensado bien.

De los que sí dedicaron un poco más de tiempo a la actividad de Análisis, tan importante para poder resolver un problema, los sujetos S05, S07 y Exp tuvieron éxito, mientras que

²Aunque hay que reconocer que existen investigaciones que NO encuentran evidencia de tal relación, como las de Houtz, Montgomery, Kirkpatrick, y Feldhusen (1979) y la de Demir Barutcu (2017).

los sujetos S02 y S04 no tuvieron éxito en resolver su problema a pesar de haber dedicado un poco más de tiempo al Análisis. Ver cuadro 4.5.

Cuadro 4.5.: Sujetos que dedicaron mucho o poco tiempo de Análisis del problema.

	Éxito	Fracaso
Poco o nulo tiempo de Análisis del problema	S08, S09, S11	S01, S03, S06, S10, S12, S13, S14
Más tiempo de Análisis del problema	S05, S07, Exp	S02, S04

Fuente: Elaboración propia.

Pareciera indicar que dedicar poco tiempo de análisis implica un alto riesgo de fracasar.

El Sujeto S06

El sujeto S06 es muy peculiar ya que requirió de mucha ayuda (ver figura 4.50 en la página 173). Después de cada una de las intervenciones de ayuda el sujeto parecía estar mejor encaminado, sin embargo a cada momento presentaba nuevas dudas y conflictos cognitivos acerca de cómo proceder a continuación, de tal forma que no podía avanzar por sí mismo. Finalmente, después de comprender esto, el investigador decidió dar por terminada la sesión después de convencerse de que el sujeto no era realmente capaz de resolver el problema por sí sólo en un tiempo razonable, a pesar de que el sujeto aparentaba estar siempre a punto de resolverlo.

Tiempo de Planificación de la solución

Excluyendo ahora los sujetos S06 y S11 por sus particulares características, podemos relacionar positivamente el tiempo dedicado a la planificación de la solución al inicio del tiempo disponible para resolver el problema con el éxito en resolverlo. Esto es porque los sujetos que dedicaron poco tiempo a la planificación de la solución al inicio del tiempo disponible son los S01, S02, S03, S04, S05, S10, S12 y S14; y de estos sólo el S05 tuvo éxito. Mientras tanto, los que sí dedicaron un poco más de tiempo a la planificación de la solución cerca del inicio del tiempo disponible son los sujetos S07, S08, S09, S13 y Exp, de los cuales sólo el S13 no tuvo éxito en resolver su problema. Ver cuadro 4.6.

Cuadro 4.6.: Sujetos que dedicaron mucho o poco tiempo de Planificación de la Solución.

	Éxito	Fracaso
Poco o nulo tiempo de Planificación de la Solución	S05	S01, S02, S03, S04, S10, S12, S14
Más tiempo de Planificación de la Solución	S07, S08, S09, Exp	S13

Fuente: Elaboración propia.

Una vez más, esto corrobora lo planteado por Schoenfeld (2016) acerca de que los novatos «se casan» (es decir que se comprometen casi ciegamente) con una idea que intentan desarrollar pero sin reflexionar seriamente si realmente hay buenas probabilidades de que los conduzca a una solución correcta.

Cambios abruptos

Excluyendo al sujeto S11, otra característica notoria en algunos sujetos es que realizan algunos o varios cambios abruptos entre diferentes actividades cognitivas, es decir, realizar cierta actividad por un corto período de tiempo y luego pasar rápidamente a otra. Los sujetos que realizaron dos o más de estos cambios son S01, S04, S05, S06, S07, S08 y S09. De ellos, sólo S01, S04 y S06 no tuvieron éxito, los demás lograron resolver su problema. Aquellos que sólo realizaron uno o ningún cambio abrupto son S02, S03, S10, S12, S13, S14 y Exp, de los cuales sólo el sujeto Exp (el experto) tuvo éxito. Ver cuadro 4.7.

Cuadro 4.7.: Sujetos que realizaron Cambios abruptos de actividad cognitiva.

	Éxito	Fracaso
Más de un cambio abrupto	S05, S07, S08, S09	S01, S04, S06
Uno o ningún cambio abrupto	Exp	S02, S03, S10, S12, S13, S14

Fuente: Elaboración propia.

Pareciera que estos cambios abruptos, mientras más ocurran, están correlacionados positivamente con el éxito al resolver el problema. Sin embargo, la muestra es demasiado pequeña para concluir tal cosa.

Los «no puedo»

Se registraron también los instantes en los que los alumnos expresaban de viva voz que no podían resolver el problema que les fue asignado. Como se mencionó antes, este es un fenómeno del que no hablan ni Schoenfeld (2015, 2016) ni Granberg (2016), pero observada en varios de los sujetos aquí analizados.

Algunas de las expresiones usadas a demás de «No, no puedo resolverlo», fueron:

- «Sí sé mi algoritmo pero no logro plasmarlo»
Recuérdese la cita de Knuth (1974, p. 327) de que uno no entiende realmente algo (por ejemplo un problema) hasta que pueda expresarlo como un algoritmo. Es decir que si uno realmente entiende algo, entonces es capaz de expresarlo como un algoritmo, y también, aplicando la contrarrecíproca: Si uno no es capaz de expresar un algoritmo, entonces no entiende realmente lo que debe hacer. Luego esta frase usada por algunos alumnos en realidad (aplicando la contrarrecíproca) indica que no entienden realmente el problema, y por eso no logran plasmarlo como algoritmo.
- «No me acuerdo cómo se resuelve»
Esto indica que este estudiante no ha entendido que la actividad se trata de construir

un algoritmo que resuelva el problema, y no de buscar en un catálogo el algoritmo exacto que lo resuelva.

- «Se me revuelve toda la cabeza»

Esta frase, junto con la reacción física que la acompañó, indican un nivel de ansiedad, angustia e indefensión muy alta, insoportable para el alumno.

Los sujetos S02, S06, S10 y S13 lo hicieron una vez. El sujeto S12 lo hizo tres veces. Todos ellos no tuvieron éxito al resolver su problema. No hubo ningún sujeto que afirmara no poder resolver el problema y tuviera éxito al hacerlo. De aquellos que no expresaron no poder resolver su problema, 10 sujetos, seis de ellos tuvieron éxito mientras que sólo cuatro no lo lograron. Ver cuadro 4.8.

Cuadro 4.8.: Sujetos que expresaron o no, no poder resolver su problema.

	Éxito	Fracaso
Dijeron «No puedo»		S02, S06, S10, S12, S13
No dijeron «No puedo»	S05, S07, S08, S09, S11, Exp	S01, S03, S04, S14

Fuente: Elaboración propia.

La investigación longitudinal de [Arens y cols. \(2020\)](#) como se mencionó antes, reporta una correlación recíproca positiva entre las notas de los alumnos en matemática y su auto-eficacia matemática. También reporta una relación unidireccional positiva entre las notas previas obtenidas en matemática y un posterior auto-concepto matemático. Sin embargo no encuentran evidencia de que un alto auto-concepto matemático esté correlacionado con posteriores altas notas en matemática.

Si extrapolamos los resultados de [Arens y cols. \(2020\)](#) a esta investigación, podríamos decir que aquellos sujetos que mostraron claramente tener baja auto-eficacia matemática y bajo auto-concepto matemático son así debido a sus bajas notas en evaluaciones o exámenes previos.

Como nos indican [Twenge, Catanese, y Baumeister \(2002\)](#), es indiscutible que patrones de comportamiento contraproducente (entre los que se encuentra el auto-derrotismo) realmente se presentan en poblaciones supuestamente normales y mentalmente sanas, y que la actividad auto-derrotista es real pero no necesariamente intencional. [Twenge y cols. \(2002\)](#) también nos dice que este tipo de comportamiento contraproducente (entre ellos el auto-derrotismo) se caracteriza porque la persona busca un resultado positivo o un beneficio, pero en la búsqueda de este resultado positivo produce uno negativo de una de dos formas: (a) Un resultado negativo está enlazado a uno positivo, y la persona principalmente busca el positivo, pero de este modo trae el malo con este. (b) La segunda forma es aplicar una estrategia en la que «el tiro sale por la culata». También nos dicen que estos mecanismos implican alguna falla en la auto-regulación.

El auto-derrotismo identificado aquí es del primer tipo. Algunos alumnos al parecer están tan bloqueados o ansiosos o angustiados o estresados que sólo quieren terminar con esa sensación desagradable. Sin embargo trae como resultado no resolver el problema y no ganar el puntaje correspondiente.

El Sujeto experto, Exp

Finalmente, el sujeto Exp presenta un patrón de actividad con algunas similitudes pero sensiblemente diferente al de los alumnos en muchos aspectos. Primero realiza un proceso de análisis antes de comenzar a implementar (escribir código), similar a los sujetos S03, S07, S08, S09, S12 y S13, pero significativamente más prolongado que estos en términos relativos al tiempo total de cada sesión. No requirió de ayuda ya que aunque realmente no sabía cómo resolverlo en un primer instante, comprendió rápidamente el problema y fue construyendo la solución poco a poco. No comenzó a implementar sin saber qué debía escribir, como lo hicieron los sujetos S01, S02, S03, S04, S10, S12 y S14. También es visiblemente menor la cantidad de tiempo que necesitó para la resolución con respecto a los alumnos (excluyendo al sujeto S11 ya que no tenía realmente un problema).

Además, este sujeto dedicó una considerable cantidad (proporcional) de tiempo a Construir la solución, cosa que no hicieron los alumnos. Alternó la construcción de la solución con momentos de implementación, y luego de implementación con verificación, parecido a los sujetos S03, S04, S05, S10 y S14, pero de una forma más sistemática y equilibrada.

Finalmente, este sujeto no realizó cambios abruptos ni expresó en ningún momento que no podía resolver el problema asignado.

Se puede decir que el patrón de este sujeto se ajusta a lo expresado por Schoenfeld (1985, 2016) y Granberg (2016) acerca de que los expertos dedican más tiempo a pensar el problema, cerca de la mitad del tiempo.

5. Conclusiones, Discusión y Recomendaciones

5.1. Conclusiones

5.1.1. Conclusiones de la parte cuantitativa

En base a los resultados cuantitativos presentados en la sub-sección 4.1.3 (ver figuras 4.12 en la página 135 y 4.13 en la página 139), se concluye lo siguiente (ver sección 1.5 en la página 7):

1. Se **confirma** la hipótesis de que el nivel de desarrollo de *Pensamiento Algorítmico* de los estudiantes de las carreras de Licenciatura en Matemática y Licenciatura en Estadística, impartidas por la Escuela de Matemática de la Facultad de Ciencias Naturales y Matemática de la Universidad de El Salvador, está positivamente correlacionado con la cantidad de materias relacionadas con la programación de computadoras que han aprobado.
Esto debido a que se encontró una correlación positiva moderada (0.533) entre las variables PAP y matProg. Es decir que mientras más materias de programación de computadoras se aprueban, aumenta también el nivel de desarrollo de *Pensamiento Algorítmico*. Ver gráfico 4.17 en la página 142. Esta correlación se vuelve más significativa si consideramos que la correlación entre las variables RP y matProg es muy baja (0.155).
2. Se **confirma** la hipótesis de que el nivel de desarrollo de *Pensamiento Algorítmico* de los estudiantes de las carreras de Licenciatura en Matemática y Licenciatura en Estadística, impartidas por la Escuela de Matemática de la Facultad de Ciencias Naturales y Matemática de la Universidad de El Salvador, está positivamente correlacionado con la cantidad total de materias universitarias aprobadas.
Esto debido a que se encontró una correlación positiva moderada (0.505) entre las variables PAP y matTot. Es decir que mientras más materias de la carrera se aprueban, aumenta también el nivel de desarrollo de *Pensamiento Algorítmico*, aunque con menor impacto que con las materias específicas de programación de computadoras. Ver gráfico 4.32 en la página 155. Esta correlación se vuelve más significativa si consideramos que la correlación entre las variables RP y matTot es muy baja (0.203).
3. Se **refuta** la hipótesis de que el nivel de desarrollo de *Pensamiento Algorítmico* de los estudiantes de las carreras de Licenciatura en Matemática y Licenciatura en Estadística, impartidas por la Escuela de Matemática de la Facultad de Ciencias Naturales

y Matemática de la Universidad de El Salvador, está positivamente correlacionado con su Coeficiente de Unidades de Mérito.

Esto debido a que se encontró una correlación positiva pero poco significativa (0.350) entre las variables PAP y cum. Es decir que obtener buenas notas no está fuertemente relacionado con un mayor nivel de desarrollo de *Pensamiento Algorítmico*. Ver gráfico 4.26 en la página 150. A manera de comparación, la correlación entre las variables RP y cum es aún más baja (0.263).

4. Se **refuta** la hipótesis de que el nivel de desarrollo de *Pensamiento Algorítmico* de los estudiantes de las carreras de Licenciatura en Matemática y Licenciatura en Estadística, impartidas por la Escuela de Matemática de la Facultad de Ciencias Naturales y Matemática de la Universidad de El Salvador, está positivamente correlacionado con la edad de los alumnos.

Esto debido a que se encontró una correlación positiva pero poco significativa (0.337) entre las variables PAP y edad. Es decir que ser más o menos joven no está fuertemente relacionado con un menor o mayor nivel de desarrollo de *Pensamiento Algorítmico*. Ver gráfico 4.27 en la página 151. Sin embargo, esta correlación es más fuerte que entre las variables RP y edad que es casi nula (0.041).

5. Se **confirma** la hipótesis de que el nivel de desarrollo de *Pensamiento Algorítmico* de los estudiantes de las carreras de Licenciatura en Matemática y Licenciatura en Estadística, impartidas por la Escuela de Matemática de la Facultad de Ciencias Naturales y Matemática de la Universidad de El Salvador, es independiente de su género (es decir que es equivalente en hombres y en mujeres).

Esto debido a que no se encontró una correlación significativa entre las variables PAP y gen (0.135), y tampoco se encontraron diferencias significativas entre los grupos de hombres y mujeres. Es decir, al separar en grupos diferentes a los hombres de las mujeres, no se encontró una diferencia estadísticamente significativa. Ver gráfico 4.31 en la página 154. Sin embargo, esta correlación es más fuerte que entre las variables RP y gen que es prácticamente nula (0.086). Ver gráfico 4.39 en la página 161.

6. Se **refuta** la hipótesis de que el nivel de desarrollo de *Pensamiento Algorítmico* de los estudiantes de las carreras de Licenciatura en Matemática y Licenciatura en Estadística, impartidas por la Escuela de Matemática de la Facultad de Ciencias Naturales y Matemática de la Universidad de El Salvador, se manifiesta diferente cuando se les solicita desarrollar un algoritmo que cuando se les solicita desarrollar unas instrucciones.

Esto debido a que no se encontró una correlación significativa entre las variables PAP y alg (0.027), y tampoco se encontraron diferencias significativas entre los grupos de alumnos con alg igual a 0 (Falso) y alg igual a 1 (verdadero). Es decir, que si se pide un algoritmo, o si se pide unas instrucciones, los resultados son equivalentes. Ver gráfico 4.24 en la página 149.

5.1.2. Interpretación de resultados de la parte cualitativa

La mayoría de los sujetos analizados presentan algunos rasgos característicos de los resolutores novatos de problemas. Entre ellos los siguientes:

- Comienzan a escribir, es decir, intentan implementar una solución del problema, sin saber muy bien qué deben escribir.
- Presentan muchas inseguridades sobre su propio conocimiento.
- No dedican suficiente tiempo a pensar en el problema, entenderlo, y resolverlo matemáticamente, antes de emprender una implementación algorítmica.
- Tienden a comprometerse sin mayor justificación con una idea inicial, y cuando esta no funciona, se quedan «atorados» tratando de «arreglarla» en lugar de replantearse aplicar otra idea diferente.

Además se observaron otros problemas de aprendizaje, con diferente nivel de gravedad, en algunos sujetos:

- Tienen muchas lagunas en el conocimiento que se supone que proveen las materias precedentes (prerrequisitos). Algunos de estos conocimientos fueron o debieron ser ejercitados en la materia que están cursando.
- Se detectó mucha inestabilidad cognitiva (cambios abruptos entre diferentes tareas cognitivas) durante la resolución de los problemas.
- Se observó mucho auto-derrotismo ante problemas que en teoría no deberían representar grandes dificultades.

Algunos de los sujetos analizados presentan también incipientemente algunos rasgos característicos de los resolutores expertos de problemas. Entre ellos los siguientes:

- No intentar resolver el problema inmediatamente después de leer el enunciado sin haberlo pensado un poco.
- Dedicar tiempo suficiente a entender el problema (análisis), antes de emprender un intento de solución.
- Dedicar tiempo suficiente a la construcción de una solución correcta en vez de perder el tiempo tratando de arreglar soluciones incorrectas.

En resumen, los estudiantes de la población analizada, aún al estar a punto de finalizar las tres materias relacionadas a la programación de computadoras presentan un desempeño regular en la resolución de problemas de pequeña escala cuya solución es un algoritmo, y en general les causa ansiedad o angustia.

5.1.3. Metainferencias (Triangulación) de los Resultados

Los sujetos de la parte cualitativa son más o menos equivalentes a los sujetos de la parte cuantitativa con matProg igual a 2 ó 3 (ver figura 4.17 en la página 142), por lo que son también más o menos comparables.

En este sentido, el nivel de desarrollo de *Pensamiento Algorítmico* reflejado en la variable PAP (ver figuras 4.9 en la página 129, 4.17 en la página 142 y 4.32 en la página 155) de la parte cuantitativa, especialmente en el grupo de alumnos con matProg igual a 2 ó 3, es consistente con los resultados de la parte cualitativa en lo que se refiere a las dificultades en el análisis y construcción de la solución de problemas cuya solución es un algoritmo. Esto puede verse en la figura 4.18 en la página 143, donde se presenta el detalle de las sub-variables asociadas al análisis de los problemas de la parte cuantitativa p2-input, p2-output, p2-idsub, p4-input, p4-dibu, p4-output y p4-idsub. Los valores de estas sub-variables tienen mucha dispersión.

Así mismo, también es consistente con el incipiente avance de algunos alumnos en lo que se refiere a su habilidad de resolver problemas cuya solución es un algoritmo. Esto puede apreciarse en la figura 4.19 en la página 144, específicamente con las sub-variables p2-solsub, p2-intsub, p2-extre, p4-solsub y p4-intsub.

5.2. Discusión y recomendaciones

5.2.1. Sobre la revisión bibliográfica

Es necesario realizar más investigación epistemológica y praxeológica sobre la dualidad algoritmo-concepto de la matemática (ver Lovász, 1988; Pyzara, 2012; Sfard, 1991).

Las definiciones conceptuales y operacionales de los constructos *Pensamiento Algorítmico* y *Pensamiento Computacional* no están completamente determinadas, y es importante seguir investigando sobre ellos, sus beneficios, sus desafíos, dificultades, obstáculos, y cómo incorporarlos (o empotrarlos) en los diferentes sistemas educativos. Sin embargo, tal como sugieren Voogt y cols. (2015), tal vez no deberíamos buscar unas definiciones definitivas, sino más bien encontrar similitudes y relaciones en las diferentes investigaciones.

A pesar de que la literatura científica indica que tampoco está claro cómo incorporar, y menos cómo empotrar, el *Pensamiento Algorítmico* y el *Pensamiento Computacional* en la Enseñanza de la Matemática el autor de esta tesis considera que las formas de hacer matemática están profundamente ampliadas en la actualidad gracias las posibilidades que ofrecen las computadoras, y que se necesita del fomento del *Pensamiento Computacional*, y por lo tanto también del *Pensamiento Algorítmico*, entre los jóvenes (y no tan jóvenes) matemáticos para poder aprovechar esta ampliación.

5.2.2. Sobre los resultados de la parte cuantitativa

La correlación entre las variables PAP y matProg es mucho más baja de lo esperado. Esto confirma la dificultad de desarrollar el *Pensamiento Algorítmico*, y que se corresponde con los bajos resultados a nivel internacional reportados por McCracken y cols. (2001), Guzdial y Soloway (2002), Kinnunen y Malmi (2006), Arévalo-Mercado y cols. (2019) y Vargas y cols. (2013). Ver sección 1.1.

La correlación entre las variables RP y matTot es muy baja (ver figura 4.43), y considerando que los problemas 1 y 3 planteados en la parte cuantitativa no eran particularmente difíciles (ver páginas 69 y 70), pero que sí requerían de habilidades de modelamiento matemático, se podría inferir que el estilo de problemas que los alumnos deben resolver en las materias de la carrera no requieren de habilidades de modelamiento. Esta idea en particular es respaldada porque la variable p3-mod muestra un bajo rendimiento en la habilidad de modelamiento en un problema geométrico, y la variable p1-ciu muestra una total independencia de la cantidad de materias que los estudiantes han aprobado. Ver figuras 4.42 en la página 164 y 4.41 en la página 163.

Sin embargo se requiere más evidencia para poder confirmar esa conjetura. Por ejemplo se necesitaría diseñar un instrumento que mida con más precisión la habilidad de modelar en una muestra seleccionada por conglomerados a través de todas las cohortes de estudiantes, o bien a través de todas las materias o de los ciclos académicos (semestres o años).

Para lograr desarrollo del *Pensamiento Algorítmico* es necesario el fortalecimiento de la habilidad de Resolución de problemas y de Modelamiento. Sobre la fuerte relación entre la resolución de problemas y la habilidad de algoritmizar nos hablan Delgado Rubí (1999), Futschek (2006) y Modeste (2012b). Y sobre la relación del modelamiento y la habilidad de algoritmizar nos hablan Ambrus (2014), Grozdev y Terzieva (2015) y Gravemeijer y cols. (2017). Ver las sub-secciones 2.2.3 y 2.2.7.

La dispersión de valores de PAP y RP a lo largo de la carrera (matTot) es muy alta. Ver las figuras 4.33 en la página 156 y 4.44 en la página 166. Esto sólo apoya la idea de que tanto el desarrollo del *Pensamiento Algorítmico* como la habilidad de modelamiento matemático no son proporcionados de forma consistente por las materias cursadas en la carrera. Parecen depender de otras variables, entre ellas, tal vez de «la inteligencia» y/o de la formación previa, y/o de las técnicas de estudio aplicadas por los estudiantes.

La correlación entre las variables PAP y RP con cum es muy baja. En esto se coincide con la investigación de Doleck y cols. (2017) que tampoco encontró correlación entre *Pensamiento Algorítmico* y rendimiento académico. Igualmente la correlación entre las variables PAP y RP con edad también es muy baja. Esto una vez más apoya la conjetura el párrafo anterior.

Parece haber ligeramente mayor rendimiento en las mujeres (ver gráficos 4.31 y 4.39), pero hace falta mayor exploración al respecto.

La falta de diferencias de puntajes de la variable PAP y la variable alg (ver figura 4.24) podría explicarse al extrapolar lo descubierto por Madeo y Bird (1990) sobre el hecho de que tener que especificar por sí mismos los elementos del lenguaje produce que los alumnos tengan menor rendimiento en la solución de este tipo de problemas. Sin embargo, esto también indicaría que los alumnos de los grupos P1, P2 y P3, a pesar de haber aprobado materias donde aprenden cómo expresar soluciones algorítmicas en lenguajes algorítmicos (diagramas de flujo y pseudocódigo) y otro lenguaje formal, no han sido en su mayoría capaces de aplicar esos conocimientos, porque posiblemente ya no los recuerdan un semestre o un año después. Ver los bajos puntajes de las variables p2-flujo, p4-flujo, p2-sin y p4-sin en la figura 4.20 en la página 145.

Finalmente hay que decir que hay mucha información valiosa en los instrumentos que respondieron los alumnos en la parte cuantitativa y que ameritan mucha mayor investigación, por ejemplo los siguientes:

- Interferencia entre distintos modos de pensamiento matemático durante la resolución de un problema.
Por ejemplo, aplicar pensamiento combinatorio en lugar de pensamiento algorítmico o en lugar de pensamiento algebraico.
- Deshonestidad académica por parte de algunos estudiantes.
- La forma misma de modelar, especialmente lo referente a descartar los datos y la información irrelevante de aquello que sí es relevante para poder resolver el problema.

5.2.3. Sobre los resultados de la parte cualitativa

Sobre la Metodología, hay que reconocer la importancia de la práctica de analizar el pensamiento matemático a través de grabaciones de video. Este es un ejercicio que vale la pena introducir en la formación de profesores de matemática. Sin embargo, también hay que considerar, como informan [Ulusoy y Çakıroğlu \(2018\)](#), la dificultad de desarrollar la habilidad de detectar el pensamiento matemático a través de grabaciones de video.

Sobre los resultados propiamente, también hay mucho que decir. Primero, que se llegó a prácticamente los mismos resultados que la preprueba que realizó [Blanco Hamad \(2016\)](#): «existen dificultades al momento de analizar, interpretar y abstraerse para dar solución a una situación problémica, pues los estudiantes comienzan a resolver dicha situación sin tener totalmente claro lo que se les exige, sin haber analizado y determinado las condiciones y exigencias de la misma. Esto conlleva, en muchas ocasiones, a algoritmizar un problema distinto al que se plantea, o a que la solución algorítmica que se obtenga no los resuelva totalmente.» (p. 33), y «la mayoría de los estudiantes no encontró una vía correcta para resolver este problema, presentando dificultades asociadas al insuficiente tiempo que dedicaron al análisis e interpretación del mismo.» (p. 38)

Es posible que la inestabilidad cognitiva, es decir, los cambios abruptos identificados en el proceder de los alumnos durante la resolución de un problema cuya solución es un algoritmo, estén en realidad más relacionados con una agilidad mental alta, razón por la cual estarían correlacionados con el éxito, y menos relacionados con confusión o falta de entendimiento. También habría que indagar en el efecto de la ansiedad matemática en la resolución de este tipo de problemas matemáticos. Ver [Monje, Pérez-Tyteca, y Castro \(2012\)](#).

Hay que investigar mucho más las características y las razones del auto-derrotismo detectado. Puede haber muchos factores, como la situación de cambio abrupto de educación presencial a educación a distancia debido a la pandemia de COVID-19. También podría deberse a otros factores como la ansiedad, la vergüenza, el estrés, la falta de preparación para la evaluación, baja auto-estima, y la baja auto-eficacia matemática en general.

Según la investigación de Arens y cols. (2020), como se mencionó en la sección de resultados, podríamos decir que aquellos sujetos que mostraron claramente tener baja auto-eficacia matemática y bajo auto-concepto matemático son así debido a sus bajas notas en evaluaciones o exámenes previos. Así que al auto-derrotismo podríamos agregar como factor causal adicional las bajas calificaciones previas en este tipo de problemas.

Sobre auto-eficacia, Dökmecioğlu, Taş, y Yerdelen (2018) reporta que las diferentes técnicas de enseñanza constructivista promueven la auto-eficacia. Esto podría promover la consolidación a más largo plazo los conocimientos adquiridos en cada materia (promover aprendizaje significativo).

A pesar de haber aprobado muchas materias de la carrera (ver cuadro 4.3 en la página 167), los alumnos presentan muchos rasgos de resolutor novato de problemas. Hace falta una indagación más profunda para determinar la efectividad de las materias en ese sentido.

Finalmente, hay que reconocer y valorar los avances de los estudiantes en materia de resolución de problemas y modelamiento matemático, más allá del aprendizaje memorísticos. Sin embargo, esto es muy difícil ya que se requiere de un examen muy profundo, casi íntimo, de los procesos cognitivos que aplican los alumnos y su evolución en el tiempo.

5.2.4. Sobre la evaluación por rúbricas en la parte cuantitativa

A pesar de los beneficios del uso de rúbricas para la evaluación educativa (ver Andrade, 2005), la validez de las rúbricas en última instancia, como de cualquier otro instrumento de evaluación, depende de muchos factores, muchos de ellos subjetivos, como advierte Kane (2013). Sin embargo, el autor de esta tesis considera que una buena rúbrica, desarrollada a partir de principios bien fundamentados, siempre es bienvenida ya que su mero desarrollo puede propiciar una discusión científica productiva.

Crespo y Nicol (2003) enfatizan la importancia de las entrevistas para entender el pensamiento matemático de los alumnos. En esta tesis no se aplicaron entrevistas, principalmente debido a las restricciones de recursos tecnológicos de los estudiantes durante el confinamiento por la pandemia de COVID-19.

En la revisión bibliográfica de la investigación de Grover (2017, p. 271) se advierte que los diferentes tipos de evaluación usados en las publicaciones revisadas, lo que incluye a las rúbricas, son valiosos para la investigación y para proveer una vista holística del aprendizaje de los estudiantes, pero que pueden no ser prácticas para medir, especialmente los contenidos usados en las aulas de clase de escuelas promedio.

Finalmente, hay que reconocer, como hace Grover (2017, p. 285), que ningún tipo de evaluación por sí sola puede capturar los aspectos cognitivos, afectivos y de transferencia del aprendizaje. Cada tipo de evaluación sirve a un propósito, y juntos [diferentes tipos de evaluación], proveen una vista integrada del aprendizaje de los estudiantes. Y tal como muestra la investigación de Grover (2017), no todas las evaluaciones funcionan igual de bien para demostrar el aprendizaje de todos los estudiantes.

«La forma que tenemos de evaluar marca inexorablemente a nuestros estudiantes, en la escuela y a lo largo de toda su vida, y con ello se contribuye a crear una sociedad u otra.

[...] No es lo mismo completar un test “neutro” con respuestas verdaderas y falsas (corregido por un lector óptico, que es más barato), que desarrollar una prueba donde [los alumnos] tengan que buscar alternativas creativas ante un dilema de una situación real.» (Murillo y Hidalgo, 2015, p. 5)

Referencias Bibliográficas

- Aho, A. V. (2012). Computation and computational thinking. *The Computer Journal*, 55(7), 832–835. doi: 10.1093/comjnl/bxs074
- Aho, A. V., y Ullman, J. D. (1992). *Foundations of computer science*. Rockville, MD: Computer Science Press. Descargado de <http://infolab.stanford.edu/~ullman/focs.html>
- Ambrus, G. (2014). Problem solving and Modelling – Traditions and Possibilities in Hungarian Mathematics Education. En A. Ambrus y É. Vásárhelyi (Eds.), *15th ProMath Problem Solving in Mathematics Education Conference* (pp. 7–17). Eger, Hungría: Mathematics Teaching and Education Center. Descargado de <https://helda.helsinki.fi/bitstream/handle/10138/230168/ProMath2013.pdf?sequence=1#page=7>
- Andrade, H. G. (2005). Teaching With Rubrics: The Good, the Bad, and the Ugly. *College Teaching*, 53(1), 27–31. doi: 10.3200/CTCH.53.1.27-31
- Anghileri, J. (2001). What are we trying to achieve in teaching standard calculating procedures? En M. van den Heuvel-Panhuizen (Ed.), *Proceedings of the 25th Conference of the International Group for the Psychology of Mathematics Education (PME 25)* (Vol. 2, pp. 41–48). Utrecht, The Netherlands: International Group for the Psychology of Mathematics Education. Descargado de <http://files.eric.ed.gov/fulltext/ED466950.pdf#page=543>
- Arens, A. K., Frenzel, A. C., y Goetz, T. (2020). Self-Concept and Self-Efficacy in Math: Longitudinal Interrelations and Reciprocal Linkages with Achievement. *The Journal of Experimental Education*. doi: 10.1080/00220973.2020.1786347
- Arévalo-Mercado, C. A., Estrada-Rentería, B. G., y Muñoz-Andrade, E. L. (2019). El efecto de la teoría de carga cognitiva en el aprendizaje de la programación básica. *Entorno*(67), 169–176. doi: 10.5377/entorno.v0i67.7500
- Artigue, M. (2003). ¿Qué se puede aprender de la investigación educativa en el nivel universitario? *Boletín de la asociación matemática venezolana*, 10(2), 117–134. Descargado de <http://emis.ams.org/journals/BAMV/conten/vol10/bamv2003-2.pdf#page=25>
- Artopoulos, A. (2013). El docente traductor: claves para la integración de tecnología en la escuela. *Revista Linhas*, 14(27), 59–82. doi: 10.5965/198472381427201359
- Association for Computing Machinery, Code.org, Computer Science Teachers Association, Cyber Innovation Center, y National Math and Science Initiative. (2016). *K–12 Computer Science Framework*. Descargado de <https://k12cs.org/wp-content/uploads/2016/09/K%E2%80%9312-Computer-Science-Framework.pdf>
- Avsec, S., y Szewczyk-Zakrzewska, A. (2018). Engineering students' self-efficacy and goal orientations in relation to their engineering design ability. *Global Journal of*

- Engineering Education*, 20(2), 85–90. Descargado de <http://www.wiete.com.au/journals/GJEE/Publish/vol20no2/01-Avsec-S.pdf>
- Bandura, A. (2001). Social Cognitive Theory: An Agentic Perspective. *Annual Review of Psychology*, 52(1), 1–26. doi: 10.1146/annurev.psych.52.1.1
- Baranauskas, M. C. C., y Carbajal, M. L. (2017). The social nature of programming: Children and fluency. En *International Conference on Human-Computer Interaction* (pp. 291–308).
- Barash, A., y Klein, R. (1996). Seventh grades students' algorithmic, intuitive and formal knowledge of multiplication and division of non negative rational numbers. En L. Puig y A. Gutierrez (Eds.), *Proceedings of the 20th Conference of the International Group for the Psychology of Mathematics Education (PME 20)* (Vol. 2, pp. 35–42). Valencia, Spain: International Group for the Psychology of Mathematics Education. Descargado de <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.852.5664&rep=rep1&type=pdf#page=44>
- Barr, D., Harrison, J., y Conery, L. (2011). Computational Thinking: A Digital Age Skill for Everyone. *Learning & Leading with Technology*, 38(6), 20–23. Descargado de <http://files.eric.ed.gov/fulltext/EJ918910.pdf>
- Barr, V., y Stephenson, C. (2011). Bringing Computational Thinking to K-12: What is Involved and What is the Role of the Computer Science Education Community? *ACM Inroads*, 2(1), 48–54. doi: 10.1145/1929887.1929905
- Batanero Bernabeu, M. d. C. (2013). Sentido estadístico - Componentes y desarrollo. *Probabilidad Condicionada: Revista de didáctica de la Estadística*(1), 55–64. Descargado de <https://dialnet.unirioja.es/descarga/articulo/5487196.pdf>
- Bell, T., Alexander, J., Freeman, I., y Grimley, M. (2009). Computer science unplugged: School students doing real computing without computers. *The New Zealand Journal of Applied Computing and Information Technology*, 13(1), 20–29.
- Blanco Hamad, A. (2016). *Sistema Básico de Habilidades para la Algoritmización Computacional* (Tesis de Master, Universidad de Oriente, Santiago de Cuba, Cuba). Descargado de https://www.researchgate.net/publication/334138906_SISTEMA_BASICO_DE_HABILIDADES_PARA_LA_ALGORITMIZACION_COMPUTACIONAL
- Blanco Hamad, A., Salgado Castillo, A., y Alonso Berenguer, I. (2016). Habilidades para la algoritmización computacional en la Licenciatura en Educación: Especialidad Educación Laboral-Informática. *Maestro y Sociedad. Revista Electrónica para maestros y profesores*, 13(1), 18–30. Descargado de <http://hdl.handle.net/10760/33534>
- Bloom, B. S., y Krathwohl, D. R. (1956). *Taxonomy of educational objectives. Vol. 1: Cognitive domain* (Vol. 1). New York: McKay.
- Böhm, C., y Jacopini, G. (1966). Flow Diagrams, Turing Machines and Languages with Only Two Formation Rules. *Communications of ACM*, 9(5), 366–371. doi: 10.1145/355592.365646
- Booker, M. J. (2007). A roof without walls: Benjamin Bloom's taxonomy and the misdirection of American education. *Academic Questions*, 20(4), 347–355. doi: 10.1007/s12129-007-9031-9
- Bordignon, F., y García-Marín, D. (2018). Principios del pensamiento computacional. En R. Aparici y D. G. Marín (Eds.), *Comunicar y educar en el mundo que viene* (Segunda ed., pp. 177–191). Barcelona: GEDISA.

- Borovik, A. (2011). Information technology in university-level mathematics teaching and learning: a mathematician's point of view. *Research in Learning Technology*, 19(1), 73–85. doi: 10.1080/09687769.2010.548504
- Boufi, A., y Skaftourou, F. (2004). From formal to semi-informal algorithms: the passage of a classroom into a new mathematical reality. En *The 28th International Conference of the International Group for the Psychology of Mathematics Education* (pp. 151–158). Bergen, Norway. Descargado de <https://files.eric.ed.gov/fulltext/ED489701.pdf>
- Breda, A., y Dos Santos, J. M. D. S. (2019). Spherical tiling with GeoGebra. New Results, Challenges and Open Problems. *Resonance*, 24(8), 861–873. doi: 10.1007/s12045-019-0849-6
- Brolley, L. (2015a). Exploring the enduring gap between undergraduate math and professional practice: the case of computer programming. En T. G. Bartell, K. N. Bieda, R. T. Putnam, K. Bradfield, y H. Dominguez (Eds.), *Proceedings of the 37th annual meeting of the North American Chapter of the International Group for the Psychology of Mathematics Education (PMENA 37)*. East Lansing, MI: Michigan State University.
- Brolley, L. (2015b). *La programmation informatique dans la recherche et la formation en mathématiques au niveau universitaire* (Tesis de Master, Université de Montréal, Montréal, Canadá). Descargado de <https://papyrus.bib.umontreal.ca/xmlui/handle/1866/12574>
- Brolley, L. (2016). Mathematics students' experiences with programming. En M. B. Wood, E. E. Turner, M. Civil, y J. A. Eli (Eds.), *Proceedings of the 38th annual meeting of the North American Chapter of the International Group for the Psychology of Mathematics Education*. Tucson, AZ: The University of Arizona.
- Brolley, L., Buteau, C., y Muller, E. (2017). (Legitimate peripheral) computational thinking in mathematics. En *Tenth Congress of the European Society for Research in Mathematics Education (CERME 10)*. Dublin, Ireland. Descargado de <https://hal.archives-ouvertes.fr/hal-01946353>
- Brolley, L., Caron, F., y Saint-Aubin, Y. (2018). Levels of programming in mathematical research and university mathematics education. *International Journal of Research in Undergraduate Mathematics Education*, 4(1), 38–55. doi: 10.1007/s40753-017-0066-1
- Bubnó, K. T., y Takács, V. L. (2014). Solving word problems by computer programming. En A. Ambrus y É. Vásárhelyi (Eds.), *15th ProMath Problem Solving in Mathematics Education Conference* (pp. 193–208). Eger, Hungría: Mathematics Teaching and Education Center. Descargado de <https://helda.helsinki.fi/bitstream/handle/10138/230168/ProMath2013.pdf?sequence=1#page=193>
- Bulgar, S. (2003). Using research to inform practice: children make sense of division of fractions. En N. A. Pateman, B. J. Dougherty, y J. T. Zilliox (Eds.), *Proceedings of the 2003 Joint Meeting of PME (27) and PMENA (25)* (pp. 157–164). Hawai'i: CRDG, College of Education, University of Hawai'i. Descargado de <https://files.eric.ed.gov/fulltext/ED500921.pdf>
- Bureau of Labor Statistics. (2020). *Employment projections* (Inf. Téc.). Descargado de <http://www.bls.gov/emp/tables.htm>
- Buteau, C., Jarvis, D. H., y Lavicza, Z. (2014). On the integration of Computer Algebra

- Systems (CAS) by Canadian mathematicians: Results of a national survey. *Canadian Journal of Science, Mathematics and Technology Education*, 14(1), 35–57. doi: 10.1080/14926156.2014.874614
- Buteau, C., Muller, E., Marshall, N., Sacristán, A. I., y Mgombelo, J. (2016). Undergraduate mathematics students appropriating programming as a tool for modelling, simulation, and visualization: A case study. *Digital Experiences in Mathematics Education*, 2(2), 142–166. doi: 10.1007/s40751-016-0017-5
- Büyükköroğlu, T., Çetin, N., Deniz, A., Düzce, S., Mahir, N., y Üreyen, M. (2006). The effect of computers on teaching the limit concept. *International Journal for Mathematics Teaching and Learning*. Descargado de <http://www.cimt.org.uk/journal/buyukkoroglu.pdf>
- Castaño-García, J. (2008). Una aproximación al proceso de comprensión de los numerales por parte de los niños: relaciones entre representaciones mentales y representaciones semióticas. *Universitas Psychologica*, 7(3), 895–905. Descargado de <https://www.redalyc.org/pdf/647/64770321.pdf>
- Caswell, T. A., Droettboom, M., Lee, A., Hunter, J. D., Firing, E., Stansby, D., ... Ivanov, P. (2019, dic). *matplotlib/matplotlib: REL: v3.1.2*. Zenodo. doi: 10.5281/zenodo.3563226
- Cebrián de la Serna, M., y Monedero Moya, J. J. (2014). Evolución en el diseño y funcionalidad de las rúbricas: desde las rúbricas 'cuadradas' a las erúbricas federadas. *REDU. Revista de Docencia Universitaria*, 12(1), 81–98. doi: 10.4995/redu.2014.6408
- Centro de Innovación y Calidad de la Educación. (2012). *Informe Diagnóstico Integral a estudiantes que culminan el Ciclo Básico curso 2011- 2012* (Inf. Téc.). La Habana, Cuba: Universidad de las Ciencias Informáticas.
- Chmielewski, L. J., Janowicz, M., Gawdzik, G., y Orłowski, A. (2017). Contiguous line segments in the Ulam spiral: Experiments with larger numbers. En *International Conference on Artificial Intelligence and Soft Computing* (pp. 463–472). doi: 10.1007/978-3-319-59063-9_41
- Companioni Madeu, M. (2005). *Alternativa Didáctica para la Solución de Problemas “no rutinarios” en Cuarto Grado* (Tesis Doctoral, Instituto Superior Pedagógico “José Martí”, Camagüey). Descargado de <http://rediuc.reduc.edu.cu/jspui/bitstream/123456789/159/1/Maximiliano%20Companioni%20Masdeu.pdf>
- Computer Science Teachers Association. (2017). *CS Standards*. Descargado de <https://www.csteachers.org/page/standards>
- Congacha Aushay, E. P., Santillán Castillo, J. R., Guerra Salazar, J. E., y Barba Vera, R. G. (2018). Use of an Informatic Application as a Didactic Strategy for the Development of Academic Aptitudes. *Revista Educación*, 42(2), 398–413. doi: 10.15517/revedu.v42i2.27472
- Corradini, I., Lodi, M., y Nardelli, E. (2017). Conceptions and Misconceptions About Computational Thinking Among Italian Primary School Teachers. En *Proceedings of the 2017 ACM Conference on International Computing Education Research* (pp. 136–144). New York, NY, USA: ACM. Descargado de <http://doi.acm.org/10.1145/3105726.3106194> doi: 10.1145/3105726.3106194
- Coscarelli, W. C., Visscher, M. O., y Schwen, T. M. (1976). Algorithmization and Transfer of Learning. En *Annual Meeting of the American Educational Research Association*

- (pp. 19–23). San Francisco, California. Descargado de <http://files.eric.ed.gov/fulltext/ED121608.pdf>
- Crespo, S., y Nicol, C. (2003). Learning to Investigate Students' Mathematical Thinking: The Role of Student Interviews. En N. A. Pateman, B. J. Dougherty, y J. T. Zilliox (Eds.), *Proceedings of the 2003 Joint Meeting of PME (27) and PMENA (25)* (Vol. 2, pp. 261–268). Hawai'i: CRDG, College of Education, University of Hawai'i. Descargado de <https://files.eric.ed.gov/fulltext/ED500934.pdf>
- Czerkawski, B. C., y Lyman III, E. W. (2015). Exploring Issues About Computational Thinking in Higher Education. *TECHTRENDS*, 59(2), 57–65. doi: 10.1007/s11528-015-0840-3
- D'Ambrosio, U. (1999). Literacy, Matheracy, and Technoracy: A Trivium for Today. *Mathematical Thinking and Learning*, 1(2), 131–153. doi: 10.1207/s15327833mtl0102_3
- Darwazeh, A. N., y Branch, R. M. (2015). A revision to the revised Bloom's taxonomy. En M. Simonson (Ed.), *Proceedings of The 38th Annual Convention of the Association for Educational Communications and Technology* (Vol. 2, pp. 220–225). Indianapolis. Descargado de <https://files.eric.ed.gov/fulltext/ED570118.pdf#page=29>
- Davis, G. E., y McGowen, M. A. (2001). Jennifer's Journey: Seeing and Remembering Mathematical Connections in a Pre-service Elementary Teachers Course. En M. van den Heuvel-Panhuizen (Ed.), *Proceedings of the 25th Conference of the International Group for the Psychology of Mathematics Education (PME 25)* (Vol. 2, pp. 305–312). Utrecht, The Netherlands: International Group for the Psychology of Mathematics Education. Descargado de <https://files.eric.ed.gov/fulltext/ED466960.pdf>
- d'Azevedo Breda, A. M., y Dos Santos, J. M. D. S. (2018). Spherical geometry and spherical tilings with geogebra. *Journal for Geometry and Graphics*, 22(2), 283–299. Descargado de <http://www.heldermann-verlag.de/jgg/jgg22/j22h2bred.pdf>
- DeJarnette, A. F. (2016). Students' use of a computer programming environment to represent distance as a function of time. En M. B. Wood, E. E. Turner, M. Civil, y J. A. Eli (Eds.), *Proceedings of the 38th annual meeting of the North American Chapter of the International Group for the Psychology of Mathematics Education (PMENA 38)* (pp. 1537–1540). Tucson, AZ: ERIC.
- DeJarnette, A. F. (2018). Students' Conceptions of Sine and Cosine Functions When Representing Periodic Motion in a Visual Programming Environment. *Journal for Research in Mathematics Education*, 49(4), 390–423. doi: 10.5951/jresmetheduc.49.4.0390
- Delgado Rubí, J. R. (1998). Las habilidades generales matemáticas. En H. Hernández (Ed.), *Cuestiones de Didáctica de la Matemática*. Rosario, Argentina: Homo Sapiens.
- Delgado Rubí, J. R. (1999). *La enseñanza de la resolución de problemas matemáticos. Dos elementos fundamentales para lograr su eficacia: la estructuración sistémica del contenido de estudio y el desarrollo de las habilidades generales matemáticas* (Tesis Doctoral, Instituto Superior Politécnico "José Antonio Echeverría", La Habana, Cuba). Descargado de https://www.researchgate.net/publication/261872113_La_ensenanza_de_la_resolucion_de_problemas_matematicos_Dos_elementos_fundamentales_para_lograr_su_eficacia_la_estructuracion_sistemica_del_contenido_de_estudio_y_el_desarrollo_de_las_habiliades_genera

- Demir Barutcu, C. (2017). The Relationship Between Problem Solving and Creative Thinking Skills among Nursing Students. *International Journal of Psychology and Educational Studies*, 4, 34–41. doi: 10.17220/ijpes.2017.02.004
- Denning, P. J. (2017). Remaining trouble spots with computational thinking. *Communications of the ACM*, 60(6), 33–39. doi: 10.1145/2998438
- DeWindt-King, A. M., y Goldin, G. A. (2001). A study of children’s visual imagery in solving problems with fractions. En M. van den Heuvel-Panhuizen (Ed.), *Proceedings of the 25th Conference of the International Group for the Psychology of Mathematics Education (PME 25)* (Vol. 2, pp. 345–352). Utrecht, The Netherlands: International Group for the Psychology of Mathematics Education. Descargado de <http://files.eric.ed.gov/fulltext/ED466950.pdf#page=847>
- Dijkstra, E. W. (1968). Letters to the Editor: Go-To Statement Considered Harmful. *Communications of ACM*, 11(3), 147–148. doi: 10.1145/362929.362947
- Dijkstra, E. W. (1974). Programming as a Discipline of Mathematical Nature. *The American Mathematical Monthly*, 81(6), 608–612. doi: 10.1080/00029890.1974.11993624
- diSessa, A. A. (2018). Computational Literacy and “The Big Picture” Concerning Computers in Mathematics Education. *Mathematical Thinking and Learning*, 20(1), 3–31. doi: 10.1080/10986065.2018.1403544
- Doğan, A. (2020). Algorithmic Thinking in Primary Education. *International Journal of Progressive Education*, 16(4), 286–301. doi: 10.29329/ijpe.2020.268.18
- Dökmecioğlu, B., Taş, Y., y Yerdelen, S. (2018). Predicting Students’ Self-Efficacy towards Learning Science by Constructivist Learning Environment Perceptions. *Mediterranean Journal of Educational Research*, 12(24), 85–97. doi: 10.29329/mjer.2018.147.5
- Doleck, T., Bazalais, P., Lemay, D. J., Saxena, A., y Basnet, R. B. (2017). Algorithmic thinking, cooperativity, creativity, critical thinking, and problem solving: exploring the relationship between computational thinking skills and academic performance. *Journal of Computers in Education*, 4(4), 355–369. doi: 10.1007/s40692-017-0090-9
- Dos Santos, J. M. D. S., y Batista Trocado, A. E. (2016). GeoGebra as a Learning Mathematical Environment. *Revista do Instituto GeoGebra Internacional de São Paulo*, 5(1), 05–22. Descargado de <https://revistas.pucsp.br/index.php/IGISP/article/view/26795>
- Douadi, B., Tahar, B., y Hamid, S. (2012). Smart edutainment game for algorithmic thinking. *Procedia – Social and Behavioral Sciences*, 31, 454–458. doi: 10.1016/j.sbspro.2011.12.085
- Dubinsky, E., y Tall, D. (2002). Advanced Mathematical Thinking and the Computer. En D. Tall (Ed.), *Advanced Mathematical Thinking* (pp. 231–243). New York: Kluwer Academic Publishers. Descargado de https://link.springer.com/content/pdf/10.1007/0-306-47203-1_14.pdf
- Durand-Guerrier, V., Meyer, A., y Modeste, S. (2019). Didactical Issues at the Interface of Mathematics and Computer Science. En G. Hanna, D. A. Reid, y M. de Villiers (Eds.), *Proof Technology in Mathematics Research and Teaching* (pp. 115–138). Cham: Springer International Publishing. doi: 10.1007/978-3-030-28483-1_6
- Eysenck, M. W., y Keane, M. T. (2010). *Cognitive psychology: A student’s handbook*. EEUU: Psychology press.
- Fernández Millán, E. (2013). Razonamiento Combinatorio y el currículo español. *Proba-*

- bilidad Condicionada: Revista de didáctica de la Estadística*(1), 539–545. Descargado de <https://dialnet.unirioja.es/descarga/articulo/5487251.pdf>
- Ferrer Vicente, M. (2000). *La resolución de problemas en la estructuración de un sistema de habilidades matemáticas en la escuela media cubana* (Tesis Doctoral, Instituto Superior Pedagógico “Frank País García”, Santiago de Cuba, Cuba). Descargado de <https://www.eumed.net/tesis-doctorales/2010/mfv/mfv.zip>
- Forehand, M. (2010). Bloom’s taxonomy. En M. Orey (Ed.), *Emerging perspectives on learning, teaching, and technology* (pp. 41–47). Global Text. Descargado de https://textbookequity.org/Textbooks/Orey_Emergin_Perspectives_Learning.pdf
- Freiman, V., Broley, L., Buteau, C., y Vasilyeva, N. (2018). Report from working group focused on research-based understandings of the interplay between the affordances of computational thinking and mathematics. En *Online Proceedings of the Computational Thinking in Mathematics Education Symposium*. Oshawa, Canada. Descargado de http://ctmath.ca/wp-content/uploads/2018/10/Symposium_Working-Report_-Frieman-Et-al.pdf
- Futschek, G. (2006). Algorithmic Thinking: The Key for Understanding Computer Science. En R. T. Mittermeir (Ed.), *Informatics Education – The Bridge between Using and Understanding Computers. ISSEP 2006. Lecture Notes in Computer Science* (Vol. 4226, pp. 159–168). Berlin, Heidelberg: Springer-Verlag. Descargado de https://publik.tuwien.ac.at/files/PubDat_140308.pdf doi: 10.1007/11915355_15
- Futschek, G. (2007). Logo-like learning of basic concepts of algorithms, having fun with algorithms. En I. Kalas (Ed.), *Proceedings of Eurologo*. Bratislava: Comenius University. Descargado de http://publik.tuwien.ac.at/files/pub-inf_4696.pdf
- Futschek, G., y Moschitz, J. (2010). Developing algorithmic thinking by inventing and playing algorithms. En *Proceedings of the 2010 Constructionist Approaches to Creative Learning, Thinking and Education: Lessons for the 21st Century (Constructionism 2010)* (pp. 1–10). Descargado de http://publik.tuwien.ac.at/files/PubDat_187461.pdf
- Gallagher, J. P. (1979). Cognitive/information processing psychology and instruction: Reviewing recent theory and practice. *Instructional Science*, 8(4), 393–414. doi: 10.1007/BF00117014
- Gallardo Romero, J. (2004). *Diagnóstico y evaluación de la comprensión del conocimiento matemático. El caso del algoritmo estándar escrito para la multiplicación de números naturales* (Tesis Doctoral, Universidad de Málaga, Málaga, España). Descargado de <http://funes.uniandes.edu.co/625/2/Gallardo2004Diagnostico.pdf>
- Gallardo Romero, J., y González, J. L. (2007). Una aproximación operativa al diagnóstico y la evaluación de la comprensión del conocimiento matemático. *PNA. Revista de Investigación en Didáctica de la Matemática*, 1(1). Descargado de <http://revistaseug.ugr.es/index.php/pna/article/view/6217>
- Gayo Avello, D., Cernuda del Río, A., Cueva Lovelle, J. M., Díaz Fondón, M. Á., García Fuente, M. d. P. A., y Redondo López, J. M. (2003). Reflexiones y experiencias sobre la enseñanza de POO como único paradigma. En *IX Jornadas sobre la Enseñanza Universitaria de la Informática (JENUI’2003)*. Cádiz (España): Thompson. Descargado de <http://hdl.handle.net/10651/30804>

- Geda, G., y Biró, C. (2020). A Possible Way to Develop Algorithmic Thinking. *Acta Didactica Napocensia*, 13(1), 19–28. Descargado de <https://files.eric.ed.gov/fulltext/EJ1266754.pdf> doi: 10.24193/adn.13.1.3
- Gee, J. P. (2008). A sociocultural perspective on opportunity to learn. En P. A. Moss, D. C. Pullin, J. P. Gee, E. H. Haertel, y L. J. Young (Eds.), *Assessment, equity, and opportunity to learn* (pp. 76–108). Cambridge, USA: Cambridge University Press.
- Google, y Gallup. (2015). *Images of computer science: Perceptions among students, parents and educators in the US* (Inf. Téc.). Mountain View, California. Descargado de <http://hdl.voced.edu.au/10707/394238>
- Granberg, C. (2016). Discovering and addressing errors during mathematics problem-solving – A productive struggle? *The Journal of Mathematical Behavior*, 42, 33–48. doi: 10.1016/j.jmathb.2016.02.002
- Gravemeijer, K., Stephan, M., Julie, C., Lin, F.-L., y Ohtani, M. (2017). What Mathematics Education May Prepare Students for the Society of the Future? *International Journal of Science and Mathematics Education*, 15, 105–123. doi: 10.1007/s10763-017-9814-6
- Grover, S. (2017). Assessing algorithmic and computational thinking in K-12: Lessons from a middle school classroom. En P. J. Rich y C. B. Hodges (Eds.), *Emerging research, practice, and policy on computational thinking* (pp. 269–288). Springer. Descargado de https://www.researchgate.net/profile/Shuchi-Grover-2/publication/316442298_Assessing_Algorithmic_and_Computational_Thinking_in_K-12_Lessons_from_a_Middle_School_Classroom/links/5a9ad824aca2721e3f301247/Assessing-Algorithmic-and-Computational-Thinking-in-K-12-Lessons-from-a-Middle-School-Classroom.pdf doi: 10.1007/978-3-319-52691-1_17
- Grover, S., y Pea, R. (2013). Computational Thinking in K–12: A Review of the State of the Field. *Educational Researcher*, 42(1), 38–43. doi: 10.3102/0013189X12463051
- Grover, S., y Pea, R. (2018). Computational thinking: A competency whose time has come. En S. Sentance, E. Barendsen, y C. Schulte (Eds.), *Computer science education: Perspectives on teaching and learning in school* (cap. 3). London, UK: Bloomsbury Publishing. Descargado de https://www.researchgate.net/profile/Shuchi-Grover-2/publication/322104135_Computational_Thinking_A_Compentency_Whose_Time_Has_Come/links/5a457813a6fdcce1971a5ce5/Computational-Thinking-A-Compentency-Whose-Time-Has-Come.pdf
- Grozdev, S., y Terzieva, T. (2011). Research of the concept of algorithmic thinking in teaching computer science. En *The International Scientific-Practical Conference “Informatization of Education–2011”* (pp. 112–119).
- Grozdev, S., y Terzieva, T. (2015). A didactic model for developmental training in computer science. *Journal of Modern Education Review*, 5(5), 470–480. doi: 10.15341/jmer(2155-7993)/05.05.2015/005
- Guzdial, M., y Soloway, E. (2002). Teaching the Nintendo generation to program. *Communications of the ACM*, 45(4), 17–21. doi: 10.1145/505248.505261
- Heirdsfield, A. M. (2003). “Spontaneous” mental computation Strategies. En N. A. Pateman, B. J. Dougherty, y J. T. Zilliox (Eds.), *Proceedings of the 2003 Joint Meeting of PME (27) and PMENA (25)* (pp. 55–62). Hawai’i: CRDG, College of Education, University

- of Hawai'i. Descargado de https://eprints.qut.edu.au/1059/1/heirdsfield_60576E73.pdf
- Hernández, H. (1989). *El perfeccionamiento de la enseñanza de la Matemática en la Educación Superior Cubana. Experiencia en Álgebra Lineal* (Tesis Doctoral no publicada). La Habana.
- Hernández, H. (1993). Sistema Básico de Habilidades Matemáticas. En *Didáctica de la Matemática. Artículos para el Debate*. Quito, Ecuador: EPN.
- Hernández Sampieri, R., Fernández Collado, C., y Baptista Lucio, P. (2014). *Metodología de la Investigación* (Sexta Edición ed.). McGrawhill - Interamericana Editores.
- Hershkowitz, R., y Kieran, C. (2001). Algorithmic and meaningful ways of joining together representatives within the same mathematical activity: an experience with graphing calculators. En M. van den Heuvel-Panhuizen (Ed.), *Proceedings of the 25th Conference of the International Group for the Psychology of Mathematics Education (PME 25)* (Vol. 1, pp. 96–107). Utrecht, The Netherlands: International Group for the Psychology of Mathematics Education. Descargado de <http://files.eric.ed.gov/fulltext/ED466950.pdf#page=148>
- Hidayat, T., Susilaningsih, E., y Kurniawan, C. (2018). The effectiveness of enrichment test instruments design to measure students' creative thinking skills and problem solving. *Thinking Skills and Creativity*, 29, 161–169. doi: 10.1016/j.tsc.2018.02.011
- Hoffmann, R. (2001). Computer simulations in mathematics education. En M. van den Heuvel-Panhuizen (Ed.), *Proceedings of the 25th Conference of the International Group for the Psychology of Mathematics Education (PME 25)* (Vol. 1, p. 316). Utrecht, The Netherlands: International Group for the Psychology of Mathematics Education. Descargado de <http://files.eric.ed.gov/fulltext/ED466950.pdf#page=356>
- Horne, M. (2003). Gender differences in the early years in Addition and subtraction. En N. A. Pateman, B. J. Dougherty, y J. T. Zilliox (Eds.), *Proceedings of the 2003 Joint Meeting of PME (27) and PMENA (25)* (Vol. 3, pp. 79–86). Hawai'i: CRDG, College of Education, University of Hawai'i. Descargado de <https://files.eric.ed.gov/fulltext/ED500988.pdf>
- Houtz, J. C., Montgomery, C., Kirkpatrick, L., y Feldhusen, J. E. (1979). Relationship among measures of evaluation ability (problem solving), creative thinking, and intelligence. *Contemporary Educational Psychology*, 4(1), 47–54. doi: 10.1016/0361-476X(79)90026-2
- Hoyles, C., y Noss, R. (2015). Revisiting programming to enhance mathematics learning. En *Math+Coding Symposium*. Canada: Western University.
- Hsu, C.-C., y Wang, T.-I. (2018). Applying game mechanics and student-generated questions to an online puzzle-based game learning system to promote algorithmic thinking skills. *Computers & Education*, 121, 73–88. doi: 10.1016/j.compedu.2018.02.002
- Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, 9(3), 90–95. doi: 10.1109/MCSE.2007.55
- Hurlburt, G. F. (2018). Thinking Critically about Algorithmic Thinking. *IT Professional*, 20(2), 5–10. doi: 10.1109/MITP.2018.021921644
- International Society Technology Education, y Computer Science Teachers Association. (2011). *Operational Definition of Computational Thinking for K–12 Educa-*

- tion. Descargado de <http://www.iste.org/docs/ct-documents/computational-thinking-operational-definition-flyer.pdf>
- Israel, M., y Lash, T. (2019). From classroom lessons to exploratory learning progressions: mathematics + computational thinking. *Interactive Learning Environments*, 28(3), 362–382. doi: 10.1080/10494820.2019.1674879
- Joma, S. (2020, 29 de dic). *Estudiantes salvadoreñas crean prototipo de robots para buscar víctimas en terremotos*. Descargado de <https://www.elsalvador.com/noticias/nacional/robotica-ciencia-estudiantes-crean-prototipo-para-buscar-victimas-terremotos/790954/2020/>
- Jönsson, A., y Svingby, G. (2007). The use of scoring rubrics: Reliability, validity and educational consequences. *Educational Research Review*, 2(2), 130–144. Descargado de <http://www.sciencedirect.com/science/article/pii/S1747938X07000188> doi: 10.1016/j.edurev.2007.05.002
- Jun, S., Han, S., Kim, H., y Lee, W. (2014). Assessing the computational literacy of elementary students on a national level in Korea. *Educational Assessment, Evaluation and Accountability*, 26(4), 319–332. doi: 10.1007/s11092-013-9185-7
- Juškevičienė, A. (2020). Developing Algorithmic Thinking Through Computational Making. En D. Gintautas, B. Jolita, y K. Janusz (Eds.), *Data Science: New Issues, Challenges and Applications* (Vol. 869, pp. 183–197). Springer. doi: 10.1007/978-3-030-39250-5_10
- Kaiser, G. (2001). A description from Germany. En M. van den Heuvel-Panhuizen (Ed.), *Proceedings of the 25th Conference of the International Group for the Psychology of Mathematics Education (PME 25)* (Vol. 1, pp. 164–169). Utrecht, The Netherlands: International Group for the Psychology of Mathematics Education. Descargado de <http://files.eric.ed.gov/fulltext/ED466950.pdf#page=212>
- Kalelioğlu, F., y Gülbahar, Y. (2014). The Effect of Instructional Techniques on Critical Thinking and Critical Thinking Dispositions in Online Discussion. *Educational Technology & Society*, 17(1), 248–258. Descargado de <https://www.jstor.org/stable/pdf/jeductechsoci.17.1.248.pdf>
- Kalelioğlu, F., Gülbahar, Y., y Kukul, V. (2016). A framework for computational thinking based on a systematic research review. *Baltic Journal Modern Computing*, 4(3), 583–596. Descargado de <https://www.bjmc.lu.lv/contents/vol-42016-no-3/>
- Kane, M. T. (2013). Validating the Interpretations and Uses of Test Scores. *Journal of Educational Measurement*, 50(1), 1–73. Descargado de <https://onlinelibrary.wiley.com/doi/abs/10.1111/jedm.12000>
- Kátai, Z. (2015). The challenge of promoting algorithmic thinking of both sciences-and humanities-oriented learners. *Journal of Computer Assisted Learning*, 31(4), 287–299. doi: 10.1111/jcal.12070
- Khamis, H. (2008). Measures of association: how to choose? *Journal of Diagnostic Medical Sonography*, 24(3), 155–162. doi: 10.1177/8756479308317006
- Kieran, C., y Guzmán, J. (2003). The spontaneous emergence of elementary Number-theoretic concepts and techniques In interaction with computing technology. En N. A. Pateman, B. J. Dougherty, y J. T. Zilliox (Eds.), *Proceedings of the 2003 Joint Meeting of PME (27) and PMENA (25)* (pp. 141–148). Hawai'i: CRDG, College of Education, University of Hawai'i. Descargado de <https://files.eric.ed.gov/>

- fulltext/ED500994.pdf
- Kilhamn, C., y Bråting, K. (2019). Algebraic thinking in the shadow of programming. En U. T. Jankvist, M. van den Heuvel-Panhuizen, y M. Veldhuis (Eds.), *Eleventh Congress of the European Society for Research in Mathematics Education (CERME 11)* (Vol. TWG03). Utrecht, Netherlands: Freudenthal Group. Descargado de <https://hal.archives-ouvertes.fr/hal-02429028>
- Kinnunen, P., y Malmi, L. (2006). Why Students Drop out CS1 Course? En *Proceedings of the Second International Workshop on Computing Education Research* (pp. 97–108). New York, NY, USA: ACM. doi: 10.1145/1151588.1151604
- Klein, R., y Hoffmann, R. (2004). Do computers promote solving problems ability in elementary school? En *The 28th International Conference of the International Group for the Psychology of Mathematics Education*. Bergen, Norway. Descargado de <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.503.1328>
- Knuth, D. E. (1974). Computer Science and its Relation to Mathematics. *The American Mathematical Monthly*, 81(4), 323–343. doi: 10.1080/00029890.1974.11993556
- Knuth, D. E. (1985). Algorithmic Thinking and Mathematical Thinking. *The American Mathematical Monthly*, 92(3), 170–181. doi: 10.1080/00029890.1985.11971572
- Korkmaz, Ö., Çakir, R., y Özden, M. Y. (2017). A validity and reliability study of the computational thinking scales (CTS). *Computers in Human Behavior*, 72, 558–569. doi: 10.1016/j.chb.2017.01.005
- Kurland, D. M., Clement, C. A., Mawby, R., y Pea, R. D. (2002). Mapping the Cognitive Demands of Learning to Program. En *Thinking: The Second International Conference*. Routledge.
- Lee, I. A., y Malyn-Smith, J. (2019). Computational Thinking Integration Patterns Along the Framework Defining Computational Thinking from a Disciplinary Perspective. *Journal of Science Education and Technology*, 29(1), 9–18. doi: 10.1007/s10956-019-09802-x
- Leonard, J., Buss, A., Unertl, A., y Mitchell, M. (2016). Using robotics and game design to promote pathways to STEM. En M. B. Wood, E. E. Turner, M. Civil, y J. A. Eli (Eds.), *Proceedings of the 38th annual meeting of the North American Chapter of the International Group for the Psychology of Mathematics Education (PMENA 38)* (pp. 1487–1494). Tucson, AZ: The University of Arizona. Descargado de <https://files.eric.ed.gov/fulltext/ED583748.pdf>
- Li, H., Liu, J., Zhang, D., y Liu, H. (2020). Examining the relationships between cognitive activation, self-efficacy, socioeconomic status, and achievement in mathematics: A multi-level analysis. *British Journal of Educational Psychology*, 91(1), 101–126. doi: 10.1111/bjep.12351
- Lipka, J., Andrew-Ihrke, D., Koester, D., Zinger, V., Olson, M., Yanez, E., y Rubinstein, D. (2015). Indigenous knowledge provides an elegant way to teach the foundations of mathematics. En T. G. Bartell, K. N. Bieda, R. T. Putnam, K. Bradfield, y H. Dominguez (Eds.), *Proceedings of the 37th annual meeting of the North American Chapter of the International Group for the Psychology of Mathematics Education (PMENA 37)* (pp. 2–18). East Lansing, MI: Michigan State University. Descargado de <https://files.eric.ed.gov/fulltext/ED584219.pdf>
- Lockwood, E., DeJarnette, A. F., Asay, A., y Thomas, M. (2016). Algorithmic Thinking: An

- Initial Characterization of Computational Thinking in Mathematics. En M. B. Wood, E. E. Turner, M. Civil, y J. A. Eli (Eds.), *Proceedings of the 38th annual meeting of the North American Chapter of the International Group for the Psychology of Mathematics Education (PMENA 38)* (pp. 1588–1595). Tucson, AZ: ERIC. Descargado de <https://files.eric.ed.gov/fulltext/ED583797.pdf>
- López García, J. C. (2014). *Actividades de aula con Scratch que favorecen el uso del pensamiento algorítmico. El caso del grado 3 en el INSA* (Tesis de Master, UNIVERSIDAD ICESI, Cali, Colombia). Descargado de <http://eduteka.icesi.edu.co/pdfdir/tesis-juan-carlos-lopez.pdf>
- Lovász, L. (1988). Algorithmic mathematics: an old aspect with a new emphasis. En *Proc. 6th International Congress on Math. Education, Budapest* (pp. 67–78). Descargado de <https://web.cs.elte.hu/~%20lovasz/icme.pdf>
- Lovász, L. (2008). Trends in Mathematics, and How they could Change Education? En *Conférence Européenne «Le Futur de l'Education Mathématique en Europe»*. Lisbonne. Descargado de <http://www.cs.elte.hu/~%20lovasz/lisbon.pdf>
- Maass, K., Geiger, V., Romero Ariza, M., y Goos, M. (2019). The Role of Mathematics in interdisciplinary STEM education. *ZDM Mathematics Education*, 51(6), 869–884. doi: 10.1007/s11858-019-01100-5
- Madeo, L. A., y Bird, D. A. (1990). The Effect of User-Specified Language Elements on Learning to Program. *Journal of Research on Computing in Education*, 22(3), 336–347. doi: 10.1080/08886504.1990.10781925
- Marais, C., y Bradshaw, K. (2015). Problem-solving ability of first year cs students: A case study and intervention. En *Proceedings of the 44th Conference of the Southern African Computers Lecturers' Association*.
- Martínez Rodríguez, M. Á. (1999). El enfoque sociocultural en el estudio del desarrollo y la educación. *Revista electrónica de investigación educativa*, 1(1), 16–37. Descargado de <https://dialnet.unirioja.es/descarga/articulo/244264.pdf>
- Martínez-López, L. G., y Gualdrón-Pinto, E. (2018). Fortalecimiento del pensamiento variacional a través de una intervención mediada con TIC en estudiantes de grado noveno. *Revista de Investigación, Desarrollo e Innovación*, 9(1), 91–102. doi: 10.19053/20278306.v9.n1.2018.8156
- Mathematical Reviews, y zbMATH. (2020). *MSC2020-Mathematics Subject Classification System* (Inf. Téc.). Descargado de <https://mathscinet.ams.org/msnhtml/msc2020.pdf>
- McCracken, M., Almstrum, V., Diaz, D., Guzdia, M., Hagan, D., Kolikant, Y. B.-D., ... Wilusz, T. (2001). A multi-national, multi-institutional study of assessment of programming skills of first-year CS students. En *Working Group Reports from ITiCSE on Innovation and Technology in Computer Science Education* (pp. 125–180). New York, NY, USA: ACM. doi: 10.1145/572133.572137
- Mertler, C. A. (2000). Designing scoring rubrics for your classroom. *Practical assessment, research, and evaluation*, 7(1). doi: 10.7275/gcy8-0w24
- Micelli, M. L., y Crespo, C. R. (2012). Ábacos de América prehispánica. *Revista Latinoamericana de Etnomatemática*, 5(1), 159–190. Descargado de <https://www.revista.etnomatemática.org/index.php/RevLatEm/article/view/41>
- Mingus, T. T. Y., y Grassl, R. M. (1998). Algorithmic and recursive thinking: Current beliefs

- and their implications for the future. En L. J. Morrow y M. J. Kenney (Eds.), *The teaching and learning of algorithms in school mathematics* (NCTM 1998 Yearbook ed., pp. 32–43). Reston, VA: National Council of Teacher of Mathematics.
- Ministerio de Educación, Ciencia y Tecnología de El Salvador. (2008). *Programas de Estudio - Matemática - Segundo ciclo de Educación Básica* (Inf. Téc.). Descargado de <https://www.mined.gob.sv/descargas/category/870-programas-de-estudio-para-4-5-grado-y-6-grado.html>
- Ministerio de Educación, Ciencia y Tecnología de El Salvador. (2018). *Programas de Estudio - Matemática - Tercer ciclo de Educación Básica* (Inf. Téc.). Descargado de <https://www.mined.gob.sv/descargas/send/869-programas-de-estudio-para-tercer-ciclo-de-educacion-basica/5529-matematica-tercer-0-ciclo.html>
- Misfeldt, M., y Ejsing-Duun, S. (2015). Learning mathematics through programming: An instrumental approach to potentials and pitfalls. En K. Krainer y N. Vondrová (Eds.), *Ninth Congress of the European Society for Research in Mathematics Education (CERME 9)* (pp. 2524–2530). Prague, Czech Republic. Descargado de <https://hal.archives-ouvertes.fr/hal-01289367>
- Modeste, S. (2012a). *Enseigner l'algorithme pour quoi? Quelles nouvelles questions pour les mathématiques? Quels apports pour l'apprentissage de la preuve?* (Tesis Doctoral, Université de Grenoble, Grenoble, Francia). Descargado de <https://tel.archives-ouvertes.fr/tel-00783294>
- Modeste, S. (2012b). La pensée algorithmique: apports d'un point de vue extérieur aux mathématiques. En J. L. Dorier y S. Coutat (Eds.), *Enseignement Des Mathématiques et Contrat Social : Enjeux et Défis pour le 21e Siècle – Actes du Colloque Espace Mathématique Francophone 2012* (pp. 467–479). Descargado de <https://www.emf.unige.ch/files/6514/5320/2307/EMF2012GT3MODESTE.pdf>
- Modeste, S. (2016). Impact of Informatics on Mathematics and Its Teaching. En F. Gadducci y M. Tivosanis (Eds.), *History and Philosophy of Computing* (pp. 243–255). Cham: Springer International Publishing. doi: 10.1007/978-3-319-47286-7_17
- Modeste, S., Gravier, S., y Ouvrier-Bufferet, C. (2010). Algorithmique et apprentissage de la preuve. *Repères IREM*, 79, 51–72. Descargado de http://yves.chevallard.free.fr/spip/spip/IMG/pdf/DDM_-_UE1_-_YC_-_TD2.pdf
- Monje, J., Pérez-Tyteca, P., y Castro, E. (2012). Resolución de problemas y ansiedad matemática: profundizando en su relación. *Unión. Revista Iberoamericana de Educación Matemática*, 32, 45–62.
- Murillo, F. J., y Hidalgo, N. (2015). Dime Cómo Evalúas y Te Diré Qué Sociedad Construyes. *Revista Iberoamericana de Evaluación Educativa*, 8(1), 5–9. Descargado de <http://www.rinace.net/riee/numeros/vol8-num1/editorial.pdf>
- National Council of Teachers of Mathematics. (1989). *Curriculum and evaluation standards for school mathematics*. Reston, VA: NCTM.
- National Research Council. (1989). *Everybody counts: A report to the nation on the future of mathematics education*. Washington, DC: National Academies Press.
- Navas-López, E. A. (2017). Una Aproximación Artística a los Conjuntos de Julia por medio de Programación de Computadoras. En R. Acuña Chacón y R. Solís Ortega (Eds.), *X Congreso Internacional sobre la Enseñanza de la Matemática Asistida por Compu-*

- tadora (X CIEMAC)* (pp. 61–88). San José, Costa Rica: Escuela de Matemática - TEC. Descargado de <http://funes.uniandes.edu.co/17392/>
- Ng, B. K., y Hu, C. (2006). Use web-based simulation to learn trigonometric curves. *International Journal for Mathematics Teaching and Learning*. Descargado de <http://www.cimt.org.uk/journal/chunhu.pdf>
- Ordóñez-Ortega, O., Gualdrón-Pinto, E., y Amaya-Franky, G. (2019). Pensamiento variacional mediado con baldosas algebraicas y manipuladores virtuales. *Revista de Investigación, Desarrollo e Innovación*, 9(2), 347–362. doi: 10.19053/20278306.v9.n2.2019.9180
- Orłowski, A., y Chmielewski, L. J. (2018). Ulam Spiral and Prime-Rich Polynomials. En *International Conference on Computer Vision and Graphics* (pp. 522–533). doi: 10.1007/978-3-030-00692-1_45
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. New York: Basic Books. Descargado de <https://dl.acm.org/doi/pdf/10.5555/1095592>
- Parra, C. A., y Delgado, M. S. (2017). *Propuesta didáctica mediada por recursos digitales educativos que promueva el uso del pensamiento algorítmico* (Tesis de Master, Universidad de Nariño, Nariño, Colombia). Descargado de <http://sired.udenar.edu.co/4239/1/Informe%20Final.pdf>
- Plerou, A., y Vlamos, P. (2016). Algorithmic thinking and mathematical learning difficulties classification. *American Journal of Applied Psychology*, 5(5), 22.
- Porto Solano, R., Silvera Sarmiento, A., Garcés Giraldo, L. E., Porto Solano, A., Suarez López, D., y Arboleda López, A. (2017). Marco de enseñanza basado en OVAs para el adiestramiento en la introducción de los algoritmos (OLIA). *Etic@net: Revista científica electrónica de Educación y Comunicación en la Sociedad del Conocimiento*, 17(1), 60–75. Descargado de <https://revistaseug.ugr.es/index.php/eticanet/article/view/11915>
- Pyzara, A. (2012). Algorithmisation in teaching mathematics. *Didactica Mathematicae*, 34, 51–68. Descargado de http://pldml.icm.edu.pl/pldml/element/bwmeta1.element.ojs-doi-10_14708_dm_v34i0_417/c/417-467.pdf doi: 10.14708/dm.v34i0.417
- Pyzara, A. (2014). Creating an algorithm of a real-life situation as a form of mathematical modelling. *Didactics of Mathematics*, 11, 25–42. doi: 10.15611/dm.2014.11.03
- R Core Team. (2020). R: A Language and Environment for Statistical Computing [Manual de software informático]. Vienna, Austria. Descargado de <https://www.R-project.org/>
- Ramadhan, H. A. (2000). Programming by discovery. *Journal of Computer Assisted Learning*, 16(1), 83–93. doi: 10.1046/j.1365-2729.2000.00118.x
- Rasmussen, C., Zandieh, M., King, K., y Teppo, A. (2005). Advancing Mathematical Activity: A Practice-Oriented View of Advanced Mathematical Thinking. *Mathematical Thinking and Learning*, 7(1), 51–73. doi: 10.1207/s15327833mtl0701_4
- Redón, Á. B., y Serrano Aroca, Á. (2018). Aprendizaje de las matemáticas a través del lenguaje de programación R en Educación Secundaria. *Educación matemática*, 30(1), 133–162. doi: 10.24844/em3001.05
- Riesco, M., Fondón, M. D., Álvarez, D., López, B., Cernuda, A., y Juan, A. (2014). Informática: materia esencial en la educación obligatoria del siglo XXI. *ReVisión*, 7(3).

Descargado de <http://hdl.handle.net/10651/30092>

- Rojas Ayala, L. G. (2018). *Propuesta del uso de programas matemáticos para mejorar las capacidades de operaciones básicas en estudiantes del primer ciclo de ingenierías en la Universidad Politécnica Amazónica de Bagua – Amazonas 2016* (Tesis Doctoral, Universidad César Vallejo, Chiclayo, Perú). Descargado de <https://hdl.handle.net/20.500.12692/25282>
- Romberg, T. (2001). Mathematics goals and achievement in the United States. En M. van den Heuvel-Panhuizen (Ed.), *Proceedings of the 25th Conference of the International Group for the Psychology of Mathematics Education (PME 25)* (Vol. 1, pp. 180–185). Utrecht, The Netherlands: International Group for the Psychology of Mathematics Education. Descargado de <http://files.eric.ed.gov/fulltext/ED466950.pdf#page=228>
- Rowland, T., Huckstep, P., y Thwaites, A. (2004). Reflecting on prospective elementary teachers' mathematics content knowledge: the case of laura. En *The 28th International Conference of the International Group for the Psychology of Mathematics Education* (Vol. 4, pp. 121–128). Bergen, Norway. Descargado de <https://files.eric.ed.gov/fulltext/ED489631.pdf>
- Ruiz Ledesma, E. F., y Valdemoros Álvarez, M. E. (2004). Connections between qualitative and quantitative thinking about proportion: the case of paulina. En *The 28th International Conference of the International Group for the Psychology of Mathematics Education*. Bergen, Norway. Descargado de <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.487.6481&rep=rep1&type=pdf>
- Sadykova, O. V., y Il'bahtin, G. G. (2019). The Definition of Algorithmic Thinking. En *Proceedings of the International Session on Factors of Regional Extensive Development (FRED 2019)* (pp. 419–422). Atlantis Press. doi: 10.2991/fred-19.2020.85
- Sadykova, O. V., y Usolzev, A. (2018). On the concept of algorithmic thinking. En *SHS Web Conferences – International Conference on Advanced Studies in Social Sciences and Humanities in the Post-Soviet Era (ICPSE 2018)* (Vol. 55). doi: 10.1051/shsconf/20185503016
- Salgado Castillo, A., y Alonso Berenguer, I. (2017). Computational algorithmization: Limitations in problem solving skills in computational sciences majors at University of Oriente. *International Journal of Education and Development using ICT*, 13(2), 166–188. Descargado de <https://www.learntechlib.org/d/180639/>
- Schoenfeld, A. H. (1985). *Mathematical problem solving*. New York: Academic Press.
- Schoenfeld, A. H. (2013). Reflections on problem solving theory and practice. *The Mathematics Enthusiast*, 10(1), 9–34. Descargado de <https://scholarworks.umt.edu/tme/vol10/iss1/3/>
- Schoenfeld, A. H. (2015). Summative and Formative Assessments in Mathematics Supporting the Goals of the Common Core Standards. *Theory Into Practice*, 54(3), 183–194. doi: 10.1080/00405841.2015.1044346
- Schoenfeld, A. H. (2016). Learning to think mathematically: Problem solving, metacognition, and sense making in mathematics (Reprint). *Journal of Education*, 196(2), 1–38. doi: 10.1177/002205741619600202
- Selby, C. C., y Woollard, J. (2013). Computational Thinking: The Developing Definition. En *18th annual conference on innovation and technology in computer science education*.

- University of Southampton (E-prints). Descargado de <https://eprints.soton.ac.uk/356481/>
- Serhan, D. (2006). The effect of graphing calculators use on students' understanding of the derivative at a point. *International Journal for Mathematics Teaching and Learning*. Descargado de <http://www.cimt.org.uk/journal/serhan.pdf>
- Sfard, A. (1991). On the dual nature of mathematical conceptions: Reflections on processes and objects as different sides of the same coin. *Educational studies in mathematics*, 22(1), 1–36. doi: 10.1007/BF00302715
- Shalyto, A. A. (2001). Logic control and “reactive” systems: algorithmization and programming. *Automation and remote control*, 62(1), 1–29. Descargado de http://is.ifmo.ru/articles_en/shalyto_articles/1999.pdf doi: 10.1023/A:1002837232103
- Shefer, O. P., Nosova, L. S., y Lebedeva, T. N. (2018). A Modern Methodology for Teaching Programming at a University. *Scientific and Technical Information Processing*, 45(2), 81–86. doi: 10.3103/S0147688218020077
- Shute, V. J., Sun, C., y Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, 22, 142–158. Descargado de <http://www.sciencedirect.com/science/article/pii/S1747938X17300350> doi: 10.1016/j.edurev.2017.09.003
- Sinclair, N., y Patterson, M. (2018). The Dynamic Geometrisation of Computer Programming. *Mathematical Thinking and Learning*, 20(1), 54–74. doi: 10.1080/10986065.2018.1403541
- Stavely, A. M. (2014). *Programming and Mathematical Thinking: A Gentle Introduction to Discrete Math Featuring Python*. Socorro, New Mexico, USA: The New Mexico Tech Press.
- Sutherland, R. (2001). Reaction to “Algorithmic and meaningful ways of joining together representatives within the same mathematical activity”. En M. van den Heuvel-Panhuizen (Ed.), *Proceedings of the 25th Conference of the International Group for the Psychology of Mathematics Education (PME 25)* (Vol. 1, pp. 108–110). Utrecht, The Netherlands: International Group for the Psychology of Mathematics Education. Descargado de <http://files.eric.ed.gov/fulltext/ED466950.pdf#page=160>
- Tall, D. (2002). *Advanced Mathematical Thinking* (D. Tall, Ed.). New York: Kluwer Academic Publishers.
- Tall, D. (2009). Information Technology and Mathematics Education: Enthusiasms, Possibilities and Realities. *Colección Digital Eudoxus*, 1(6). Descargado de <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.481.7415&rep=rep1&type=pdf>
- Tang, X., Yin, Y., Lin, Q., Hadad, R., y Zhai, X. (2020). Assessing computational thinking: A systematic review of empirical studies. *Computers & Education*, 148, 1–22. doi: 10.1016/j.compedu.2019.103798
- Téllez Ramírez, M. (2019). Pensamiento Computacional: Una competencia del siglo XXI. *Revista Científica de Publicación del Centro Psicopedagógico y de Investigación en Educación Superior*, 6(1), 24–32. Descargado de <http://ojs.cepies.umsa.bo/index.php/RCV/article/view/28>
- Thom, J. S., y McGarvey, L. M. (2015). The act and artifact of drawing(s): observing geo-

- metric thinking with, in, and through children's drawings. *ZDM*, 47(3), 465–481. doi: 10.1007/s11858-015-0697-0
- Treviño, E., Fraser, P., Meyer, A., Morawietz, L., Inostroza, P., y Naranjo, E. (2015). *Tercer Estudio Regional Comparativo y Explicativo (TERCE 2013)*. Descargado de <https://unesdoc.unesco.org/ark:/48223/pf0000243532>
- Tucker, A., Deek, F. P., Jones, J., McCowan, D., Stephenson, C., y Verno, A. (2003). *A model curriculum for K-12 Computer Science: Final Report of the ACM K-12 Task Force Curriculum Committee* (Inf. Téc.). New York, NY, USA: Association for Computing Machinery. Descargado de <https://dl.acm.org/doi/book/10.1145/2593247> doi: 10.1145/2593247
- Twenge, J. M., Catanese, K. R., y Baumeister, R. F. (2002). Social exclusion causes self-defeating behavior. *Journal of personality and social psychology*, 83(3), 606–615. doi: 10.1037/0022-3514.83.3.606
- Ulusoy, F., y Çakıroğlu, E. (2018). Using video cases and small-scale research projects to explore prospective mathematics teachers' noticing of student thinking. *EURASIA Journal of Mathematics, Science and Technology Education*, 14(11), em1571. doi: 10.29333/ejmste/92020
- Valverde, G., y Näslund-Hadley, E. (2010). La condición de la educación en matemáticas y ciencias naturales en América Latina y el Caribe. *Banco Interamericano de Desarrollo*. Descargado de <https://publications.iadb.org/es/publicacion/14587/la-condicion-de-la-educacion-en-matematicas-y-ciencias-naturales-en-america>
- Vargas, A., Blanco, R., Pérez, O., y Rodríguez, E. (2013). Desarrollo de la Habilidad Algoritmizar en el Álgebra Lineal. En R. Flores (Ed.), *Acta Latinoamericana de Matemática Educativa 26* (pp. 1625–1631). México: Comité Latinoamericano de Matemática Educativa. Descargado de <http://funes.uniandes.edu.co/4453/1/VargasDesarrolloALME2013.pdf>
- Vásquez Hernández, A. P., y Moreno Gatica, D. A. (2016). Nudos y patrones: Una forma de expresión matemática de Guatemala y Costa Rica. En E. Mariscal y cols. (Eds.), *Acta Latinoamericana de Matemática Educativa 29* (pp. 1177–1182). México: Comité Latinoamericano de Matemática Educativa. Descargado de <http://funes.uniandes.edu.co/11820/1/Vasquez2016Nudos.pdf>
- Vera Olivera, H. (2019). Influencia de la Capacitación en Algorítmica en el Nivel de Pensamiento Computacional de los estudiantes de la Universidad Global del Cusco, 2018. *Universidad Global del Cusco*. Descargado de <http://repositorio.uglobal.edu.pe/handle/uglobal/27>
- Voogt, J., Fisser, P., Good, J., Mishra, P., y Yadav, A. (2015). Computational thinking in compulsory education: Towards an agenda for research and practice. *Education and Information Technologies*, 20(4), 715–728. doi: 10.1007/s10639-015-9412-6
- Vorhölter, K., Greefrath, G., Ferri, R. B., Leiß, D., y Schukajlow, S. (2019). Mathematical modelling. En *Traditions in German-speaking mathematics education research* (pp. 91–114). Springer, Cham. doi: 10.1007/978-3-030-11069-7_4
- Voskoglou, M. G., y Buckley, S. (2012). Problem solving and computational thinking in a learning environment. *Egyptian Computer Science Journal*, 36(4). Descargado de <https://arxiv.org/abs/1212.0750>

- Waterman, K., Goldsmith, L., Pasquale, M., Goldenberg, E. P., Malyn-Smith, J., DeMallie, A., y Lee, I. A. (2018). Integrating computational thinking into elementary mathematics and science curriculum materials and instruction. En *Conference Proceedings: the Future of Education*.
- Weber, K. (2003). A procedural route toward understanding The concept of proof. En N. A. Pateman, B. J. Dougherty, y J. T. Zilliox (Eds.), *Proceedings of the 2003 Joint Meeting of PME (27) and PMENA (25)* (pp. 395–401). Hawai'i. Descargado de <https://files.eric.ed.gov/fulltext/ED501161.pdf>
- Weber, K. (2004). A framework for describing the processes that undergraduates use to construct proofs. En *The 28th International Conference of the International Group for the Psychology of Mathematics Education* (pp. 425–432). Bergen, Norway. Descargado de <https://files.eric.ed.gov/fulltext/ED489676.pdf>
- Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., y Wilensky, U. (2016, 01). Defining Computational Thinking for Mathematics and Science Classrooms. *Journal of Science Education and Technology*, 25(1), 127–147. doi: 10.1007/s10956-015-9581-5
- Wickham, H., y Stryjewski, L. (2011). *40 years of boxplots*. Descargado de <http://vita.had.co.nz/papers/boxplots.pdf>
- Wilf, H. S. (2008). Mathematics: an experimental science. En T. Gowers, J. Barrow-Green, y I. Leader (Eds.), *The Princeton companion to mathematics*. Princeton University Press. Descargado de https://www.math.upenn.edu/~wilf/website/Mathematics_AnExperimentalScience.pdf
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33–35. doi: 10.1145/1118178.1118215
- Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, 366(1881), 3717–3725. doi: 10.1098/rsta.2008.0118
- Wing, J. M. (2010). Computational Thinking: What and Why? *Link Magazine*. Descargado de <http://www.cs.cmu.edu/%7ECompThink/papers/TheLinkWing.pdf>
- Wing, J. M. (2017). Computational thinking's influence on research and education for all. *Italian Journal of Educational Technology*, 25(2), 7–14. doi: 10.17471/2499-4324/922
- Wong, G. K., y Jiang, S. (2018). Computational Thinking Education for Children: Algorithmic Thinking and Debugging. En *2018 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE)* (pp. 328–334). doi: 10.1109/TALE.2018.8615232
- Yurkovich, N., Seben, V., y Mar'yan, M. (2017). Fractal approach to teaching physics and computer modeling. *Journal of Science Education*, 18(2), 117–120. Descargado de <http://chinaxjy.com/downloads/V18-2017-2/V18-2017-2-17.pdf>
- Ziatdinov, R., y Musa, S. (2012). Rapid Mental Computation System as a Tool for Algorithmic Thinking of Elementary School Students Development. *European Researcher*, 25(7), 1105–1110. Descargado de <http://www.erjournal.ru/pdf.html?n=1342467174.pdf>

A. Instrumentos de recolección de datos de la parte cuantitativa



UNIVERSIDAD DE EL SALVADOR
FACULTAD DE CIENCIAS NATURALES Y MATEMÁTICA
ESCUELA DE MATEMÁTICA

Modelamiento Matemático en la Resolución de Problemas septiembre-2018

Objetivo: Identificar las estrategias usadas para la resolución de ciertos tipos de problemas matemáticos.

DATOS PERSONALES

Año de ingreso a la Universidad: _____

Número de materias aprobadas: _____

Edad: _____

Número de materias de programación aprobadas: _____

CUM: _____

Género: Masculino Femenino

Indicaciones generales:

- Por favor intente resolver en las páginas adicionales todos los problemas que se le presentan a continuación. Cada problema en página separada. Plasmee por escrito todas las ideas que le surjan durante su proceso de resolución. Utilice dibujos, diagramas, tablas y todos los recursos que desee.
- No borre nada de lo que escriba, pero si cree que no es el procedimiento correcto, simplemente táchelo y no lo borre. Si necesita más papel, solicítelo.
- Si cree que falta información precisa para resolverlos, siéntase en la libertad de suponer valores para los datos que considere faltantes, y deje por escrito que lo hará y por qué.
- Puede resolverlos en el orden que usted prefiera.

PROBLEMA 1

En un país llamado Alicialandia existen nueve ciudades importantes: Almeida, Brunswick, Charlestown, Dendera, Edinburgo, Fukushima, Galium, Hormisdas e Idílicum. Se le pide que encuentre la ruta más corta para ir en carretera desde Almeida hasta Idílicum.

Se sabe que la distancia desde Almeida a Charlestown es de 2km, de Brunswick a Charlestown es de 3km, de Charlestown a Dendera es de 6km, de Brunswick a Edinburgo son 5km, de Charlestown a Edinburgo son 4km, de Dendera a Edinburgo son 2km, de Dendera a Fukushima son 5km, de Edinburgo a Fukushima son 4km, de Edinburgo a Galium son 8km, de Galium a Idílicum son 5km, de Fukushima a Hormisdas es 1km, de Fukushima a Idílicum son 6km, y finalmente de Hormisdas a Idílicum son 3km.

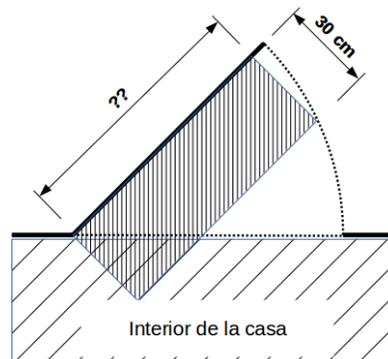
Las demás distancias no se han mencionado porque no hay carretera entre esas parejas de ciudades. Por ejemplo entre Brunswick y Galium no hay carretera.

PROBLEMA 2

En una escuela que sirve desde séptimo grado hasta noveno grado se va cada dos meses a un viaje que incluye la visita a un museo y a una función de teatro. La directora de la escuela ha establecido como norma que a estos viajes debe ir un adulto responsable por cada 15 alumnos, y que cada viaje sea organizado por un profesor diferente (o sea que se van rotando), por lo que cada profesor pasa varios meses (incluso años) sin organizar un viaje y cuando les toca de nuevo ya no recuerdan qué pasos seguir. Desarrolle unas instrucciones simples que le permitan a cualquier organizador que las lea calcular cuánto costará el viaje (esto para poder establecer cuánto deberá aportar cada estudiante).

PROBLEMA 3

Una familia se acaba de mudar a una casa y desean convertirla en tienda. Por lo que están en proceso de acondicionar la casa (la parte de enfrente) para tal fin. Entre otras cosas, necesitan colocar una superficie plana en la puerta-balcón (que mide 80cm de ancho) para colocar y mostrar los productos a los clientes, también debe servir para que los clientes puedan colocar el dinero que pagan. Sin embargo hay un problema: la puerta se abre hacia afuera y la superficie a colocar no se puede poner por el lado de afuera ya que obstaculizaría el paso peatonal por enfrente de la casa (además de tener mayor seguridad frente a clientes desconocidos). Ver el dibujo adjunto.

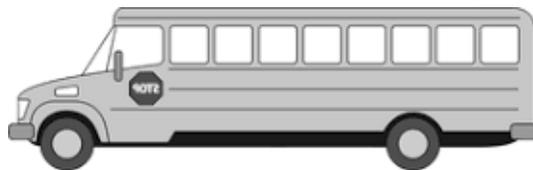


La familia ha decidido que 30cm son espacio suficiente para colocar los productos que la tienda venderá (botellas de aceite, sobres de café instantáneo, huevos, libras de azúcar, etc.).

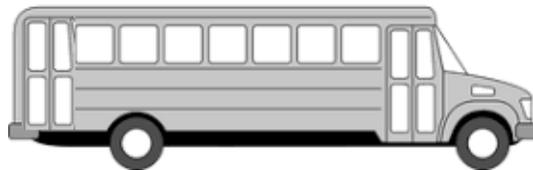
Se le pide que encuentre una expresión que calcule el ancho máximo que puede tener la superficie para realizar el intercambio con los clientes y que la puerta se pueda cerrar sin problema.

PROBLEMA 4

Como usted sabe, a la mayoría de los buses que ingresan a El Salvador para trabajar en el transporte colectivo les quitan los asientos originales y les ponen otros con menos espacio entre sí para aumentar su capacidad y disminuir su confort.



La Empresa «Apretados S.A. de C.V.» se dedica a ofrecer este servicio de modificación a las empresas de transporte colectivo cuando estas «traen un nuevo bus» (que ya sabemos que no sólo son usados sino que también están descartados en otros países).



Escriba unas instrucciones genéricas para que el gerente operativo (el jefe de los obreros) realice el trabajo de calcular cuántos asientos y a qué distancia deben colocarse. Asuma que los asientos originales ya han sido retirados y los «nuevos» están en un almacén cercano ya armados y listos para poner. Puesto que la empresa se dedica a esto, tiene una provisión casi ilimitada de asientos «nuevos».

MUCHAS GRACIAS

Se le agradece profundamente por haber participado en este estudio. Si tiene alguna observación, por favor hágala:

.....
.....
.....
.....



UNIVERSIDAD DE EL SALVADOR
FACULTAD DE CIENCIAS NATURALES Y MATEMÁTICA
ESCUELA DE MATEMÁTICA

Modelamiento Matemático en la Resolución de Problemas

septiembre-2018

Objetivo: Identificar las estrategias usadas para la resolución de ciertos tipos de problemas matemáticos.

DATOS PERSONALES

Año de ingreso a la Universidad: _____

Número de materias aprobadas: _____

Edad: _____

Número de materias de programación aprobadas: _____

CUM: _____

Género: Masculino

Femenino

Indicaciones generales:

- Por favor intente resolver en las páginas adicionales todos los problemas que se le presentan a continuación. Cada problema en página separada. Plasmee por escrito todas las ideas que le surjan durante su proceso de resolución. Utilice dibujos, diagramas, tablas y todos los recursos que desee.
- No borre nada de lo que escriba, pero si cree que no es el procedimiento correcto, simplemente táchelo y no lo borre. Si necesita más papel, solicítelo.
- Si cree que falta información precisa para resolverlos, siéntase en la libertad de suponer valores para los datos que considere faltantes, y deje por escrito que lo hará y por qué.
- Puede resolverlos en el orden que usted prefiera.

PROBLEMA 1

En un país llamado Alicialandia existen nueve ciudades importantes: Almeida, Brunswick, Charlestown, Dendera, Edinburgo, Fukushima, Galium, Hormisdas e Idílicum. Se le pide que encuentre la ruta más corta para ir en carretera desde Almeida hasta Idílicum.

Se sabe que la distancia desde Almeida a Charlestown es de 2km, de Brunswick a Charlestown es de 3km, de Charlestown a Dendera es de 6km, de Brunswick a Edinburgo son 5km, de Charlestown a Edinburgo son 4km, de Dendera a Edinburgo son 2km, de Dendera a Fukushima son 5km, de Edinburgo a Fukushima son 4km, de Edinburgo a Galium son 8km, de Galium a Idílicum son 5km, de Fukushima a Hormisdas es 1km, de Fukushima a Idílicum son 6km, y finalmente de Hormisdas a Idílicum son 3km.

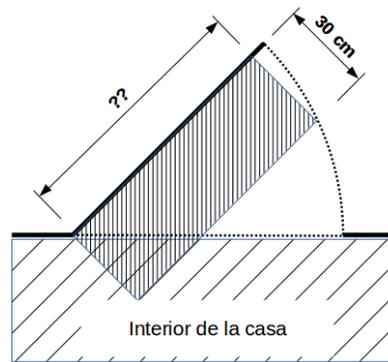
Las demás distancias no se han mencionado porque no hay carretera entre esas parejas de ciudades. Por ejemplo entre Brunswick y Galium no hay carretera.

PROBLEMA 2

En una escuela que sirve desde séptimo grado hasta noveno grado se va cada dos meses a un viaje que incluye la visita a un museo y a una función de teatro. La directora de la escuela ha establecido como norma que a estos viajes debe ir un adulto responsable por cada 15 alumnos, y que cada viaje sea organizado por un profesor diferente (o sea que se van rotando), por lo que cada profesor pasa varios meses (incluso años) sin organizar un viaje y cuando les toca de nuevo ya no recuerdan qué pasos seguir. Desarrolle un algoritmo simple que le permita a cualquier organizador que lo lea calcular cuánto costará el viaje (esto para poder establecer cuánto deberá aportar cada estudiante).

PROBLEMA 3

Una familia se acaba de mudar a una casa y desean convertirla en tienda. Por lo que están en proceso de acondicionar la casa (la parte de enfrente) para tal fin. Entre otras cosas, necesitan colocar una superficie plana en la puerta-balcón (que mide 80cm de ancho) para colocar y mostrar los productos a los clientes, también debe servir para que los clientes puedan colocar el dinero que pagan. Sin embargo hay un problema: la puerta se abre hacia afuera y la superficie a colocar no se puede poner por el lado de afuera ya que obstaculizaría el paso peatonal por enfrente de la casa (además de tener mayor seguridad frente a clientes desconocidos). Ver el dibujo adjunto.

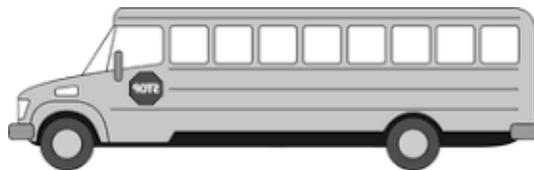


La familia ha decidido que 30cm son espacio suficiente para colocar los productos que la tienda venderá (botellas de aceite, sobres de café instantáneo, huevos, libras de azúcar, etc.).

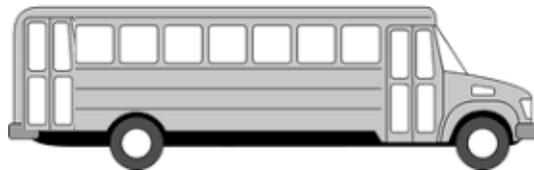
Se le pide que encuentre una expresión que calcule el ancho máximo que puede tener la superficie para realizar el intercambio con los clientes y que la puerta se pueda cerrar sin problema.

PROBLEMA 4

Como usted sabe, a la mayoría de los buses que ingresan a El Salvador para trabajar en el transporte colectivo les quitan los asientos originales y les ponen otros con menos espacio entre sí para aumentar su capacidad y disminuir su confort.



La Empresa «Apretados S.A. de C.V.» se dedica a ofrecer este servicio de modificación a las empresas de transporte colectivo cuando estas «traen un nuevo bus» (que ya sabemos que no sólo son usados sino que también están descartados en otros países).



Escriba un algoritmo para que el gerente operativo (el jefe de los obreros) realice el trabajo de calcular cuántos asientos y a qué distancia deben colocarse. Asuma que los asientos originales ya han sido retirados y los «nuevos» están en un almacén cercano ya armados y listos para poner. Puesto que la empresa se dedica a esto, tiene una provisión casi ilimitada de asientos «nuevos».

MUCHAS GRACIAS

Se le agradece profundamente por haber participado en este estudio. Si tiene alguna observación, por favor hágala:

.....
.....
.....
.....

B. Lista de problemas usados para la parte cualitativa

Esta es la lista de problemas asignados a los sujetos investigados en la parte cualitativa:

S01

Escriba un algoritmo que dada una lista de números devuelva la Mediana de los números.

S02

Escriba un algoritmo que dada una lista de números devuelva la posición en que se encuentra el elemento más cercano al promedio de todos los datos.

S03

Escriba un algoritmo que dada una lista de números (que se asume ya está ordenada) y un número aparte, busque el número aparte en la lista por el método de búsqueda binaria, y si se encuentra debe devolver el índice correspondiente. Si el número no está presente en la lista, la función debe devolver -1.

S04

Escriba un algoritmo que dada una cadena de texto devuelva True si la cadena es un palíndromo y False si no lo es. Un palíndromo es una palabra que es igual de derecha a izquierda y de izquierda a derecha. Omite considerar la ocurrencia de espacios en blanco y caracteres extraños.

S05

Escriba un algoritmo que dada una cadena de texto, devuelva una copia de la cadena en la que han desplazado los caracteres hacia la izquierda (de manera cíclica). Por ejemplo, si la cadena de entrada es "hola a todos" la salida será "ola a todosh".

S06

Escriba un algoritmo que dada una cadena de texto que sólo contiene unos y ceros que representa un número entero binario puro, devuelva el valor numérico que representa. Por ejemplo, si la entrada es “101” la salida debe ser el número 5, si la entrada es “1000” la salida debe ser el número 8.

Nota: No puede usar la conversión de la función `int`.

S07

Escriba un algoritmo que dada una lista de números y un número aparte, busque el número aparte en la lista de derecha a izquierda, y si se encuentra debe devolver el índice correspondiente. Si el número no está presente en la lista, la función debe devolver -1.

S08

Un número primo es aquel número natural que es divisible únicamente por sí mismo y la unidad. Escriba un algoritmo que dado un número entero mayor que 1 devuelva `True` si el número es primo y `False` si no lo es. Todos los números enteros menores o iguales que la unidad se consideran no primos.

S09

Escriba un programa que calcule la función doble factorial de un número natural. El doble factorial del número entero no negativo n es:

$$n!! = \begin{cases} 1 & n = 0 \vee n = 1 \\ (n-2)!! \cdot n & n > 1 \end{cases}$$

S10

La sucesión de Fibonacci comienza con 1 y 1, y cada término siguiente es la suma de los dos anteriores: $\{1, 1, 2, 3, 5, 8, 13, 21, \dots\}$. Escriba un algoritmo que dado un número entero mayor que 1 devuelva `True` si el número es parte de la sucesión de Fibonacci y `False` si no lo es. Todos los números enteros menores que la unidad no pertenecen a la sucesión.

S11

Dados dos números enteros positivos llamados Mayor y Menor, para obtener el máximo común divisor (MCD) de ambos por el Método de Euclides se procede así:

a) Se realiza la división entera del Mayor entre el Menor para calcular el Residuo.

- b) Si el Residuo es cero, el MCD es el valor de la variable llamada Menor.
- c) Pero si Residuo no es cero, el valor de Menor pasa a Mayor y el valor de Residuo pasa a Menor.
- d) Repetir el paso a).

Escriba un algoritmo que dados dos números enteros positivos, devuelva su MCD.

S12

Escriba un algoritmo que devuelva la aproximación de la raíz cuadrada de un número real positivo z por el método de Newton-Raphson, es decir, utilizando la fórmula recurrente:

$$r_{i+1} = \frac{\frac{z}{r_i} + r_i}{2}$$

El algoritmo recibirá como entradas el valor positivo z , la aproximación inicial r_0 (también positiva), y la precisión p de la solución, es decir la diferencia máxima entre la i -ésima aproximación y la $(i + 1)$ -ésima para aceptar a esta $(i + 1)$ -ésima como la mejor aproximación. Este último valor suele ser muy pequeño, como 10^{-9} o similar.

S13

Escriba un algoritmo que dado un número entero positivo menor que cien, imprima el número en letras.

S14

Escriba un algoritmo que dado un número entero positivo n imprima los factoriales desde 0 hasta n con el formato siguiente:

```
0!=1
1!=1
2!=1*2=2
3!=1*2*3=6
4!=1*2*3*4=24
...
```

Exp

Un capicúa o número palíndromo es aquel que se escribe igual de derecha a izquierda que de izquierda a derecha. Por ejemplo: 909, 1991 y 2002. Escriba un algoritmo que dado un número entero positivo, devuelva True si el número es capicúa y False si no lo es.

Nota: No puede convertir el número a cadena y procesarlo de esa forma.



UNIVERSIDAD DE EL SALVADOR
FACULTAD DE CIENCIAS NATURALES Y MATEMÁTICA
CARRERA: LICENCIATURA EN MATEMÁTICA
PLAN DE ESTUDIOS 2003

L10940

CICLO I	CICLO II	CICLO III	CICLO IV	CICLO V	CICLO VI	CICLO VII	CICLO VIII	CICLO IX	CICLO X	CICLO XI
1	5	9	13	17	21	25	29	33	37	
MBL1109	CLM1109	CLM2109	CLM3109	CLM4109	AVL1109	ILM1109	LAM1109	TML1109	SLM1109	
Matemática Básica	Cálculo I	Cálculo II	Cálculo III	Cálculo IV	Análisis Vectorial	Metodología de la Investigación	Análisis Matemático	Teoría de la Medida	Seminario de Matemática	
4	B4	14	54	134	174	214	134	294	34	
2	6	10	14	18	22	26	30	34	38	
LML1109	ALL1109	ALL2109	IIL1109	PLM1109	PLM2109	ANL1109	VAC1109	TOP1109		
Lógica Matemática	Algebra Lineal I	Algebra Lineal II	Introducción a la Informática	Programación I	Programación II	Análisis Numérico I	Cálculo de Variable Compleja	Topología	Electiva II	
4	B4	64	24	144	184	13,184	174	294	(*)	
3	7	11	15	19	23	27	31	35	39	
GEL1109	GEL2109	ELM1109	ELM2109	EDL1109	EDL2109	AML1109	AML2109	LMO1109		
Geometría I	Geometría II	Estadística I	Estadística II	Ecuaciones Diferenciales I	Ecuaciones Diferenciales II	Algebra Moderna I	Algebra Moderna II	Métodos de Optimización I	Electiva III	
4	B4	34	54	134	194	164	274	10,184	(*)	
4	8	12	16	20	24	28	32	36	40	
	MDL1109	TEN1109	TEN1109	FLM1109	FLM2109	FLM3109	FLM3109			
Humanística I	Introducción a la Matemática Discreta	Humanística II	Teoría del Número	Física I	Física II	Física III	Física III	Electiva I	Electiva IV	
4	B4	24	24	(*)4	94	244	284	(*)4	(*)	

NC	Número correlativo
C	Unidades Valorativas
NA	Nombre de asignatura
P	Prerrequisito
UV	Bachillerato

Nombre de la carrera: Licenciatura en Matemática-L10940
Total de materias: 40
Total de Unidades Valorativas del Plan: 160
 Lista de las materias electivas (optativas) por ciclo: consultar
 Acuerdo de Consejo Superior Universitario: # 128-99-2003(V-3.4),30-01-2003
 Duración: 5 años
 Fecha de aprobación: 30 de enero de 2003
Obligatorias: 33 Electivas: 4 Humanísticas: 3 Complementarias: 0

D. Scripts para la generación de los gráficos de la parte cuantitativa

Para realizar los gráficos de la sección 4.1 se implementaron los siguientes scripts en Lenguaje Python¹ v3.8.2, utilizando principalmente la biblioteca Matplotlib² diseñada por Hunter (2007), concretamente en su versión 3.1.2, de Caswell y cols. (2019).

Listado de código D.1: Columnas del archivo de datos

```
1 #!/usr/bin/python3
2 #coding: latin1
3 """
4 Esta es la lista de las columnas de datos del
5 archivo de datos cuantitativos 'datos-cuanti.csv'.
6 """
7
8 #Datos de clasificación
9 ID = 0 #Identificador único del instrumento
10 INGRESO = 1 #año de ingreso
11 EDAD = 2
12 CUM = 3 #Coeficiente de Unidades de Mérito
13 MATTOT = 4 #Cantidad de materias de la carrera aprobadas
14 MATPROG = 5 #Cantidad de materias de Programación aprobadas
15 GRUPO = 6 #según matProg
16 GEN = 7 #género
17 ALG = 8 #Se le pidió hacer algoritmos o instrucciones
18
19 #Subvariables de PR
20 P1_GRAF = 9
21 P1_CIU = 10
22 P1_CARR = 11
23 P1_SOL = 12
24 P1_RES = 13
25
26 P3_MOD = 14
27 P3_SOL = 15
28 P3_RES = 16
29
30 RP = 17 #Promedio ponderado
31
32 #Subvariables de PAP
33 P2_INPUT = 18
34 P2_OUTPUT = 19
```

¹<https://www.python.org/>

²<https://matplotlib.org/>

```

35 P2_IDSUB = 20
36 P2_SOLSUB = 21
37 P2_INTSUB = 22
38 P2_NORM = 23
39 P2_EXTRE = 24
40 P2_FLUJO = 25
41 P2_SIN = 26
42
43 P4_INPUT = 27
44 P4_DIBU = 28
45 P4_OUTPUT = 29
46 P4_IDSUB = 30
47 P4_SOLSUB = 31
48 P4_INTSUB = 32
49 P4_NORM = 33
50 P4_EXTRE = 34
51 P4_FLUJO = 35
52 P4_SIN = 36
53
54 PAP = 37 #Promedio ponderado
55
56 #Eliminadas del archivo CSV:
57 #SUG = 38 #sugerencias hechas por los alumnos
58 #OBSERVACIONES = 39 #observaciones del investigador

```

Listado de código D.2: Configuración de fuente del documento

```

1 #!/usr/bin/python3
2 #coding: latin1
3 """Configuración de Fuente.
4 Por defecto, matplotlib usa una fuente 'sans'.
5 """
6 import matplotlib.pyplot as plt
7 plt.rcParams.update({
8     "text.usetex": True,
9     "font.family": "serif",
10 })

```

Listado de código D.3: Código de regresión lineal para usar en varios gráficos

```

1 #!/usr/bin/python3
2 #coding: latin1
3 """Regresión Lineal"""
4 def regresionLineal(X,Y):
5     """Esta función devuelve los coeficientes (m,b) de una recta de
6     regresión
7     lineal,  $Y = mX+b$ 
8     """
9     n = len(X)
10    if n!=len(Y): raise ValueError("Los vectores no son del mismo tamaño")
11    xy=0
12    x=0
13    y=0

```

```

13     xx=0
14     for i in range(n):
15         x += X[i]
16         y += Y[i]
17         xy += X[i]*Y[i]
18         xx += X[i]*X[i]
19     m = (n*xy - x*y) / (n*xx - x*x)
20     b = (y - m*x)/n
21     return m,b

```

Listado de código D.4: Código de tabla de coeficientes de Correlación de Spearman y de Kendall

```

1  #!/usr/bin/python3
2  #coding: latin1
3  """Mapa de calor de las correlaciones.
4
5  Con código adaptado de:
6  https://matplotlib.org/stable/gallery/images\_contours\_and\_fields/
7  image_annotated_heatmap.html
8  """
9  import numpy as np
10 import matplotlib
11 import matplotlib.pyplot as plt
12 #import fuente #Este gráfico se ve mejor con fuentes 'sans'
13 def heatmap(data, row_labels, col_labels, ax=None,
14             cbar_kw={}, cbarlabel="", **kwargs):
15
16     if not ax:
17         ax = plt.gca()
18
19     # Plot the heatmap
20     im = ax.imshow(data,
21                   norm=matplotlib.colors.Normalize(vmin=0.0, vmax=1.0),
22                   **kwargs
23                )
24
25     # Create colorbar
26     #cbar = ax.figure.colorbar(im, ax=ax, **cbar_kw)
27     #cbar.ax.set_ylabel(cbarlabel, rotation=-90, va="bottom")
28
29     # We want to show all ticks...
30     ax.set_xticks(np.arange(data.shape[1]))
31     ax.set_yticks(np.arange(data.shape[0]))
32     # ... and label them with the respective list entries.
33     ax.set_xticklabels(col_labels)
34     ax.set_yticklabels(row_labels)
35
36     # Let the horizontal axes labeling appear on top.
37     ax.tick_params(top=True, bottom=False,
38                   labeltop=True, labelbottom=False)
39
40     # Rotate the tick labels and set their alignment.

```

```

51 plt.setp(ax.get_xticklabels(), rotation=-30, ha="right",
52         rotation_mode="anchor")
53
54 ax.set_xticks(np.arange(data.shape[1]+1)-.5, minor=True)
55 ax.set_yticks(np.arange(data.shape[0]+1)-.5, minor=True)
56 ax.grid(which="minor", color="w", linestyle='-', linewidth=3)
57 ax.tick_params(which="minor", bottom=False, left=False)
58
59 return im
60
61 def annotate_heatmap(im, dataTexto=None, dataColor=None, valfmt="{x:.3
62 f}",
63                    textcolors=("white", "black"),
64                    **textkw):
65     if not isinstance(dataTexto, (list, np.ndarray)):
66         dataTexto = im.get_array()
67     if not isinstance(dataColor, (list, np.ndarray)):
68         dataColor = im.get_array()
69
70     # Set default alignment to center, but allow it to be
71     # overwritten by textkw.
72     kw = dict(horizontalalignment="center",
73               verticalalignment="center")
74     kw.update(textkw)
75
76     # Get the formatter in case a string is supplied
77     if isinstance(valfmt, str):
78         valfmt = matplotlib.ticker.StrMethodFormatter(valfmt)
79
80     # Loop over the data and create a 'Text' for each "pixel".
81     # Change the text's color depending on the data.
82     texts = []
83     for i in range(dataTexto.shape[0]):
84         for j in range(dataTexto.shape[1]):
85             #print(im.norm(dataColor[i, j]))
86             kw.update(color=textcolors[int(0.2<im.norm(dataColor[i, j
87 ])<0.8)])
88             if i==j:
89                 text = im.axes.text(j, i, "1", **kw)
90             else:
91                 text = im.axes.text(j, i, valfmt(dataTexto[i, j], None
92 ), **kw)
93             texts.append(text)
94
95     return texts
96
97 def conversion(originales, pMedio=0.5, maxValor=1.0):
98     """Esta función cambia los valores de la matriz original de
99     correlaciones según los colores del mapa de colores 'RdYlBu'
100     de Matplotlib para distinguir los coeficientes de
101     correlación de Spearman (en tonos de rojo) de los de Kendall
102     (en tonos de azul).
103     """

```

```

92     conv = []
93     i = 0
94     while i<len(originales):
95         f = []
96         j = 0
97         while j<len(originales[i]):
98             if i==j: #diagonal principal, blanco
99                 f.append(pMedio)
100            elif i>j: #triángulo inferior, Spearman, rojo
101                f.append(pMedio*(abs(originales[i][j])-maxValor)/-
102                    maxValor)
103            else: #i<j: #triángulo superior, Kendall, azul
104                f.append((1.0-pMedio)*abs(originales[i][j])/maxValor+
105                    pMedio)
106            j = j+1
107            i = i+1
108            conv.append(f)
109            # for f in conv:
110            #     for d in f:
111            #         # print("{:0.3f}\t".format(d), end=' ')
112            #     print()
113            return np.array(conv)
114
115 nombre="coefs-correlacionales"
116 variables = [
117     "edad", "cum", "matTot", "matProg", "RP", "PAP",
118 ]
119
120 correlaciones = np.array([ #Estos datos vienen del script en R
121     [1.00000,0.11984,0.39302,0.39778,0.03106,0.25484],
122     [0.16712,1.00000,0.18094,0.17605,0.18351,0.23904],
123     [0.50777,0.26763,1.00000,0.63075,0.13300,0.35874],
124     [0.47278,0.22635,0.75812,1.00000,0.11869,0.41898],
125     [0.04084,0.26284,0.20261,0.15451,1.00000,0.25979],
126     [0.33690,0.34991,0.50470,0.53255,0.38389,1.00000],
127 ])
128
129 mapaColores = conversion(correlaciones, 0.55, 0.76)
130
131 fig, ax = plt.subplots()
132
133 #Lista de Mapas de Colores:
134 #https://matplotlib.org/stable/tutorials/colors/colormaps.html
135 im = heatmap(mapaColores, variables, variables, ax=ax,
136             cmap="RdYlBu", cbarlabel="Correlaciones")
137 texts = annotate_heatmap(im, dataTexto=correlaciones, dataColor=
138     mapaColores)
139
140 fig.tight_layout()
141 #print('Guardando archivo', nombre); plt.savefig(nombre+'.pdf')
142 #print('Guardando archivo', nombre); plt.savefig(nombre+'.png');
143 plt.show()

```

Listado de código D.5: Histogramas de las variables PAP y RP

```

1  #!/usr/bin/python3
2  #coding: latin1
3  """
4  Este programa dado el archivo de datos cuantitativos genera
5  histogramas para las variables:
6  PAP y RP con diferentes anchos.
7  """
8  from columns import *
9  import matplotlib.pyplot as plt
10 import fuente
11
12 lineas = open('../datos-cuanti.csv').readlines()
13 pa = []
14 pr = []
15
16 for l in lineas[1:]: #Omitir encabezados
17     datos = l.split(',')
18     if datos[GRUPO] in ('P0', 'P1', 'P2', 'P3'):
19         pa.append(float(datos[PAP]))
20         pr.append(float(datos[RP]))
21     else:
22         raise ValueError(datos[GRUPO])
23
24 anchos_histograma = [5,10,15]
25 titulos = [
26     'Histogramas de la Variable  $\{\{PAP\}\}$  con anchos de {}
27     puntos'.format(anchos_histograma),
28     'Histogramas de la Variable  $\{\{RP\}\}$  con anchos de {}
29     puntos'.format(anchos_histograma),
30 ]
31 colores = ['green', 'lightgreen', 'lightblue',]
32
33 etiquetas = [
34     'Pensamiento Algorítmico ( $\{\{PAP\}\}$ )',
35     'Resolución de Problemas ( $\{\{RP\}\}$ )'
36 ]
37 nombres = ['PAP', 'RP']
38 minEjeX = 0
39 maxEjeX = 100
40
41 etiquetasX = [list(range(minEjeX, maxEjeX+a, a)) for a in
42     anchos_histograma]
43
44 # Graficar -----
45 for ix, ejeX in enumerate((pa,pr)):
46     fig, axs = plt.subplots(nrows=len(anchos_histograma), ncols=1,
47         figsize=(8, 9),
48     )
49
50     # Histogramas
51     for ix2, ancho in enumerate(anchos_histograma):
52         axs[ix2].hist(ejeX,
53             bins=etiquetasX[ix2],

```

```

51         color=colores[ix2]
52     )
53
54     # Configurando los gráficos
55     for ix2, ax in enumerate(axes):
56         ax.xaxis.grid(True)
57         ax.yaxis.grid(True)
58         ax.set_ylabel('Frecuencias')
59         ax.set_xlim(0,100)
60         ax.set_xticks(etiquetasX[ix2])
61         #ax.set_title(titulos[ix][ix2], size=14)
62     axes[len anchos_histograma)-1].set_xlabel(etiquetas[ix])
63     axes[0].set_title(titulos[ix],
64                       size=14
65     )
66
67     fig.tight_layout()
68     #print('Guardando archivo', nombres[ix]); plt.savefig(nombres[ix]
69     #print('Guardando archivo', nombres[ix]); plt.savefig(nombres[ix]
70     plt.show()

```

Listado de código D.6: Histogramas de matTot, ingreso, edad y cum

```

1  #!/usr/bin/python3
2  #coding: latin1
3  """
4  Este programa dado el archivo de datos cuantitativos genera
5  histogramas para las variables:
6  .- matTot,
7  .- (año de) ingreso,
8  .- edad y
9  .- CUM
10 con diferentes anchos.
11 """
12 from columns import *
13 import matplotlib.pyplot as plt
14 import fuente
15
16 lineas = open('../datos-cuanti.csv').readlines()
17 matTot = []
18 ingreso = []
19 edad = []
20 cum = []
21
22 for l in lineas[1:]: #Omitir encabezados
23     datos = l.split(',')
24     if datos[GRUPO] in ('P0', 'P1', 'P2', 'P3'):
25         matTot.append(int(datos[MATTOT]))
26         ingreso.append(int(datos[INGRESO]))
27         edad.append(int(datos[EDAD]))
28         cum.append(float(datos[CUM]))
29     else:

```

```

30         raise ValueError(datos[GRUPO])
31
32 anchos_histograma = [
33     [1,2,3],
34     [1,2,3],
35     [1,2,3],
36     [0.2, 0.25, 0.5]
37 ]
38 titulos = [
39     'Histogramas de la Variable  $\text{\$matTot}\$$  con anchos {}
40     materias'.format(anchos_histograma[0]),
41     'Histogramas de la Variable Año de  $\text{\$ingreso}\$$  con anchos
42     {} años'.format(anchos_histograma[1]),
43     'Histogramas de la Variable  $\text{\$edad}\$$  con anchos {} años'.
44     .format(anchos_histograma[2]),
45     'Histogramas de la Variable  $\text{\$cum}\$$  con anchos {} puntos'
46     .format(anchos_histograma[3]),
47 ]
48 colores = ['green', 'lightgreen', 'lightblue']
49
50 etiquetas = [
51     'Cantidad total de Materias aprobadas ( $\text{\$matTot}\$$ )',
52     'Año de ingreso a la universidad ( $\text{\$ingreso}\$$ )',
53     ' $\text{\$edad}\$$ ',
54     'Coeficiente de Unidades de Mérito ( $\text{\$cum}\$$ )',
55 ]
56 nombres = ['matTot', 'ingreso', 'edad', 'cum']
57 minEjeX = [min(matTot), min(ingreso), min(edad), 6.0]
58 maxEjeX = [max(matTot), max(ingreso), max(edad), 10.0]
59 #minEjeX = [1, 2002, 17, 6.0]
60 #maxEjeX = [35, 2018, 33, 10.0]
61 etiquetasX = [
62     [list(range(minEjeX[0], maxEjeX[0]+a, a)) for a in
63     anchos_histograma[0]],
64     [list(range(minEjeX[1], maxEjeX[1]+a, a)) for a in
65     anchos_histograma[1]],
66     [list(range(minEjeX[2], maxEjeX[2]+a, a)) for a in
67     anchos_histograma[2]],
68     [[6.0+k*a for k in range(int((10.0-6.0)/a)+1)] for a in
69     anchos_histograma[3]],
70 ]
71
72 # Graficar -----
73 for ix, ejeX in enumerate((matTot, ingreso, edad, cum)):
74     fig, axs = plt.subplots(nrows=len(anchos_histograma[ix]), ncols=1,
75     figsize=(8, 9),
76     )
77
78     # Histogramas
79     for ix2, ancho in enumerate(anchos_histograma[ix]):
80         axs[ix2].hist(ejeX,
81             bins=etiquetasX[ix][ix2],
82             color=colores[ix2])

```

```

76     )
77
78     # Configurando los gráficos
79     for ix2, ax in enumerate(axes):
80         ax.xaxis.grid(True)
81         ax.yaxis.grid(True)
82         ax.set_ylabel('Frecuencias')
83         #ax.set_xlim(0,100)
84         ax.set_xticks(etiquetasX[ix][ix2])
85         #ax.set_title(titulos[ix][ix2], size=14)
86     axes[len(anchos_histograma[ix])-1].set_xlabel(etiquetas[ix])
87     axes[0].set_title(titulos[ix],
88                       size=14)
89 )
90
91 fig.tight_layout()
92 #print('Guardando archivo', nombres[ix]); plt.savefig(nombres[ix]
93               +''.pdf');
94 #print('Guardando archivo', nombres[ix]); plt.savefig(nombres[ix]
95               +''.png')
96 plt.show()

```

Listado de código D.7: Gráficos de pastel de matProg, alg y gen

```

1  #!/usr/bin/python3
2  #coding: latin1
3  """
4  Este programa dado el archivo de datos cuantitativos genera
5  gráficos de pastel para las variables:
6  .- matProg,
7  .- alg y
8  .- gen
9  """
10 from columns import *
11 import matplotlib.pyplot as plt
12 import fuente
13
14 def func(pct,todas):
15     '''Para incluir el valor relativo y absoluto
16     en el diagrama de pastel.
17     '''
18     absoluto = int(round(pct*sum(todas)/100.))
19     cad = "{:1.1f}\%□({:d})".format(pct, absoluto)
20     return cad
21
22
23 lineas = open('../datos-cuanti.csv').readlines()
24 matProg = [0,0,0,0] #P0, P1, P2, P3
25 alg = [0,0] #No, Sí
26 gen = [0,0] #Masculino, Femenino
27
28 for l in lineas[1:]: #Omitir encabezados
29     datos = l.split(',')
30     if datos[GRUPO] in ('P0', 'P1', 'P2', 'P3'):

```

```

31     try:
32         v = int(datos[MATPROG])
33         matProg[v] = matProg[v] + 1
34
35         v = int(datos[ALG])
36         alg[v] = alg[v] + 1
37
38         if datos[GEN]=='m':
39             gen[0] = gen[0] + 1
40         elif datos[GEN]=='f':
41             gen[1] = gen[1] + 1
42         else:
43             raise ValueError("Género: {}".format(datos[GEN]))
44     except:
45         print(datos[:SUG])
46         exit()
47     else:
48         raise ValueError(datos[GRUPO])
49
50     titulos = [
51         'Proporciones de la Variable  $\text{matProg}$ ',
52         'Proporciones de la Variable  $\text{alg}$ ',
53         'Proporciones de la Variable  $\text{gen}$ ',
54     ]
55     etiquetas = [
56         ['0 materias', '1 materia', '2 materias', '3 materias'],
57         ['Instrucciones', 'Algoritmo'],
58         ['Masculino', 'Femenino']
59     ]
60     tamanios = [(4.5, 4.5), (3,3), (3,3) ]
61     colores = ['green', 'lightgreen', 'lightblue', 'blue']
62     nombres = ['matProg', 'alg', 'gen']
63
64
65     # Graficar -----
66     for ix, datos in enumerate((matProg,alg,gen)):
67         fig, axs = plt.subplots(
68             figsize=tamanios[ix],
69         )
70
71         wedges, texts, autotexts = axs.pie(
72             datos, labels=etiquetas[ix],
73             shadow=True,
74             # autopct = "%1.1f%%",
75             autopct=lambda pct: func(pct, datos),
76             wedgeprops=dict(width=0.7),
77             explode=[0.01 for t in etiquetas[ix]],
78             startangle=-30,
79             # rotatelabels=True,
80             # center=(-10,0),
81             colors = colores,
82         )
83         plt.axis('equal')
84

```

```

85     axs.legend(
86         wedges, etiquetas[ix],
87         #title="Tipos",
88         loc="upper_left",
89         #labelspacing=1.5,
90     ) # https://matplotlib.org/stable/api/\_as\_gen/matplotlib.axes.Axes
        .legend.html
91
92     #plt.setp(texts, weight="bold")
93     plt.title(titulos[ix], size=16)
94
95     fig.tight_layout()
96     print('Guardando archivo', nombres[ix]); plt.savefig(nombres[ix]+'
97         .pdf');
97     #print('Guardando archivo', nombres[ix]); plt.savefig(nombres[ix]
98         ]+'.png')
98     #plt.show()

```

Listado de código D.8: Correlación entre cum y edad

```

1  #!/usr/bin/python3
2  #coding: latin1
3  """
4  Este programa dado el archivo de datos cuantitativos genera
5  un gráfico de dispersión para las variables cum~edad
6  con histogramas incorporados.
7  """
8  from regresion import regresionLineal
9  from columnas import *
10 import matplotlib.pyplot as plt
11 import fuente
12 import numpy as np
13
14 lineas = open('../datos-cuanti.csv').readlines()
15
16 cum = []
17 edad = []
18
19 for l in lineas[1:]: #Omitir encabezados
20     datos = l.split(',')
21     cum.append(float(datos[CUM]))
22     edad.append(int(datos[EDAD]))
23
24 titulo = 'Variable_\$\\mathsf{cum}\$_según_la_variable_\$\\mathsf{edad}\$'
25 nombre = 'cum~edad'
26
27 # definiciones de los ejes
28 izq, ancho = 0.1, 0.65
29 abajo, alto = 0.1, 0.65
30 espacio = 0.015
31
32 rect_scatter = [izq, abajo, ancho, alto]
33 rect_histx = [izq, abajo + alto + espacio, ancho, 0.2]
34 rect_histy = [izq + ancho + espacio, abajo, 0.2, alto]

```

```

35
36 # crear la figura
37 fig = plt.figure(figsize=(8, 7))
38
39 ax = fig.add_axes(rect_scatter)
40 ax_histx = fig.add_axes(rect_histx, sharex=ax)
41 ax_histy = fig.add_axes(rect_histy, sharey=ax)
42
43 # quitar las etiquetas
44 ax_histx.tick_params(axis="x", labelbottom=False)
45 ax_histy.tick_params(axis="y", labelleft=False)
46
47 # gráfico de dispersión:
48 ax.scatter(edad, cum)
49
50 minEdad = min(edad)
51 maxEdad = max(edad)
52
53 # ancho del histograma
54 ancho_barras_edad = 1
55 ancho_barras_cum = 0.1
56
57 binsEdad = list(range(minEdad, maxEdad + ancho_barras_edad,
58     ancho_barras_edad))
59 binsCUM = np.arange(min(cum), max(cum) + ancho_barras_cum,
60     ancho_barras_cum)
61
62 nx,bx,px = ax_histx.hist(edad, bins=binsEdad)
63 ny,by,py = ax_histy.hist(cum, bins=binsCUM, orientation='horizontal')
64
65 #Recta de regresión lineal:
66 m,b = regresionLineal(edad,cum)
67 ax.plot([minEdad,maxEdad],[b+m*minEdad,b+m*maxEdad],'-', color='k')
68 plt.suptitle(titulo+' $y = \{0.2f\}x + \{0.2f\}$ '.format(m,b), size=16, y
69     =0.997)
70
71 # Configurando ambos gráficos
72 ax.xaxis.grid(True)
73 ax.yaxis.grid(True)
74 ax.set_xlabel('Edad')
75 ax.set_ylabel('Coeficiente de Unidades de Mérito ( $\mathsf{cum}$ )')
76 ax.set_xticks(range(minEdad, maxEdad+1))
77 #valoresCUM = np.arange(6.5, 10.25, 0.25)
78 valoresCUM = np.arange(6.6, 10.2, 0.2)
79 ax.set_yticks(valoresCUM)
80 ax_histx.xaxis.grid(True)
81 ax_histx.yaxis.grid(True)
82 ax_histy.xaxis.grid(True)
83 ax_histy.yaxis.grid(True)
84 ax_histx.set_yticks(range(0, int(max(nx))+1,3))
85 ax_histy.set_xticks(range(0, int(max(ny))+1,2))
86 ax_histy.set_yticks(valoresCUM)
87
88 fig.tight_layout()

```

```

86 #print('Guardando archivo', nombre); plt.savefig(nombre+'.pdf');
87 #print('Guardando archivo', nombre); plt.savefig(nombre+'.png')
88 plt.show()

```

Listado de código D.9: Correlación entre matProg y matTot

```

1  #!/usr/bin/python3
2  #coding: latin1
3  """
4  Este programa dado el archivo de datos cuantitativos genera
5  un gráfico de dispersión para las variables matProg~matTot.
6  """
7  from regresion import regresionLineal
8  from columnas import *
9  import matplotlib.pyplot as plt
10 import fuente
11
12 lineas = open('../datos-cuanti.csv').readlines()
13
14 matProg = []
15 matTot = []
16
17 for l in lineas[1:]: #Omitir encabezados
18     datos = l.split(',')
19     matProg.append(float(datos[MATPROG]))
20     matTot.append(int(datos[MATTOT]))
21
22 titulo = 'Variable_\$\\mathsf{matProg}\$_según_la_variable_\$\\mathsf{
23     matTot}\$'
24 nombre = 'matProg~matTot'
25
26 #Graficar -----
27 fig, axs = plt.subplots(nrows=1, ncols=1, figsize=(10, 4))
28
29 axs.scatter(matTot, matProg)
30 minMaterias = min(matTot)
31 maxMaterias = max(matTot)
32 #Recta de regresión lineal:
33 m,b = regresionLineal(matTot,matProg)
34 axs.plot([minMaterias,maxMaterias],[b+m*minMaterias,b+m*maxMaterias],
35     '-', color='k')
36 plt.title(titulo+'_\$/_\$y={:0.2f}x+{:0.2f}\$'.format(m,b), size=16)
37
38 # Configurando el gráfico
39 axs.xaxis.grid(True)
40 axs.yaxis.grid(True)
41 axs.set_xlabel('Total_de_Materias_cursadas_\$\\mathsf{matTot}\$')
42 axs.set_ylabel('Materias_de_Programación_aprobadas_\$\\mathsf{matProg}\$
43     )')
44 axs.set_xticks(range(minMaterias, maxMaterias+1))
45 axs.set_yticks(range(4))
46
47 #print('Guardando archivo', nombre); plt.savefig(nombre+'.pdf');
48 #print('Guardando archivo', nombre); plt.savefig(nombre+'.png')

```

```
46 plt.show()
```

Listado de código D.10: Relaciones entre las variables PAP, gen, alg y matProg

```
1  #!/usr/bin/python3
2  #coding: latin1
3  """
4  Este programa dado el archivo de datos cuantitativos genera
5  gráficos de caja y de violín para las
6  variables PAP~gen~matProg y para PAP~alg~matProg.
7  """
8  from columns import *
9  import matplotlib.pyplot as plt
10 import fuente
11
12 lineas = open('../datos-cuanti.csv').readlines()
13
14 pa_p0_m = []
15 pa_p0_f = []
16 pa_p1_m = []
17 pa_p1_f = []
18 pa_p2_m = []
19 pa_p2_f = []
20 pa_p3_m = []
21 pa_p3_f = []
22 pa_p0_alg = []
23 pa_p0_ins = []
24 pa_p1_alg = []
25 pa_p1_ins = []
26 pa_p2_alg = []
27 pa_p2_ins = []
28 pa_p3_alg = []
29 pa_p3_ins = []
30
31 for l in lineas[1:]: #Omitir encabezados
32     datos = l.split(',')
33     if datos[GRUPO] == 'P0':
34         if datos[GEN] == 'm':
35             pa_p0_m.append(float(datos[PAP]))
36         elif datos[GEN] == 'f':
37             pa_p0_f.append(float(datos[PAP]))
38         else:
39             raise ValueError(datos[ID]+'-'+datos[GRUPO]+'-'+datos[GEN
40                               ])
41         if datos[ALG] == '1':
42             pa_p0_alg.append(float(datos[PAP]))
43         elif datos[ALG] == '0':
44             pa_p0_ins.append(float(datos[PAP]))
45         else:
46             raise ValueError(datos[ID]+'-'+datos[GRUPO]+'-'+datos[ALG
47                               ])
48     elif datos[GRUPO] == 'P1':
49         if datos[GEN] == 'm':
50             pa_p1_m.append(float(datos[PAP]))
```

```

49     elif datos[GEN] == 'f':
50         pa_p1_f.append(float(datos[PAP]))
51     else:
52         raise ValueError(datos[ID]+'-'+datos[GRUPO]+'-'+datos[GEN
53         ])
54     if datos[ALG] == '1':
55         pa_p1_alg.append(float(datos[PAP]))
56     elif datos[ALG] == '0':
57         pa_p1_ins.append(float(datos[PAP]))
58     else:
59         raise ValueError(datos[ID]+'-'+datos[GRUPO]+'-'+datos[ALG
60         ])
61     elif datos[GRUPO] == 'P2':
62         if datos[GEN] == 'm':
63             pa_p2_m.append(float(datos[PAP]))
64         elif datos[GEN] == 'f':
65             pa_p2_f.append(float(datos[PAP]))
66         else:
67             raise ValueError(datos[ID]+'-'+datos[GRUPO]+'-'+datos[GEN
68             ])
69         if datos[ALG] == '1':
70             pa_p2_alg.append(float(datos[PAP]))
71         elif datos[ALG] == '0':
72             pa_p2_ins.append(float(datos[PAP]))
73         else:
74             raise ValueError(datos[ID]+'-'+datos[GRUPO]+'-'+datos[ALG
75             ])
76     elif datos[GRUPO] == 'P3':
77         if datos[GEN] == 'm':
78             pa_p3_m.append(float(datos[PAP]))
79         elif datos[GEN] == 'f':
80             pa_p3_f.append(float(datos[PAP]))
81         else:
82             raise ValueError(datos[ID]+'-'+datos[GRUPO]+'-'+datos[GEN
83             ])
84         if datos[ALG] == '1':
85             pa_p3_alg.append(float(datos[PAP]))
86         elif datos[ALG] == '0':
87             pa_p3_ins.append(float(datos[PAP]))
88         else:
89             raise ValueError(datos[ID]+'-'+datos[GRUPO]+'-'+datos[ALG
90             ])
91     else:
92         raise ValueError(datos[GRUPO])
93
94 datos_para_grafico_gen = [
95     pa_p0_m, pa_p0_f,
96     pa_p1_m, pa_p1_f,
97     pa_p2_m, pa_p2_f,
98     pa_p3_m, pa_p3_f,
99 ]
100 datos_para_grafico_alg = [
101     pa_p0_alg, pa_p0_ins,
102     pa_p1_alg, pa_p1_ins,

```

```

97     pa_p2_alg, pa_p2_ins,
98     pa_p3_alg, pa_p3_ins,
99 ]
100 etiquetasX = [
101     ['0/masc.', '0/feme.',
102     '1/masc.', '1/feme.',
103     '2/masc.', '2/feme.',
104     '3/masc.', '3/feme.'],
105     ['0/algo.', '0/inst',
106     '1/algo.', '1/inst',
107     '2/algo.', '2/inst',
108     '3/algo.', '3/inst']
109 ]
110 etiquetasX = [
111     ['0/masc.\n({})'.format(str(len(pa_p0_m))),
112     '0/feme.\n({})'.format(str(len(pa_p0_f))),
113     '1/masc.\n({})'.format(str(len(pa_p1_m))),
114     '1/feme.\n({})'.format(str(len(pa_p1_f))),
115     '2/masc.\n({})'.format(str(len(pa_p2_m))),
116     '2/feme.\n({})'.format(str(len(pa_p2_f))),
117     '3/masc.\n({})'.format(str(len(pa_p3_m))),
118     '3/feme.\n({})'.format(str(len(pa_p3_f)))]],
119     ['0/algo.\n({})'.format(str(len(pa_p0_alg))),
120     '0/inst\n({})'.format(str(len(pa_p0_ins))),
121     '1/algo.\n({})'.format(str(len(pa_p1_alg))),
122     '1/inst\n({})'.format(str(len(pa_p1_ins))),
123     '2/algo.\n({})'.format(str(len(pa_p2_alg))),
124     '2/inst\n({})'.format(str(len(pa_p2_ins))),
125     '3/algo.\n({})'.format(str(len(pa_p3_alg))),
126     '3/inst\n({})'.format(str(len(pa_p3_ins)))]
127 ]
128 titulos = [
129     '$\mathsf{PAP}$ segun $\mathsf{gen}$ y $\mathsf{matProg}$',
130     '$\mathsf{PAP}$ segun $\mathsf{alg}$ y $\mathsf{matProg}$'
131 ]
132 ejeX = [
133     'Cantidad de Materias de Programación aprobadas ($\mathsf{matProg}$
134     y Género ($\mathsf{gen}$)',
135     'Cantidad de Materias de Programación aprobadas ($\mathsf{matProg}$
136     y variable $\mathsf{alg}$'
137 ]
138 nombres = ['PAP~gen~matProg', 'PAP~alg~matProg']
139 colores = [
140     ['pink', 'lightblue']*4,
141     ['lightgreen', 'lightcoral']*4
142 ]
143 secuencia_graficos = (
144     datos_para_grafico_gen,
145     datos_para_grafico_alg,
146 )
147 # Graficar -----
148 for ix, datos_grafico in enumerate(secuencia_graficos):
149     fig, axs = plt.subplots(nrows=1, ncols=1, figsize=(6, 4))

```



```

15 P1 = 1
16 P2 = 2
17 P3 = 3
18
19 datos_grafico = {
20     P2_INPUT   : [[], [], [], []],
21     P2_OUTPUT  : [[], [], [], []],
22     P2_IDSUB   : [[], [], [], []],
23     P2_SOLSUB  : [[], [], [], []],
24     P2_INTSUB  : [[], [], [], []],
25     P2_NORM    : [[], [], [], []],
26     P2_EXTRE   : [[], [], [], []],
27     P2_FLUJO   : [[], [], [], []],
28     P2_SIN     : [[], [], [], []],
29
30     P4_INPUT   : [[], [], [], []],
31     P4_DIBU    : [[], [], [], []],
32     P4_OUTPUT  : [[], [], [], []],
33     P4_IDSUB   : [[], [], [], []],
34     P4_SOLSUB  : [[], [], [], []],
35     P4_INTSUB  : [[], [], [], []],
36     P4_NORM    : [[], [], [], []],
37     P4_EXTRE   : [[], [], [], []],
38     P4_FLUJO   : [[], [], [], []],
39     P4_SIN     : [[], [], [], []],
40 }
41
42 for l in lineas[1:]: #Omitir encabezados
43     datos = l.split(',')
44     if datos[GRUPO] == 'P0':
45         g = P0
46     elif datos[GRUPO] == 'P1':
47         g = P1
48     elif datos[GRUPO] == 'P2':
49         g = P2
50     elif datos[GRUPO] == 'P3':
51         g = P3
52     else:
53         raise ValueError(datos[GRUPO])
54
55     datos_grafico[P2_INPUT][g].append(int(datos[P2_INPUT]) if datos[
56         P2_INPUT] != '' else 0)
57     datos_grafico[P2_OUTPUT][g].append(int(datos[P2_OUTPUT]) if datos[
58         P2_OUTPUT] != '' else 0)
59     datos_grafico[P2_IDSUB][g].append(int(datos[P2_IDSUB]) if datos[
60         P2_IDSUB] != '' else 0)
61     datos_grafico[P2_SOLSUB][g].append(int(datos[P2_SOLSUB]) if datos[
62         P2_SOLSUB] != '' else 0)
63     datos_grafico[P2_INTSUB][g].append(int(datos[P2_INTSUB]) if datos[
64         P2_INTSUB] != '' else 0)
65     datos_grafico[P2_NORM][g].append(int(datos[P2_NORM]) if datos[
66         P2_NORM] != '' else 0)
67     datos_grafico[P2_EXTRE][g].append(int(datos[P2_EXTRE]) if datos[
68         P2_EXTRE] != '' else 0)

```

```

62 datos_grafico[P2_FLUJO][g].append(int(datos[P2_FLUJO]) if datos[
    P2_FLUJO]!='' else 0)
63 datos_grafico[P2_SIN][g].append(int(datos[P2_SIN]) if datos[P2_SIN
    ]!='' else 0)
64
65 datos_grafico[P4_INPUT][g].append(int(datos[P4_INPUT]) if datos[
    P4_INPUT]!='' else 0)
66 datos_grafico[P4_DIBU][g].append(int(datos[P4_DIBU]) if datos[
    P4_DIBU]!='' else 0)
67 datos_grafico[P4_OUTPUT][g].append(int(datos[P4_OUTPUT]) if datos[
    P4_OUTPUT]!='' else 0)
68 datos_grafico[P4_IDSUB][g].append(int(datos[P4_IDSUB]) if datos[
    P4_IDSUB]!='' else 0)
69 datos_grafico[P4_SOLSUB][g].append(int(datos[P4_SOLSUB]) if datos[
    P4_SOLSUB]!='' else 0)
70 datos_grafico[P4_INTSUB][g].append(int(datos[P4_INTSUB]) if datos[
    P4_INTSUB]!='' else 0)
71 datos_grafico[P4_NORM][g].append(int(datos[P4_NORM]) if datos[
    P4_NORM]!='' else 0)
72 datos_grafico[P4_EXTRE][g].append(int(datos[P4_EXTRE]) if datos[
    P4_EXTRE]!='' else 0)
73 datos_grafico[P4_FLUJO][g].append(int(datos[P4_FLUJO]) if datos[
    P4_FLUJO]!='' else 0)
74 datos_grafico[P4_SIN][g].append(int(datos[P4_SIN]) if datos[P4_SIN
    ]!='' else 0)
75
76 variables = [
77     '$\mathsf{p2\_input}$',
78     '$\mathsf{p2\_output}$',
79     '$\mathsf{p2\_idsub}$',
80     '$\mathsf{p2\_solsub}$',
81     '$\mathsf{p2\_intsub}$',
82     '$\mathsf{p2\_norm}$',
83     '$\mathsf{p2\_extre}$',
84     '$\mathsf{p2\_flujo}$',
85     '$\mathsf{p2\_sin}$',
86     '$\mathsf{p4\_input}$',
87     '$\mathsf{p4\_dibu}$',
88     '$\mathsf{p4\_output}$',
89     '$\mathsf{p4\_idsub}$',
90     '$\mathsf{p4\_solsub}$',
91     '$\mathsf{p4\_intsub}$',
92     '$\mathsf{p4\_norm}$',
93     '$\mathsf{p4\_extre}$',
94     '$\mathsf{p4\_flujo}$',
95     '$\mathsf{p4\_sin}$',
96 ]
97 nombre = 'PA2+PA4~matProg-'
98 etiquetaEjeY = 'Cantidad de Materias\nde Programación aprobadas ($\
    \mathsf{matProg}$)'
99 colores = ['pink', 'lightgreen', 'lightblue', 'lightcoral']
100 posiciones = [
101     (0,0), (1,0),
102     (0,1), (1,1),

```

```

103 ]
104
105 dimensiones = [
106     [P2_INPUT, None,      P4_INPUT, P4_DIBU],
107     [P2_OUTPUT, P2_IDSUB, P4_OUTPUT, P4_IDSUB],
108     [P2_SOLSUB, P2_INTSUB, P4_SOLSUB, P4_INTSUB],
109     [P2_NORM, P2_EXTRE,   P4_NORM, P4_EXTRE],
110     [P2_FLUJO, P2_SIN,    P4_FLUJO, P4_SIN],
111 ]
112 ix_s_nombres = [
113     [0,-1,9,10],
114     [1,2,11,12],
115     [3,4,13,14],
116     [5,6,15,16],
117     [7,8,17,18],
118 ]
119
120 # Graficar -----
121 for grupo_graf in range(5):
122     fig, axs = plt.subplots(nrows=2, ncols=2, figsize=(8, 6))
123     for dimension, ix, dim in zip(dimensiones[grupo_graf], posiciones,
124                                 ix_s_nombres[grupo_graf]):
125         if dimension:
126             # gráfico de violín
127             axs[ix].violinplot(datos_grafico[dimension],
128                               # showmeans=True,
129                               showmedians=True,
130                               # widths=0.8,
131                               )
132             # gráfico de caja
133             grfCaja = axs[ix].boxplot(datos_grafico[dimension],
134                                     patch_artist=True, #para ponerles color
135                                     #notch=True,
136                                     )
137             for patch, color in zip(grfCaja['boxes'], colores):
138                 patch.set_facecolor(color)
139                 patch.set_alpha(0.6)
140
141             #Medias
142             for ix2, d in enumerate(datos_grafico[dimension]):
143                 axs[ix].scatter([ix2+1], [sum(d)/len(d)], zorder=3,
144                                 color='k')
145
146             # Configurando gráfico
147             axs[ix].yaxis.grid(True)
148             axs[ix].set_ylabel(variables[dim])
149             axs[ix].set_yticks(range(0, 101, 10))
150             plt.setp(axs[ix],
151                     xticks = list(range(1,4+1)),
152                     xticklabels=range(4) #0,1,2,3
153                     )
154         else:
155             axs[1,0].remove()

```

```

155         axs[0,0].set_xlabel(etiquetaEjeY)
156     axs[posiciones[+1]].set_xlabel(etiquetaEjeY)
157     axs[posiciones[-1]].set_xlabel(etiquetaEjeY)
158
159     fig.tight_layout()
160     #print('Guardando archivo', nombre+str(grupo_graf)); plt.savefig(
161         nombre+str(grupo_graf)+'.pdf')
162     #print('Guardando archivo', nombre+str(grupo_graf)); plt.savefig(
163         nombre+str(grupo_graf)+'.png');
164     plt.show()

```

Listado de código D.12: Detalle de las sub-variables de PAP según matTot

```

1  #!/usr/bin/python3
2  #coding: latin1
3  """
4  Este programa dado el archivo de datos cuantitativos genera
5  gráficos de dispersión para las subvariables de PAP
6  respecto de la variable matTot en diferentes archivos.
7  """
8  from regresion import regresionLineal
9  from columnas import *
10 import matplotlib.pyplot as plt
11 import fuente
12
13 lineas = open('../datos-cuanti.csv').readlines()
14
15 matTot = []
16 datos_grafico = {
17     P2_INPUT : [],
18     P2_OUTPUT : [],
19     P2_IDSUB : [],
20     P2_SOLSUB : [],
21     P2_INTSUB : [],
22     P2_NORM : [],
23     P2_EXTRE : [],
24     P2_FLUJO : [],
25     P2_SIN : [],
26
27     P4_INPUT : [],
28     P4_DIBU : [],
29     P4_OUTPUT : [],
30     P4_IDSUB : [],
31     P4_SOLSUB : [],
32     P4_INTSUB : [],
33     P4_NORM : [],
34     P4_EXTRE : [],
35     P4_FLUJO : [],
36     P4_SIN : [],
37 }
38
39 for l in lineas[1:]: #Omitir encabezados
40     datos = l.split(',')
41     matTot.append(int(datos[MATTOT]))

```

```

42
43   datos_grafico[P2_INPUT].append(int(datos[P2_INPUT]) if datos[
44     P2_INPUT]!='' else 0)
45   datos_grafico[P2_OUTPUT].append(int(datos[P2_OUTPUT]) if datos[
46     P2_OUTPUT]!='' else 0)
47   datos_grafico[P2_IDSUB].append(int(datos[P2_IDSUB]) if datos[
48     P2_IDSUB]!='' else 0)
49   datos_grafico[P2_SOLSUB].append(int(datos[P2_SOLSUB]) if datos[
50     P2_SOLSUB]!='' else 0)
51   datos_grafico[P2_INTSUB].append(int(datos[P2_INTSUB]) if datos[
52     P2_INTSUB]!='' else 0)
53   datos_grafico[P2_NORM].append(int(datos[P2_NORM]) if datos[P2_NORM
54     ]!='' else 0)
55   datos_grafico[P2_EXTRE].append(int(datos[P2_EXTRE]) if datos[
56     P2_EXTRE]!='' else 0)
57   datos_grafico[P2_FLUJO].append(int(datos[P2_FLUJO]) if datos[
58     P2_FLUJO]!='' else 0)
59   datos_grafico[P2_SIN].append(int(datos[P2_SIN]) if datos[P2_SIN]!=
60     '' else 0)
61
62   datos_grafico[P4_INPUT].append(int(datos[P4_INPUT]) if datos[
63     P4_INPUT]!='' else 0)
64   datos_grafico[P4_DIBU].append(int(datos[P4_DIBU]) if datos[P4_DIBU
65     ]!='' else 0)
66   datos_grafico[P4_OUTPUT].append(int(datos[P4_OUTPUT]) if datos[
67     P4_OUTPUT]!='' else 0)
68   datos_grafico[P4_IDSUB].append(int(datos[P4_IDSUB]) if datos[
69     P4_IDSUB]!='' else 0)
70   datos_grafico[P4_SOLSUB].append(int(datos[P4_SOLSUB]) if datos[
71     P4_SOLSUB]!='' else 0)
72   datos_grafico[P4_INTSUB].append(int(datos[P4_INTSUB]) if datos[
73     P4_INTSUB]!='' else 0)
74   datos_grafico[P4_NORM].append(int(datos[P4_NORM]) if datos[P4_NORM
75     ]!='' else 0)
76   datos_grafico[P4_EXTRE].append(int(datos[P4_EXTRE]) if datos[
77     P4_EXTRE]!='' else 0)
78   datos_grafico[P4_FLUJO].append(int(datos[P4_FLUJO]) if datos[
79     P4_FLUJO]!='' else 0)
80   datos_grafico[P4_SIN].append(int(datos[P4_SIN]) if datos[P4_SIN]!=
81     '' else 0)
82
83 minMat = 0 #min(matTot)
84 maxMat = (max(matTot)//5)*5+5
85 variables = [
86   '$\mathsf{p2\_input}$',
87   '$\mathsf{p2\_output}$',
88   '$\mathsf{p2\_idsub}$',
89   '$\mathsf{p2\_solsub}$',
90   '$\mathsf{p2\_intsub}$',
91   '$\mathsf{p2\_norm}$',
92   '$\mathsf{p2\_extre}$',
93   '$\mathsf{p2\_flujo}$',
94   '$\mathsf{p2\_sin}$',
95   '$\mathsf{p4\_input}$',

```

```

77     '$\mathsf{p4\_dibu}$',
78     '$\mathsf{p4\_output}$',
79     '$\mathsf{p4\_idsub}$',
80     '$\mathsf{p4\_solsub}$',
81     '$\mathsf{p4\_intsub}$',
82     '$\mathsf{p4\_norm}$',
83     '$\mathsf{p4\_extre}$',
84     '$\mathsf{p4\_flujo}$',
85     '$\mathsf{p4\_sin}$',
86 ]
87 nombre = 'PA2+PA4~matTot-'
88 etiquetaEjeY = 'Cantidad de Materias \naprobadas ($\mathsf{matTot}$)',
89 posiciones = [
90     (0,0), (1,0),
91     (0,1), (1,1),
92 ]
93
94 dimensiones = [
95     [P2_INPUT, None,      P4_INPUT, P4_DIBU],
96     [P2_OUTPUT, P2_IDSUB, P4_OUTPUT, P4_IDSUB],
97     [P2_SOLSUB, P2_INTSUB, P4_SOLSUB, P4_INTSUB],
98     [P2_NORM, P2_EXTRE,   P4_NORM, P4_EXTRE],
99     [P2_FLUJO, P2_SIN,    P4_FLUJO, P4_SIN],
100 ]
101 ixs_nombres = [
102     [0,-1,9,10],
103     [1,2,11,12],
104     [3,4,13,14],
105     [5,6,15,16],
106     [7,8,17,18],
107 ]
108
109 # Graficar -----
110 for grupo_graf in range(5):
111     fig, axs = plt.subplots(nrows=2, ncols=2, figsize=(8, 6))
112     for dimension, ix, dim in zip(dimensiones[grupo_graf], posiciones,
113     ixs_nombres[grupo_graf]):
114         if dimension:
115             # gráfico de dispersión
116             axs[ix].scatter(matTot, datos_grafico[dimension])
117
118             # Recta de regresión lineal:
119             m,b = regresionLineal(matTot, datos_grafico[dimension])
120             axs[ix].plot([minMat, maxMat], [b+m*minMat, b+m*maxMat], '- ',
121                 color='k')
122
123             # Configurando gráfico
124             axs[ix].yaxis.grid(True)
125             axs[ix].xaxis.grid(True)
126             axs[ix].set_ylabel(variables[dim])
127             axs[ix].set_yticks(range(0, 101, 10))
128             axs[ix].set_xlim(minMat, maxMat)
129             plt.setp(axs[ix],

```

```

129         xticks = list(range(minMat,maxMat+1)),
130         xticklabels=[',',1]+[(str(n) if n%5==0 else '') for n
131             in range(2,maxMat+5)]
132     )
133     else:
134         axs[1,0].remove()
135         axs[0,0].set_xlabel(etiquetaEjeY)
136     axs[posiciones[+1]].set_xlabel(etiquetaEjeY)
137     axs[posiciones[-1]].set_xlabel(etiquetaEjeY)
138
139     fig.tight_layout()
140     #print('Guardando archivo', nombre+str(grupo_graf)); plt.savefig(
141         nombre+str(grupo_graf)+'.pdf')
142     #print('Guardando archivo', nombre+str(grupo_graf)); plt.savefig(
143         nombre+str(grupo_graf)+'.png');
144     plt.show()

```

Listado de código D.13: Correlación entre PAP y RP

```

1  #!/usr/bin/python3
2  #coding: latin1
3  """
4  Este programa dado el archivo de datos cuantitativos genera
5  un gráfico de dispersión para las variables PAP~RP
6  con histogramas incorporados.
7  """
8  from regresion import regresionLineal
9  from columnas import *
10 import matplotlib.pyplot as plt
11 import fuente
12
13 lineas = open('../datos-cuanti.csv').readlines()
14
15 pa = []
16 rp = []
17
18 for l in lineas[1:]: #Omitir encabezados
19     datos = l.split(',')
20     pa.append(float(datos[PAP]))
21     rp.append(float(datos[RP]))
22
23 titulo = 'Variable  $\square$   $\square$  según la variable  $\square$ '
24 nombre = 'PAP~RP'
25
26 # definiciones de los ejes
27 izq, ancho = 0.1, 0.65
28 abajo, alto = 0.1, 0.65
29 espacio = 0.005
30
31 rect_scatter = [izq, abajo, ancho, alto]
32 rect_histx = [izq, abajo + alto + espacio, ancho, 0.2]
33 rect_histy = [izq + ancho + espacio, abajo, 0.2, alto]
34

```

```

35 # crear la figura
36 fig = plt.figure(figsize=(8, 8))
37
38 ax = fig.add_axes(rect_scatter)
39 ax_histx = fig.add_axes(rect_histx, sharex=ax)
40 ax_histy = fig.add_axes(rect_histy, sharey=ax)
41
42 # quitar las etiquetas
43 ax_histx.tick_params(axis="x", labelbottom=False)
44 ax_histy.tick_params(axis="y", labelleft=False)
45
46 # gráfico de dispersión:
47 ax.scatter(rp, pa)
48
49 # ancho del histograma
50 ancho_barras = 10
51
52 bins = list(range(0, 100 + ancho_barras, ancho_barras))
53 nx,bx,px = ax_histx.hist(rp, bins=bins)
54 ny,by,py = ax_histy.hist(pa, bins=bins, orientation='horizontal')
55
56 #Recta de regresión lineal:
57 m,b = regresionLineal(rp,pa)
58 ax.plot([0,100],[b,b+m*100],'-', color='k')
59 plt.suptitle(titulo+' $y = {:.2f}x + {:.2f}$ '.format(m,b), size=16, y
60             =0.99)
61
62 # Configurando ambos gráficos
63 ax.xaxis.grid(True)
64 ax.yaxis.grid(True)
65 ax.set_xlabel('Resolución de Problemas ( $\mathsf{RP}$ )')
66 ax.set_ylabel('Pensamiento Algorítmico ( $\mathsf{PAP}$ )')
67 ax.set_xticks(range(0, 101, 10))
68 ax.set_yticks(range(0, 101, 10))
69 ax_histx.xaxis.grid(True)
70 ax_histx.yaxis.grid(True)
71 ax_histy.xaxis.grid(True)
72 ax_histy.yaxis.grid(True)
73 ax_histx.set_yticks(range(0, int(max(nx))+1,2))
74 ax_histy.set_xticks(range(0, int(max(ny))+1,2))
75
76 fig.tight_layout()
77 #print('Guardando archivo', nombre); plt.savefig(nombre+'.pdf');
78 #print('Guardando archivo', nombre); plt.savefig(nombre+'.png')
79 plt.show()

```

Listado de código D.14: Relación entre PAP, RP y alg

```

1 #!/usr/bin/python3
2 #coding: latin1
3 """
4 Este programa dado el archivo de datos cuantitativos genera gráficos
5 de caja y de violín para las variables PAP~alg y para PR~alg.
6 """

```

```

6 from columns import *
7 import matplotlib.pyplot as plt
8 import fuente
9
10 lineas = open('../datos-cuanti.csv').readlines()
11
12 pa_alg = []
13 pa_noalg = []
14 rp_alg = []
15 rp_noalg = []
16
17 for l in lineas[1:]: #Omitir encabezados
18     datos = l.split(',')
19     if datos[ALG] == '1':
20         rp_alg.append(float(datos[RP]))
21         pa_alg.append(float(datos[PAP]))
22     elif datos[ALG] == '0':
23         rp_noalg.append(float(datos[RP]))
24         pa_noalg.append(float(datos[PAP]))
25     else:
26         raise ValueError(datos[ID]+'-'+datos[GRUPO]+'-'+datos[ALG])
27
28 datos_para_grafico_pa = [
29     pa_alg, pa_noalg
30 ]
31 datos_para_grafico_rp = [
32     rp_alg, rp_noalg
33 ]
34 variables = [
35     'Pensamiento Algorítmico ( $\text{PAP}$ )',
36     'Resolución de Problemas ( $\text{RP}$ )',
37 ]
38 titulos = [
39     'Variable  $\text{PAP}$  según la variable  $\text{alg}$ ',
40     'Variable  $\text{RP}$  según la variable  $\text{alg}$ ',
41 ]
42 nombres = ['PAP~alg', 'RP~alg']
43 colores = ['lightblue', 'lightgreen']
44
45 secuencia_graficos = (
46     datos_para_grafico_pa,
47     datos_para_grafico_rp,
48 )
49 # Graficar -----
50 for ix, datos_grafico in enumerate(secuencia_graficos):
51     fig, axs = plt.subplots(nrows=1, ncols=1, figsize=(5, 5))
52
53     # gráfico de violín
54     partes = axs.violinplot(datos_grafico,
55         # showmeans=True,
56         # showmedians=True,
57         showextrema=False,
58         # quantiles=[0.25,0.75], #No está en la versión 3.1.2... :(
59     )

```

```

60     for ix2, p in enumerate(partes['bodies']):
61         p.set_facecolor(colores[ix2])
62         p.set_edgecolor('black')
63         #p.set_alpha(0.9)
64
65     # gráfico de caja
66     grfCaja = axs.boxplot(datos_grafico,
67         patch_artist=True, #para ponerles color
68         #notch=True,
69     )
70     for patch, color in zip(grfCaja['boxes'], colores):
71         patch.set_facecolor(color)
72         patch.set_alpha(0.6)
73
74     #Medias
75     for ix2, d in enumerate(datos_grafico):
76         axs.scatter([ix2+1], [sum(d)/len(d)], zorder=3, color='k')
77
78     # Configurando ambos gráficos
79     axs.yaxis.grid(True)
80     axs.set_xlabel('Algoritmo_{}_Instrucciones_({}\mathsf{alg}\$)')
81     axs.set_yticks(range(0, 101, 10))
82     axs.set_ylabel(variables[ix])
83     plt.setp(axs,
84         xticks = list(range(1, len(datos_grafico)+1)),
85         xticklabels=['Algoritmo_({})'.format(len(datos_grafico[0])), '
86         Instrucciones_({})'.format(len(datos_grafico[1]))]
87     )
88     plt.title(titulos[ix], size=16)
89
90     fig.tight_layout()
91     #print('Guardando archivo', nombres[ix]); plt.savefig(nombres[ix]
92     #+'pdf');
93     #print('Guardando archivo', nombres[ix]); plt.savefig(nombres[ix]
94     #+'png')
95     plt.show()

```

Listado de código D.15: Relación entre PAP, RP y gen

```

1  #!/usr/bin/python3
2  #coding: latin1
3  """
4  Este programa dado el archivo de datos cuantitativos genera
5  gráficos de caja y de violín para las
6  variables PAP~gen y para PR~gen.
7  """
8  from columns import *
9  import matplotlib.pyplot as plt
10 import fuente
11
12 lineas = open('../datos-cuanti.csv').readlines()
13
14 pa_masculino = []
15 pa_femenino = []

```

```

16 rp_masculino = []
17 rp_femenino = []
18
19 rp1_masculino = []
20 pa2_masculino = []
21 rp3_masculino = []
22 pa4_masculino = []
23
24 rp1_femenino = []
25 pa2_femenino = []
26 rp3_femenino = []
27 pa4_femenino = []
28 for l in lineas[1:]: #Omitir encabezados
29     datos = l.split(',')
30     if datos[GEN] == 'm':
31         rp_masculino.append(float(datos[RP]))
32         pa_masculino.append(float(datos[PAP]))
33
34         rp1_masculino.append(int(datos[P1_SOL]) if datos[P1_SOL]!=''
35                               else 0.0)
36         pa2_masculino.append(int(datos[P3_SOL]) if datos[P3_SOL]!=''
37                               else 0.0)
38         rp3_masculino.append(int(datos[P2_SOLSUB]) if datos[P2_SOLSUB]
39                               !='' else 0.0)
40         pa4_masculino.append(int(datos[P4_SOLSUB]) if datos[P4_SOLSUB]
41                               !='' else 0.0)
42     elif datos[GEN] == 'f':
43         rp_femenino.append(float(datos[RP]))
44         pa_femenino.append(float(datos[PAP]))
45
46         rp1_femenino.append(int(datos[P1_SOL]) if datos[P1_SOL]!=''
47                               else 0.0)
48         pa2_femenino.append(int(datos[P3_SOL]) if datos[P3_SOL]!=''
49                               else 0.0)
50         rp3_femenino.append(int(datos[P2_SOLSUB]) if datos[P2_SOLSUB]
51                               !='' else 0.0)
52         pa4_femenino.append(int(datos[P4_SOLSUB]) if datos[P4_SOLSUB]
53                               !='' else 0.0)
54     else:
55         raise ValueError(datos[ID]+'-'+datos[GRUPO]+'-'+datos[GEN])
56
57 datos_para_grafico_pa = [
58     pa_masculino, pa_femenino
59 ]
60 datos_para_grafico_rp = [
61     rp_masculino, rp_femenino
62 ]
63 datos_para_grafico_rp1 = [
64     rp1_masculino, rp1_femenino
65 ]
66 datos_para_grafico_pa2 = [
67     pa2_masculino, pa2_femenino
68 ]
69 datos_para_grafico_rp3 = [

```

```

62     rp3_masculino, rp3_femenino
63 ]
64 datos_para_grafico_pa4 = [
65     pa4_masculino, pa4_femenino
66 ]
67 variables = [
68     'Pensamiento_Algorítmico_($\mathsf{PAP}\$)',
69     'Resolución_de_Problemas_($\mathsf{RP}\$)',
70     'Solución_del_problema_2_($\mathsf{p2\_solsub}\$)',
71     'Solución_del_problema_4_($\mathsf{p4\_solsub}\$)',
72     'Solución_del_problema_1_($\mathsf{p1\_sol}\$)',
73     'Solución_del_problema_3_($\mathsf{p3\_sol}\$)',
74 ]
75 titulos = [
76     'Variable_$\mathsf{PAP}\$_según_la_variable_$\mathsf{gen}\$',
77     'Variable_$\mathsf{RP}\$_según_la_variable_$\mathsf{gen}\$',
78     'Solución_de_Problema_2_($\mathsf{p2\_solsub}\$)_según_$\mathsf{gen}\$',
79     'Solución_de_Problema_4_($\mathsf{p4\_solsub}\$)_según_$\mathsf{gen}\$',
80     'Solución_de_Problema_1_($\mathsf{p1\_sol}\$)_según_$\mathsf{gen}\$',
81     'Solución_de_Problema_3_($\mathsf{p3\_sol}\$)_según_$\mathsf{gen}\$',
82 ]
83 nombres = [
84     'PAP~gen', 'RP~gen',
85     'PA2~gen', 'PA4~gen', 'RP1~gen', 'RP3~gen',
86 ]
87 colores = ['lightblue', 'pink']
88
89 secuencia_graficos = (
90     datos_para_grafico_pa, datos_para_grafico_rp,
91     #datos_para_grafico_pa2, datos_para_grafico_pa4,
92     #datos_para_grafico_rp1, datos_para_grafico_rp3,
93 )
94 # Graficar -----
95 for ix, datos_grafico in enumerate(secuencia_graficos):
96     fig, axs = plt.subplots(nrows=1, ncols=1, figsize=(5, 5))
97
98     # gráfico de violín
99     partes = axs.violinplot(datos_grafico,
100         #showmeans=True,
101         #showmedians=True,
102         showextrema=False,
103         #quantiles=[0.25,0.75], #No está en la versión 3.1.2... :(
104     )
105     for ix2, p in enumerate(partes['bodies']):
106         p.set_facecolor(colores[ix2])
107         p.set_edgecolor('black')
108         #p.set_alpha(0.9)
109
110     # gráfico de caja
111     grfCaja = axs.boxplot(datos_grafico,

```

```

112         patch_artist=True, #para ponerles color
113         #notch=True,
114     )
115     for patch, color in zip(grfCaja['boxes'], colores):
116         patch.set_facecolor(color)
117         patch.set_alpha(0.6)
118
119     #Medias
120     for ix2, d in enumerate(datos_grafico):
121         axs.scatter([ix2+1], [sum(d)/len(d)], zorder=3, color='k')
122
123     # Configurando ambos gráficos
124     axs.yaxis.grid(True)
125     axs.set_xlabel('Género_□( $\mathsf{gen}$ )')
126     axs.set_yticks(range(0, 101, 10))
127     axs.set_ylabel(variables[ix])
128     plt.setp(axs,
129             xticks = list(range(1, len(datos_grafico)+1)),
130             xticklabels=['Masculino_□({})'.format(len(datos_grafico[0])), '
131                         Femenino_□({})'.format(len(datos_grafico[1]))]
132     )
133     plt.title(titulos[ix], size=16)
134
135     fig.tight_layout()
136     #print('Guardando archivo', nombres[ix]); plt.savefig(nombres[ix
137     ]+'.pdf');
138     #print('Guardando archivo', nombres[ix]); plt.savefig(nombres[ix
139     ]+'.png')
140     plt.show()

```

Listado de código D.16: Relación entre PAP, RP y gen en orden ascendente

```

1  #!/usr/bin/python3
2  #coding: latin1
3  """
4  Este programa dado el archivo de datos cuantitativos genera
5  gráficos de líneas para las variables PAP~gen y para RP~gen
6  ordenando PAP y RP ascendentemente.
7  """
8  from columns import *
9  import matplotlib.pyplot as plt
10 import fuente
11
12 lineas = open('../datos-cuanti.csv').readlines()
13
14 pa_masculino = []
15 pa_femenino = []
16 rp_masculino = []
17 rp_femenino = []
18
19 for l in lineas[1:]: #Omitir encabezados
20     datos = l.split(',')
21     if datos[GEN] == 'm':
22         rp_masculino.append(float(datos[RP]))

```

```

23     pa_masculino.append(float(datos[PAP]))
24 elif datos[GEN] == 'f':
25     rp_femenino.append(float(datos[RP]))
26     pa_femenino.append(float(datos[PAP]))
27 else:
28     raise ValueError(datos[ID]+'-'+datos[GRUPO]+'-'+datos[GEN])
29
30 pa_masculino.sort()
31 pa_femenino.sort()
32 rp_masculino.sort()
33 rp_femenino.sort()
34
35 datos_para_grafico_pa = [
36     pa_masculino, pa_femenino
37 ]
38 datos_para_grafico_rp = [
39     rp_masculino, rp_femenino
40 ]
41 variables = [
42     'Pensamiento Algorítmico ( $\text{PAP}$ )',
43     'Resolución de Problemas ( $\text{RP}$ )',
44 ]
45 titulos = [
46     'Variable  $\text{PAP}$  según la variable  $\text{gen}$  orden ascendente',
47     'Variable  $\text{RP}$  según la variable  $\text{gen}$  orden ascendente',
48 ]
49 nombres = [
50     'PAP~gen-lineas', 'RP~gen-lineas',
51 ]
52 colores = ['royalblue', 'deeppink']
53
54 secuencia_graficos = (
55     datos_para_grafico_pa, datos_para_grafico_rp,
56 )
57 # Graficar -----
58 for ix, datos_grafico in enumerate(secuencia_graficos):
59     fig, axs = plt.subplots(nrows=1, ncols=1, figsize=(6, 4))
60
61     # Gráfico de líneas simple:
62     axs.plot(range(1, len(datos_grafico[0])+1), datos_grafico[0], label="Hombres", marker='o', markersize=2, color=colores[0])
63     axs.plot(range(1, len(datos_grafico[1])+1), datos_grafico[1], label="Mujeres", marker='o', markersize=2, color=colores[1])
64
65     # Configurando ambos gráficos
66     axs.yaxis.grid(True)
67     axs.xaxis.grid(True)
68     axs.set_xlabel('Cantidad de Alumnos')
69     axs.set_xticks([1]+list(range(5, len(datos_grafico[0])+5, 5)))
70     axs.set_yticks(range(0, 101, 10))
71     axs.set_ylabel(variables[ix])
72

```

```

73     axs.legend()
74     plt.title(titulos[ix], size=16)
75
76     fig.tight_layout()
77     #print('Guardando archivo', nombres[ix]); plt.savefig(nombres[ix]
78     #print('Guardando archivo', nombres[ix]); plt.savefig(nombres[ix]
79     plt.show()

```

Listado de código D.17: Relación entre PAP, RP y gen en orden ascendente con corrección de escala

```

1  #!/usr/bin/python3
2  #coding: latin1
3  """
4  Este programa dado el archivo de datos cuantitativos genera
5  gráficos de líneas para las variables PAP~gen y para RP~gen
6  ordenando PAP y RP ascendentemente y haciendo un «estiramiento»
7  de la escala horizontal para que ambos gráficos coincidan
8  debido a que tienen diferente cantidad de datos.
9  """
10 from columns import *
11 import matplotlib.pyplot as plt
12 import fuente
13
14 lineas = open('../datos-cuanti.csv').readlines()
15
16 pa_masculino = []
17 pa_femenino = []
18 rp_masculino = []
19 rp_femenino = []
20
21 for l in lineas[1:]: #Omitir encabezados
22     datos = l.split(',')
23     if datos[GEN] == 'm':
24         rp_masculino.append(float(datos[RP]))
25         pa_masculino.append(float(datos[PAP]))
26     elif datos[GEN] == 'f':
27         rp_femenino.append(float(datos[RP]))
28         pa_femenino.append(float(datos[PAP]))
29     else:
30         raise ValueError(datos[ID]+'-'+datos[GRUPO]+'-'+datos[GEN])
31
32 pa_masculino.sort()
33 pa_femenino.sort()
34 rp_masculino.sort()
35 rp_femenino.sort()
36
37 datos_para_grafico_pa = [
38     pa_masculino, pa_femenino
39 ]
40 datos_para_grafico_rp = [
41     rp_masculino, rp_femenino

```

```

42 ]
43 xMasculino = list(range(1, len(pa_masculino)+1))
44 xFemenino = list(map(
45     lambda x: (len(pa_masculino)-1)*(x-1.0)/(len(pa_femenino)-1)+1,
46     range(1, len(pa_femenino)+1)
47 ))
48
49 variables = [
50     'Pensamiento Algorítmico ( $\mathsf{PAP}$ )',
51     'Resolución de Problemas ( $\mathsf{RP}$ )',
52 ]
53 titulos = [
54     'Variable  $\mathsf{PAP}$  según la variable  $\mathsf{gen}$  en orden ascendente \n con corrección de escala',
55     'Variable  $\mathsf{RP}$  según la variable  $\mathsf{gen}$  en orden ascendente \n con corrección de escala',
56 ]
57 nombres = [
58     'PAP~gen-lineas2', 'RP~gen-lineas2',
59 ]
60 colores = ['royalblue', 'deeppink']
61
62 secuencia_graficos = (
63     datos_para_grafico_pa, datos_para_grafico_rp,
64 )
65 # Graficar -----
66 for ix, datos_grafico in enumerate(secuencia_graficos):
67     fig, axs = plt.subplots(nrows=1, ncols=1, figsize=(8, 5))
68
69     # Gráfico de líneas simple:
70     axs.plot(xMasculino, datos_grafico[0], label="Hombres", marker='o',
71             , markersize=3, color=colores[0])
72     axs.plot(xFemenino, datos_grafico[1], label="Mujeres", marker='o',
73             , markersize=3, color=colores[1])
74
75     # Configurando ambos gráficos
76     axs.yaxis.grid(True)
77     axs.xaxis.grid(True)
78     axs.set_xlabel('Alumnos, con cantidad de mujeres equiparadas a la cantidad de hombres')
79     axs.set_yticks(range(0, 101, 5))
80     axs.set_ylabel(variables[ix])
81     plt.setp(axs,
82             xticks = list(range(1, len(datos_grafico[0])+1)),
83             xticklabels=[],
84             )
85
86     axs.legend()
87     plt.title(titulos[ix], size=16)
88
89     fig.tight_layout()
90     #print('Guardando archivo', nombres[ix]); plt.savefig(nombres[ix]
91     #+'.pdf');
92     #print('Guardando archivo', nombres[ix]); plt.savefig(nombres[ix]

```

```

90     ]+'.png')
    plt.show()

```

Listado de código D.18: Relación entre PAP, RP y matProg

```

1  #!/usr/bin/python3
2  #coding: latin1
3  """
4  Este programa dado el archivo de datos cuantitativos genera
5  gráficos de caja y de violín para las
6  variables PAP~matProg y para PR~matProg.
7  """
8  from columns import *
9  import matplotlib.pyplot as plt
10 import fuente
11
12 lineas = open('../datos-cuanti.csv').readlines()
13
14 rp_p0 = []
15 rp_p1 = []
16 rp_p2 = []
17 rp_p3 = []
18 pa_p0 = []
19 pa_p1 = []
20 pa_p2 = []
21 pa_p3 = []
22
23 for l in lineas[1:]: #Omitir encabezados
24     datos = l.split(',')
25     if datos[GRUPO] == 'P0':
26         rp_p0.append(float(datos[RP]))
27         pa_p0.append(float(datos[PAP]))
28     elif datos[GRUPO] == 'P1':
29         rp_p1.append(float(datos[RP]))
30         pa_p1.append(float(datos[PAP]))
31     elif datos[GRUPO] == 'P2':
32         rp_p2.append(float(datos[RP]))
33         pa_p2.append(float(datos[PAP]))
34     elif datos[GRUPO] == 'P3':
35         rp_p3.append(float(datos[RP]))
36         pa_p3.append(float(datos[PAP]))
37     else:
38         raise ValueError(datos[GRUPO])
39
40 datos_para_grafico_pa = [
41     pa_p0, pa_p1, pa_p2, pa_p3
42 ]
43 datos_para_grafico_rp = [
44     rp_p0, rp_p1, rp_p2, rp_p3
45 ]
46 variables = ['Pensamiento_□Algorítmico_□($\mathsf{PAP}$)', 'Resolución_□
47     de_□Problemas_□($\mathsf{RP}$)']
48 titulos = ['Variable_□$\mathsf{PAP}$_□según_□la_□variable_□$\mathsf{matProg}$',
49     'Variable_□$\mathsf{RP}$_□según_□la_□variable_□$\mathsf{matProg}$']

```

```

48 nombres = ['PAP~matProg', 'RP~matProg']
49 colores = ['pink', 'lightgreen', 'lightblue', 'lightcoral']
50 secuencia_graficos = (
51     datos_para_grafico_pa, datos_para_grafico_rp
52 )
53 #Graficar -----
54 for ix, datos_grafico in enumerate(secuencia_graficos):
55     fig, axs = plt.subplots(nrows=1, ncols=1, figsize=(6, 6))
56
57     # gráfico de violín
58     axs.violinplot(datos_grafico,
59         #showmeans=True,
60         showmedians=True,
61         #widths=0.8,
62     )
63
64     # gráfico de caja
65     grfCaja = axs.boxplot(datos_grafico,
66         patch_artist=True, #para ponerles color
67     )
68     for patch, color in zip(grfCaja['boxes'], colores):
69         patch.set_facecolor(color)
70         patch.set_alpha(0.6)
71
72     #Medias
73     for ix2, d in enumerate(datos_grafico):
74         axs.scatter([ix2+1], [sum(d)/len(d)], zorder=3, color='k')
75
76     # Configurando ambos gráficos
77     axs.yaxis.grid(True)
78     axs.set_xlabel('Cantidad de Materias de Programación aprobadas ($\
79     mathsf{matProg}$)')
80     axs.set_ylabel(variables[ix])
81     axs.set_yticks(range(0, 101, 10))
82     plt.setp(axs,
83         xticks = list(range(1, len(datos_grafico)+1)),
84         xticklabels=range(4) #0,1,2,3
85     )
86     plt.title(titulos[ix], size=16)
87
88     fig.tight_layout()
89     #print('Guardando archivo', nombres[ix]); plt.savefig(nombres[ix
90     ]+'.pdf');
91     #print('Guardando archivo', nombres[ix]); plt.savefig(nombres[ix
92     ]+'.png')
93     plt.show()

```

Listado de código D.19: Relaciones entre PAP, RP y matTot, edad y cum

```

1 #!/usr/bin/python3
2 #coding: latin1
3 """
4 Este programa dado el archivo de datos cuantitativos genera gráficos
  de dispersión para las variables:

```

```

5  PAP~matTot, PR~matTot,
6  PAP~edad, PR~edad,
7  PAP~cum, PR~cum.
8  """
9  from regresion import regresionLineal
10 from columnas import *
11 import matplotlib.pyplot as plt
12 import fuente
13
14 lineas = open('../datos-cuanti.csv').readlines()
15 materias = []
16 edad = []
17 cum = []
18 pa = []
19 pr = []
20
21 for l in lineas[1:]: #Omitir encabezados
22     datos = l.split(',')
23     if datos[GRUPO] in ('P0', 'P1', 'P2', 'P3'):
24         matTotales = int(datos[MATTOT])
25
26         materias.append(matTotales)
27         edad.append(int(datos[EDAD]))
28         cum.append(float(datos[CUM]))
29         pa.append(float(datos[PAP]))
30         pr.append(float(datos[RP]))
31     else:
32         raise ValueError(datos[GRUPO])
33
34 # definiciones de los ejes
35 izq, ancho = 0.1, 0.65
36 abajo, alto = 0.1, 0.65
37 espacio = 0.015
38
39 rect_scatter = [izq, abajo, ancho, alto]
40 rect_histx = [izq, abajo + alto + espacio, ancho, 0.2]
41 rect_histy = [izq + ancho + espacio, abajo, 0.2, alto]
42
43 titulos = [
44     ['Variable_ $\mathsf{PAP}$ según la variable_ $\mathsf{matTot}$', '
45     Variable_ $\mathsf{RP}$ según la variable_ $\mathsf{matTot}$'],
46     ['Variable_ $\mathsf{PAP}$ según la variable_ $\mathsf{edad}$', '
47     Variable_ $\mathsf{RP}$ según la variable_ $\mathsf{edad}$'],
48     ['Variable_ $\mathsf{PAP}$ según la variable_ $\mathsf{cum}$', '
49     Variable_ $\mathsf{RP}$ según la variable_ $\mathsf{cum}$'],
50 ]
51
52 etiquetas = [
53     'Cantidad de materias aprobadas por los alumnos_ ($\mathsf{matTot}$)
54     ',
55     'Edad de los alumnos_ ($\mathsf{edad}$)',
56     'CUM de los alumnos_ ($\mathsf{cum}$)',
57 ]
58
59 nombres = [
60     ['PAP~matTot', 'RP~matTot'],

```

```

55     ['PAP~edad', 'RP~edad'],
56     ['PAP~cum', 'RP~cum']
57 ]
58 minEjeX = [0, min(edad)-1, 6]
59 maxEjeX = [
60     (max(materias)//5)*5+5,
61     max(edad),
62     10
63 ]
64 etiquetasX = [
65     ['',1]+[(str(n) if n%5==0 else '') for n in range(2,max(materias)
66     +5)],
67     range(minEjeX[1],max(edad)+1),
68     [str(c/2.0+6.0) for c in range(9)],
69 ]
70 # anchos del histograma
71 anchoY = 5
72 anchosX = [3, 1, 0.25]
73
74 binsX = [
75     list(range(minEjeX[0], maxEjeX[0]+anchosX[0], anchosX[0])),
76     list(range(minEjeX[1], maxEjeX[1]+anchosX[1], anchosX[1])),
77     [6.0+k*anchosX[2] for k in range(int((10.0-6.0)/anchosX[2])+1)]
78 ]
79
80 variablesY = ['Pensamiento_□Algorítmico_□($\mathsf{PAP}$)', 'Resolución_□
81     de_□Problemas_□($\mathsf{RP}$)']
82
83 colores = ['lightgreen', 'lightblue']
84
85 # Graficar -----
86 for ix_var, ejeX in enumerate((materias, edad, cum)):
87     for ix_dim, ejeY in zip(range(2), (pa, pr)):
88         # crear la figura
89         fig = plt.figure(figsize=(8, 8))
90
91         ax = fig.add_axes(rect_scatter)
92         ax_histx = fig.add_axes(rect_histx, sharex=ax)
93         ax_histy = fig.add_axes(rect_histy, sharey=ax)
94
95         # quitar las etiquetas
96         ax_histx.tick_params(axis="x", labelbottom=False)
97         ax_histy.tick_params(axis="y", labelleft=False)
98
99         # gráfico de dispersión
100        ax.scatter(ejeX, ejeY,
101                c=colores[ix_dim]
102        )
103
104        # *****
105        # binsEdad = list(range(minEdad, maxEdad + ancho_barras_edad,
106        #     ancho_barras_edad))
107        # binsCUM = np.arange(min(cum), max(cum) + ancho_barras_cum,
108        #     ancho_barras_cum)

```

```

105     nx,bx,px = ax_histx.hist(ejeX, bins=binsX[ix_var])
106     ny,by,py = ax_histy.hist(ejeY, bins=range(0,100+anchoY, anchoY)
        , orientation='horizontal')
107     ax_histx.xaxis.grid(True)
108     ax_histx.yaxis.grid(True)
109     ax_histy.xaxis.grid(True)
110     ax_histy.yaxis.grid(True)
111     ax_histx.set_yticks(range(0, int(max(nx))+1,3))
112     ax_histy.set_xticks(range(0, int(max(ny))+1,2))
113
114
115     #Recta de regresión lineal:
116     m, b = regresionLineal(ejeX, ejeY)
117     ax.plot([minEjeX[ix_var], maxEjeX[ix_var]], [b+minEjeX[ix_var]
        ]*m, b+maxEjeX[ix_var]*m], '-')
118
119     # Configurando el gráfico
120     ax.xaxis.grid(True)
121     ax.yaxis.grid(True)
122     ax.set_ylabel(variablesY[ix_dim])
123     ax.set_yticks(range(0, 101, 10))
124     ax.set_ylim(bottom=0, top=100)
125     ax.set_xlim(minEjeX[ix_var], maxEjeX[ix_var])
126
127     ax.set_xlabel(etiquetas[ix_var])
128     plt.setp(ax,
129             xticks = range(minEjeX[ix_var], maxEjeX[ix_var]+1) if
                ix_var!=2 else [c/2.0+6.0 for c in range(9)],
130             xticklabels=etiquetasX[ix_var]
131         )
132
133     #Título
134     plt.suptitle(titulos[ix_var][ix_dim]+'□/□$y={:0.2f}x+{:0.2f}$',
135                 .format(m,b),
136                 size=14, y=0.99
137             )
138
139     fig.tight_layout()
140     #print('Guardando archivo', nombres[ix_var][ix_dim]); plt.
        savefig(nombres[ix_var][ix_dim]+'pdf');
141     #print('Guardando archivo', nombres[ix_var][ix_dim]); plt.
        savefig(nombres[ix_var][ix_dim]+'png')
    plt.show()

```

Listado de código D.20: Dispersión de PAP, RP según matTot

```

1  #!/usr/bin/python3
2  #coding: latin1
3  """
4  Este programa dado el archivo de datos cuantitativos genera
5  gráficos de caja y de violín para las
6  variables PAP~matTot y para PR~matTot.
7  """
8  from columnas import *

```

```

9 import matplotlib.pyplot as plt
10 import fuente
11
12 lineas = open('../datos-cuanti.csv').readlines()
13
14 rp = {}
15 pa = {}
16
17 for l in lineas[1:]: #Omitir encabezados
18     datos = l.split(',')
19     matTot = int(datos[MATTOT])
20     if matTot not in rp: #todavía no está ese valor
21         rp[matTot] = []
22         pa[matTot] = []
23     rp[matTot].append(float(datos[RP ]))
24     pa[matTot].append(float(datos[PAP]))
25
26 temp = rp.keys()
27 minMaterias = min(temp)
28 maxMaterias = max(temp)
29 datos_para_grafico_pa = []
30 datos_para_grafico_rp = []
31
32 for m in range(minMaterias, maxMaterias+1):
33     datos_para_grafico_pa.append(pa.get(m, [0]))
34     datos_para_grafico_rp.append(rp.get(m, [0]))
35
36 variables = ['Pensamiento_□Algorítmico_□($\mathsf{PAP}$)', 'Resolución_□
de_□Problemas_□($\mathsf{RP}$)']
37 titulos = ['Variable_□$\mathsf{PAP}$_□según_□la_□variable_□$\mathsf{matTot}$
$', 'Variable_□$\mathsf{RP}$_□según_□la_□variable_□$\mathsf{matTot}$']
38 nombres = ['PAP~matTot-cajas', 'RP~matTot-cajas']
39 #colores = ['pink', 'lightgreen', 'lightblue', 'lightcoral']
40 secuencia_graficos = (
41     datos_para_grafico_pa, datos_para_grafico_rp
42 )
43 #Graficar -----
44 for ix, datos_grafico in enumerate(secuencia_graficos):
45     fig, axs = plt.subplots(nrows=1, ncols=1, figsize=(9, 6))
46
47     # gráfico de violín
48     axs.violinplot(datos_grafico,
49         #showmeans=True,
50         showmedians=True,
51         #widths=0.8,
52     )
53
54     # gráfico de caja
55     grfCaja = axs.boxplot(datos_grafico,
56         patch_artist=True, #para ponerles color
57     )
58     for patch in grfCaja['boxes']:
59         patch.set_facecolor('lightgreen')
60         patch.set_alpha(0.6)

```

```

61     #for patch, color in zip(grfCaja['boxes'], colores):
62         #patch.set_facecolor(color) #¿Un color a cada caja??
63         #patch.set_alpha(0.6)
64
65     #Medias
66     for ix2, d in enumerate(datos_grafico):
67         if d != [0]:
68             axs.scatter([ix2+1], [sum(d)/len(d)], zorder=3, color='k')
69
70     # Configurando ambos gráficos
71     axs.yaxis.grid(True)
72     axs.set_xlabel('Cantidad de Materias Totales aprobadas ( $\mathit{matTot}$ )')
73     axs.set_ylabel(variables[ix])
74     axs.set_yticks(range(0, 101, 10))
75     plt.setp(axs,
76             xticks = list(range(1, len(datos_grafico)+1)),
77             xticklabels=range(minMaterias, maxMaterias+1) #
78     )
79     plt.title(titulos[ix], size=16)
80
81     fig.tight_layout()
82     print('Guardando archivo', nombres[ix]); plt.savefig(nombres[ix]+'
83             .pdf');
84     #print('Guardando archivo', nombres[ix]); plt.savefig(nombres[ix]
85             +' .png')
86     #plt.show()

```

Listado de código D.21: Detalle de las sub-variables de RP según matProg

```

1  #!/usr/bin/python3
2  #coding: latin1
3  """
4  Este programa dado el archivo de datos cuantitativos genera
5  gráficos de caja y de violín para las subvariables de RP
6  respecto de la variable matProg en un sólo archivo.
7  """
8  from columns import *
9  import matplotlib.pyplot as plt
10 import fuente
11
12 lineas = open('../datos-cuanti.csv').readlines()
13
14 P0 = 0
15 P1 = 1
16 P2 = 2
17 P3 = 3
18
19 datos_grafico = {
20     P1_GRAF: [[], [], [], []],
21     P1_CIU : [[], [], [], []],
22     P1_CARR: [[], [], [], []],
23     P1_SOL : [[], [], [], []],
24     P1_RES : [[], [], [], []],

```

```

25     P3_MOD : [[],[],[],[]],
26     P3_SOL : [[],[],[],[]],
27     P3_RES : [[],[],[],[]],
28 }
29
30 for l in lineas[1:]: #Omitir encabezados
31     datos = l.split(',')
32     if datos[GRUPO] == 'P0':
33         g = P0
34     elif datos[GRUPO] == 'P1':
35         g = P1
36     elif datos[GRUPO] == 'P2':
37         g = P2
38     elif datos[GRUPO] == 'P3':
39         g = P3
40     else:
41         raise ValueError(datos[GRUPO])
42
43     datos_grafico[P1_GRAF][g].append(int(datos[P1_GRAF]) if datos[
44         P1_GRAF]!='' else 0)
45     datos_grafico[P1_CIU][g].append(int(datos[P1_CIU]) if datos[
46         P1_CIU]!='' else 0)
47     datos_grafico[P1_CARR][g].append(int(datos[P1_CARR]) if datos[
48         P1_CARR]!='' else 0)
49     datos_grafico[P1_SOL][g].append(int(datos[P1_SOL]) if datos[
50         P1_SOL]!='' else 0)
51     datos_grafico[P1_RES][g].append(int(datos[P1_RES]) if datos[
52         P1_RES]!='' else 0)
53     datos_grafico[P3_MOD][g].append(int(datos[P3_MOD]) if datos[
54         P3_MOD]!='' else 0)
55     datos_grafico[P3_SOL][g].append(int(datos[P3_SOL]) if datos[
56         P3_SOL]!='' else 0)
57     datos_grafico[P3_RES][g].append(int(datos[P3_RES]) if datos[
58         P3_RES]!='' else 0)
59
60 variables = [
61     '$\mathsf{p1\_graf}$',
62     '$\mathsf{p1\_ciu}$',
63     '$\mathsf{p1\_carr}$',
64     '$\mathsf{p1\_sol}$',
65     '$\mathsf{p1\_res}$',
66     '$\mathsf{p3\_mod}$',
67     '$\mathsf{p3\_sol}$',
68     '$\mathsf{p3\_res}$',
69 ]
70 nombre = 'RP1+RP3~matProg'
71 colores = ['pink', 'lightgreen', 'lightblue', 'lightcoral']
72 posiciones = [
73     (0,0), (0,1),
74     (1,0), (1,1),
75     (2,0), (2,1),
76     (3,0), (3,1),
77 ]

```

```

71 # Graficar -----
72 fig, axs = plt.subplots(nrows=4, ncols=2, figsize=(8, 11))
73 for dimension, ix, dim in zip(range(P1_GRAF, P3_RES+1), posiciones,
74     range(8)):
75     # gráfico de violín
76     axs[ix].violinplot(datos_grafico[dimension],
77         #showmeans=True,
78         showmedians=True,
79         #widths=0.8,
80     )
81     # gráfico de caja
82     grfCaja = axs[ix].boxplot(datos_grafico[dimension],
83         patch_artist=True, #para ponerles color
84         #notch=True,
85     )
86     for patch, color in zip(grfCaja['boxes'], colores):
87         patch.set_facecolor(color)
88         patch.set_alpha(0.6)
89
90     #Medias
91     for ix2, d in enumerate(datos_grafico[dimension]):
92         axs[ix].scatter([ix2+1], [sum(d)/len(d)], zorder=3, color='k')
93
94     # Configurando gráfico
95     axs[ix].yaxis.grid(True)
96     axs[ix].set_ylabel(variables[dim])
97     axs[ix].set_yticks(range(0, 101, 10))
98     plt.setp(axs[ix],
99         xticks = list(range(1,4+1)),
100         xticklabels=range(4) #0,1,2,3
101     )
102     axs[posiciones[-2]].set_xlabel('Cantidad de Materias \nde Programación
103     aprobadas (\mathsf{matProg})')
104     axs[posiciones[-1]].set_xlabel('Cantidad de Materias \nde Programación
105     aprobadas (\mathsf{matProg})')
106     #plt.suptitle("Detalle de \mathsf{RP} según \mathsf{matProg}",
107         size=16, y=1.005)
108
109     fig.tight_layout()
110     #print('Guardando archivo', nombre); plt.savefig(nombre+'.pdf');
111     #print('Guardando archivo', nombre); plt.savefig(nombre+'.png')
112     plt.show()

```

Listado de código D.22: Detalle de las sub-variables de RP según matTot

```

1 #!/usr/bin/python3
2 #coding: latin1
3 """
4 Este programa dado el archivo de datos cuantitativos genera
5 gráficos de dispersión para las subvariables de PR
6 respecto de la variable matTot en un sólo archivo.
7 """
8 from regresion import regresionLineal

```

```

9  from columns import *
10 import matplotlib.pyplot as plt
11 import fuente
12
13 lineas = open('../datos-cuanti.csv').readlines()
14
15 matTot = []
16 datos_grafico = {
17     P1_GRAF: [],
18     P1_CIU : [],
19     P1_CARR: [],
20     P1_SOL : [],
21     P1_RES : [],
22     P3_MOD : [],
23     P3_SOL : [],
24     P3_RES : [],
25 }
26
27 for l in lineas[1:]: #Omitir encabezados
28     datos = l.split(',')
29     matTot.append(int(datos[MATTOT]))
30
31     datos_grafico[P1_GRAF].append(int(datos[P1_GRAF]) if datos[P1_GRAF]
32     != '' else 0)
33     datos_grafico[P1_CIU].append(int(datos[P1_CIU]) if datos[P1_CIU]
34     != '' else 0)
35     datos_grafico[P1_CARR].append(int(datos[P1_CARR]) if datos[P1_CARR]
36     != '' else 0)
37     datos_grafico[P1_SOL].append(int(datos[P1_SOL]) if datos[P1_SOL]
38     != '' else 0)
39     datos_grafico[P1_RES].append(int(datos[P1_RES]) if datos[P1_RES]
40     != '' else 0)
41     datos_grafico[P3_MOD].append(int(datos[P3_MOD]) if datos[P3_MOD]
42     != '' else 0)
43     datos_grafico[P3_SOL].append(int(datos[P3_SOL]) if datos[P3_SOL]
44     != '' else 0)
45     datos_grafico[P3_RES].append(int(datos[P3_RES]) if datos[P3_RES]
46     != '' else 0)
47
48 minMat = 0 #min(matTot)
49 maxMat = (max(matTot)//5)*5+5
50 variables = [
51     '$\mathsf{p1\_graf}$',
52     '$\mathsf{p1\_ciu}$',
53     '$\mathsf{p1\_carr}$',
54     '$\mathsf{p1\_sol}$',
55     '$\mathsf{p1\_res}$',
56     '$\mathsf{p3\_mod}$',
57     '$\mathsf{p3\_sol}$',
58     '$\mathsf{p3\_res}$',
59 ]
60 nombre = 'RP1+RP3~matTot'
61 posiciones = [
62     (0,0), (0,1),

```

```

55     (1,0), (1,1),
56     (2,0), (2,1),
57     (3,0), (3,1),
58 ]
59
60 # Graficar -----
61 fig, axs = plt.subplots(nrows=4, ncols=2, figsize=(8, 11))
62 for dimension, ix, dim in zip(range(P1_GRAF, P3_RES+1), posiciones,
63     range(8)):
64     # gráfico de dispersión
65     axs[ix].scatter(matTot, datos_grafico[dimension])
66
67     # Recta de regresión lineal:
68     m,b = regresionLineal(matTot, datos_grafico[dimension])
69     axs[ix].plot([minMat, maxMat], [b+m*minMat, b+m*maxMat], '-', color='k
70         ')
71
72     # Configurando gráfico
73     axs[ix].yaxis.grid(True)
74     axs[ix].xaxis.grid(True)
75     axs[ix].set_ylabel(variables[dim])
76     axs[ix].set_yticks(range(0, 101, 10))
77     axs[ix].set_xlim(minMat, maxMat)
78     plt.setp(axs[ix],
79         xticks = list(range(minMat, maxMat+1)),
80         xticklabels=['', 1]+[(str(n) if n%5==0 else '') for n in range
81             (2, maxMat+5)]
82     )
83     axs[posiciones[-2]].set_xlabel('Cantidad de Materias \naprobadas (\mathsf{matTot})')
84     axs[posiciones[-1]].set_xlabel('Cantidad de Materias \naprobadas (\mathsf{matTot})')
85     #plt.suptitle("Detalle de  $\mathsf{RP}$  según  $\mathsf{matTot}$ ", size
86         =16, y=1.005)
87
88 fig.tight_layout()
89 #print('Guardando archivo', nombre); plt.savefig(nombre+'.pdf');
90 #print('Guardando archivo', nombre); plt.savefig(nombre+'.png')
91 plt.show()

```

E. Script para la generación de los Diagramas tipo Schoenfeld en la parte cualitativa

En la sección 3.13 en la página 117 se explica la forma del archivo de datos de entrada para este programa, y en la sección 3.12 en la página 111 se explican los gráficos que genera como salida, y ahí también se explica su interpretación.

Para realizar los gráficos de la sección 4.2 se implementó el siguiente script en Lenguaje Python¹ v3.8.2, utilizando principalmente la biblioteca Matplotlib² diseñada por Hunter (2007), concretamente en su versión 3.1.2, de Caswell y cols. (2019).

Listado de código E.1: Script para generación de gráficos de Actividades Cognitivas tipo Schoenfeld

```
1 #!/usr/bin/python3
2 #coding: latin1
3 """
4 Este programa dado el archivo de datos cualitativos genera gráficos
5 tipo Schoenfeld.
6 """
7 import matplotlib.pyplot as plt
8 plt.rcParams.update({
9     "font.family": "serif",
10 })
11 ACT = [
12     'le', #Leer
13     'an', #Analizar
14     'co', #Corregir conocimiento previo
15     're', #Recordar conocimiento previo
16     'nu', #Generar nuevo conocimiento
17     'so', #Construir solución
18     'aj', #Ajustar solución
19     'im', #Implementar solución
20     've', #Verificar solución
21     '#RE', #Resultado - Problema resuelto o no resuelto
22 ]
23
24 ACT_TEXTO = [
25     "Leer\n\n",
```

¹<https://www.python.org/>

²<https://matplotlib.org/>

```

26     "Analizar\n\n",
27     "Corregir_Conoc\n\n",
28     "Recordar_Conoc\n\n",
29     "Crear_Conoc\n\n",
30     "Const_Solución\n\n",
31     "Ajustar_Sol\n\n",
32     "Implementar\n\n",
33     "Verificar\n\n",
34 #     "Resultado\n\n",
35 ]
36
37 TIEMPO = 0
38 SECUENCIA = 1
39 VALOR = 2
40
41 propiedades_graficos = dict(
42     color='black',
43     linewidth=2,
44     drawstyle='steps-post',
45     label="Secuencia_de_actividades_cognitivas",
46 )
47 propiedades_ayuda = dict(
48     #marker='P',
49     marker='o',
50     linewidth=0, #para que sólo dibuje los puntos
51     markersize=5,
52     color='red',
53     label='Pide_o_Recibe_Ayuda',
54 )
55 propiedades_linea_ayuda = dict(
56     color=propiedades_ayuda['color'],
57     linewidth=propiedades_graficos['linewidth'],
58     #label="Duración de la ayuda", #No debe salir en la leyenda
59 )
60 propiedades_ayuda_texto = dict(
61     s='F1',
62     color=propiedades_ayuda['color'],
63     size=10,
64     #fontfamily="Latin Modern Roman",
65     horizontalalignment='center',
66     fontstretch='ultra-condensed',
67 )
68 propiedades_falla = dict(
69     #marker='$\ddot{\$}', #dibujar una unidad abajo
70     color=propiedades_graficos['color'],
71     marker='$/$',
72     #color='chocolate',
73     linewidth=0,
74     markersize=8,
75     label='Inicio_de_Falla_técnica_o_de_comunicación',
76 )
77 propiedades_linea_falla = dict(
78     color='white',
79     linewidth=propiedades_graficos['linewidth'],

```

```

80     #label="Duración de la falla", #No debe salir en la leyenda
81 )
82 propiedades_nopuedo = dict(
83     marker='o',
84     linewidth=0,
85     markersize=5,
86     color='blue',
87     label='Dice que no puede',
88 )
89 propiedades_nopuedo_texto = dict(
90     s='x_x',
91     color=propiedades_nopuedo['color'],
92     size=15,
93     fontfamily="Liberation_Sans",
94     horizontalalignment='center',
95     fontstretch='ultra-condensed',
96 )
97 propiedades_eje_vertical_texto = dict(
98     #color='black',
99     size=15,
100    fontfamily="Latin_Modern_Roman",
101    horizontalalignment='right',
102    verticalalignment='center',
103    fontstretch='ultra-condensed',
104 )
105 propiedades_resuelto = dict(
106    marker='*',
107    linewidth=0,
108    markersize=18,
109    color='green',
110    label='Problema resuelto',
111 )
112 propiedades_noresuelto = dict(
113    marker='X',
114    linewidth=0,
115    markersize=16,
116    color='green',
117    label='Problema no resuelto',
118 )
119
120
121 def convertir(act):
122     """Convierte un texto de actividad en un número.
123     """
124     if act=='le': return 0.5 #Si es cero no sale en el gráfico, no sé
125         por qué...
126     if act=='an': return 1.5
127     if act=='co': return 2.5
128     if act=='re': return 3.5
129     if act=='nu': return 4.5
130     if act=='so': return 5.5
131     if act=='aj': return 6.5
132     if act=='im': return 7.5
133     if act=='ve': return 8.5

```

```

133     return 0
134
135
136 lineas = open('datos-cuali.csv').readlines()
137
138 tiempos_norm = {} #Valores de tiempo de 'actividades_norm'
139 tiempos_todos = {} #Todas las marcas de tiempo
140 actividades_norm = {} #Actividades normales le~an~co~re~nu~so~aj~im~ve
    representadas numéricamente
141 secuencia_norm = {} #Valores correlativos para gráfico secundario.
    Secuencia en lugar de tiempo.
142
143 act_ayuda = {} #'ay' -- pedir o recibir ayuda
144 act_falla = {} #'te' -- falla técnica
145 act_nopuedo = {} #'no' -- indicar que no puede
146 act_resuelto = {} #'pr' -- problema resuelto
147 act_noresuelto = {} #'pn' -- problema no resuelto
148
149
150 esteSujeto = ''
151 for l in lineas[1:]: #Omitir encabezados
152     sujeto, str_tiempo, str_actividad = l.split(',')
153     str_actividad = str_actividad[:-1] #quitar el '\n' final
154     tiempoDecimal = float(str_tiempo)
155
156     if sujeto != '': #Es el mismo sujeto de la línea anterior del
        archivo
157         cont_secuencia = 0
158         esteSujeto = sujeto
159         #Valores del eje horizontal en escala de tiempo (minutos y
            segundos):
160         tiempos_norm[esteSujeto] = []
161         #Valores del eje horizontal en secuencia lineal (correlativo
            de cada actividad cognitiva reportada):
162         secuencia_norm[esteSujeto] = []
163
164         #Valores del eje vertical de las actividades cognitivas:
165         actividades_norm[esteSujeto] = []
166
167         #Otras actividades, (tiempoDecimal, secuencia, valor-eje-
            vertical):
168         act_ayuda[esteSujeto] = ([], [], [])
169         act_falla[esteSujeto] = ([], [], [])
170         act_nopuedo[esteSujeto] = ([], [], [])
171         act_resuelto[esteSujeto] = ([], [], [])
172         act_noresuelto[esteSujeto] = ([], [], [])
173
174         tiempos_todos[esteSujeto] = []
175
176         actividad = convertir(str_actividad)
177         if actividad!=0: #Las actividades normales
178             tiempos_norm[esteSujeto] .append(tiempoDecimal)
179             secuencia_norm[esteSujeto] .append(cont_secuencia)
180

```

```

181     actividades_norm[esteSujeto].append(actividad)
182
183     tiempos_todos[esteSujeto] .append(tiempoDecimal)
184 else:
185     #valores especiales
186     if str_actividad == 'pr':
187         ultimaActividad = actividades_norm[esteSujeto][-1]
188
189         act_resuelto[esteSujeto][TIEMPO] .append(tiempoDecimal)
190         act_resuelto[esteSujeto][SECUENCIA].append(cont_secuencia
191             +0.5)
192         act_resuelto[esteSujeto][VALOR] .append(ultimaActividad
193             )
194
195         tiempos_todos[esteSujeto] .append(tiempoDecimal)
196
197         #Agregar línea hasta el final
198         tiempos_norm[esteSujeto] .append(tiempoDecimal)
199         secuencia_norm[esteSujeto] .append(cont_secuencia)
200         actividades_norm[esteSujeto].append(ultimaActividad)
201     elif str_actividad == 'pn':
202         ultimaActividad = actividades_norm[esteSujeto][-1]
203
204         act_noresuelto[esteSujeto][TIEMPO] .append(tiempoDecimal
205             )
206         act_noresuelto[esteSujeto][SECUENCIA].append(
207             cont_secuencia+0.5)
208         act_noresuelto[esteSujeto][VALOR] .append(
209             ultimaActividad)
210
211         tiempos_todos[esteSujeto] .append(tiempoDecimal)
212
213         #Agregar línea hasta el final
214         tiempos_norm[esteSujeto] .append(tiempoDecimal)
215         secuencia_norm[esteSujeto] .append(cont_secuencia)
216         actividades_norm[esteSujeto].append(ultimaActividad)
217     elif str_actividad == 'ay':
218         act_ayuda[esteSujeto][TIEMPO] .append(tiempoDecimal)
219         act_ayuda[esteSujeto][SECUENCIA] .append(cont_secuencia
220             +0.5)
221         act_ayuda[esteSujeto][VALOR] .append(actividades_norm
222             [esteSujeto][-1])
223         tiempos_todos[esteSujeto] .append(tiempoDecimal)
224     elif str_actividad == 'te':
225         act_falla[esteSujeto][TIEMPO] .append(tiempoDecimal)
226         act_falla[esteSujeto][SECUENCIA].append(cont_secuencia
227             +0.5)
228         act_falla[esteSujeto][VALOR] .append(actividades_norm[
229             esteSujeto][-1])
230         tiempos_todos[esteSujeto] .append(tiempoDecimal)
231     elif str_actividad == 'no':
232         act_nopuedo[esteSujeto][TIEMPO] .append(tiempoDecimal)
233         act_nopuedo[esteSujeto][SECUENCIA].append(cont_secuencia
234             +0.5)

```

```

225         act_nopuedo[esteSujeto][VALOR]         .append(actividades_norm
226             [esteSujeto][-1])
227         tiempos_todos[esteSujeto]             .append(tiempoDecimal)
228     cont_secuencia += 1
229
230 # Graficar
231 for sujeto in tiempos_norm.keys():
232     #for sujeto in (list(tiempos_norm.keys())[14],): #para pruebas
233     fig, axs = plt.subplots(2, 1,
234         figsize=(14, 10), #para los sujetos reales
235         #figsize=(7,10) #para los ejemplos
236     )
237     #plt.suptitle('Sujeto: '+sujeto,
238     axs[0].set_title('Sujeto:□'+sujeto,
239         size=20,
240     )
241
242     #Primer gráfico, en escala de tiempo (minutos) -----
243     axs[0].axhspan(0, 1, xmin=-1, color='powderblue')
244     axs[0].axhspan(1, 2, xmin=-1, color='khaki')
245     axs[0].axhspan(2, 5, xmin=-1, color='lightgray')
246     axs[0].axhspan(5, 7, xmin=-1, color='lightgreen')
247     axs[0].axhspan(7, 9, xmin=-1, color='lightblue')
248     # https://matplotlib.org/stable/gallery/color/named_colors.html
249     axs[0].plot(
250         tiempos_norm[sujeto], actividades_norm[sujeto],
251         **propiedades_graficos,
252     )
253     axs[0].set_xlabel('Tiempo□(minutos)')
254     axs[0].grid(True)
255     axs[0].set_ylim(-0.5,9.5)
256     axs[0].set_yticks(range(10))
257     axs[0].yaxis.set_ticklabels(ACT_TEXTO)
258     #axs[0].yaxis.set_ticklabels([])
259     axs[0].set_xticks(range(int(tiempos_norm[sujeto][-1]+1)+1))
260
261     axs[0].plot(act_ayuda[sujeto][TIEMPO], act_ayuda[sujeto][VALOR],
262         **propiedades_ayuda,
263     )
264     i = 1
265     for t,v in zip(act_ayuda[sujeto][TIEMPO], act_ayuda[sujeto][VALOR
266         ]):
267         #Buscar en tiempos_todos[sujeto] dónde termina la ayuda:
268         while i<len(tiempos_todos[sujeto]) and tiempos_todos[sujeto][i
269             ] <= t:
270             i += 1
271         #Dibujar la línea roja:
272         #print(sujeto,t,tiempos_todos[sujeto][i])
273         axs[0].plot([t,tiempos_todos[sujeto][i]], [v,v], **
274             propiedades_linea_ayuda)

```

```

275 )
276 i = 1
277 for t,v in zip(act_falla[sujeto][TIEMPO], act_falla[sujeto][VALOR
278 ]):
279     #Buscar en tiempos_todos[sujeto] dónde termina la falla:
280     while i<len(tiempos_todos[sujeto]) and tiempos_todos[sujeto][i
281 ] <= t:
282         i += 1
283     #Dibujar la línea blanca:
284     axs[0].plot([t,tiempos_todos[sujeto][i]], [v,v], **
285 propiedades_linea_falla)
286 axs[0].plot(act_nopuedo[sujeto][TIEMPO], act_nopuedo[sujeto][VALOR
287 ],
288 **propiedades_nopuedo ,
289 )
290 axs[0].plot(act_resuelto[sujeto][TIEMPO], act_resuelto[sujeto][
291 VALOR],
292 **propiedades_resuelto ,
293 )
294 axs[0].plot(act_noresuelto[sujeto][TIEMPO], act_noresuelto[sujeto
295 ][VALOR],
296 **propiedades_noresuelto ,
297 )
298 axs[0].legend(
299     markerscale=0.8,
300     loc='lower_right',
301 ) # https://matplotlib.org/stable/api/\_as\_gen/matplotlib.axes.Axes
302 .legend.html
303
304 #Segundo gráfico, en secuencia correlativa -----
305 axs[1].axhspan(0, 1, xmin=-1, color='powderblue')
306 axs[1].axhspan(1, 2, xmin=-1, color='khaki')
307 axs[1].axhspan(2, 5, xmin=-1, color='lightgray')
308 axs[1].axhspan(5, 7, xmin=-1, color='lightgreen')
309 axs[1].axhspan(7, 9, xmin=-1, color='lightblue')
310 # https://matplotlib.org/stable/gallery/color/named\_colors.html
311 axs[1].plot(
312     secuencia_norm[sujeto], actividades_norm[sujeto],
313     **propiedades_graficos ,
314 )
315 axs[1].set_xlabel('Secuencia_(total_{ }_tareas_cognitivas)'.format(
316     len(secuencia_norm[sujeto]) +
317     len(act_ayuda[sujeto][SECUENCIA]) +
318     ##len(act_falla[sujeto][SECUENCIA]) + #La falla no es una
319     actividad del alumno
320     len(act_nopuedo[sujeto][SECUENCIA])
321 ))
322 axs[1].grid(True)
323 axs[1].set_ylim(-0.5,9.5)
324 axs[1].set_yticks(range(10))
325 axs[1].grid(linewidth = 1, color = 'dimgray')
326 axs[1].yaxis.set_ticklabels(ACT_TEXTO)
327 #axs[1].yaxis.set_ticklabels([])

```

```

321     axs[1].set_xticks(range(int(secuencia_norm[sujeto][-1])+2))
322
323     #Comentar para que aparezcan los números correlativos en el eje
324     horizontal:
325     axs[1].xaxis.set_ticklabels([])
326
327     axs[1].plot(act_ayuda[sujeto][SECUENCIA], act_ayuda[sujeto][VALOR
328     ],
329     **propiedades_ayuda,
330     )
331     for t,v in zip(act_ayuda[sujeto][SECUENCIA], act_ayuda[sujeto][
332     VALOR]):
333         axs[1].plot([t-0.5,t+0.5], [v,v], **propiedades_linea_ayuda)
334
335     axs[1].plot(act_falla[sujeto][SECUENCIA], act_falla[sujeto][VALOR
336     ],
337     **propiedades_falla,
338     )
339     for t,v in zip(act_falla[sujeto][SECUENCIA], act_falla[sujeto][
340     VALOR]):
341         axs[1].plot([t-0.5,t+0.5], [v,v], **propiedades_linea_falla)
342
343     axs[1].plot(act_nopuedo[sujeto][SECUENCIA], act_nopuedo[sujeto][
344     VALOR],
345     **propiedades_nopuedo,
346     )
347     axs[1].plot(act_resuelto[sujeto][SECUENCIA], act_resuelto[sujeto][
348     VALOR],
349     **propiedades_resuelto,
350     )
351     axs[1].plot(act_noresuelto[sujeto][SECUENCIA], act_noresuelto[
352     sujeto][VALOR],
353     **propiedades_noresuelto,
354     )
355
356     i = 0
357     while i < len(act_ayuda[sujeto][VALOR]):
358         axs[0].text(
359             act_ayuda[sujeto][TIEMPO][i],
360             act_ayuda[sujeto][VALOR][i]+0.6,
361             **propiedades_ayuda_texto,
362         )
363         axs[1].text(
364             act_ayuda[sujeto][SECUENCIA][i],
365             act_ayuda[sujeto][VALOR][i]+0.6,
366             **propiedades_ayuda_texto,
367         )
368         i += 1
369
370     i = 0
371     while i < len(act_nopuedo[sujeto][VALOR]):
372         axs[0].text(
373             act_nopuedo[sujeto][TIEMPO][i],
374             act_nopuedo[sujeto][VALOR][i]-1,

```

```

367         **propiedades_nopuedo_texto ,
368     )
369     axs[1].text(
370         act_nopuedo[sujeto][SECUENCIA][i],
371         act_nopuedo[sujeto][VALOR][i]-1,
372         **propiedades_nopuedo_texto ,
373     )
374     i += 1
375
376     fig.tight_layout()
377     #print('Guardando archivo de', sujeto); plt.savefig(sujeto+'.pdf')
378     ;
379     print('Guardando archivo de', sujeto); plt.savefig(sujeto+'.png')
380     #plt.show(); break #cuando sólo quiero ver uno

```

F. Meta-análisis cuantitativo de las Referencias Bibliográficas

Esta tesis incluye 220 referencias, de las cuales se analizó cuantitativamente:

1. El tipo de documento (artículo de revista científica, artículo de memoria de congreso científico, libro, tesis, capítulo de libro científico, etc.);
2. El año de publicación;
3. El género de los autores.

La lista de referencias está en un archivo de formato BibTeX como el siguiente ejemplo, de Buteau y cols. (2014) y de Lockwood y cols. (2016):

```
1 @article{buteau2014integration ,
2   author = "Buteau, Chantal and Jarvis, Daniel H and Lavicza, Zsolt",
3   doi = "10.1080/14926156.2014.874614",
4   journal = "Canadian Journal of Science, Mathematics and Technology
5     Education",
6   number = "1",
7   pages = "35-57",
8   publisher = "Taylor \& Francis",
9   title = "{On the integration of Computer Algebra Systems (CAS) by
10     Canadian mathematicians: Results of a national survey}",
11   volume = "14",
12   x-color = "#009966",
13   year = "2014",
14   z-gen = "f,m,m"
15 }
16
17 @inproceedings{lockwood2016algorithmic ,
18   address = "Tucson, AZ",
19   author = "Lockwood, Elise and DeJarnette, Anna F. and Asay, Autumn
20     and Thomas, Matthew",
21   booktitle = "{Proceedings of the 38th annual meeting of the North
22     American Chapter of the International Group for the Psychology
23     of Mathematics Education (PMENA 38)}",
24   editor = "Wood, M. B. and Turner, E. E. and Civil, M. and Eli, J.
25     A.",
26   pages = "1588-1595",
27   title = "{Algorithmic Thinking: An Initial Characterization of
28     Computational Thinking in Mathematics.}",
29   url = "https://files.eric.ed.gov/fulltext/ED583797.pdf",
30   x-color = "#009966",
31   year = "2016",
```

25
26

```
z-gen = "f,f,f,m"  
}
```

Además de los campos básicos propios de cada tipo de documento, se agregó el campo `x-color` para indicar que ese registro sí ha sido usado en este documento, ya que el archivo de referencias contiene registros de referencias que no se usaron, pero que en algún momento fueron considerados en la revisión bibliográfica. Es decir que los registros que no tienen este campo no fueron usados en este documento.

Además, se agregó el campo `z-gen` para indicar el género de cada uno de los autores, «f» para femenino y «m» para masculino. Hay algunos autores cuyo género no se pudo determinar, por lo que se les colocó una letra «d» (desconocido), y esos no cuentan en el análisis del género.

La figura F.1 presenta la Cantidad y proporción de referencias según cada tipo de documento. Puede verse que un poco más de la mitad de referencias son a Artículos de revistas científicas. Más de una cuarta parte son a Artículos de memorias de congresos científicos (Proceedings). El resto son Capítulos de libros científicos, tesis, libros y otros.

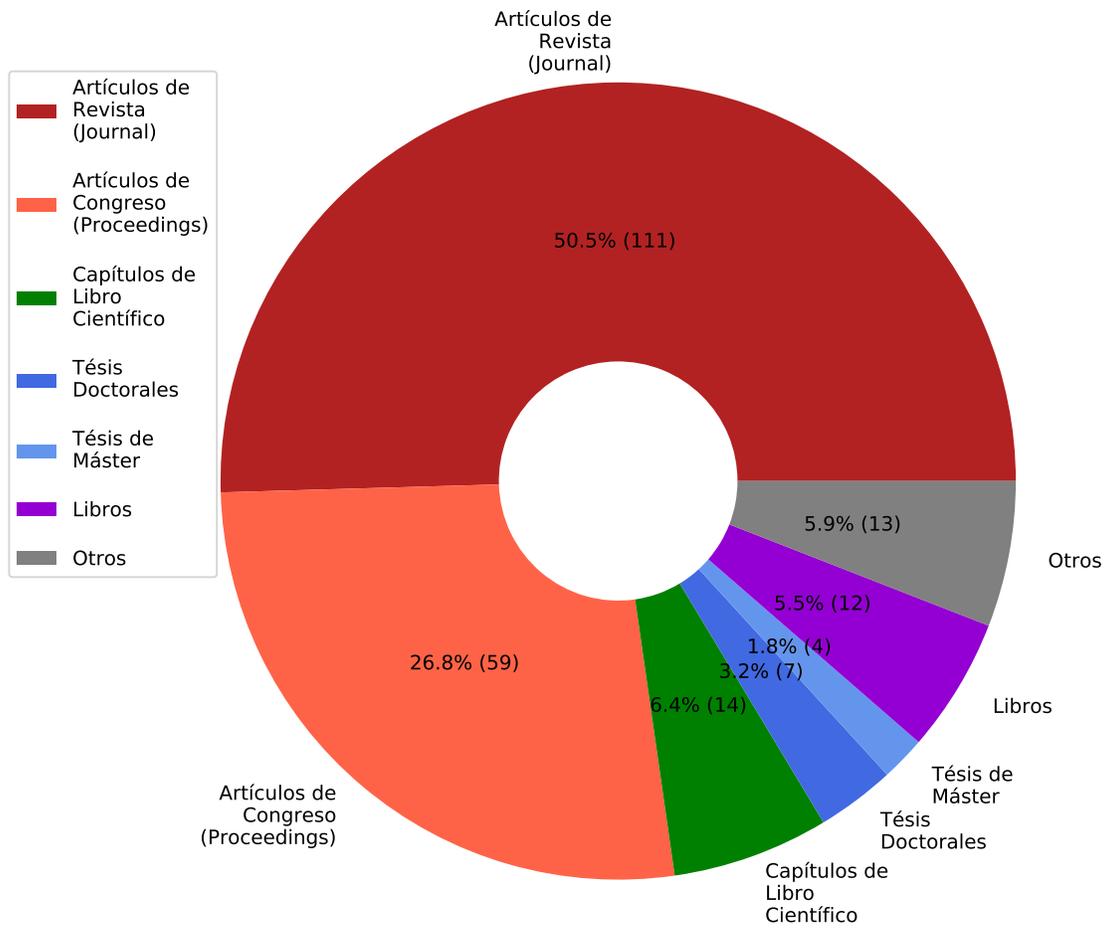


Figura F.1.: Tipos de Referencias Bibliográficas.
Fuente: Elaboración propia.

La figura E.2 muestra la Cantidad de referencias por año de publicación. Los puntos negros indican la cantidad de referencias de ese año específico. El histograma presenta la misma información excepto que agrupa las publicaciones cada tres años. La línea vertical azul larga indica la media aritmética del año de las publicaciones. Las líneas rojas cortas indican respectivamente el Cuartil 1, el Cuartil 2 (la mediana) y el Cuartil 3.

Así que se comprueba que la mitad de referencias son de 2013 o más recientes, y la cuarta parte más reciente es de a lo sumo de 2017.

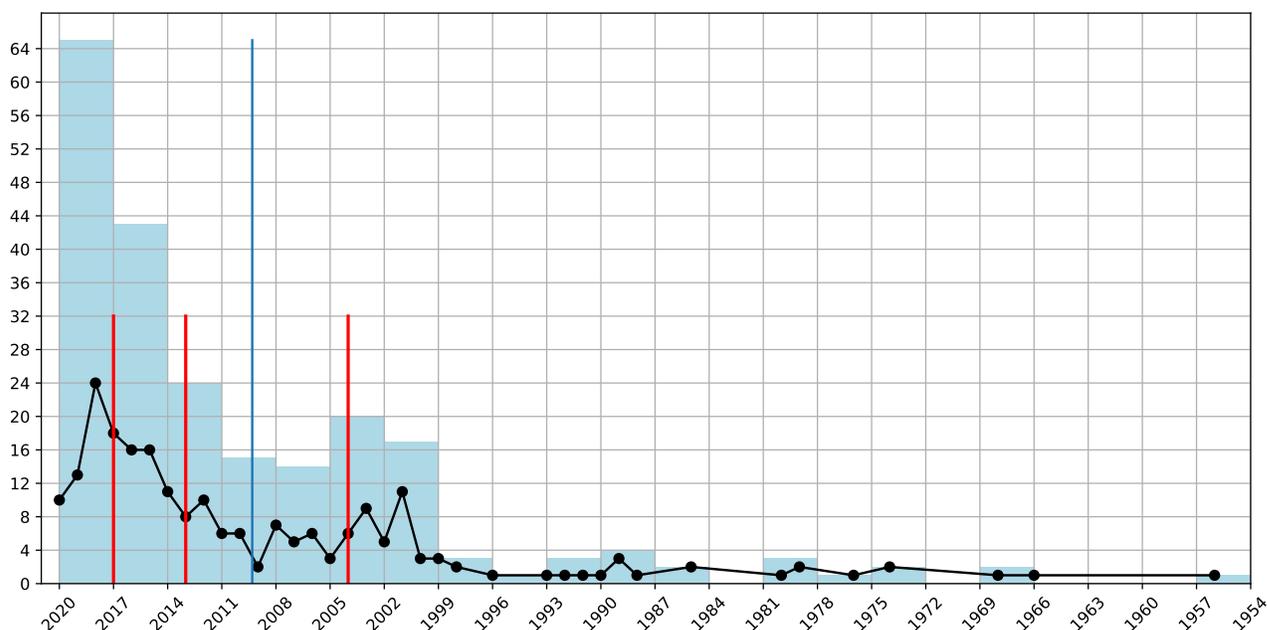


Figura E.2.: Cantidad de Referencias por año.

Fuente: Elaboración propia.

La cantidad de autores citados por año y género está representada en la figura E.3 en la página siguiente. Se intentó identificar el género de los autores de 204 referencias de las 220, especificándolo en el campo z-gen de su registro BibTeX. Los 16 que no tienen campo z-gen son autores institucionales.

En total, estos 204 documentos tienen 435 autores, de los cuales se logró identificar el género de 419, mientras que de 16 fue imposible. La búsqueda se hizo manualmente buscando fotografías de los autores en redes sociales académicas y/o en las páginas web de las instituciones educativas en las que trabajan. En el caso de la mayoría de los autores iberoamericanos y anglosajones, el género se pudo inferir a partir del nombre completo. Hubo un pequeño porcentaje de autores cuya foto no pudo ser encontrada, por lo que se aplicó la estrategia de colocar su nombre en un buscador de imágenes de la web, y si las fotos resultantes sólo se correspondían con un género, se infirió que el autor era del mismo género.

De estos 419 autores cuyo género sí se pudo identificar, 237 son hombres y 182 son mujeres, y su distribución se presenta en el gráfico de la figura E.3 en la página siguiente.

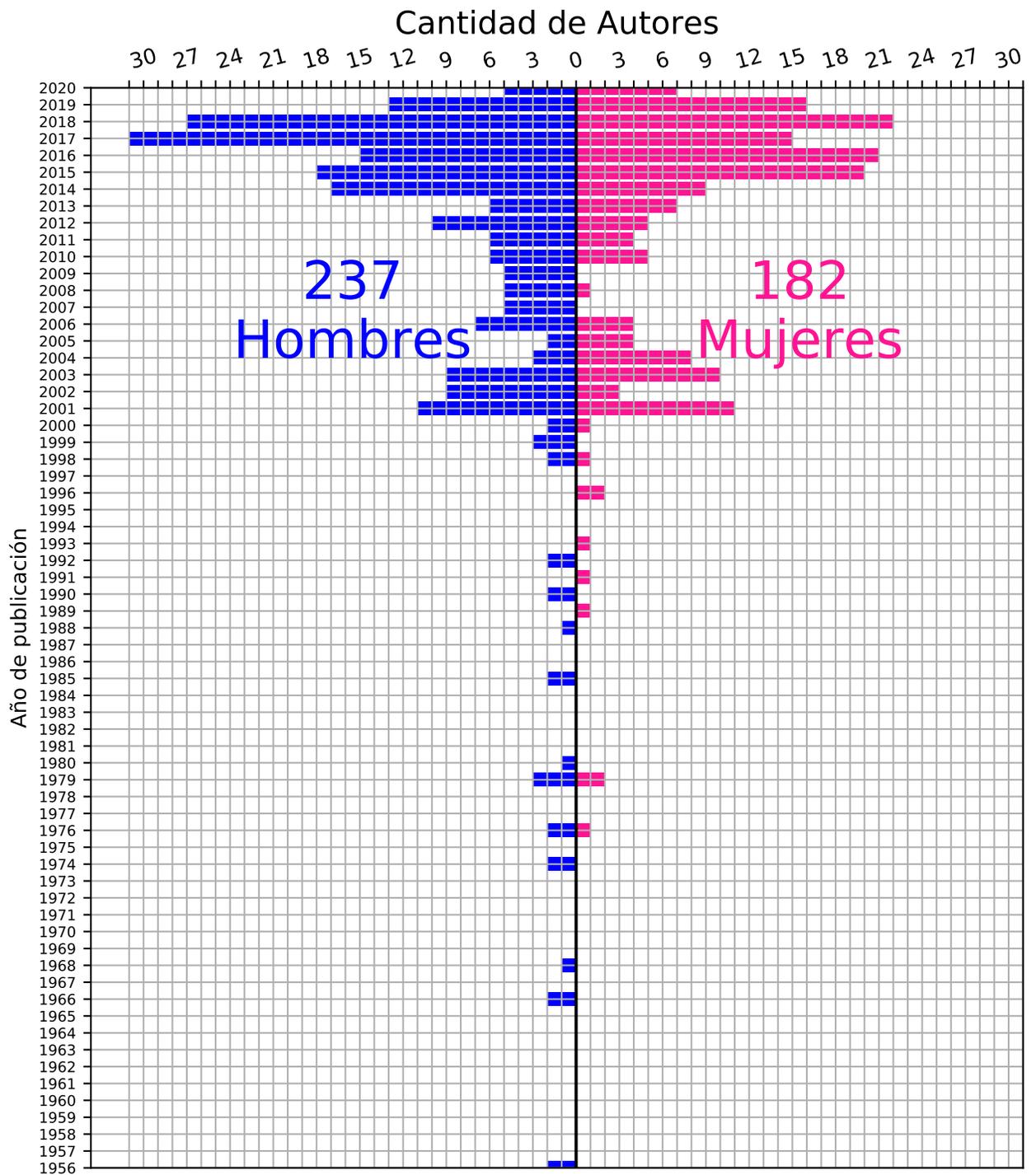


Figura E3.: Cantidad de Autores por Género por año.
Fuente: Elaboración propia.

G. Código fuente de los scripts para generar los gráficos del análisis de las Referencias Bibliográficas

Para realizar estos gráficos se implementaron los siguientes scripts en Lenguaje Python¹ v3.8.2, utilizando principalmente la biblioteca Matplotlib² diseñada por Hunter (2007), concretamente en su versión 3.1.2, de Caswell y cols. (2019).

Listado de código G.1: Script para generación de gráfico de Referencias por Tipo

```
1  #!/usr/bin/python3
2  #coding: latin1
3  """
4  Este programa dado el archivo de referencias bibliográficas genera un
5  gráfico de la cantidad de referencias por tipo.
6
7  Requiere del paquete 'bibtool'
8  """
9  import matplotlib.pyplot as plt
10 import subprocess
11 import shlex
12 import math
13
14 ARCHIVO = "../referencias.bib"
15 FILTRAR_X_COLOR = True *****
16
17 #comando para extraer entradas con campo 'x-color':
18 filtro = "bibtool - -select{x-color}" + ARCHIVO
19
20 nofiltro = "bibtool" + ARCHIVO
21
22 Tipos = [
23     "Article",
24     "InProceedings",
25     "InCollection",
26     "PhDThesis",
27     "MastersThesis",
28     "Book",
29     "Otro"
30 ]
```

¹<https://www.python.org/>

²<https://matplotlib.org/>

```

31 Tipos_es = [
32     "Artículos de \nRevista \n(Journal)",
33     "Artículos de \nCongreso \n(Proceedings)",
34     "Capítulos de \nLibro \nCientífico",
35     "Tesis \nDoctorales",
36     "Tesis de \nMáster",
37     "Libros",
38     "Otros"
39 ]
40 Colores = [ # https://matplotlib.org/stable/gallery/color/named\_colors.html
41     'firebrick',
42     'tomato',
43     'green',
44     'royalblue',
45     'cornflowerblue',
46     'darkviolet',
47     'gray',
48 ]
49
50 def func(pct, todas):
51     '''Para incluir el valor relativo y absoluto
52     en el diagrama de pastel.
53     '''
54     absoluto = int(round(pct*sum(todas)/100.))
55     return "{:1.1f}% \n({:d})".format(pct, absoluto)
56
57 cuentaTipos = dict([(t,0) for t in Tipos])
58
59
60 p = subprocess.run(shlex.split(filtro if FILTRAR_X_COLOR else nofiltro
61 ),
62     text=True, capture_output=True
63 )
64 #Toda la cadena con todo el archivo BibTeX con entradas con campo 'x-
65     color'
66 entradas=p.stdout
67 registros = entradas.split("\n@")
68 i = 0
69 while i<len(registros):
70     r = registros[i]
71     pos_llave = r.find("{")
72     if pos_llave == -1:
73         i += 1
74         continue
75     t = r[:pos_llave]
76     if t in Tipos:
77         cuentaTipos[t] += 1
78     else:
79         #print("TIPO: -{}".format(t))
80         cuentaTipos["Otro"] += 1
81         #print(i, r[:r.find(',')].strip())
82         i += 1

```

```

82
83
84 cuentas = [cuentaTipos[t] for t in Tipos]
85 cantRefs = sum(cuentas)
86 #print(cuentas)
87 print("Cantidad de Referencias:", cantRefs)
88
89 #print(cuentaTipos)
90 #print(Tipos)
91 #print(cuentas)
92
93 fig, ax = plt.subplots(
94     figsize=(10, 9),
95     #figsize=(6, 5),
96 )
97
98 wedges, texts, autotexts = plt.pie(
99     cuentas, labels=Tipos_es,
100     #shadow=True,
101     #autopct="%1.1f%%",
102     autopct=lambda pct: func(pct, cuentas),
103     wedgeprops=dict(width=0.7),
104     #explode=[0.01 for t in Tipos],
105     startangle=0,
106     #rotatelabels=True,
107     #center=(-10,0),
108     colors = Colores,
109 )
110 plt.axis('equal')
111
112 ax.legend(
113     wedges, Tipos_es,
114     #title="Tipos",
115     loc="upper left",
116     labelspacing=1.5,
117 ) # https://matplotlib.org/stable/api/_as_gen/matplotlib.axes.Axes.
    legend.html
118
119 #plt.setp(autotexts, size=8, weight="bold")
120
121 #plt.title('Tipos de Referencias', size=20)
122 fig.tight_layout()
123 #plt.savefig('refs-por-tipo.pdf')
124 plt.show()

```

Listado de código G.2: Script para generación de gráfico de Referencias por año

```

1 #!/usr/bin/python3
2 #coding: latin1
3 """
4 Este programa dado el archivo de referencias
5 bibliográficas genera un gráfico de la cantidad
6 de referencias por año.
7

```

```

8  Requiere del paquete 'bibtool'
9  """
10
11 import matplotlib.pyplot as plt
12 import subprocess
13 import shlex
14 import sys
15
16 ARCHIVO = "../referencias.bib"
17 CANT_ANNIOS = 3 #ancho del histograma, ¿3, 4 o 5?
18 FILTRAR_X_COLOR = True #*****
19
20 #comando para extraer entradas con campo 'x-color':
21 filtro = "bibtool --select{x-color}" + ARCHIVO
22
23 nofiltro = "bibtool" + ARCHIVO
24
25 listaAños = {}
26 todosAños = []
27
28 p = subprocess.run(shlex.split(filtro if FILTRAR_X_COLOR else nofiltro
29 ),
30 text=True, capture_output=True
31 )
32 #Toda la cadena con todo el archivo BibTeX con entradas con campo 'x-
33 color'
34 entradas=p.stdout
35 registros = entradas.split("\n@")
36 contYear=0
37 i = 0
38 while i<len(registros):
39     r = registros[i]
40     bloques = r.split(",")
41     #print(i, bloques[0])
42     año = 0
43     for bloque in bloques:
44         try:
45             nombre, valor = bloque.split("=")
46         except:
47             continue
48         nombre = nombre.strip()
49         try:
50             valor = valor.split('"')[1]
51         except:
52             continue
53
54         if nombre=="year":
55             contYear += 1
56             valor = int(valor)
57             año = valor
58             todosAños.append(valor)
59             if valor in listaAños:
60                 listaAños[valor] += 1

```

```

60         else:
61             listaAnnios[valor] = 1
62     if annio == 0 and len(bloques)>1:
63         print("ESTE REGISTRO NO TIENE AÑO (delimitado por comillas
64             dobles):", bloques[0], file=sys.stderr)
65     i += 1
66     #print(contYear)
67     #print(listaAnnios)
68     #Listas para el gráfico de líneas ++++++
69     annios = sorted(listaAnnios)
70     annios.reverse()
71     x = []
72     y = []
73     cantRefs = 0
74     for a in annios:
75         x.append(a)
76         y.append(listaAnnios[a])
77         cantRefs += listaAnnios[a]
78     if (contYear!=cantRefs):
79         raise Exception("No salen las cuentas")
80     print("Cantidad de Referencias:", cantRefs)
81
82     #Histograma: ++++++
83     #Se crea al revés para garantizar que el último año sea el mayor
84     divisiones = list(range(max(todosAnnios), min(todosAnnios)-CANT_ANNIOS
85         , -CANT_ANNIOS))
86     #Averguar el máximo valor de frecuencia del gráfico, para que la
87     #escala sea siempre entera
88     maxFreqAnnio = 0
89     i=len(divisiones)-1
90     while i > 0:
91         cant = 0
92         #print(list(range(divisiones[i], divisiones[i-1])), i) #no incluye
93         #el último año
94         for a in range(divisiones[i],divisiones[i-1]):
95             if a in listaAnnios:
96                 cant += listaAnnios[a]
97         if i==1 and divisiones[0] in listaAnnios:
98             cant += listaAnnios[divisiones[0]]
99         if maxFreqAnnio < cant:
100             maxFreqAnnio = cant
101         i -= 1
102     #print(maxFreqAnnio)
103     #print(divisiones)
104     #Es necesario revertirlo para el parámetro bins del histograma
105     divisiones.reverse()
106
107     fig,ax = plt.subplots(figsize=(12,7))
108     plt.axes(
109         xlim=(divisiones[-1]+1,divisiones[0]),

```

```

110     xticks=divisiones,
111     yticks=list(map(lambda x: 4*x, range((maxFreqAnio+4)//4))),
112     #yticks=range(maxFreqAnio+1),
113 )
114 plt.xticks(rotation=45)
115 plt.grid() #activar la cuadrícula
116
117 plt.plot(x,y,marker='o', color='black')
118 plt.hist(todosAnios, bins=divisiones,
119         color='lightblue',
120         #orientation='horizontal',
121 )
122 tantosAnios = len(todosAnios)
123 media = sum(todosAnios)/tantosAnios
124 plt.plot([media,media],[0,maxFreqAnio])
125 todosAnios.sort()
126 cuartiles = dict(
127     color='red',
128     linewidth=2,
129 )
130 plt.plot([todosAnios[len(todosAnios)//4]*2,[0,maxFreqAnio//2], #
131         Q1
132         **cuartiles)
133 plt.plot([todosAnios[len(todosAnios)//2]*2,[0,maxFreqAnio//2], #
134         mediana
135         **cuartiles)
136 plt.plot([todosAnios[len(todosAnios)*3//4]*2,[0,maxFreqAnio//2], #
137         Q3
138         **cuartiles)
139
140 fig.tight_layout()
141 #plt.title('Cantidad de Referencias por año', size=20)
142 #plt.savefig('refs-por-annio.pdf')
143 plt.show()

```

Listado de código G.3: Script para generación de gráfico de Autores por Género

```

1  #!/usr/bin/python3
2  #coding: latin1
3  """
4  Este programa dado el archivo de referencias bibliográficas
5  genera un gráfico de la cantidad de referencias por año y
6  por género.
7
8  Requiere del paquete 'bibtool'
9  """
10
11 import matplotlib.pyplot as plt
12 import subprocess
13 import sys
14 import shlex
15
16 ARCHIVO = "../referencias.bib"
17 CAMPO_GENERO = 'z-gen'

```

```

18 FILTRAR_X_COLOR = True #*****
19
20 #comando para extraer entradas con campo 'x-color':
21 filtro1 = "bibtool --select{x-color}" + ARCHIVO
22
23 nofiltro = "bibtool" + ARCHIVO
24
25 #comando para extraer entradas con campo de género
26 filtro2 = "bibtool --select{" + CAMPO_GENERO + "}"
27
28 listaAnniosH = {}
29 listaAnniosM = {}
30 todosAnnios = []
31
32 #Filtro de color
33 p1 = subprocess.run(shlex.split(filtro1 if FILTRAR_X_COLOR else
34     nofiltro),
35     text=True, capture_output=True
36 )
37 #Filtro de género
38 p2 = subprocess.run(shlex.split(filtro2), text=True, capture_output=
39     True, input=p1.stdout)
40 #Toda la cadena con todo el archivo BibTeX con entradas con campo 'x-
41     color' y 'z-gen'
42 entradas=p2.stdout
43 #print(entradas)
44
45 registros = entradas.split("\n@")
46 cantConGen = 0
47 autoresSinGen = 0
48 for r in registros:
49     pos_llave = r.find("{")
50
51     bloques = r.split("=")
52     pos_coma = bloques[0].rfind(',')
53     nombre_registro = r[pos_llave+1:pos_coma].strip()
54     bloques[-1] += ',' #para que todos los bloques tengan la misma
55     estructura
56
57     annio = 0
58     generos=[]
59     i = 1
60     while i < len(bloques):
61         try:
62             nombre = bloques[i-1][bloques[i-1].rfind(',')+1:].strip()
63             valor = bloques[i][:bloques[i].rfind(',')].split('"')[1]
64             #Tanto el valor del campo 'year' como el del género
65             #deben estar delimitados por comillas dobles
66         except:
67             #print(nombre)
68             #print(valor)
69             i += 1

```

```

68         continue
69
70     if nombre=="year":
71         annio = int(valor)
72     if nombre==CAMPO_GENERO:
73         generos = valor.split(',')
74
75     i += 1
76
77     if annio!=0 and generos!=[]: #Se encontraron los dos campos
78         cantConGen += 1
79         #print(nombre_registro)
80         if annio not in todosAnnios:
81             todosAnnios.append(annio)
82         if annio not in listaAnniosM:
83             listaAnniosM[annio] = 0
84         if annio not in listaAnniosH:
85             listaAnniosH[annio] = 0
86         for g in generos:
87             if g.lower()=='f':
88                 listaAnniosM[annio] +=1
89             elif g.lower()=='m':
90                 listaAnniosH[annio] +=1
91             else:
92                 autoresSinGen += 1
93                 print("Género indefinido", g,"», en", r[:pos_llave],
94                       nombre_registro, file=sys.stderr)
95                 #print(r[:pos_llave])
96                 #print(generos)
97
98     #print(listaAnniosH)
99     #print(listaAnniosM)
100     print("Cantidad de Referencias con género:", cantConGen)
101     print("Cantidad de autores sin género identificado:", autoresSinGen)
102
103     #cantAuts = sum([listaAnniosH[a]+listaAnniosM[a] for a in todosAnnios
104                    ])
105     #print(cantAuts)
106
107     #Listas para el gráfico de barras ++++++
108     cantMujeres = []
109     cantHombres = []
110     cantAuts = 0
111     maxAutsPorAnno = 0
112     for a in todosAnnios:
113         cantMujeres.append(listaAnniosM[a])
114         cantHombres.append(-listaAnniosH[a])
115         cantAuts += (listaAnniosM[a]+listaAnniosH[a])
116         # print(a)
117         # print(listaAnniosM[a])
118         # print(listaAnniosH[a])
119         # print(cantAuts)
120
121     if maxAutsPorAnno < cantMujeres[-1]:

```

```

120         maxAutsPorAnnio = cantMujeres[-1]
121     if maxAutsPorAnnio < listaAñosH[a]:
122         maxAutsPorAnnio = listaAñosH[a]
123     print("Cantidad de Autores con género identificado:", cantAuts)
124     print("Cantidad de Hombres:", -sum(cantHombres))
125     print("Cantidad de Mujeres:", sum(cantMujeres))
126
127     #-----
128     fig,ax = plt.subplots(figsize=(7,8))
129     ax.barh(todosAños, cantMujeres,
130            align='center',
131            color='deeppink',
132            ecolor='black'
133            )
134     ax.barh(todosAños, cantHombres,
135            align='center',
136            color='blue',
137            ecolor='black'
138            )
139     ax.plot([0,0],[min(todosAños),max(todosAños)], color='black')
140     ax.set_yticks(list(range(min(todosAños),max(todosAños)+1)))
141     plt.yticks(fontsize='x-small')
142     plt.xticks(fontsize='small')
143     eje = list(range(-maxAutsPorAnnio,maxAutsPorAnnio+1))
144     ax.set_xticks(eje)
145     ax.set_xticklabels([str(abs(x)) if x%3==0 else '' for x in eje])
146     ax.xaxis.set_ticks_position('top')
147     plt.xticks(rotation=15)
148     #ax.set_xlabel('Autores',size=15, ) #Lo pone abajo
149     ax.set_ylabel('Año de publicación', size=10, )
150     plt.ylim(min(todosAños),max(todosAños))
151     ax.grid(True)
152     pos_texto = maxAutsPorAnnio/2
153     plt.text(pos_texto, 2004, #podría ser sum(todosAños)/len(
154            todosAños),
155            str(sum(cantMujeres))+'\nMujeres',
156            color='deeppink',
157            size=25,
158            horizontalalignment='center'
159            )
160     plt.text(-pos_texto, 2004, #podría ser sum(todosAños)/len(
161            todosAños),
162            str(-sum(cantHombres))+'\nHombres',
163            color='blue',
164            size=25,
165            horizontalalignment='center'
166            )
167     #-----
168     plt.title('Cantidad de Autores', size=15)
169     fig.tight_layout()
170     plt.savefig('refs-por-gen.pdf')
171     #plt.show()

```

H. Software Utilizado

Esta es una lista no exhaustiva del software utilizado para realizar esta investigación, este documento, y su correspondiente presentación:

- Editores principales:
 - LyX, v2.3.4.2, <https://www.lyx.org/>
 - LibreOffice, v6.4.5.2, <https://www.libreoffice.org/>
 - KBibTex, v0.8.1, <https://userbase.kde.org/KBibTeX>
 - Xournal, v0.4.8.2016, <http://xournal.sourceforge.net>
- Lenguajes de programación:
 - Python, v3.8.2, <https://www.python.org/>, con Matplotlib, v3.1.2, <https://matplotlib.org/>
 - R, v3.6.3, <https://www.r-project.org/>
- IDE:
 - Geany, v1.36, <https://www.geany.org/>
- Editor de archivos de texto plano:
 - xed, v2.6.2, <https://github.com/linuxmint/xed>
- \LaTeX :
 - TeX, v3.14159265
 - TeXLive, v2019.20200218, <http://www.tug.org/texlive/>
- Renderizador de la presentación:
 - Hovercraft!, v2.8, <https://github.com/regebro/hovercraft>
 - Impress.JS, <https://github.com/impress/impress.js/>
 - reStructuredText, <https://docutils.sourceforge.io/rst.html>
- Visor de archivos PDF:
 - Xreader, v2.6.4, <https://github.com/linuxmint/xreader/>
- Navegadores web:
 - Mozilla Firefox, v80.0.1, <https://www.mozilla.org/es-ES/>
 - Chromium, v92.0.4515.13, <https://www.chromium.org/Home>
- Manipulación de imágenes:

- ImageMagick, v6.9.10-23, <https://imagemagick.org/>
- GIMP, v2.10.18, <https://www.gimp.org/>
- Shotwell, v0.30.10, <https://wiki.gnome.org/Apps/Shotwell>
- Todo corriendo sobre Linux Mint 20 (Ulyana), <https://linuxmint.com/>