

UNIVERSIDAD DE EL SALVADOR  
FACULTAD MULTIDISCIPLINARIA ORIENTA  
DEPARTAMENTO DE INGENIERÍA Y ARQUITECTURA



**TRABAJO DE GRADO:**

DESARROLLO DE UN SISTEMA INFORMÁTICO DE REGISTRO, EMISIÓN E  
INSPECCIÓN DE CREDENCIALES SANITARIAS OTORGADAS A LOS  
MANIPULADORES DE ALIMENTOS EN LA UNIDAD COMUNITARIA DE SALUD  
FAMILIAR DE SAN FRANCISCO GOTERA EN EL DEPARTAMENTO DE  
MORAZÁN.

**PRESENTADO POR:**

DUGLAS ENRIQUE DÍAZ BARAHONA  
LUIS FERNANDO HERNÁNDEZ CASTILLO  
ERICK ADALBERTO LÓPEZ JOYA  
MARÍA YANCI MARTÍNEZ GARCÍA

**PARA OPTAR AL TÍTULO DE:**

INGENIERO DE SISTEMAS INFORMÁTICOS

**DOCENTE ASESOR:**

ING. MDU. JULIO CÉSAR ARIAS GUEVARA

CIUDAD UNIVERSITARIA, ENERO 2021  
SAN MIGUEL, EL SALVADOR, CENTRO AMÉRICA

**UNIVERSIDAD DE EL SALVADOR**

**AUTORIDADES**

**RECTOR:**

Msc. ROGER ARMANDO ARIAS

**VICERECTOR ACADÉMICO:**

PhD. RAÚL ERNESTO AZCUNAGA LOPEZ

**VICERECTOR ADMINISTRATIVO:**

ING. JUAN ROSA QUINTANILLA

**SECRETARIO GENERAL:**

ING. FRANCISCO ALARCON

**FISCAL GENERAL:**

LIC. RAFAEL HUMBERTO PEÑA MARIN

**DEFENSOR DE LOS DERECHOS UNIVERSITARIOS:**

LIC. LUIS ANTONIO MEJIA LIPE

**FACULTAD MULTIDISCIPLINARIA ORIENTAL**

**AUTORIDADES**

**DECANO:**

LIC. CRISTOBAL HERNAN RÍOS BENÍTEZ

**VICEDECANO:**

LIC. OSCAR VILLALOBOS

**SECRETARIO INTERINO:**

LIC. ISRAEL LÓPEZ MIRANDA

**DIRECTOR GENERAL DE PROCESOS DE GRADUACIÓN:**

LIC. JORGE PASTOR FUENTES CABRERA

**DEPARTAMENTO DE INGENIERÍA Y ARQUITECTURA**

**AUTORIDADES**

**JEFE DEL DEPARTAMENTO:**

ING. JUAN ANTONIO GRANILLO COREAS

**COORDINADORA DE LA CARRERA DE INGENIERÍA DE SISTEMAS**

**INFORMÁTICOS:**

ING. LIGIA ASTRID HERNÁNDEZ BONILLA

**COORDINADORA DE PROCESOS DE GRADUACIÓN:**

ING. MILAGRO DE MARÍA ROMERO DE GARCÍA.

**TRIBUNAL EVALUADOR**

**JURADO ASESOR:**

ING. MDU. JULIO CÉSAR ARIAS GUEVARA

**DOCENTE JURADO CALIFICADOR:**

ING. LIGIA ASTRID HERNÁNDEZ BONILLA

**DOCENTE JURADO CALIFICADOR:**

ING. HECTOR JAVIER PAIZ RAMOS

## AGRADECIMIENTOS

*En primer lugar, agradecer a Dios, por bendecirme siempre a lo largo de este proceso académico y de la misma forma en mi vida en general, por permitirme gozar de buena salud y por haber puesto en mi camino excelentes amistades y así poder hacerle frente a cada dificultad y cada reto que se me ha presentado y nunca rendirme.*

*Agradezco a mis padres, Rosa Barahona y Jaime Díaz, y a mi hermana Lorena Díaz, quienes siempre han estado ahí brindándome su apoyo incondicional en cada etapa de mi vida, siempre han creído en mí y sin ellos no habría podido alcanzar este tan anhelado logro.*

*Agradezco a mis abuelas María Barahona y Cleotilde Díaz, ambas me han apoyado en cada una de mis decisiones, aunque en algunas no siempre estaban de acuerdo, pero sabían que era por mi bien; ellas me han brindado su amor y cariño incondicional, han sido un pilar fundamental en mi vida, han sido como una madre para mí, siempre agradeceré a Dios por bendecirme con tenerlas.*

*Agradezco a mis grandes amigos y compañeros Yanci Martínez y Luis Hernández, ellos son como hermanos para mí, nos conocimos básicamente desde que comenzamos nuestra carrera, a pesar de todas nuestras locuras siempre nos hemos apoyado mutuamente, nunca dejaron que me diera por vencido, y sabíamos que íbamos a llegar a este día tan importante, el día que culmina una gran etapa de nuestras vidas, para darle inicio a otra, y con fe en Dios que seguiremos siendo amigos y teniendo grandes momentos como los que hemos tenido hasta hoy. También agradezco a mi compañero Erick López, que se nos unió en este proceso y a pesar de todo siempre nos ha colaborado.*

*Agradezco a todos mis docentes especialmente a nuestro asesor de tesis, Ingeniero Julio Cesar Arias, siempre apoyándonos y guiándonos a lo largo de este proceso de tesis y a lo largo de nuestra carrera y que ha compartido su gran conocimiento para culminar esta etapa académica de una manera exitosa.*

***Duglas Diaz***

## AGRADECIMIENTOS

*Agradezco principalmente a Dios y mis padres Berta Castillo y Nelson Hernández y a mi familia, a mis compañeros y amigos de tesis Erick López, Duglas Díaz y en especial a mi novia Yanci Martínez por siempre animarme y apoyarme, para cada uno de ustedes por ser parte de este proceso además de brindarme su hospitalidad, humildad y respeto.*

*Agradezco a Wilber Méndez por tu amistad y haberme ayudado a comprender las artes de la programación. Doy agradecimientos especiales a Jonathan centeno, Roberto Carlos, Rebeka Díaz, Carlos González, Fátima Arias, Wilber Aguilar, Jeremías Luna, Maricela Moreira, Cristina Rivera, Delmis, Gilberto, también para los Ingenieros, Licenciados y profesores; mis buenos deseos para ustedes y muchas gracias por todo.*

*Luis Hernández*

## AGRADECIMIENTOS

*Primero que nada quiero darle gracias a Dios por que sin él nada hubiese sido posible, ya que el ha derramado muchas bendiciones en mi vida y ha estado conmigo a lo largo de toda la carrera. A la virgen María por haber intercedido por mi y para que pueda culminar mi carrera.*

*A mi mamá Miriam Elizabeth Joya de López quien me ha apoyado desde el principio y hasta hoy en día, ha estado conmigo en todo momento motivándome, dándome fuerzas y esperanzas para seguir. Ella ha dado todo para que yo pueda cumplir mi sueño, siempre creyó en mi y se esforzó para que yo pueda estar aquí.*

*A mi papá Juan José López Gomez quien también me apoyo mucho durante toda mi carrera motivándome y alentándome a seguir. Trabando duro para que yo pueda terminar mis estudios y mis hermanos Walter, Juan y Katherine que me ayudaron muchas veces ya sea con consejos o como motivación para lograr culminar mis estudios.*

*A mis compañeros de tesis, Yanci Martínez, Luis Hernández y Duglas Díaz quienes me ayudaron mucho en mi proceso de graduación ya que me motivaron a seguir y me apoyaron en los momentos más difíciles.*

*A todos mis compañeros y amigos que han aportado de una u otra forma en mi persona a lo largo de todos los años de la carrera para que yo pueda lograr llegar hasta donde estoy.*

*Agradezco también a todos los docentes que han contribuido en mi formación a lo largo de todos los ciclos. A nuestro docente asesor el Ing. Julio Arias, también a los docentes del jurado observador la Ing. Ligia Hernández y el Ing. Hector Paíz.*

*También quiero agradecer a mi hija Sofía Valentina quien me ha servido de motivación en este año tan duro que ha sido el 2020, ella me ha regalado un rayo de luz y esperanza para seguir.*

***Erick Joya***

## AGRADECIMIENTOS

*Primera mente darle gracias a Dios por bendecirme y haberme permitido llegar hasta aquí y a mi mamá Consuelo García por haberme apoyado y ayudado siempre ya que sin su ayuda nunca hubiese logrado llegar tan lejos gracias a mi familia por estar siempre ahí cuando necesite su ayuda.*

*Agradezco en especial a mi novio Luis Hernández que ha sido mi mejor amigo compañero de alegrías y tristezas mi salvación cuando todo parecía no tener arreglo y por su ayuda incondicional y apoyo también tengo mucho que agradecer a mi gran amigo que es como un hermano para mí Douglas Díaz hemos estado en las buenas y en las malas para poder lograr lo que un día pareció algo inalcanzable pero gracias a Dios logramos terminar nuestros estudios.*

*Agradezco a Erick López por ayudarnos en la realización de nuestra tesis y nuestro asesor Ing. Julio Cesar Arias ya que siempre estuvo brindando su ayuda en el transcurso de todos estos años y a todos los Ingenieros y Licenciados gracias por impartir sus conocimientos y también gracias a todas aquellas personas que me brindaron su ayuda cuando lo necesite.*

**Yanci Martínez**

## RESUMEN

En este documento se presenta el proceso de desarrollo de un sistema informático para el área de saneamiento ambiental en la Unidad Comunitaria de Salud Familiar de San Francisco Gotera, donde se dan a conocer los datos de interés sobre la institución, para conocer el impacto que esta herramienta tendrá sobre el manejo automatizado de los procesos en el área de saneamiento ambiental específicamente en la manipulación de alimentos.

Se abordan de forma general los beneficios que proporcionara indirectamente a la población; además se dan conocer las herramientas que se utilizaron para el desarrollo del software que se adaptan a las necesidades señaladas del área de Saneamiento Ambiental. Se dan a conocer las factibilidades para la implementación de este Sistema Informático, así como también se detallan cada una de las características con las que contará y las tareas que podrá realizar.

Se presentan las interfaces gráficas para tener una visión clara del diseño del sistema informático, también se incluye los manuales de usuario, de instalación y de programación para tener una documentación completa de su funcionamiento, que coadyuven con las modificaciones que se requieran a futuro y que estas puedan ser realizadas por cualquier otro programador.

## ABSTRAC

In this document is introduced the process of developing a computer system for the area of environmental sanitation at La Unidad Comunitaria de Salud Familiar de San Francisco Gotera, where is introduced the data of interest about the institution, in order to know the impact that this tool will have on the automated management of processes in the area of environmental sanitation specifically in food handling.

The benefits that it will indirectly provide to the population are addressed in a general way; In addition, the tools that were used for the development of the software that adapt to the indicated needs of the Environmental Sanitation area are introduced as well. The feasibilities for the implementation of this Computer System are introduced, as well as each of the characteristics that it will have and the tasks that it will be able to perform are also detailed.

The graphic interfaces are presented to have a clear vision of the design of the computer system, the user, installation and programming manuals are also included to have a complete documentation of its operation, which will contribute to the modifications that are required in the future and that these can be done by any other programmer

# Índice general

<b>Introducción</b>	<b>I</b>
<b>1. Estudio Preliminar</b>	<b>1</b>
1.1. Definición del Tema . . . . .	1
1.2. Descripción de la Institución . . . . .	1
1.2.1. Unidad Comunitaria de Salud Familiar de San Francisco Gotera . . . . .	1
1.2.2. Organigrama de la institución . . . . .	5
1.2.3. Croquis de ubicación del proyecto . . . . .	6
1.3. Situación Problemática . . . . .	7
1.4. Enunciado del problema . . . . .	9
1.5. Objetivos . . . . .	9
1.5.1. Objetivo General . . . . .	9
1.5.2. Objetivos Específicos . . . . .	9
1.6. Justificación . . . . .	10
1.7. Alcances . . . . .	12
1.8. Limitaciones . . . . .	13
1.9. Delimitación . . . . .	13
1.9.1. Espacial . . . . .	13
1.9.2. Temporal . . . . .	13
<b>2. Marco de Referencia</b>	<b>14</b>
2.1. Marco Histórico . . . . .	14
2.1.1. Origen del Ministerio de Salud . . . . .	14
2.2. Marco Contextual . . . . .	17
2.2.1. Condiciones del sistema de salud de El Salvador . . . . .	17
2.2.2. Estructura y Cobertura . . . . .	17
2.2.3. Recurso Humano del MINSAL . . . . .	18
2.2.4. Manipulación de Alimentos en El Salvador . . . . .	20
2.2.5. Capacitaciones sobre higiene de alimentos . . . . .	21
2.3. Marco Teórico . . . . .	22
2.3.1. Aplicaciones Web o Webbapps . . . . .	22
2.3.2. Ventajas de la aplicaciones web . . . . .	23
2.3.3. Aplicación Móvil . . . . .	23
2.3.3.1. Desarrollo de aplicaciones móviles. . . . .	24
2.3.3.2. Desarrollo de software. . . . .	24
2.3.3.3. Tipos de aplicaciones móviles. . . . .	25
2.3.3.4. Características de las aplicaciones móviles. . . . .	26

2.3.4.	<b>Estratetegias de prueba para webapps</b>	30
2.3.5.	Internet	34
2.3.6.	Web	35
2.3.7.	URL	35
2.3.8.	Servidor Web	36
2.3.9.	Cliente Web	37
2.3.10.	Sistema Operativo Android	37
2.3.11.	Lenguaje Unificado de Modelado (UML)	38
2.3.12.	Modelado de caso de uso	38
2.3.12.1.	Elementos principales del diagrama de caso de uso	39
2.3.13.	Modelo Entidad-Relación	46
2.3.14.	Diagrama de despliegue	50
2.3.14.1.	Ventajas y Desventajas	50
2.3.15.	Diagrama de Secuencia	51
2.3.15.1.	Principales elementos del diagrama de secuencia	52
2.3.16.	Estimación de costo de un sistema de software	56
2.3.16.1.	Método punto función	56
2.3.16.2.	Utilidades del método de puntos de función	57
2.3.16.3.	Método de estimación por Puntos de Casos de Uso	58
2.3.17.	Linode	59
2.4.	Herramientas para el desarrollo del sistema	61
2.4.1.	Sistema operativo requerido para Servidor Web	61
2.4.2.	Sistema operativo requerido para Cliente Web	61
2.4.3.	HTML	62
2.4.4.	CSS	62
2.4.5.	JavaScript	62
2.4.6.	PHP	63
2.4.7.	MVC	63
2.4.8.	GitHub	64
2.4.9.	Visual Studio Code	65
2.4.10.	PHPMyAdmin	66
2.5.	Herramientas para el desarrollo de la aplicación móvil	68
2.5.1.	Flutter	68
2.5.2.	Dart	68
2.5.3.	Android SDK	70
2.5.4.	Android Studio	70
2.5.5.	Firebase	71
2.5.6.	Patrones de diseño	72
2.6.	Marco legal	75
2.6.1.	Artículos relacionados a la manipulación de alimentos	75
2.6.1.1.	Educación sanitaria	75
2.6.1.2.	Salud del manipulador	75
2.6.2.	Ley de propiedad intelectual	76
2.6.3.	Licencia de Software	77
2.6.4.	Código fuente	77
2.7.	Marco Conceptual	79

### 3. Metodología de la Investigación

3.1. Metodología de la Investigación . . . . .	86
3.1.1. Investigación descriptiva . . . . .	86
3.1.2. Investigación Tecnológica . . . . .	87
3.2. Metodología de Desarrollo . . . . .	88
3.3. Población y muestra . . . . .	91
3.3.1. Población . . . . .	91
3.3.2. Método de muestreo . . . . .	91
3.3.3. Muestra en cadena o por redes . . . . .	92
3.4. Técnicas e instrumentos de investigación . . . . .	93
3.5. Técnicas e instrumentos para el análisis de los datos . . . . .	95
3.5.1. Procedimiento para la validación de los datos . . . . .	95
3.5.2. Procedimiento de recolección de datos . . . . .	95
3.5.3. Procedimientos para procesar e interpretar los datos . . . . .	97
3.6. Análisis e interpretación de resultados . . . . .	98
3.6.1. Tabulación de la encuesta . . . . .	98
3.6.2. Entrevistas . . . . .	107
<b>4. Desarrollo del software . . . . .</b>	<b>109</b>
4.1. Determinación de requerimientos . . . . .	109
4.1.1. Requerimientos funcionales . . . . .	109
4.1.2. Requerimientos no funcionales . . . . .	113
4.2. Estimación de esfuerzo por medio del método Puntos de Casos de Uso . . . . .	115
4.2.1. Cálculo del esfuerzo . . . . .	119
4.3. Estudio de Factibilidad . . . . .	122
4.3.1. Factibilidad Técnica . . . . .	122
4.3.2. Factibilidad Financiera . . . . .	128
4.3.3. Factibilidad Operativa . . . . .	138
4.4. Diseño del sistema informático . . . . .	140
4.4.1. Diagrama de Casos de uso . . . . .	140
4.4.1.1. Súper usuario . . . . .	140
4.4.1.2. Administrador . . . . .	141
4.4.1.3. Manipuladores . . . . .	142
4.4.2. Narración de casos de uso . . . . .	143
4.4.3. Diagrama de Secuencia . . . . .	154
4.4.3.1. Diagrama de secuencias inicio de sesión . . . . .	156
4.4.3.2. Interfaz de Inicio de sesión . . . . .	157
4.4.3.3. Diagrama de secuencias procesos de usuario . . . . .	158
4.4.3.4. Interfaz de Usuario . . . . .	159
4.4.4. Diagrama de Secuencia Reporte . . . . .	163
4.4.4.1. Interfaz de Reportes . . . . .	164
4.4.4.2. Diagrama de secuencia proceso de asistencia . . . . .	165
4.4.4.3. Interfaz de Asistencia . . . . .	166
4.4.4.4. Diagrama de secuencia proceso de inspección . . . . .	169
4.4.4.5. Interfaz de Inspección . . . . .	170
4.4.4.6. Diagrama de secuencia proceso de Bitácora . . . . .	174
4.4.4.7. Interfaz de bitácora . . . . .	175
4.4.4.8. Diagrama de secuencia proceso de Credenciales . . . . .	176
4.4.4.9. Interfaz de Credenciales . . . . .	177

4.4.4.10.	Diagrama de secuencia proceso de Manipuladores . . . . .	179
4.4.4.11.	Interfaz de Manipuladores . . . . .	180
4.4.4.12.	Diagrama de secuencia proceso de Establecimientos . . . . .	184
4.4.4.13.	Interfaz de Establecimientos . . . . .	185
4.4.4.14.	Diagrama de secuencia proceso de Exámenes . . . . .	189
4.4.4.15.	Interfaz de Exámenes . . . . .	190
4.4.5.	Diagrama de Clases . . . . .	193
4.4.6.	Diagramas de la base de datos . . . . .	194
4.4.6.1.	Diagrama Entidad - Relación (ER) . . . . .	196
4.4.6.2.	Diagrama Físico . . . . .	197
4.4.6.3.	Diagrama Lógico . . . . .	198
4.4.6.4.	Diagrama Conceptual . . . . .	199
4.5.	Diccionario de datos . . . . .	200
4.5.1.	Uso del diccionario de datos . . . . .	200
4.5.2.	Diccionario de la base de datos UCSF . . . . .	201
4.6.	Diseño de la aplicación Móvil . . . . .	214
4.6.1.	Aplicación móvil de los Inspectores de Saneamiento Ambiental de la UCSF. . . . .	214
4.6.1.1.	Diagrama de secuencias de la aplicación móvil para los Inspectores . . . . .	217
4.6.2.	Aplicación Móvil de los Manipuladores de Alimentos de San Francisco Gotera Morazán . . . . .	218
4.6.2.1.	Diagrama de secuencias de la aplicación móvil para los manipuladores . . . . .	218
4.7.	Metodología para la programación . . . . .	220
4.7.1.	Programación orientada a objetos . . . . .	220
4.8.	Estándar de desarrollo . . . . .	223
4.8.1.	Estándar de programación . . . . .	223
4.8.1.1.	PSR-1.Codificación estándar básica . . . . .	223
4.8.1.2.	PSR-2 y PSR-12. Guía de estilos de codificación . . . . .	224
4.8.2.	Estructura de ficheros utilizando modelo, vista controlador . . . . .	231
4.8.3.	REST API . . . . .	235
4.8.4.	Estándares de base de datos . . . . .	236
<b>5.</b>	<b>Plan de implementación</b>	<b>238</b>
5.1.	Documentación . . . . .	239
5.1.1.	Manual de usuario . . . . .	239
5.1.2.	Manual de programación . . . . .	240
5.1.3.	Manual de instalación . . . . .	240
5.1.4.	Plan de implementación . . . . .	241
5.1.5.	Estructura Organizativa . . . . .	242
5.1.6.	Descripción de la estructura organizativa . . . . .	242
5.1.7.	Recursos a utilizar . . . . .	243
<b>6.</b>	<b>Conclusiones y Recomendaciones</b>	<b>247</b>
6.1.	Conclusiones . . . . .	247
6.2.	Recomendaciones . . . . .	249
	<b>Bibliografía</b>	<b>250</b>

<b>Anexos</b>	<b>252</b>
<b>A. Entrevista</b>	<b>253</b>
<b>B. Encuestas</b>	<b>255</b>
<b>C. Formato de los permisos para manipuladores</b>	<b>259</b>
<b>D. Presentación de sistema y cartas de aceptación U.C.S.F</b>	<b>260</b>

# Índice de tablas

2.3.1. Principales elementos del diagrama de secuencias . . . . .	55
4.1.1. Requerimientos Funcionales . . . . .	112
4.1.2. Requerimientos No Funcionales . . . . .	114
4.2.1. Actores y casos de uso asociados . . . . .	115
4.2.2. Factores de peso de los actores . . . . .	116
4.2.3. Factores de peso de los caso de uso . . . . .	117
4.2.4. Factores de complejidad técnica . . . . .	118
4.2.5. Factores de ambiente . . . . .	119
4.2.6. Distribución genérica del esfuerzo . . . . .	120
4.2.7. Distribución genérica del esfuerzo . . . . .	121
4.3.1. Especificaciones de Hardware de UCSF para el sistema web . . . . .	123
4.3.2. Especificaciones de los dispositivos móvil de los inspectores de UCSF para la app . . . . .	124
4.3.3. Especificaciones de Hardware para el desarrollo . . . . .	125
4.3.4. Especificaciones de los dispositivos móvil para la ejecución de las pruebas	126
4.3.5. Herramientas para la construcción del sistema web . . . . .	127
4.3.6. Herramientas para la construcción de la Aplicación Móvil . . . . .	127
4.3.7. Costos de Recursos tecnológicos . . . . .	129
4.3.8. Costos de Material Fungible . . . . .	130
4.3.9. Actores y casos de uso asociados . . . . .	130
4.3.10. Factores de peso de los actores . . . . .	131
4.3.11. Factores de peso de los caso de uso . . . . .	131
4.3.12. Factores de complejidad técnica . . . . .	133
4.3.13. Factores de ambiente . . . . .	134
4.3.14. Distribución genérica del esfuerzo . . . . .	135
4.3.15. Distribución genérica del esfuerzo . . . . .	136
4.3.16. Costo de operación . . . . .	137
4.3.17. Costos Varios por Servicios . . . . .	137
4.3.18. Costo Total del Proyecto . . . . .	138
4.5.1. Tablas de la base de datos . . . . .	202
4.5.2. Tabla asistencias de BD UCSF . . . . .	203
4.5.3. Tabla Credenciales de BD UCSF . . . . .	203
4.5.4. Tabla establecimientos de BD UCSF . . . . .	205
4.5.5. Tabla exámenes de BD UCSF . . . . .	206
4.5.6. Tabla fecha capacitaciones de BD UCSF . . . . .	207
4.5.7. Tabla inspecciones de BD UCSF . . . . .	208
4.5.8. Tabla log de BD UCSF . . . . .	209

4.5.9. Tabla manipuladores de BD UCSF . . . . .	211
4.5.10. Tabla tipo usuarios de BD UCSF . . . . .	211
4.5.11. Tabla usuarios de BD UCSF . . . . .	213
4.8.1. Rest API . . . . .	235
5.1.1. Recursos necesarios para la realización de la capacitación . . . . .	243
5.1.2. Cronograma de implementación . . . . .	245

# Índice de figuras

1.2.1.	Organigrama UCSF San Francisco Gotera Morazán . . . . .	5
1.2.2.	Ubicación de UCSF de San Francisco Gotera, Centro América . . . . .	6
1.2.3.	Ubicación de UCSF de San Francisco Gotera, El Salvador . . . . .	6
1.2.4.	Ubicación de UCSF de San Francisco Gotera, Morazán . . . . .	6
2.3.1.	Ejemplo de una relación de asociación . . . . .	42
2.3.2.	Ejemplo de una relación de extensión . . . . .	43
2.3.3.	Ejemplo de una relación de usos . . . . .	44
2.3.4.	Ejemplo de una relación de dependencia . . . . .	45
2.3.5.	Ejemplo de una relación de herencia . . . . .	47
3.6.1.	Porcentaje de Manipuladores con permiso . . . . .	98
3.6.2.	Manipuladores que han olvidado renovar credenciales . . . . .	99
3.6.3.	Porcentaje de manipuladores con Smartphone . . . . .	100
3.6.4.	Sistema operativos móviles usado por los manipuladores . . . . .	101
3.6.5.	Porcentaje de acceso a internet . . . . .	102
3.6.6.	Porcentaje de conexión a internet . . . . .	103
3.6.7.	Porcentaje de instalar aplicaciones móviles . . . . .	104
3.6.8.	Porcentaje de método para conexión a internet . . . . .	105
3.6.9.	Manipuladores que consideran útil una aplicación móvil . . . . .	106
4.4.1.	Diagrama de Casos de uso de Súper Usuario . . . . .	140
4.4.2.	Diagrama de Casos de uso de administrador . . . . .	141
4.4.3.	Diagrama de Casos de uso de manipuladores . . . . .	142
4.4.4.	Narración de caso de uso: Ingresar al sistema . . . . .	143
4.4.5.	Narración de caso de uso: Registrar Usuarios . . . . .	144

4.4.6. Narración de caso de uso: Gestionar Usuarios . . . . .	145
4.4.7. Narración de caso de uso: Registrar Manipuladores . . . . .	146
4.4.8. Narración de caso de uso: Gestionar Manipuladores . . . . .	147
4.4.9. Narración de caso de uso: Registrar Establecimientos . . . . .	148
4.4.10. Narración de caso de uso: Gestionar Establecimientos . . . . .	149
4.4.11. Narración de caso de uso: Búsqueda de Registros . . . . .	150
4.4.12. Narración de caso de uso: Generación de Reportes . . . . .	151
4.4.13. Narración de caso de uso: Inspección de establecimientos . . . . .	152
4.4.14. Narración de caso de uso: Resultados de exámenes . . . . .	153
4.4.15. Diagrama de Secuencia: proceso general de inscripción de un nuevo manipulador de alimentos . . . . .	155
4.4.16. Diagrama de Secuencia: proceso de inicio de sesión . . . . .	156
4.4.17. Interfaz: Inicio de Sesión . . . . .	157
4.4.18. Interfaz: Pagina principal . . . . .	157
4.4.19. Diagrama de Secuencia: proceso de usuario . . . . .	158
4.4.20. Interfaz de Usuario búsqueda . . . . .	159
4.4.21. Interfaz de Usuario índice . . . . .	159
4.4.22. Interfaz de ver Usuario . . . . .	160
4.4.23. Interfaz de Registrar Usuario . . . . .	160
4.4.24. Interfaz de Editar Usuario . . . . .	161
4.4.25. Interfaz de Desactivar Usuario . . . . .	161
4.4.26. Interfaz de confirmación para Desactivar Usuario . . . . .	162
4.4.27. Diagrama de Secuencia: proceso de reportes . . . . .	163
4.4.28. Interfaz de Reportes . . . . .	164
4.4.29. Interfaz de Reporte generado . . . . .	164
4.4.30. Diagrama de Secuencia: proceso de asistencia . . . . .	165
4.4.31. Interfaz de asistencia búsqueda . . . . .	166
4.4.32. Interfaz de asistencia . . . . .	166
4.4.33. Interfaz de Ver asistencia . . . . .	167
4.4.34. Interfaz de Editar asistencia . . . . .	167
4.4.35. Interfaz de Inasistencia . . . . .	168
4.4.36. Interfaz de Fecha capacitaciones . . . . .	168

4.4.37. Diagrama de Secuencia: proceso de inspección . . . . .	169
4.4.38. Interfaz de inspección búsqueda . . . . .	170
4.4.39. Interfaz de inspección lista . . . . .	170
4.4.40. Interfaz de inspección desactivadas . . . . .	171
4.4.41. Interfaz de Establecimientos . . . . .	171
4.4.42. Interfaz de Agregar nueva inspección . . . . .	172
4.4.43. Interfaz de ver inspección . . . . .	172
4.4.44. Interfaz de actualizar . . . . .	173
4.4.45. Diagrama de Secuencia: proceso de bitácora . . . . .	174
4.4.46. Interfaz de buscar bitácora . . . . .	175
4.4.47. Interfaz ver registro de bitácora . . . . .	175
4.4.48. Diagrama de Secuencia: proceso de credenciales . . . . .	176
4.4.49. Interfaz de Búsqueda credenciales . . . . .	177
4.4.50. Interfaz Credenciales aptas . . . . .	177
4.4.51. Interfaz Credenciales no aptas . . . . .	178
4.4.52. Interfaz Ver Credenciales . . . . .	178
4.4.53. Diagrama de Secuencia: proceso de Manipulador . . . . .	179
4.4.54. Interfaz de Manipuladores Activos . . . . .	180
4.4.55. Interfaz de Búsqueda Manipuladores . . . . .	180
4.4.56. Interfaz de Lista Manipuladores Desactivados . . . . .	181
4.4.57. Interfaz de Manipuladores Desactivados . . . . .	181
4.4.58. Interfaz de Nuevo Manipuladores . . . . .	182
4.4.59. Interfaz de Ver Manipuladores . . . . .	182
4.4.60. Interfaz de Actualizar Manipuladores . . . . .	183
4.4.61. Diagrama de Secuencia: proceso de Establecimiento . . . . .	184
4.4.62. Interfaz de Establecimientos . . . . .	185
4.4.63. Interfaz de Búsqueda Establecimientos . . . . .	185
4.4.64. Interfaz de Lista Establecimientos Desactivados . . . . .	186
4.4.65. Interfaz de Establecimientos Desactivados . . . . .	186
4.4.66. Interfaz de Nuevo Establecimiento . . . . .	187
4.4.67. Interfaz de Ver Establecimiento . . . . .	187
4.4.68. Interfaz de Actualizar Establecimientos . . . . .	188

4.4.69. Diagrama de Secuencia: proceso de Exámenes . . . . .	189
4.4.70. Interfaz de Búsqueda de Exámenes . . . . .	190
4.4.71. Interfaz de Exámenes aptos . . . . .	190
4.4.72. Interfaz de Exámenes No aptos . . . . .	191
4.4.73. Interfaz de Ver Exámenes . . . . .	191
4.4.74. Interfaz de Actualizar Exámenes . . . . .	192
4.4.75. Diagrama de Clases . . . . .	194
4.4.76. Diagrama Entidad - Relación . . . . .	196
4.4.77. Diagrama Físico . . . . .	197
4.4.78. Diagrama Lógico . . . . .	198
4.4.79. Diagrama Conceptual . . . . .	199
4.6.1. Diseño de la aplicación móvil de los inspectores . . . . .	214
(a). Inicio de sesión inspectores . . . . .	214
(b). Home de app inspectores . . . . .	214
(c). Home de app inspectores . . . . .	214
4.6.2. Diseño de la aplicación móvil de los inspectores . . . . .	215
(a). Establecimiento . . . . .	215
(b). Listado establecimientos . . . . .	215
(c). Búsqueda establecimientos . . . . .	215
4.6.3. Diseño de la aplicación móvil de los inspectores . . . . .	215
(a). Datos Establecimiento . . . . .	215
(b). Datos de inspección . . . . .	215
(c). Test de evaluación . . . . .	215
4.6.4. Diseño de la aplicación móvil de los inspectores . . . . .	216
(a). Procesar datos . . . . .	216
(b). Guardar datos . . . . .	216
(c). Registro guardado correctamente . . . . .	216
4.6.5. Diagrama de Secuencia: App Inspectores . . . . .	217
4.6.6. Diagrama de Secuencia: App Manipuladores . . . . .	218
4.6.7. Diseño de la aplicación móvil de los manipuladores . . . . .	219
(a). Ingreso de usuario . . . . .	219
(b). Inicio . . . . .	219

(c).	Inicio . . . . .	219
4.6.8.	Diseño de la aplicación móvil de los manipuladores . . . . .	219
(a).	Comprobar . . . . .	219
(b).	Ajustes . . . . .	219
(c).	Cargando configuración . . . . .	219
4.8.1.	directorios principales en MVC . . . . .	232
4.8.2.	carpetas base . . . . .	232
4.8.3.	estructura interna del directorio public . . . . .	233
4.8.4.	estructura interna del directorio app . . . . .	234
4.8.5.	Servicios web . . . . .	235
5.1.1.	Estructura organizativa de las personas involucradas en la implementación del sistema web . . . . .	242
C.0.1.	Ejemplo de formato de permiso otorgado a los manipuladores . . . . .	259
D.0.1.	Reunión con inspectores de UCSF . . . . .	260
D.0.2.	Carta de solicitud de la Directora . . . . .	261
D.0.3.	Carta de aceptación del sistema Directora . . . . .	262
D.0.4.	Carta de aceptación del sistema Inspector Técnico . . . . .	263

# INTRODUCCIÓN

Los sistemas informáticos y las Tecnologías de la Información y de la Comunicación (TIC) han transformado la manera en que las organizaciones trabajan en los últimos años. La demanda de este tipo de servicios aumenta cada vez debido a que las organizaciones se están enfocando siempre hacia importantes mejoras, como una mayor productividad y competitividad, la automatización de los procesos, una mejor toma de decisiones, etc.

Con el desarrollo del software presentado en esta documentación se pretende la mejora de los servicios que se prestan en el área de saneamiento ambiental específicamente a los usuarios relacionados con la manipulación de alimentos, en la actualidad esto se gestionan con el apoyo de las hojas de cálculo.

Con el desarrollo del software presentado en esta documentación se pretende la mejora de los servicios que se prestan en el área de saneamiento ambiental específicamente a los usuarios relacionados con la manipulación de alimentos, en la actualidad esto se gestionan con el apoyo de las hojas de cálculo.

En el primer capítulo de este documento se presenta la problemática actual con respecto al manejo de la información de los manipuladores de alimentos, haciendo enfoque en la importancia de la utilización de un sistema informático para la buena gestión de esta, también se presentan los objetivos, los alcances y limitaciones que conllevaron el desarrollo de este proyecto.

En el segundo capítulo se presenta una reseña histórica de la Unidad Comunitaria de Salud Familiar de San Francisco Gotera así como datos importantes del Ministerio de Salud relacionados con la manipulación de alimentos. También se presentan las herramientas que se utilizaron en el desarrollo del proyecto, así como datos legales para respaldar la investigación.

En el capítulo III se da a conocer la metodología de investigación descriptiva, la cual fue utilizada para el desarrollo de esta investigación, esta permitió principalmente conocer las características de la población que conllevaron las etapas de estudio. También se

presenta la metodología de desarrollo de cascada, que fue la utilizada en para el desarrollo de software; esta tiene un enfoque metodológico que ordena rigurosamente las etapas del ciclo de vida del software, de forma tal que el inicio de cada etapa debe esperar a la finalización de la inmediatamente anterior. En este capítulo también se da a conocer el método en cadena o redes, que es el método de muestreo que se utilizó para seleccionar la población de estudio. También se presentan las técnicas e instrumentos de investigación y los instrumentos para el análisis de los datos obtenidos.

El cuarto capítulo se enfoca en el desarrollo del software; se determinan los requerimientos funcionales y no funcionales del software para dar a conocer cuáles son las tareas que realizara el sistema y la forma en que lo realizara. Se hace un estudio de factibilidades para determinar qué tan viable era el desarrollo del proyector en tres áreas principales: técnica, económica y operativa.

También se presentan los diagramas utilizados para tener un mejor entendimiento de las funciones del sistema acompañando estos de las interfaces gráficas para tener una mejor visión del producto final.

En el capítulo V se elaboró el plan de implementación del proyecto, incluyendo en este los manuales de usuario, programación e instalación, para dar soporte a futuras mejoras o modificaciones que deseen implementar los administradores del sistema y para culminar en el capítulo VI se presentan las conclusiones y recomendaciones.

# Capítulo 1

## Estudio Preliminar

### 1.1. Definición del Tema

Desarrollo de un Sistema Informático de Registro, Emisión e Inspección de Credenciales Sanitarias Otorgadas a los Manipuladores de Alimentos en la Unidad Comunitaria de Salud Familiar de San Francisco Gotera en el Departamento de Morazán.

### 1.2. Descripción de la Institución

#### *1.2.1. Unidad Comunitaria de Salud Familiar de San Francisco Gotera*

La Unidad Comunitaria de Salud Familiar de San Francisco Gotera se fundó el 29 de noviembre del 2011, y se inauguró formalmente el 14 de febrero del 2012; iniciando su funcionamiento con un total de treinta y cuatro personas empleadas, destacadas en diferentes puestos de trabajo:<sup>1</sup>

- Siete médicos,
- Tres enfermeras,

---

<sup>1</sup>Historia de Unidad Comunitaria de Salud Familiar de San Francisco Gotera Información Proporcionada por Inspectores del área de Saneamiento Ambiental (Javier Hernández y Humberto Sánchez )

- Tres auxiliares de enfermería,
- Ocho promotores de salud,
- Un polivalente,
- Un ordenanza,
- Tres estadísticos,
- Dos Inspectores de saneamiento,
- Dos promotores antidengue,
- Un encargado de farmacia,
- Un supervisor específico de promotores,
- Dos odontólogas y,
- Un agente urbano solidario.

Para atender una población de veinticinco mil cincuenta y tres habitantes; contando solamente con las siguientes áreas y consultorios:

1. cuatro consultorios médicos,
2. área de odontología,
3. área de saneamiento ambiental,
4. archivo,
5. bodega y,
6. farmacia.

En ese momento la ubicación de La Unidad Comunitaria de Salud Familiar se encontraba en la Alameda San Francisco, salida a San Miguel.

A medida que se fue incrementando el número de usuarios en medicina general, desde julio del 2012 se traslada la Unidad de Salud a un local más amplio y adecuado para

brindar las atenciones, se habilitan cuatro consultorios más, se incorpora el laboratorio clínico, se amplía el área de odontología; y hasta la fecha se alquilan las instalaciones donde funciona.<sup>2</sup>

Su ubicación actual es en carretera salida a Osicala, Barrio El Calvario, a 50 metros al Sur de Desvío El Amate, San Francisco Gotera.

La primera Directora fue la Dra. Nuvia Mabel Chicas de Ascencio quién asumió el cargo desde el 19 de septiembre del 2011, hasta que falleció el 07 de septiembre de 2013, razón por la cual asumió el cargo como interino el Dr. Manuel de Jesús Vásquez Viera. Actualmente el cargo de Director de la Unidad de Salud lo ocupa la Dra. Sandra de la Paz Laínez de Canales.<sup>3</sup>

Actualmente cuenta con un total de cuarenta y un empleados, destacados en el desarrollo de una oferta de servicios clínicos como lo son:

### **Ambulatorios**

- Medicina General
- Curaciones
- Inyecciones
- Pequeña Cirugía
- Inhalo-terapia
- Rehidratación Oral
- Vacunación Humana
- Odontología
- Educación para la salud
- Farmacia

---

<sup>2</sup>Ibídem, pág.2

<sup>3</sup>Idem

- Laboratorio Clínico

### **Programas**

- Atención integral en salud infantil (Toma del tamizaje neonatal)
- Atención integral en salud de la mujer (Toma de Citología y examen clínico de mama)
- Adulto mayor
- Adolescente
- Adulto hombre
- Programa TB - VIH-SIDA
- Salud ocupacional
- Salud sexual y reproductiva Extramural
- Atención a escuela saludable
- Actividades de organización y participación comunitaria
- Actividades de promoción de la salud
- Atención al medio
- Vacunación Animal
- Atención de Denuncias

Desde entonces los habitantes del municipio han tenido la ventaja de contar con su propia Unidad de Salud, con un mejor acceso a un lugar donde pasar consulta para sus problemas de salud más comunes de una manera rápida y obteniendo mejores tratamientos.

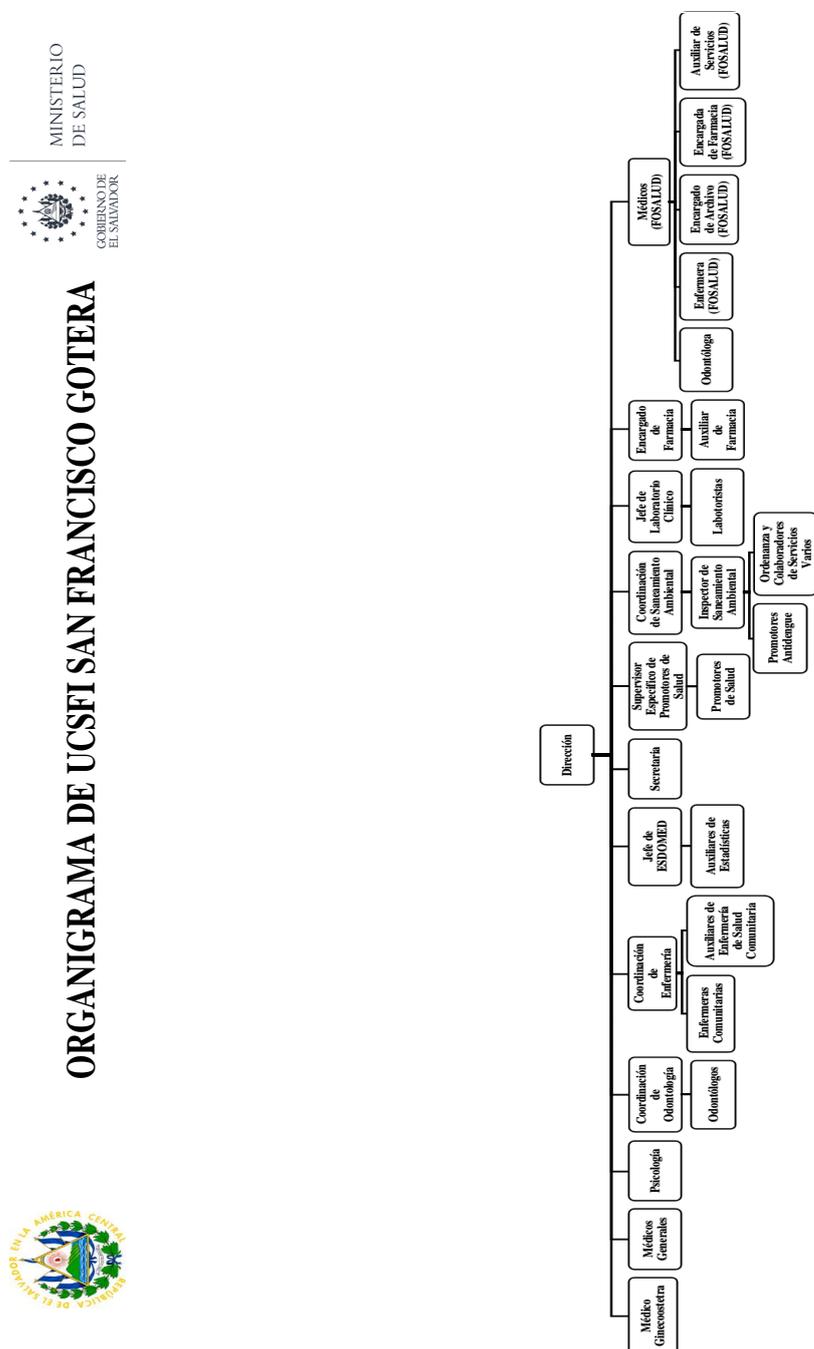
Con la apertura y la extensión de los servicios de dicho establecimiento se logró el descongestionamiento del primer nivel de atención de los servicios médicos que brinda el único hospital del municipio.<sup>4</sup>

---

<sup>4</sup>Idem

### 1.2.2. Organigrama de la institución

Organigrama de la Unidad de Salud Familiar de San Francisco Gotera<sup>5</sup>



## ORGANIGRAMA DE UCSFI SAN FRANCISCO GOTERA



Figura 1.2.1: Organigrama UCSF San Francisco Gotera Morazán

<sup>5</sup>Información de Unidad Comunitaria de Salud Familiar de San Francisco Gotera Proporcionada por Inspectores del área de Saneamiento Ambiental (Javier Hernández y Humberto Sánchez )

### 1.2.3. Croquis de ubicación del proyecto

Unidad Comunitaria de Salud Familiar de San Francisco Gotera, Morazán<sup>6</sup>



Figura 1.2.2: Ubicación de UCSF de San Francisco Gotera, Centro América

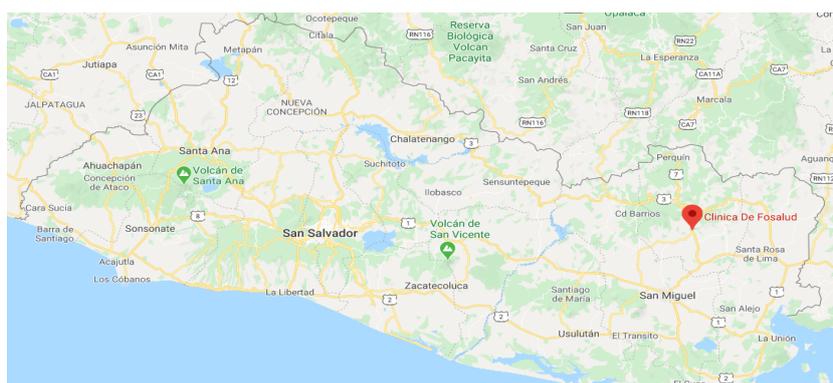


Figura 1.2.3: Ubicación de UCSF de San Francisco Gotera, El Salvador



Figura 1.2.4: Ubicación de UCSF de San Francisco Gotera, Morazán

<sup>6</sup>Imágenes Recuperadas de Mapa Google

### 1.3. Situación Problemática

En el año 2016, en El Salvador fue implementado por el Ministerio de Salud el proyecto *Capacitación sobre higiene de alimentos*, con el fin de reducir el riesgo de enfermedades transmitidas por los mismos y garantizar la salud de la población; dicho proyecto consiste en capacitar a todos los manipuladores de alimentos en el proceso de su elaboración e impulsar buenas prácticas de higiene personal, además, solicitar diversos exámenes clínicos al personal que labora en establecimientos de comida.

En la Unidad Comunitaria de Salud Familiar del municipio de San Francisco Gotera en el Departamento de Morazán, se manejan datos de alrededor de mil doscientos cincuenta manipuladores de alimentos a los cuales se les exige el cumplimiento de la normativa impuesta por el Ministerio de Salud, apegada al proyecto mencionado anteriormente. Cada comerciante que trate productos comestibles tiene que asistir a capacitaciones impartidas anualmente por el personal de la Unidad de Saneamiento Ambiental y además presentar los respectivos exámenes clínicos cada seis meses para comprobar que se encuentra apto para elaborar o manipular alimentos; para que, posteriormente si ha cumplido con todos los requisitos pueda obtener la aprobación de su credencial en la Unidad de Salud y luego dirigirse a la Alcaldía Municipal que es donde se presenta el documento que se ha proporcionado para que le provean su credencial final.

Actualmente en dicha Unidad de Salud no se cuenta con una herramienta automatizada que coadyuve en el manejo de la información concerniente a los manipuladores de alimentos, todos los datos son procesados y almacenados en hojas de cálculo y archivos físicos, lo cual afecta de manera negativa la eficiencia del personal de la Unidad, provocando que los procesos para solicitar credenciales sean lentos y engorrosos, también afecta en las actividades de campo que realizan los Inspectores de Saneamiento en los diferentes establecimientos de comida para verificar que todo se maneje bajo las normas establecidas por el Ministerio de Salud; es importante mencionar que el proceso de la entrega de las credenciales culmina en la Alcaldía Municipal, lo cual continua generando problemas de pérdida de tiempo al personal involucrado y a los interesados en la obtención

de su credencial; como hasta la fecha las autoridades pertinentes no han considerado automatizar este proceso se ha tomado a bien desarrollar un sistema informático amigable y eficiente que supla las necesidades del personal de la Unidad así como también de los involucrados en la manipulación de alimentos.

## 1.4. Enunciado del problema

¿Cómo desarrollar un sistema informático integral que mejore la gestión en los procesos de entrega de credenciales de manipulación de alimentos y la inspección de establecimientos manejados por la Unidad Comunitaria de Salud Familiar de San Francisco Gotera?

## 1.5. Objetivos

### 1.5.1. *Objetivo General*

Desarrollar un sistema para el registro, emisión e inspección de credenciales sanitarias Otorgadas a los manipuladores de alimentos en la Unidad Comunitaria de Salud Familiar del Municipio de San Francisco Gotera en El Departamento de Morazán.

### 1.5.2. *Objetivos Específicos*

- Analizar la problemática actual que tiene el área de Saneamiento Ambiental de la Unidad Comunitaria de Salud Familiar en el proyecto de Capacitación para la Higiene para proveer soluciones que ayuden a optimizar el trabajo que ahí se desarrolla.
- Diseñar una interfaz amigable que se adapte a las necesidades de La Unidad para lograr un manejo eficiente de la información.
- Codificar los requerimientos brindados por La Unidad de Salud y realizar las pruebas pertinentes para garantizar el buen funcionamiento del sistema informático.
- Crear un plan de implementación del sistema para que este disponible en el momento que la administración desee ponerlo en marcha.

## 1.6. Justificación

En el municipio de San Francisco Gotera del Departamento de Morazán, existe una cantidad considerable de trabajadores que se dedican a la manipulación de alimentos, cuyos procesos de elaboración requieren el control y la inspección por parte de las autoridades del Ministerio de Salud, partiendo de ello, surge la necesidad de un sistema informático en la Unidad Comunitaria de Salud Familiar, el cual registre toda la información de los manipuladores de alimentos, así como también los datos obtenidos en las revisiones realizadas por los Inspectores de Saneamiento; automatizar todos esos procesos es de gran interés para las autoridades administrativas de dicha institución, debido a que actualmente los procesos se llevan de forma manual, generando problemas de pérdida de información, desarrollo de procesos de forma lenta e ineficiente y además duplicación de registros.

Hasta la fecha las autoridades del Ministerio de Salud no han destinado fondos para la adquisición de un software que ayude con el manejo de las tareas mencionadas anteriormente y así prevenir los problemas que se generan día a día; con el desarrollo del sistema que se propone, se estarían beneficiando a tres colaboradores de la Unidad de Saneamiento Ambiental que son los que estarán involucrado directamente con el sistema, a mil doscientos cuarenta y tres manipuladores de alimentos entre los sectores formal e informal (Datos otorgados por la Unidad de Salud Municipal); además, a los consumidores en general que, según la última campaña de fumigación ejecutada por la misma Unidad de Saneamiento se realizó un censo, obteniendo un total de una población de veinticinco mil novecientos veinticinco personas del área urbana y rural.

El sistema informático será manejado por el Área de Saneamiento Ambiental de la Unidad de Salud en estudio, para llevar un control adecuado y una correcta inspección de los establecimientos de alimentos, además se tendrá un registro de las personas que han completado el plan de capacitación, así como de los que han cumplido con todos los requisitos establecidos en la normativa, los usuarios directos del sistema podrán clasificar por sectores los datos de los manipuladores de alimentos.

Como complemento al sistema se desarrollará una aplicación móvil destinada al uso de los Promotores encargados de la inspección de los establecimientos para evitar el problema de llevar registros en papel, y de esta forma ellos puedan actualizar información en tiempo real y esta se actualice inmediatamente en la base de datos, garantizando que la población obtendrá alimentos de buena calidad y seguros para el consumo.

Finalmente uno de los beneficios mas importantes que aportará el software es la prevención de enfermedades que adquiere la población por la ingestión de alimentos preparados de forma incorrecta e insalubre; actualmente en el municipio en cuestión el control por parte de las autoridades de saneamiento se ve limitado por la cantidad de establecimientos de alimentos que existen y en adición se tienen a manipuladores del sector informal, al poseer un personal escaso en la Unidad para el desarrollo de todos los procesos necesarios y que maneje los datos que se generan a diario, se les hace muy difícil tener un control eficiente de todos los manipuladores y es de suma importancia mencionar que las enfermedades transmitidas por los alimentos son uno de los problemas de salud pública que se presentan con más frecuencia en la vida cotidiana de la población, y ese es el impacto en la salud pública que se desea disminuir y así también el daño económico a los establecimientos por las sanciones que podrían obtener al no cumplir la normativa.

## 1.7. Alcances

El desarrollo del Sistema Informático tiene como finalidad acelerar los procesos de registro, emisión e inspección de credenciales sanitarias que se llevan a cabo en el área de saneamiento ambiental de la Unidad de Salud.

Siguiendo las etapas del modelo de desarrollo de software en cascada, el Sistema Informático será llevado hasta la fase de la elaboración del plan de implementación.

El Sistema Informático contará con las siguientes características:

- Módulo de monitoreo de los manipuladores de alimentos: llevando un registro del cumplimiento e incumplimiento de las medidas sanitarias y de las asistencias a las capacitaciones impartidas por los inspectores de Saneamiento Ambiental.
- Módulo de usuario para brindar mayor seguridad en la información y establecer privilegios de administración de acuerdo a su tipo.
- Creación de módulo donde se llevará el registro, control y monitoreo de los establecimientos de comida del sector formal, llevando un registro del cumplimiento o incumplimiento de las normativas impuestas por el MINSAL, llevadas a cabo en la respectiva inspección.
- Clasificar a los manipuladores de alimentos en sector formal e informal y realizar filtros de búsqueda que permitan llevar el control de las credenciales sanitarias, así como los que se encuentran en tratamiento médico para la obtención y los que no cumplen con los requisitos establecidos por el MINSAL.
- Se desarrollarán dos aplicaciones móviles para el sistema operativo android una para notificar a los manipuladores de alimentos con fecha anticipada a la expiración de su credencial y la otra para facilitar la inspección del sector formal, permitiendo la búsqueda del establecimiento y la verificación del historial del mismo para la realización de la respectivas evaluaciones.
- Se realizarán capacitaciones por el período de una semana, dirigidas al personal del

Área de Saneamiento Ambiental en cuanto al uso y funcionamiento de cada uno de los módulos que se incluirán en el sistema informático.

- Para el desarrollo del sistema se utilizará la metodología del Modelo en Cascada ya que facilita la planeación, la administración de proyectos, programas y portafolios.
- Se realizará un manual de instalación, manual de programación y manual de usuarios del sistema a desarrollar.

## 1.8. Limitaciones

- La Unidad Comunitaria de Salud Familiar de San Francisco Gotera en el Departamento de Morazán, cuenta solamente con dos equipos para la ejecución del sistema informático.
- Los operadores del sistema, deberán poseer como mínimo conocimientos básicos de computación.
- La institución será la responsable de cubrir los gastos de implementación del sistema.
- Por motivos de presupuesto, se limita a desarrollar la aplicación móvil solo para sistema operativo android.

## 1.9. Delimitación

### 1.9.1. *Espacial*

El desarrollo de esta investigación se llevará a cabo en La Unidad Comunitaria de Salud Familiar de San Francisco Gotera en el Departamento de Morazán.

### 1.9.2. *Temporal*

El proyecto se planeó para un periodo de nueve meses, iniciando el diez de febrero del año dos mil veinte y finalizando el diez de octubre del año dos mil veinte.

# Capítulo 2

## Marco de Referencia

### 2.1. Marco Histórico

#### *2.1.1. Origen del Ministerio de Salud*

La Institución nace el 23 de Julio de 1900, con el nombre de Consejo Superior de Salubridad, dependencia del Ministerio de Gobernación. El Consejo determinó que entre las actividades principales a realizar fueran: estadísticas médicas, saneamiento de zonas urbanas, inspecciones de víveres, higiene de rastros y mercados, construcción de cloacas y sistemas de aguas servidas, obligatoriedad de instalar letrinas, lucha contra los mosquitos, visitas a establos, fábricas y beneficios de lavar café. El primer Código de Sanidad entra en vigencia el 24 de julio del mismo año. El 15 de agosto se creó la Dirección General de Vacunación la cual en 1907 se llamó Instituto de Vacunación Contra la Viruela.<sup>7</sup>

En 1948 se creó el Ministerio de Asistencia Social, quien en 1950 pasa a ser el Ministerio de Salud Pública y Asistencia Social. En 1926 el Poder Ejecutivo a través de la Sub-secretaría de beneficencia creó el servicio de Asistencia Gratuita, con la finalidad de prestar auxilio a los enfermos de escasos recursos económicos. Servicio que fue modificado, posteriormente en 1945, según Acuerdo N<sup>o</sup> 730 del Ministerio de Salud; estableciendo que por la consulta externa se pagaría una tarifa especial, la cual, serviría para complementar

---

<sup>7</sup>Historia del Ministerio de Salud (2015) Extraído de [www.salud.gob.sv/historia/](http://www.salud.gob.sv/historia/)

los gastos de los rubros que no cubrían con el presupuesto de dicha institución.<sup>8</sup>

En 1964, dentro del Ministerio de Salud se creó la supervisión de los servicios de agua potable, a través del departamento de Saneamiento Ambiental, con la finalidad de captar, proteger y llevar agua a las comunidades rurales del país. Desde los años 70´s se comenzó a reconocer la necesidad de impulsar simultáneamente la salud y el desarrollo social, así como la concordancia en el desarrollo de los servicios, estructura y las metas de salud de cada país. Siempre durante los años setenta el Ministerio de Salud, logró dar cobertura a todo el país a través de la división organizativa de cinco regiones: Región Occidental, Región Metropolitana, Región Central, Región Oriental y Región Paracentral.<sup>9</sup>

La Región Occidental estaba constituida por los departamentos de Ahuachapán, Sonsonate y Santa Ana, quien estaba bajo la dirección de un Médico Director Regional, y era el responsable de velar por el cumplimiento de todos los programas y proyectos de los tres departamentos.

En septiembre de 2001, se lleva a cabo la desconcentración del Sistema Nacional de Salud en el que se obliga a identificar, priorizar y satisfacer las necesidades en salud de la población, lo cual es logrado a través del Sistema Básico de Salud Integral (SIBASI), el cual se considera la estructura básica operativa descentralizada del Sistema Nacional de Salud, fundamentada en la Atención Primaria en Salud, que mediante la producción de servicios integrales de salud del primer y segundo nivel de atención (El Primer nivel es aquel que esta conformado por las Unidades de Salud, el Segundo nivel comprende el Hospital San Juan de Dios) la participación ciudadana consiente y efectiva, así como, la responsabilidad de otros sectores que es la que contribuye a mejorar el nivel de salud de una población definida.

En El Salvador, los esfuerzos han sido importantes. La última revisión, al respecto, realizada duramente los últimos dos años y que ha sido recopilada de diversos escritos, van dirigidos hacia objetivos comunes, es decir, a todos los sectores sociales interesados en que se impulse e implemente el proceso de reforma en salud, con una amplia base de

---

<sup>8</sup>Idem

<sup>9</sup>Idem

concertación política - técnica, ejemplo de ello, son algunos sistemas que han desarrollado en alguna medida la intersectorialidad, la participación social, redes de atención integral, calidad, referencia y retorno, desarrollo hospitalario, entre otros; volviéndose una fortaleza para el abordaje a los problemas de salud.

Fue en el programa de gobierno 1999 - 2004, que el Ministerio de Salud Pública y Asistencia Social (MSPYAS) se propuso desarrollar acciones que estaban inmersas en la Alianza Solidaria y la Alianza por la Seguridad por lo cual se planteó la política de: “Mejorar el nivel de salud y de vida de la población salvadoreña”, propiciando las condiciones de eficiencia, eficacia, equidad, calidad y calidez, en la provisión de los servicios, haciendo énfasis en el enfoque de atención integral y tomando como estrategia básica la participación activa de todos los actores sociales y las necesidades de la comunidad.

## 2.2. Marco Contextual

### 2.2.1. *Condiciones del sistema de salud de El Salvador*

En El Salvador sigue habiendo importantes rezagos en materia de vivienda, acceso a agua potable y servicios sanitarios, y nutrición. No sorprende, por lo tanto, que en la población infantil predominen las enfermedades diarreicas, parasitaria y respiratorias, y los padecimientos asociados a la desnutrición. La mortalidad infantil en el país asciende a 17.5 por 1000 nacidos vivos. La razón de mortalidad materna es de 170 por 100 000 nacidos vivos registrados, una de las más altas de la región latinoamericana. En la población adulta, además de las enfermedades no transmisibles, hay una alta incidencia de lesiones intencionales y no intencionales. La lista de principales causas de mortalidad refleja un creciente predominio de las enfermedades no transmisibles y las lesiones. En las mujeres las primeras causas de muerte son el infarto agudo del miocardio y la diabetes mellitus. Entre los hombres las dos principales causas de muerte son las lesiones con armas y los accidentes de vehículos de motor, que juntas concentran casi 20 % de los decesos totales. La tercera causa de muerte es la insuficiencia renal crónica.<sup>10</sup>

### 2.2.2. *Estructura y Cobertura*

El sistema de salud salvadoreño está compuesto por dos sectores, el público y el privado. El sector público incluye el Ministerio de Salud Pública y Asistencia Social (MSPAS), el Instituto Salvadoreño del Seguro Social (ISSS), el Instituto Salvadoreño de Rehabilitación a los Inválidos (ISRI), Sanidad Militar, el Instituto Salvadoreño de Bienestar Magisterial (ISBM) y el Fondo Solidario para la Salud (FOSALUD).

El sector privado comprende las entidades privadas lucrativas y no lucrativas.

El ISSS y el MSPAS cubren a la mayor parte de la población salvadoreña. El ISSS es una entidad autónoma y tiene como fin la seguridad social de los trabajadores del sector formal de la economía, mientras que el MSPAS, además de ofrecer servicios de salud a la

---

<sup>10</sup>Sistema de Salud de El Salvador (junio de 2011). Extraído de <http://www.scielo.org.mx/>

población sin seguridad social, es el ente regulador del sector.

El ISRI es una institución autónoma cuya finalidad es la provisión de servicios especializados de rehabilitación a personas con discapacidad. Sanidad Militar ofrece protección social en salud a las fuerzas armadas. Finalmente, el ISBM es una entidad oficial autónoma de derecho público que cubre a la población docente y sus familias.

FOSALUD, es una entidad de derecho público vigente desde 2004 que cuenta con plena autonomía tanto en lo financiero como en lo administrativo y presupuestario adscrita al MSPAS. Esta instancia tiene como objetivos: propiciar la creación de programas especiales para ampliar la cobertura de los servicios de salud; formular y ejecutar programas integrales que atiendan las necesidades básicas de salud de la población más vulnerable del país, y fomentar campañas de educación para la salud.

El sector privado está integrado sobre todo por organizaciones lucrativas establecidas en las principales ciudades salvadoreñas. Estas organizaciones ofrecen servicios en el mercado privado y venden servicios al ISSS y al ISBM. Las organizaciones no lucrativas (ONG, iglesias y otras), por su parte, operan sobre todo en las zonas rurales de El Salvador.

### ***2.2.3. Recurso Humano del MINSAL***

El Salvador se encuentra actualmente en una situación paradójica: produce una cantidad relativamente elevada de agentes de salud, pero todavía no puede cubrir las necesidades de la población en lo que se refiere a una cobertura universal de la atención sanitaria en el país, y a la nueva estrategia nacional de salud en pro de una atención sanitaria básica completa. Entre los principales desafíos, se encuentran los siguientes:

- Desempleo y subempleo, motivados por la sobreproducción de agentes de salud en determinadas categorías, la incapacidad financiera del sector sanitario para crear contratos, y la distribución desigual de los agentes de salud en los distintos niveles de servicios.
- Un déficit del 43% en lo que se refiere a agentes de salud para los servicios de

atención primaria, unido a una distribución geográfica, demográfica e institucional no equitativa de los recursos existentes.

- Disparidad en lo que respecta a las condiciones de trabajo y los incentivos económicos entre el Instituto Salvadoreño del Seguro Social (ISSS) y el Ministerio de Salud.
- Falta de agentes de salud formados adecuadamente para poner en marcha la actual estrategia de atención sanitaria basada en la atención primaria.
- Existencia de mecanismos de coordinación y coordinación deficientes entre las instituciones del sector sanitario y los centros de formación de agentes de salud.
- Falta de un sistema de información que permita una toma de decisiones oportuna en materia de RHS para solucionar problemas y generar acciones de personal orientadas al desarrollo y la calidad de la atención prestada.
- Situación de los Recursos Humanos en Salud en el Salvador: Estudio de caso, 2010

El desarrollo de RHS abarca 4 áreas: capacidad y política de gestión; planificación y regulación de los RHS; cambio en la calidad y las condiciones laborales de los agentes de salud; y coordinación de la formación de RHS. Los objetivos relativos a estas cuatro áreas se han desarrollado en el contexto del plan estratégico nacional de salud, de 5 años de duración (Construyendo la Esperanza, Estrategias y Recomendaciones en Salud 2009-2014). A través de estos objetivos, se persigue:

- Recoger información para la gestión estratégica del desarrollo de RHS
- Planificar y coordinar la formación de RHS
- Desarrollar políticas para mejorar las condiciones y la seguridad laborales de los agentes de salud
- Mejorar la capacidad de actuación de los agentes de salud comunitarios
- Aumentar la cantidad de enfermeros y médicos especialistas disponibles
- Fortalecer la Dirección de RHS

- Reforzar la formación de médicos especialistas.

El Salvador ha creado recientemente una Dirección de Desarrollo de Recursos Humanos en Salud, y CISALUD: una comisión intersectorial de asuntos sanitarios para coordinar actividades relacionadas con la salud, entre las que se incluyen aquellas relacionadas con los RHS. En la comisión participan los Ministerios de Salud, Trabajo y Hacienda, y diversos organismos públicos.<sup>11</sup>

#### *2.2.4. Manipulación de Alimentos en El Salvador*

Cuando hablamos de manipulación de alimentos nos referimos al acto de estar presente en la diferentes etapas que estos poseen; es decir la preparación, fabricación, transformación, embazado, almacenamiento, transporte y distribución. Es de suma importancia que cuando una persona se involucra en cualquiera de las etapas mencionadas tenga un cuidado especial a fin de prevenir accidentes y enfermedades.

Todas las personas que están en contacto con los productos alimenticios durante su etapa de manipulación, reposición y recepción deben cuidar que los productos no entren en contacto con sustancias químicas o contaminantes.

El manipulador de alimentos debe vigilar las fechas de caducidad y retirar aquellos productos que hayan caducado. Además, debe controlar la higiene y limpieza de los lugares donde se almacenan los productos alimenticios.

Por lo general, las normas de seguridad exigen que los manipuladores de alimentos mantengan una escrupulosa higiene personal, que no fumen mientras se manipulan los productos, que utilicen guantes de goma u otros elementos protectores en caso de tener heridas o cortes, que usen ropa limpia y un gorro para mantener el cabello recogido.

Dada la complejidad en el proceso de producción de ciertos alimentos, las empresas requieren además de normas y regulaciones, un personal capacitado en higiene alimentaria. En base a esto, se suele exigir que todos los manipuladores de alimentos dispongan de los conocimientos necesarios para desarrollar adecuadamente las prácticas de manipulación.

---

<sup>11</sup>Extraído de [www.who.int/workforcealliance/countries/slv/es/](http://www.who.int/workforcealliance/countries/slv/es/)

En la actualidad la higiene personal y los hábitos higiénicos de los manipuladores de alimentos son las formas más efectivas de combatir las infecciones e intoxicaciones alimentarias entre los consumidores. Por eso las empresas y emprendimientos dedicados a la industria alimentaria deben poner especial atención en las normas y regulaciones de higiene que se aplican en cada establecimiento.

Como podemos ver, la higiene es un factor que debe estar presente a lo largo de todo el proceso de manipulación de los alimentos, a fin de preservar la salud, garantizar salubridad y seguridad en el consumo, evitando la transmisión de enfermedades.<sup>12</sup>

### ***2.2.5. Capacitaciones sobre higiene de alimentos***

El art. 34 de la norma técnica de alimentos establece que todas las personas que laboran en la manipulación de alimentos deben estar autorizadas para realizar tal actividad a través de los cursos que imparten los establecimientos de salud del MINSAL. La Dirección de Salud Ambiental del MINSAL son los que se encargan de capacitar a propietarios y empleados de empresas y establecimientos alimenticios con el fin de reducir el riesgo de las enfermedades que se transmiten a través de los alimentos y así garantizar la salud de la población.

En las capacitaciones que se imparten se tocan temas sobre la higiene personal, la higiene de los alimentos y las enfermedades que se transmiten por alimentos, todo con el fin de garantizarle a la población que las personas que elaboran y sirven diversos platillos manipulen los ingredientes una forma correcta. Además de las capacitaciones se les solicita exámenes médicos a todos los que participan en la elaboración de alimentos para así conocer el estado de salud y evitar la transmisión de enfermedades como parasitismo intestinal, fiebre tifoidea, hepatitis A, entre otros. Posteriormente se les entrega un carnet que certifica que la persona está capacitada y libre de enfermedades.

---

<sup>12</sup>Extraído de [www.educativo.net/articulos/en-que-consiste-la-manipulacion-de-alimentos](http://www.educativo.net/articulos/en-que-consiste-la-manipulacion-de-alimentos)

## 2.3. Marco Teórico

### 2.3.1. *Aplicaciones Web o Webbapps*

Esta categoría de software centrado en redes agrupa una amplia gama de aplicaciones. En su forma más sencilla, las webbapps son poco más que un conjunto de archivos de hipertexto vinculados que presentan información con uso de texto y gráficas limitadas. Sin embargo, desde que surgió Web 2.0, las webbapps están evolucionando hacia ambientes de cómputo sofisticados que no sólo proveen características aisladas, funciones de cómputo y contenido para el usuario final, sino que también están integradas con bases de datos corporativas y aplicaciones de negocios.<sup>13</sup>

Powell [Pow98] sugiere que los sistemas y aplicaciones basados en web “involucran una mezcla entre las publicaciones impresas y el desarrollo de software, entre la mercadotecnia y la computación, entre las comunicaciones internas y las relaciones exteriores, y entre el arte y la tecnología”. La gran mayoría de webbapps presenta los siguientes atributos:<sup>14</sup>

- **Uso intensivo de redes.** Una webbapp reside en una red y debe atender las necesidades de una comunidad diversa de clientes. La red permite acceso y comunicación mundiales (por ejemplo, internet) o tiene acceso y comunicación limitados (por ejemplo, una intranet corporativa).
- **Concurrencia.** A la webbapp puede acceder un gran número de usuarios a la vez. En muchos casos, los patrones de uso entre los usuarios finales varían mucho.
- **Carga impredecible.** El número de usuarios de la webbapp cambia en varios órdenes de magnitud de un día a otro. El lunes tal vez la utilicen cien personas, el jueves quizá 10 000 usen el sistema.
- **Rendimiento.** Si un usuario de la webbapp debe esperar demasiado (para entrar, para el procesamiento por parte del servidor, para el formado y despliegue del lado

---

<sup>13</sup>Ingeniería de software : un enfoque practico, 7th edición , Roger S. Pressman, 2010, P.7

<sup>14</sup>Ibidem P.9

del cliente), él o ella quizá decidan irse a otra parte.

- **Disponibilidad.** Aunque no es razonable esperar una disponibilidad de 100 %, es frecuente que los usuarios de webapps populares demanden acceso las 24 horas de los 365 días del año. Los usuarios en Australia o Asia quizá demanden acceso en horas en las que las aplicaciones internas de software tradicionales en Norte América no estén en línea por razones de mantenimiento.
- **Orientadas a los datos.** La función principal de muchas webapp es el uso de hiper-medios para presentar al usuario final contenido en forma de texto, gráficas, audio y vídeo. Además, las webapps se utilizan en forma común para acceder a información que existe en bases de datos que no son parte integral del ambiente basado en web (por ejemplo, comercio electrónico o aplicaciones financieras).

### *2.3.2. Ventajas de la aplicaciones web*

Hay muchos beneficios relacionados con el proceso de montar o mejorar una aplicación en Web<sup>15</sup>:

1. Aumenta el número de usuarios que se enteran de la disponibilidad de un servicio, producto, industria, persona o grupo.
2. Los usuarios tienen la posibilidad de acceder las 24 horas del día.
3. Se puede mejorar la utilidad y capacidad de uso del diseño de la interfaz.
4. Se puede expandir un sistema globalmente en vez de permanecer en el entorno local, con lo cual se puede establecer contacto con personas en ubicaciones remotas sin preocuparse por la zona horaria en la que se encuentren

### *2.3.3. Aplicación Móvil*

Las aplicaciones móviles son programas diseñados para ser ejecutados en teléfonos, tablet y otros dispositivos móviles, que permiten al usuario realizar actividades

---

<sup>15</sup>Análisis y diseño de sistemas, Kendall Kendall, 8th Edición , 2011, pag. 4

profesionales, acceder a servicios, mantenerse informado, entre otro universo de posibilidades.<sup>16</sup>

### 2.3.3.1. Desarrollo de aplicaciones móviles.

El desarrollo de aplicaciones móviles son los procedimientos y procesos establecidos que intervienen cuando se crea un software para pequeños dispositivos informáticos inalámbricos, como tabletas y teléfonos inteligentes.<sup>17</sup>

Al igual que el desarrollo de aplicaciones web, los procesos de desarrollo de aplicaciones móviles tienen sus raíces en el desarrollo de software tradicional.

### 2.3.3.2. Desarrollo de software.

Cuando se trata del desarrollo de aplicaciones móviles, se requiere acceso a kits de desarrollo de software (SDK) que permiten a los programadores diseñar y probar su aplicación de código en un entorno simulado controlado.

1. Los SDK comúnmente utilizados son:
2. Unity
3. Android SDK
4. Licencia de Desarrollador iOS (necesaria para el desarrollo de aplicaciones para iOS)

Para la creación de una aplicación móvil exitosa, los programadores deben pasar por las siguientes fases:

- **La investigación:** Refinación de la idea a través de la investigación
- **Wireframing:** Creación del marco esquelético de la aplicación
- **Evaluación de viabilidad técnica:** Teniendo en cuenta los sistemas de back-end de la aplicación
- **Prototipo:** Un prototipo rápido es el concepto de la aplicación en realidad

---

<sup>16</sup>Recuperado de <https://servisoftcorp.com/definicion-y-como-funcionan-las-aplicaciones-moviles>

<sup>17</sup>Recuperado de <https://invidgroup.com/es/que-es-el-desarrollo-de-aplicaciones-moviles/>

- **Diseño:** Incluye codificación y diseño de la interfaz
- **Desarrollo:** Desarrollo progresivo de la aplicación
- **Pruebas:** Prueba de funcionalidad y cualquier error que deba corregirse.
- **Implementación:** Presentación de la aplicación móvil final.

### 2.3.3.3. Tipos de aplicaciones móviles.

La popularidad exponencial de los teléfonos inteligentes y tabletas ha llevado al aumento de la creación de software en línea con el desarrollo de aplicaciones móviles. Los dos sistemas operativos líderes, iOS y Android, han marcado el ritmo en la estandarización de los diferentes tipos de desarrollo de aplicaciones móviles para programadores.<sup>18</sup>

Estos diferentes tipos de aplicaciones móviles incluyen:

- **Aplicaciones nativas**

Las aplicaciones nativas están diseñadas para plataformas de dispositivos específicos, ya sea Android o iOS. Se descargan o instalan a través de una tienda de aplicaciones y se accede a través de un ícono en el dispositivo. Las aplicaciones nativas están diseñadas para aprovechar al máximo las características del dispositivo como el GPS, la cámara y las listas de contactos, entre otros.

- **Aplicaciones web o HTML5**

Basándose en las tecnologías web universales y estandarizadas, como HTML5, JavaScript y CSS, las aplicaciones web se implementan como un sitio web que simplemente se ve y se siente como aplicaciones nativas. Funcionan y se ejecutan en un navegador escrito típicamente en HTML5. Estas aplicaciones sólo aprovechan las funciones de GPS y cámara de un dispositivo.

- **Aplicaciones híbridas**

Las aplicaciones híbridas de JavaScript, HTML y CSS son una combinación de

---

<sup>18</sup>Idem

aplicaciones nativas y web. Se obtienen de una tienda de aplicaciones y aprovechan las características del dispositivo, como una aplicación nativa. Al igual que la aplicación web, se accede desde un navegador y se basa en HTML. En pocas palabras, las aplicaciones híbridas se instalan como una aplicación nativa, pero operan como una aplicación web.

#### **2.3.3.4. Características de las aplicaciones móviles.**

Principales características que debe poseer las aplicaciones móviles se mencionan a continuación:<sup>19</sup>

##### **1. Interfaz simple**

El éxito de una aplicación móvil no es directamente proporcional a lo compleja que sea la aplicación. Todo lo contrario. Los usuarios demandan un interfaz simple e intuitivo. Una navegación simple basada en las pautas de diseño para la plataforma seleccionada.

##### **2. Visibilidad en IOS y Android**

Se trata de las dos principales plataformas en las que se debe estar presente. El desarrollo multiplataforma ahorra los presupuestos iniciales, permite la entrada temprana en el mercado y logra de manera efectiva la siguiente tercera característica

##### **3. Seguridad**

La seguridad es vital para el éxito de una aplicación. Así los temas importantes como la confidencialidad de los datos está protegida en el nivel legal pero también existen otros aspectos que atienden a la importancia de la seguridad como puede ser el acceso a información sensible de cada usuario.

##### **4. Funcionamiento offline de la APP**

Es un aspecto a tener en cuenta a la hora de plantear el uso y funcionalidad de una aplicación móvil. Es lógico pensar que la APP dependa del uso y consumo de

---

<sup>19</sup><https://bluumi.net/10-caracteristicas-una-aplicacion-movil-de-empresa-exito/>

datos, es decir de su conexión a Internet. Sin embargo, el acceso a determinadas funcionalidades o contenidos en modo offline puede resultar de interés para los usuarios.

## 5. Actualizaciones periódicas de la APP

Una aplicación móvil necesitará un ciclo continuo de desarrollo y por lo tanto de actualizaciones periódicas. Se debe contar con un equipo preparado para dar ese mantenimiento y esas nuevas funcionalidades que hagan crecer la APP. Todo el contenido que se ofrezca a través de una aplicación móvil deberá ser actualizado y relevante para el usuario o de lo contrario con el tiempo perderá valor.

## 6. El sistema de búsqueda

El sistema debe ser más fácil e intuitivo posible para localizar lo que se quiera dentro de una aplicación móvil.

### ■ Enfoque para las Pruebas de Software

La prueba es un conjunto de actividades que pueden planearse por adelantado y realizarse de manera sistemática. Por esta razón, durante el proceso de software, debe definirse una plantilla para la prueba del software: un conjunto de pasos que incluyen métodos de prueba y técnicas de diseño de casos de prueba específicos.<sup>20</sup>

En la literatura sobre el tema, se han propuesto algunas estrategias de prueba de software. Todas proporcionan una plantilla para la prueba y tienen las siguientes características genéricas:<sup>21</sup>

- Para realizar una prueba efectiva, debe realizar revisiones técnicas efectivas. Al hacerlo, eliminará muchos errores antes de comenzar la prueba.
- La prueba comienza en los componentes y opera “hacia afuera”, hacia la integración de todo el sistema de cómputo.
- Diferentes técnicas de prueba son adecuadas para distintos enfoques de

---

<sup>20</sup>Roger S. Pressman Ingeniería del Software: un enfoque practico, 7th edición, 2010, Pag. 384

<sup>21</sup>Ídem

ingeniería de software y en diferentes momentos en el tiempo.

- Las pruebas las realiza el desarrollador del software y (para proyectos grandes) un grupo de prueba independiente.
- Prueba y depuración son actividades diferentes, pero la depuración debe incluirse en cualquier estrategia de prueba.

Una estrategia para la prueba de software debe incluir pruebas de bajo nivel, que son necesarias para verificar que un pequeño segmento de código fuente se implementó correctamente, así como pruebas de alto nivel, que validan las principales funciones del sistema a partir de los requerimientos del cliente. Una estrategia debe proporcionar una guía para el profesional y un conjunto de guías para el jefe de proyecto. Puesto que los pasos de la estrategia de prueba ocurren cuando comienza a aumentar la presión por las fechas límite, el avance debe ser medible y los problemas deben salir a la superficie tan pronto como sea posible<sup>22</sup>.

#### ■ Estrategias de Prueba para Software Orientando a Objetos

Enunciado de manera simple, el objetivo de probar es encontrar el mayor número posible de errores con una cantidad manejable de esfuerzo aplicado durante un lapso realista. Aunque este objetivo fundamental se mantiene invariable para el software orientado a objeto, la naturaleza de este software cambia tanto la estrategia como las tácticas de la prueba.<sup>23</sup>

#### ■ Prueba de unidad en el contexto OO

Cuando se considera software orientado a objeto, el concepto de unidad cambia. La encapsulación determina la definición de clases y objetos. Esto significa que cada clase y cada instancia de una clase empaqueta los atributos (datos) y las operaciones que manipulan estos datos. Por lo general, una clase encapsulada es el foco de la prueba de unidad. No obstante, las operaciones (métodos) dentro de la clase son las unidades comprobables más pequeñas. Puesto que una clase puede contener

---

<sup>22</sup>Ídem

<sup>23</sup>Roger S. Pressman Ingeniería del Software: un enfoque practico, 7th edición, 2010, Pag. 397

algunas operaciones diferentes, y una operación particular puede existir como parte de algunas clases diferentes, las tácticas aplicadas a la prueba de unidad deben cambiar.

Ya no es posible probar una sola operación en aislamiento (la visión convencional de la prueba de unidad) sino más bien como parte de una clase. Para ilustrarlo, considere una jerarquía de clase en la que una operación X se define para la superclase y la heredan algunas subclasses. Cada subclase usa la operación X, pero se aplica dentro del contexto de los atributos y operaciones privados que se definieron para la subclase. Dado que el contexto en el que se usa la operación X varía sutilmente, es necesario probar la operación X en el contexto de cada una de las subclasses. Esto significa que por lo general no es efectivo probar la operación X en forma aislada (el enfoque de prueba de unidad convencional) en el contexto orientado a objeto.

La prueba de clase para software OO es el equivalente de la prueba de unidad para software convencional. A diferencia de la prueba de unidad del software convencional, que tiende a enfocarse sobre el detalle algorítmico de un módulo y en los datos que fluyen a través de la interfaz de módulo, la prueba de clase para software OO la dirigen las operaciones encapsuladas por la clase y el comportamiento de estado de ésta. <sup>24</sup>

### ■ Prueba de integración en el contexto OO

Puesto que el software orientado a objeto no tiene una estructura de control jerárquico obvia, las estrategias tradicionales descendente y ascendente; tienen poco significado. Además, con frecuencia es imposible integrar las operaciones una a la vez en una clase (el enfoque de integración incremental convencional) debido a las “interacciones directa e indirecta de los componentes que constituyen la clase” [Ber93].

Existen dos estrategias diferentes para la prueba de integración de los sistemas OO [Bin94b]. La primera, la prueba basada en hebra, integra el conjunto de clases requeridas para responder a una entrada o evento para el sistema. Cada hebra

---

<sup>24</sup>Ibídem, Pag. 398

se integra y prueba de manera individual. La prueba de regresión se aplica para asegurar que no ocurran efectos colaterales. El segundo enfoque de integración, la prueba basada en uso, comienza la construcción del sistema al probar dichas clases (llamadas clases independientes) que usan muy pocas clases servidor (si es que usan alguna). Después de probar las clases independientes, se prueba la siguiente capa de clases, llamadas dependientes, que usan las clases independientes. Esta secuencia de probar capas de clases dependientes continúa hasta que se construye todo el sistema.

El uso de controladores y representantes también cambia cuando se realiza la prueba de integración de los sistemas OO. Los controladores pueden usarse para probar operaciones en el nivel más bajo, y para la prueba de todos los grupos de clases. También puede usarse un controlador para sustituir la interfaz de usuario, de modo que las pruebas de funcionalidad del sistema puedan realizarse antes de la implementación de la interfaz. Los representantes (stubs) pueden usarse en situaciones donde se requiere la colaboración entre clases, pero donde una o más de las clases colaboradoras todavía no se implementan por completo.

La prueba de grupo es un paso en la prueba de integración del software OO. Aquí, un grupo de clases colaboradoras (determinadas al examinar el CRC y el modelo objeto relacional) se ejercita al diseñar casos de prueba que intentan descubrir errores en las colaboraciones.<sup>25</sup>

#### ***2.3.4. Estratetegias de prueba para webapps***

La estrategia para probar webapps adopta los principios básicos para todas las pruebas de software y aplica una estrategia y tácticas que se usan para sistemas orientados a objetos. Los siguientes pasos resumen el enfoque:<sup>26</sup>

1. El modelo de contenido para la webapp se revisa para descubrir errores.
2. El modelo de interfaz se revisa para garantizar que todos los casos de uso

---

<sup>25</sup>Roger S. Pressman Ingeniería del Software: un enfoque practico, 7th edición, 2010, Pag. 398

<sup>26</sup>Ibídem, Pag. 398

pueden adecuarse.

3. El modelo de diseño para la webapp se revisa para descubrir errores de navegación.
4. La interfaz de usuario se prueba para descubrir errores en los mecanismos de presentación y/o navegación.
5. A cada componente funcional se le aplica una prueba de unidad.
6. Se prueba la navegación a lo largo de toda la arquitectura.
7. La webapp se implementa en varias configuraciones ambientales diferentes y se prueba en su compatibilidad con cada configuración.
8. Las pruebas de seguridad se realizan con la intención de explotar vulnerabilidades en la webapp o dentro de su ambiente.
9. Se realizan pruebas de rendimiento.
10. La webapp se prueba mediante una población de usuarios finales controlada y monitoreada. Los resultados de su interacción con el sistema se evalúan por errores de contenido y navegación, preocupaciones de facilidad de uso, preocupaciones de compatibilidad, así como confiabilidad y rendimiento de la webapp.
11. Puesto que muchas webapps evolucionan continuamente, el proceso de prueba es una actividad siempre en marcha, y se realiza para apoyar al personal que usa pruebas de regresión derivadas de las pruebas desarrolladas cuando se elaboró por primera vez la webapp. En el capítulo 20 se consideran métodos para probar la webapp.

#### ■ Pruebas de Validación

Las pruebas de validación comienzan en la culminación de las pruebas de integración, cuando se ejercitaron componentes individuales, el software está completamente ensamblado como un paquete y los errores de interfaz se descubrieron y corrigieron.

En el nivel de validación o de sistema, desaparece la distinción entre software convencional, software orientado a objetos y webapps. Las pruebas se enfocan en las acciones visibles para el usuario y las salidas del sistema reconocibles por el usuario.

La validación puede definirse en muchas formas, pero una definición simple (aunque dura) es que la validación es exitosa cuando el software funciona en una forma que cumpla con las expectativas razonables del cliente. En este punto, un desarrollador de software curtido en la batalla puede protestar: “¿quién o qué es el árbitro de las expectativas razonables?”. Si se desarrolló una Especificación de requerimientos de software, en ella se describen todos los atributos del software visibles para el usuario; contiene una sección de Criterios de validación que forman la base para un enfoque de pruebas de validación<sup>27</sup>.

#### ■ Criterios de pruebas de validación

La validación del software se logra a través de una serie de pruebas que demuestran conformidad con los requerimientos. Un plan de prueba subraya las clases de pruebas que se van a realizar y un procedimiento de prueba define casos de prueba específicos que se diseñan para garantizar que: se satisfacen todos los requerimientos de funcionamiento, se logran todas las características de comportamiento, todo el contenido es preciso y se presenta de manera adecuada, se logran todos los requerimientos de rendimiento, la documentación es correcta y se satisfacen la facilidad de uso y otros requerimientos (por ejemplo, transportabilidad, compatibilidad, recuperación de error, mantenimiento).

Después de realizar cada caso de prueba de validación, existen dos posibles condiciones: 1) La característica de función o rendimiento se conforma de acuerdo con las especificaciones y se acepta, o 2) se descubre una desviación de la especificación y se crea una lista de deficiencias. Las desviaciones o errores descubiertos en esta etapa en un proyecto rara vez pueden corregirse antes de la entrega calendarizada. Con frecuencia es necesario negociar con el cliente para establecer un método para

---

<sup>27</sup>Roger S. Pressman Ingeniería del Software: un enfoque practico, 7th edición, 2010, Pag. 399

resolver deficiencias<sup>28</sup>.

### ■ Pruebas alfa y beta

Virtualmente, es imposible que un desarrollador de software prevea cómo usará el cliente realmente un programa. Las instrucciones para usarlo pueden malinterpretarse; regularmente pueden usarse combinaciones extrañas de datos; la salida que parecía clara a quien realizó la prueba puede ser ininteligible para un usuario.

Cuando se construye software a la medida para un cliente, se realiza una serie de pruebas de aceptación a fin de permitir al cliente validar todos los requerimientos. Realizada por el usuario final en lugar de por los ingenieros de software, una prueba de aceptación puede variar desde una “prueba de conducción” informal hasta una serie de pruebas planificadas y ejecutadas sistemáticamente. De hecho, la prueba de aceptación puede realizarse durante un periodo de semanas o meses, y mediante ella descubrir errores acumulados que con el tiempo puedan degradar el sistema.

Si el software se desarrolla como un producto que va a ser usado por muchos clientes, no es práctico realizar pruebas de aceptación formales con cada uno de ellos. La mayoría de los constructores de productos de software usan un proceso llamado prueba alfa y prueba beta para descubrir errores que al parecer sólo el usuario final es capaz de encontrar.

- **La prueba alfa** se lleva a cabo en el sitio del desarrollador por un grupo representativo de usuarios finales. El software se usa en un escenario natural con el desarrollador “mirando sobre el hombro” de los usuarios y registrando los errores y problemas de uso. Las pruebas alfa se realizan en un ambiente controlado.
- **La prueba beta** se realiza en uno o más sitios del usuario final. A diferencia de la prueba alfa, por lo general el desarrollador no está presente. Por tanto, la prueba beta es una aplicación “en vivo” del software en un ambiente que no puede controlar el desarrollador. El cliente registra todos los problemas

---

<sup>28</sup>Ibídem

(reales o imaginarios) que se encuentran durante la prueba beta y los reporta al desarrollador periódicamente. Como resultado de los problemas reportados durante las pruebas beta, es posible hacer modificaciones y luego preparar la liberación del producto de software a toda la base de clientes.

En ocasiones se realiza una variación de la prueba beta, llamada prueba de aceptación del cliente, cuando el software se entrega a un cliente bajo contrato. El cliente realiza una serie de pruebas específicas con la intención de descubrir errores antes de aceptar el software del desarrollador. En algunos casos (por ejemplo, un gran corporativo o sistema gubernamental) la prueba de aceptación puede ser muy formal y abarcar muchos días o incluso semanas de prueba.<sup>29</sup>

### **2.3.5. Internet**

La Internet, también llamada como “La Red”. Es todo el sistema de redes de computadora que existe mundialmente, formando un conjunto integrado por diferentes redes de cada país, donde los usuarios en cualquier parte del mundo a través de un ordenador, tienen acceso a la red en caso de contar con los permisos apropiados, pueden tener acceso a distintos tipos de información como noticias, documentos, libros integrados en formato de (imagen, texto, sonido y vídeo). Toda esta información está contenida en bases de datos alojada en servidores remotos.<sup>30</sup>

Fue concebido por la agencia de nombre ARPA (Advanced Research Projects Agency) del gobierno de los Estados Unidos en el año de 1969 y se le conocía inicialmente como ARPANET. El propósito original fue crear una red que permitiera a los investigadores del Campus poder comunicarse a través de los sistemas de cómputo con investigadores de otras Universidades.

El Internet hace uso del protocolo de comunicación TCP/IP (Transmission Control Protocol/Internet Protocol). Originalmente creados con fines militares, TCP/IP

---

<sup>29</sup>Roger S. Pressman Ingeniería del Software: un enfoque practico, 7th edición, 2010, Pag. 400

<sup>30</sup>Informática Milenium, S.A. de C.V.(s.f.). Internet. Recuperado de <https://www.informaticamilenium.com.mx/es/temas/que-es-internet.html>

representa todas las reglas de comunicación a Internet y se basa en la noción de dirección IP, es decir la idea de brindar una IP a cada equipo de la red para poder en-rutar paquetes de datos.<sup>31</sup>

### **2.3.6. Web**

La web también conocida como WWW o World Wide Web, es una fuente inagotable de conocimiento humano accesible a través de Internet, una de las características sobresalientes es el texto marcado, este es un método para referencias cruzadas instantáneas, donde ciertas palabras de texto aparecen de un color diferente al resto del documento, por lo general el texto tendrá un subrayado. El cual puede activarse mediante un click, uno es transferido al sitio o página relacionado a esa frase. En algunas ocasiones hay botones, imágenes, o porciones de imágenes.

El nacimiento de la World Wide Web, surge de un programa con el mismo nombre, por unos investigadores del Laboratorio Europeo de Física de Partículas CERN, el verano de 1991. Consistía en un conjunto de protocolos que permitían el acceso flexible y generalizado a la información almacenada en la red en diversos formatos.<sup>32</sup>

Para la exploración en la web se necesita de un software especializado denominado browser o explorador, la apariencia del sitio web puede variar ligeramente del explorador que se esté utilizando.

### **2.3.7. URL**

URL (Uniform Resource Location). Es un enlace hacia una dirección web específica, donde cada segmento de este enlace brinda diferente información, requerida por el navegador a través del cliente.

La URL describe una dirección comprendida con un esquema el cual es reconocido

---

<sup>31</sup>Diseño y Desarrollo Web - MSc César A. Delgado B.(s.f.). Protocolo TCP/IP (Transmission Control Protocol/Internet Protocol). Recuperado de [https://upanama.educativa.com/archivos/repositorio//6000/6126/html/51\\_pproto.htm](https://upanama.educativa.com/archivos/repositorio//6000/6126/html/51_pproto.htm)

<sup>32</sup>Grupo BBVA.(s.f.). Tim Berners-Lee y el origen de la Web. Recuperado de <https://www.bbvaopenmind.com/tecnologia/visionarios/tim-berners-lee-y-el-origen-de-la-web/>

por nuestros navegadores indicando el tipo de dirección a la que se desea tener acceso. Los más comunes son http y https (este es empleado cuando se necesita seguridad al sitio donde se va a requerir acceso).

Los dominios y subdominios son uno de los componentes principales en el esquema de una URL ya que lo más común es que el nombre sea de la compañía que administra el sitio web como ejemplo: Facebook, Google, entre otros. Para los subdominios se comienza escribiendo como, WWW o World Wide Web y por lo general puede omitirse.

También hay otros tipos de subdominios que están al extremo derecho del dominio, entre los más comunes son los que terminan en .edu, son los creados con fines educativos, .com son los más frecuentemente utilizado para compañías y .org utilizado para las organizaciones gubernamentales. Por último, se tienen las rutas de archivos ubicadas al extremo derecho del subíndice, las cuales dan acceso a páginas específicas del sitio.

### ***2.3.8. Servidor Web***

Un Servidor es un ordenador con características superiores a los ordenadores comunes, esto es así ya que deben dar respuestas a una gran cantidad de peticiones “http” del navegador del usuario, estos ordenadores necesitan de un programa específico para el Servidor Web, el cual gestiona las aplicaciones del lado del servidor, y establece conexiones bidireccionales, unidireccionales, síncronas o asíncronas con el cliente y generar respuestas para la aplicación del lado del cliente.<sup>33</sup>

El tipo de respuesta puede ser un código que puede ser compilado y ejecutado por el navegador web, para la transmisión de estos datos se utilizan protocolos, el que mas comúnmente se utiliza es el protocolo HTTP (HyperText Transfer Protocol) que desempeña un papel importante en la comunicación ya que contiene toda la información necesaria estableciendo criterios de semántica y sintaxis, permitiendo a los ordenadores hablar un mismo idioma.

---

<sup>33</sup>Hostingsaurio.com. (s.f.). Qué es un servidor web. Recuperado de <https://hostingsaurio.com/que-es-un-servidor-web/>

### **2.3.9. Cliente Web**

El cliente web es un programa encargado de establecer conexiones con el servidor más comúnmente utilizado es el navegador web a través de una red de telecomunicaciones. Ya que muchos servidores son capaces de ofrecer sus servicios a través de este sin la necesidad de requerir otra aplicación para hacerlo.

Existen tipos de clientes como los que tienen la capacidad almacenar datos y procesarlos, los de correo electrónico suelen ser un cliente pesado, cliente híbrido: no tiene almacenados los datos que envía al servidor, cliente liviano: el cual simplemente muestran las páginas web solicitadas por el usuario.

### **2.3.10. Sistema Operativo Android**

Es un sistema operativo móvil basado en Linux enfocado para ser utilizado en dispositivos móviles como teléfonos inteligentes, tabletas, Google TV y otros dispositivos.

Es desarrollado por la Open Handset Alliance, liderada por Google. La estructura del sistema operativo Android se compone de aplicaciones que se ejecutan en un framework Java de aplicaciones orientadas a objetos sobre el núcleo de las bibliotecas de Java en una máquina virtual Dalvik con compilación en tiempo de ejecución.

Las bibliotecas escritas en lenguaje C incluyen un administrador de interfaz gráfica, un framework OpenCore, una base de datos relacional SQLite, una Interfaz de programación de API gráfica OpenGL ES 2.0 3D, un motor de renderizado WebKit, un motor gráfico SGL, SSL y una biblioteca estándar de C Bionic.

Las aplicaciones se desarrollan habitualmente en el lenguaje Java con Android Software Development Kit (Android SDK), Existen otras herramientas de desarrollo, incluyendo un Kit de Desarrollo Nativo para aplicaciones o extensiones en C, C++ u otros lenguajes de programación.<sup>34</sup>

---

<sup>34</sup>Juan Garrido Cobo.(Enero, 2013). TFC Desarrollo de Aplicaciones Móviles

### 2.3.11. *Lenguaje Unificado de Modelado (UML)*

El Lenguaje Unificado de Modelado (UML) fue creado para forjar un lenguaje de modelado visual común y semántica y sintácticamente rico para la arquitectura, el diseño y la implementación de sistemas de software complejos, tanto en estructura como en comportamiento. UML tiene aplicaciones más allá del desarrollo de software, p. ej., en el flujo de procesos en la fabricación.<sup>35</sup>

Es comparable a los planos usados en otros campos y consiste en diferentes tipos de diagramas. En general, los diagramas UML describen los límites, la estructura y el comportamiento del sistema y los objetos que contiene.<sup>36</sup>

UML no es un lenguaje de programación, pero existen herramientas que se pueden usar para generar código en diversos lenguajes usando los diagramas UML.

UML guarda una relación directa con el análisis y el diseño orientados a objetos.

Como sabemos una imagen vale mas que mil palabras, es por ello que es necesario crear una estructura que sea comprensible por todo tipo de usuario que le interese el funcionamiento del software. Además, estos diagramas nos permitirán tener un indicio del desarrollo del software, algo palpable antes de la construcción del sistema.

### 2.3.12. *Modelado de caso de uso*

Un modelo de caso de uso es un modelo de las funciones previstas del sistema y su entorno y sirve como un contrato entre el cliente y los desarrolladores. Los casos de uso sirven como hebra de unión a lo largo del desarrollo del sistema. El mismo modelo de caso de uso es el resultado de la disciplina de requisitos y se utiliza como entrada para disciplinas de Prueba, Diseño y Análisis.<sup>37</sup>

---

<sup>35</sup>Lucidchart. *¿Qué es el lenguaje unificado de modelado (UML)?*, Estados Unidos, 2010, Recuperado de <https://www.lucidchart.com/pages/es/que-es-el-lenguaje-unificado-de-modelado-uml>

<sup>36</sup>Idem

<sup>37</sup>FOWLER, Martin. *UML distilled*, Reino Unido, Addison Wesley, 1997.

### 2.3.12.1. Elementos principales del diagrama de caso de uso

Los casos de uso se representan gráficamente con una elipse horizontal con el nombre del caso que aparece encima, debajo o dentro de la elipse. Un caso de uso representa un objetivo individual del sistema y describe una secuencia de actividades y de interacciones del usuario para tratar de alcanzar el objetivo.<sup>38</sup>

La creación de los casos de uso ha probado ser una técnica excelente para entender y documentar mejor los requerimientos del sistema.

Un caso de uso por sí solo no se considera como un requerimiento funcional, pero la historia (el escenario) que relata el caso de uso consiste en uno o más requerimientos.<sup>39</sup>

Inicialmente, los casos de uso se definen durante la etapa de los requerimientos del ciclo de vida y se refinarán adicionalmente a lo largo del ciclo de vida. Durante la identificación de los requisitos, los casos de uso se emplean para capturar la esencia de los problemas de negocios y para modelar (a un alto nivel) la funcionalidad del sistema propuesto.<sup>40</sup>

Adicionalmente, son el punto inicial para la identificación de las entidades de datos o de los objetos del sistema. Durante el análisis de requisitos, los casos de uso se refinan para modelar el uso del sistema con más detalle.<sup>41</sup>

En otras palabras, se actualizan para especificar lo que los usuarios están tratando de alcanzar y el porqué. Estos casos ayudan en la definición de los prototipos o interfaces del usuario. Durante el diseño los casos de uso se refinan para modelar cómo los usuarios usarán realmente el sistema con respecto a cualquier interfaz y a las restricciones del sistema.<sup>42</sup>

Estos tipos de casos de uso ayudan a identificar el comportamiento del objeto o

---

<sup>38</sup>Ibidem

<sup>39</sup>Ibidem

<sup>40</sup>CAMACHO, Antonio. *Herramienta para el análisis de requerimientos dentro de la pequeña empresa desarrolladora de software en Bogotá*, Colombia, Proyecto De Grado, Pontificia Universidad Javeriana, 2005.

<sup>41</sup>Ibidem

<sup>42</sup>Ibidem

del sistema y a diseñar las especificaciones de código y de interfaz, así como para servir para planear las pruebas del sistema. En la construcción, los casos de uso ayudan a los desarrolladores en la programación y en las pruebas.

Estos casos también sirven como la línea base para preparar cualquier documentación del sistema y para el usuario, además de servir como herramientas para el entrenamiento de los usuarios.

Y como los casos de uso contienen una enorme cantidad de detalle de la funcionalidad del sistema, éstos serán un recurso constante para validar al sistema. Alguno de los elementos principales son:

- **Actores**

Los casos de uso se inician o son generados por los usuarios externos llamados actores. Un actor inicia la actividad del sistema, un caso de uso, con el propósito de terminar alguna tarea de negocios que produzca algo con valor apreciable.<sup>43</sup>

Utilicemos el ejemplo de un estudiante universitario que se inscribe en las clases o materias de un ciclo determinado.

El actor sería el estudiante, y el evento de negocios, o caso de uso, sería Inscribirse en Ciclo.

Un actor representa un papel desempeñado por un usuario que interactúa con el sistema y no significa que retrate a una persona o un puesto de trabajo.

De hecho, un actor no tiene que ser humano. Puede ser una organización, otro sistema de información, un dispositivo externo tal como un sensor de calor, o aun el concepto de tiempo. Un actor se representa gráficamente como una figura de línea rotulada con el nombre del papel que juega el actor.<sup>44</sup>

Es importante observar que hay principalmente cuatro tipos de actores:

---

<sup>43</sup>Junta de Andalucía. *Guía para la redacción de casos de uso*, España, 2009, Recuperado de <http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/416>

<sup>44</sup>Ibidem

- *Actor primario de negocios*: el interesado que se beneficia principalmente de la ejecución de un caso de uso al recibir algo de valor medible u observable. El actor primario de negocios puede o no iniciar el evento de negocios. Por ejemplo, en el evento de negocios de un empleado que recibe un cheque como pago (algo con valor medible) del sistema de nómina cada viernes, el empleado no inicia el evento pero es el receptor primario del algo de valor.<sup>45</sup>
- *Actor primario del sistema*: el involucrado que tiene una interfaz directa con el sistema para iniciar u ocasionar el evento de negocios o de sistema. Los actores primarios del sistema pueden interactuar con los actores primarios de negocios con el propósito de usar el sistema real. Ellos facilitan el evento a través del uso directo del sistema para beneficio del actor primario de negocios. Los ejemplos incluyen un dependiente de una tienda de abarrotes que selecciona los artículos para el cliente que compra abarrotes, una operadora de teléfonos que desproporciona información del directorio a un cliente, y un cajero de banco que procesa una transacción bancaria. El actor principal de negocios y el actor principal de sistema pueden ser la misma persona para eventos en los cuales el actor de negocios tiene una interfaz directa con el sistema; por ejemplo, una persona que reserva la renta de un automóvil a través de un sitio Web.<sup>46</sup>
- *Actor externo servidor*: el involucrado que responde a una solicitud desde el caso de uso (por ejemplo, un buró de crédito que autoriza el pago mediante tarjeta de crédito).<sup>47</sup>
- *Actor externo receptor*: el involucrado que no es el actor primario pero que recibe algo de valor medible u observable (salida) proveniente del caso de uso (por ejemplo, un almacén que recibe una orden de embalaje para preparar un

---

<sup>45</sup>Universidad Nacional Abierta y a Distancia. *Lenguaje de Modelado Unificado*, Recuperado de <https://openclassrooms.com/en/courses/4990961-planea-tu-proyecto-con-uml/4996511-diagramas-de-casos-de-uso>

<sup>46</sup>Ibidem

<sup>47</sup>Ibidem

flete después de que un cliente ha colocado una orden).<sup>48</sup>

### ■ Relaciones

Una relación se ilustra como una línea entre dos símbolos en el diagrama de casos de uso. El significado de las relaciones puede diferir dependiendo de cómo se dibujen las líneas y qué tipos de símbolos conectan. En las siguientes secciones definiremos las diferentes relaciones que se encuentran en un diagrama de casos de uso.<sup>49</sup>

### ■ Asociaciones

Existe una relación entre un actor y un caso de uso siempre que el caso describa una interacción entre éstos. A esta relación se le denomina asociación. Como se indica en la **figura 2.3.1**, una asociación se modela como una línea continua que conecta al actor y al caso de uso. Una asociación que contiene una cabeza de flecha en el extremo que toca al caso de uso ( 1 ) indica que el caso fue iniciado por el actor en el otro extremo de la línea. Las asociaciones sin cabezas de flecha ( 2 ) indican una interacción entre el caso de uso y un actor externo servidor o receptor. Si un actor se asocia con un caso de uso, decimos que el actor se comunica con el caso. Las asociaciones pueden ser bidireccionales o unidireccionales.<sup>50</sup>



**Figura 2.3.1:** Ejemplo de una relación de asociación

### ■ Extensión

---

<sup>48</sup>Ibidem

<sup>49</sup>POINTEAU, Arthur, *Diagrama de casos de uso*, Openclassrooms, 2018, Recuperado de <https://openclassrooms.com/en/courses/4990961-planea-tu-proyecto-con-uml/4996511-diagramas-de-casos-de-uso>

<sup>50</sup>Ibidem

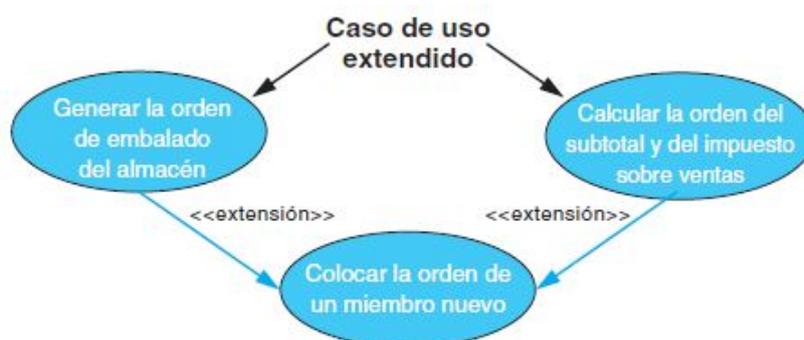
Un caso de uso puede contener una funcionalidad compleja que consiste de varios pasos que hacen difícil entender a la lógica del caso. Con objeto de simplificar el caso de uso y hacer que se entienda más fácilmente, podemos extraer los pasos más complejos para formar su propio caso.<sup>51</sup>

El caso resultante se llama un caso de uso de extensión ya que extiende la funcionalidad del caso de uso original. La relación entre el caso de uso de extensión y el que se está extendiendo se llama una relación de extensión.

Un caso de uso puede tener muchas relaciones de extensión, pero un caso de uso de extensión puede ser invocado solamente por el caso que se esté extendiendo.

Como se ilustra en la **figura 2.3.2**, la relación de extensión se representa como una línea con cabeza de flecha (ya sea continua o segmentada) que comienza en el caso de uso de extensión y que apunta al caso de uso que se está extendiendo. Cada línea de relación de extensión se rotula como “«extensión»”.

Generalmente los casos de uso de extensión no se identifican en la fase de requerimientos, sino en la de análisis.



**Figura 2.3.2:** Ejemplo de una relación de extensión

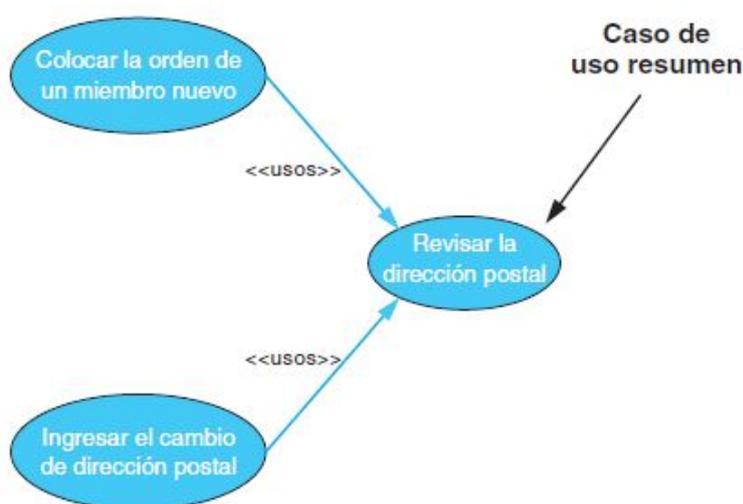
#### ■ Uso o inclusión

Muy comúnmente, usted puede descubrir dos o más casos de uso que ejecuten pasos de funcionalidad idéntica. Lo mejor es extraer estos pasos comunes para formar un caso de uso separado que sea propio llamado un caso de uso resumen. Un caso

---

<sup>51</sup>Ibidem

de uso resumen representa una forma de “reuso” y es una herramienta excelente para reducir la redundancia entre los casos de uso. Un caso de uso resumen está disponible como referencia (o uso) para cualquier otro caso de uso que requiera su funcionalidad. La relación entre el caso de uso resumen y el caso de uso que lo usa se llama una relación de uso (algunas herramientas de la modelación de casos de uso lo denominan una relación de inclusión). La relación de uso que se presenta en la **figura 2.2.3** se ilustra con una línea con cabeza de flecha (ya sea continua o segmentada) que comienza en el caso de uso oficial y que apunta al caso de uso que esté usando. Cada línea de relación de uso se rotula “«uso»”. Generalmente los casos de uso resúmenes no se identifican en la fase de requisitos, sino en la de análisis.<sup>52</sup>



**Figura 2.3.3:** Ejemplo de una relación de usos

### ■ Dependencia

Como administrador de proyecto o desarrollador líder, es de mucha ayuda saber cuáles casos de uso tienen una dependencia sobre otros casos de uso con objeto de determinar la secuencia en que es necesario desarrollar los casos de uso.

Si usamos el negocio bancario como ejemplo, el caso de uso

Hacer un retiro no puede ejecutarse hasta que haya ocurrido el caso de uso.

<sup>52</sup>IBM corp. *Directriz: Modelo de caso de uso*, 2016, Recuperado de [https://cgrw01.cgr.go.cr/rup/RUP.es/LargeProjects/core.base\\_rup/guidances/guidelines/use\\_case\\_model\\_C121CF4.html](https://cgrw01.cgr.go.cr/rup/RUP.es/LargeProjects/core.base_rup/guidances/guidelines/use_case_model_C121CF4.html)

- Abrir una cuenta bancaria.

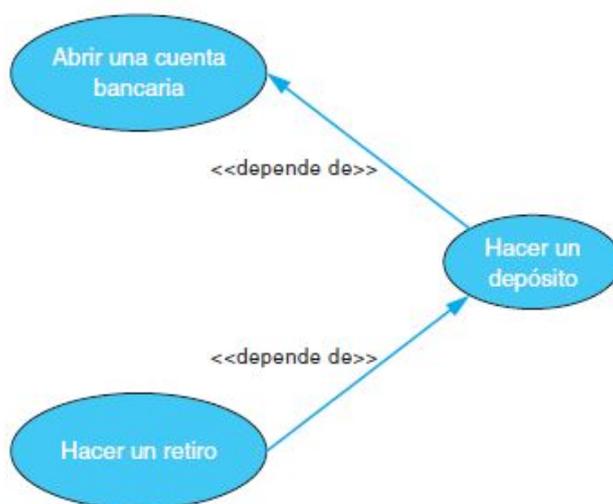
Debido a estas dependencias, el equipo de desarrollo muy probablemente escogerá desarrollar el caso de uso

- Abra una cuenta bancaria primero,
- en segundo lugar el caso de uso Haga un depósito,
- y en tercer lugar el caso de uso Haga un retiro para los propósitos de condiciones de uso y pruebas.

Un diagrama de casos de uso que modele las dependencias de caso de usos del sistema mediante el uso de la relación de dependencia proporciona un modelo que es una herramienta excelente para propósitos de planeación y de programación.<sup>53</sup>

La relación de dependencia tal como se presenta en la **figura 2.2.4** se ilustra con una línea con cabeza de flecha (ya sea continua o segmentada) que comienza en un caso de uso y que apunta al caso de uso del cual depende.

La línea de relación de dependencia se rotula “«depende de»”.



**Figura 2.3.4:** Ejemplo de una relación de dependencia

## ■ Herencia

---

<sup>53</sup>Ibidem

Cuando dos o más actores comparten un comportamiento común (en otras palabras, pueden iniciar el mismo caso de uso) lo mejor es extrapolar este comportamiento común y asignarlo a un nuevo actor resumen con objeto de reducir la comunicación redundante con el sistema.<sup>54</sup>

Por ejemplo, un usuario de una biblioteca es un miembro que tiene una tarjeta que está autorizado a “Buscar en el inventario de la biblioteca” así como a “Sacar libros en préstamo” de la biblioteca.

Como muchas bibliotecas son instituciones públicas, ellos dan la bienvenida a los visitantes para que usen sus servicios en el lugar tales como “Buscar en el inventario de la biblioteca” pero a los visitantes no se les permite la extensión de servicios (tal como “Sacar libros en préstamo”) que están reservados para los usuarios.

Al crear un actor resumen llamado cliente, del cual van a heredar el usuario y el visitante, tenemos que modelar la relación solamente una vez iniciando el caso de uso Buscar en el inventario de la biblioteca.<sup>55</sup>

En el diagrama de caso de uso se ilustra la relación herencia con el tipo de flecha mostrado en la sección “Después” de la **figura 2.2.5**.

### ***2.3.13. Modelo Entidad-Relación***

Un diagrama entidad-relación, también conocido como modelo entidad relación o ERD, es un tipo de diagrama de flujo que ilustra cómo las entidades, como personas, objetos o conceptos, se relacionan entre sí dentro de un sistema.<sup>56</sup> Los diagramas ER se usan a menudo para diseñar o depurar bases de datos relacionales en los campos de ingeniería de software, sistemas de información empresarial, educación e investigación.

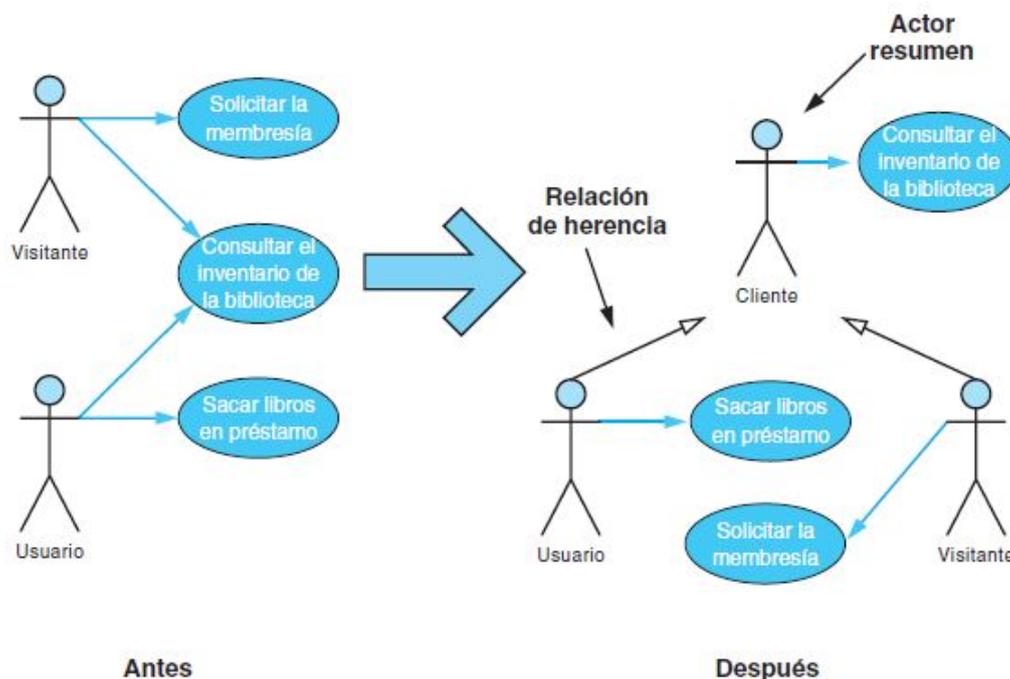
También conocidos como los ERD o modelos ER, emplean un conjunto definido de símbolos, tales como rectángulos, diamantes, óvalos y líneas de conexión para representar

---

<sup>54</sup>Ibidem

<sup>55</sup>Ibidem

<sup>56</sup>Ibidem



**Figura 2.3.5:** Ejemplo de una relación de herencia

la interconexión de entidades, relaciones y sus atributos. Son un reflejo de la estructura gramatical y emplean entidades como sustantivos y relaciones como verbos.<sup>57</sup>

Los componentes y las características de un diagrama ER Los diagramas ER se componen de entidades, relaciones y atributos. También representan la cardinalidad, que define las relaciones en términos de números. A continuación definimos cada elemento:<sup>58</sup>

- **Entidad:** Algo que se puede definir, como una persona, objeto, concepto u evento, que puede tener datos almacenados acerca de este. Piensa en las entidades como si fueran sustantivos. Por ejemplo: un cliente, estudiante, auto o producto. Por lo general se muestran como un rectángulo.
- **Tipo de entidad:** Un grupo de cosas que se pueden definir, como estudiantes o atletas, mientras que la entidad sería el estudiante o atleta específico. Otros ejemplos son clientes, autos o productos.
- **Conjunto de entidades:** Es igual que un tipo de entidad, pero se define en un

<sup>57</sup>Fundamento de las bases de datos: Modelo entidad-relación (2013) Recuperado de <https://www.genbeta.com/desarrollo/fundamento-de-las-bases-de-datos-modelo-entidad-relacion>

<sup>58</sup>Ibidem

momento determinado, como por ejemplo estudiantes que se inscribieron en una clase el primer día. Otros ejemplos son clientes que realizaron una compra en el último mes o autos registrados actualmente en Florida. Un término relacionado es una instancia, en la que una persona determinada o un auto específico podría ser una instancia del conjunto de entidades.

- **Categorías de entidades:** Las entidades se clasifican en fuertes, débiles o asociativas. Una entidad fuerte se puede definir únicamente por sus propios atributos, en cambio, una entidad débil no. Una entidad asociativa es aquella que relaciona entidades (o elementos) dentro de un conjunto de entidades.
- **Claves de entidad:** Se refiere a un atributo que únicamente define una entidad en un conjunto de entidades. Las claves de entidad se dividen en superclave, clave candidata o clave primaria.
  - **Superclave:** un conjunto de atributos (uno o más) que juntos definen una entidad en un conjunto de entidades.
  - **Clave candidata:** es una superclave mínima, es decir, contiene el menor número posible de atributos para seguir siendo una superclave. Un conjunto de entidades puede tener más de una clave candidata.
  - **Clave primaria:** es una clave candidata seleccionada por el diseñador de la base de datos para identificar únicamente al conjunto de entidades.
  - **Clave extranjera:** identifica la relación entre las entidades.
- **Relación:** Cómo las entidades interactúan o se asocian entre sí. Piensa en las relaciones como si fueran verbos. Por ejemplo, el estudiante mencionado podría inscribirse en un curso. Las dos entidades serían el estudiante y el curso, y la relación representada es el acto de inscribirse, que conecta ambas entidades de ese modo. Las relaciones se muestran, por lo general, como diamantes o etiquetas directamente en las líneas de conexión.<sup>59</sup>

---

<sup>59</sup>Ivar Jacobson, Ian Spence, Kurt Bittner. *Casos de uso 2.0, la guía para ser exitoso con los casos de uso*, Estados Unidos, IVAR JACOBSON INTERNATIONAL SA, 2013.

- **Relación recursiva:** La misma entidad participa más de una vez en la relación.<sup>60</sup>
- **Atributo:** Una propiedad o característica de una entidad. A menudo se muestra como un óvalo o círculo.
- **Atributo descriptivo:** Una propiedad o característica de una relación (frente a una entidad).
- **Categorías de los atributos:** Los atributos se clasifican en simples, compuestos y derivados, así como de valor único o de valores múltiples. Simples: significa que el valor del atributo es mínimo y ya no puede dividirse, como un número de teléfono. Compuestos: los subatributos surgen de un atributo. Derivados: los atributos se calculan o derivan de otro atributo, por ejemplo, la edad se calcula a partir de la fecha de nacimiento.
- **Valores múltiples:** Se denota más de un valor del atributo, como varios números de teléfono para una persona. Valor único: contienen solo un valor de atributo. Los tipos se pueden combinar, por ejemplo, puede haber atributos de valor único simples o atributos de múltiples valores compuestos.<sup>61</sup>
- **Cardinalidad:** Define los atributos numéricos de la relación entre dos entidades o conjuntos de entidades. Las tres relaciones cardinales principales son: uno a uno, uno a muchos y muchos a muchos. Un ejemplo de uno a uno sería un estudiante asociado a una dirección de correo electrónico. Un ejemplo de uno a muchos (o muchos a uno, en función de la dirección de la relación) sería un estudiante que se inscribe en muchos cursos, y todos esos cursos se asocian a ese estudiante en particular. Un ejemplo de muchos a muchos sería los estudiantes en grupo están asociados a múltiples miembros de la facultad y a su vez los miembros de la facultad están asociados a múltiples estudiantes.<sup>62</sup>
- **Vistas de cardinalidad:** La cardinalidad puede estar del lado opuesto o del mismo,

---

<sup>60</sup>Ibidem

<sup>61</sup>Ibidem

<sup>62</sup>Fundamento de las bases de datos: Modelo entidad-relación (2013) Recuperado de <https://www.genbeta.com/desarrollo/fundamento-de-las-bases-de-datos-modelo-entidad-relación>

en función de dónde se muestran los símbolos.

- **Restricciones de cardinalidad:** Los números máximos o mínimos que se aplican a una relación.

### 2.3.14. Diagrama de despliegue

Diagrama de Despliegue es un tipo de diagrama del Lenguaje Unificado de Modelado que muestran las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos.<sup>63</sup>

Los diagramas de despliegue son los complementos de los diagramas de componentes que, unidos, proveen la vista de implementación del sistema.<sup>64</sup>

Describen la topología del sistema la estructura de los elementos de hardware y el software que ejecuta cada uno de ellos.<sup>65</sup>

Los diagramas de despliegue representan a los nodos y sus relaciones.<sup>66</sup>

Los nodos son conectados por asociaciones de comunicación tales como enlaces de red, conexiones TCP/IP.<sup>67</sup>

#### 2.3.14.1. Ventajas y Desventajas

##### Ventajas

- Muestra un conjunto de nodos y sus relaciones.
- Se utilizan para describir la vista de despliegue estática de un sistema.
- Se relacionan con los diagramas de componentes, ya que un nodo incluye normalmente uno o mas componentes.<sup>68</sup>

##### Desventajas

---

<sup>63</sup>ECURED. *Diagrama de despliegue*, Recuperado de [https://www.ecured.cu/Diagrama\\_de\\_despliegue](https://www.ecured.cu/Diagrama_de_despliegue)

<sup>64</sup>Ibidem

<sup>65</sup>JBenaQuir. *Diagramas de Despliegue*, Enero, 2012. Disponible:<http://www.diadspg.blogspot.com/>

<sup>66</sup>Ibidem

<sup>67</sup>Ibidem

<sup>68</sup>Ingenieriasoftwaredos. *Diagrama de Despliegue*. Enero, 2012. Disponible:<http://www.ingenieriasoftwaredos.wikispaces.com>

- La posible falla en la modelación de un hardware.
- Tales sistema contienen a menudo varias versiones de componentes software, en algunos de los cuales pueden incluso emigrar de un nodo a otro.
- El diseño de tales sistema requieren tomar decisiones que permitan un cambio continuo de la topología del sistema.<sup>69</sup>

### 2.3.15. Diagrama de Secuencia

El diagrama de secuencia es un tipo de diagrama del lenguaje unificado de modelado (UML) que, a su vez, se trata de un lenguaje orientado a objetos y está compuesto por elementos gráficos.<sup>70</sup>

UML modela sistemas y procesos de la programación orientada a objetos así como procesos de negocio con el objetivo de presentar asuntos complejos de manera clara. Para ello, UML establece una notación estandarizada y recurre a formas visuales para representar un componente o comportamiento específico.<sup>71</sup>

El llamado metamodelado define las unidades lingüísticas y su significado dentro de UML y determina cómo pueden interactuar ciertos elementos entre sí y qué jerarquías existen entre las respectivas unidades lingüísticas.<sup>72</sup>

Un diagrama de secuencia es un tipo de diagrama de interacción porque describe cómo —y en qué orden— un grupo de objetos funcionan en conjunto. Tanto los desarrolladores de software como los profesionales de negocios usan estos diagramas para comprender los requisitos de un sistema nuevo o documentar un proceso existente.<sup>73</sup>

A los diagramas de secuencia en ocasiones se los conoce como diagramas de eventos o escenarios de eventos. De acuerdo a *Lucidchart*, **"hay dos tipos de diagramas de**

---

<sup>69</sup>Ibidem

<sup>70</sup>CEBALLOS, Karla. *UML: Diagrama de Secuencia*, Portafolio Digital, 2015. Recuperado de <https://ingsoftwarekarlacevallos.wordpress.com/2015/07/07/uml-diagrama-de-secuencia/>

<sup>71</sup>Ibidem

<sup>72</sup>Digital Guide. *Diagramas de secuencia: mostrar interacciones con UML*, IONOS, 2019. Recuperado de: <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/diagramas-de-secuencia/>

<sup>73</sup>Ibidem

**secuencia: los diagramas UML y los diagramas que se basan en código."**<sup>74</sup>

### 2.3.15.1. Principales elementos del diagrama de secuencia

El diagrama de secuencias hace parte de los diagramas de interacción de la especificación UML 2.1.1 que describen los aspectos dinámicos de un sistema y muestran la interacción entre los objetos de un sistema y los mensajes enviados entre ellos, ordenados según su secuencia en el tiempo.<sup>75</sup>

Los principales elementos del diagrama de secuencias especificados por Fowler y Scott (1997 y 2003) para la versión 2.0 y que siguen siendo válidos en la Superestructura UML 2.1.1 (OMG, 2007) se pueden apreciar en la tabla 2.3.1.

La semántica de los fragmentos combinados (FC) depende del operador de interacción, que contiene un cierto número de operandos y un identificador. Mediante los operadores, se pueden definir, entre otros:<sup>76</sup>

- *Alternativa* (denotado “alt”), que modela estructuras if... then... else.
- *Opción* (denotado “opt”), que modela estructuras switch. Una opción es semánticamente equivalente a un fragmento combinado alternativo donde hay un operando con contenido vacío y otro no vacío.
- *Opción* (denotado “opt”), que modela estructuras switch. Una opción es semánticamente
- *Quiebre de secuencia* (denotado por “break”), que modela una secuencia alternativa de eventos, que se procesa en lugar de todo el resto del diagrama.
- *Paralelo* (denotado por “par”), que modela procesos concurrentes.
- *Ciclo* (denotado por “loop”), el cual incluye una serie de mensajes que se repiten.

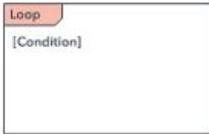
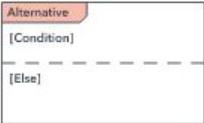
---

<sup>74</sup>Lucidchart. *Diagrama de secuencias*, Rescatado de: <https://www.lucidchart.com/pages/es/diagrama-de-secuencia>

<sup>75</sup>Ibidem

<sup>76</sup>ZAPATA, Carlos Mario; GARCÉS, Gilma Liliana. *Generación del diagrama de secuencias de UML 2.1.1 desde esquemas preconceptuales*, 2008

Símbolo	Nombre	Descripción
	Símbolo de objeto	Representa una clase u objeto en UML. El símbolo objeto demuestra cómo se comportará un objeto en el contexto del sistema. Los atributos de las clases no deben aparecer en esta figura.
	Casilla de activación	Representa el tiempo necesario para que un objeto finalice una tarea. Cuanto más tiempo lleve la tarea, más larga será la casilla de activación.
	Símbolo de actor	Muestra entidades que interactúan con el sistema o que son externas al sistema.
	Símbolo de paquete	Se usa en notación UML 2.0 para contener los elementos interactivos del diagrama. También conocida como "marco", esta figura rectangular tiene un pequeño rectángulo interior para etiquetar el diagrama.
	Símbolo de paquete	Se usa en notación UML 2.0 para contener los elementos interactivos del diagrama. También conocida como "marco", esta figura rectangular tiene un pequeño rectángulo interior para etiquetar el diagrama.

Símbolo	Nombre	Descripción
	Símbolo de línea de vida	Representa el paso del tiempo a medida que se extiende hacia abajo. Esta línea vertical discontinua representa eventos secuenciales que le ocurren a un objeto durante el proceso graficado. Las líneas de vida pueden comenzar con una figura rectangular etiquetada o un símbolo de actor.
	Símbolo de bucle de opción	Se emplea para modelar escenarios del tipo "Si... entonces...", es decir, una circunstancia que solo sucederá en determinadas condiciones.
	Símbolo de alternativas	Simboliza una decisión (que, por lo general, es mutuamente exclusiva) entre dos o más secuencias de mensajes. Para representar alternativas, emplea la figura rectangular etiquetada con una línea discontinua en su interior.
	Símbolo de mensaje sincrónico	Representados por una línea continua y una punta de flecha sólida. Este símbolo se utiliza cuando un remitente debe esperar una respuesta a un mensaje antes de proseguir. El diagrama debe mostrar el mensaje y la respuesta.

Símbolo	Nombre	Descripción
	Símbolo de mensaje asincrónico	Representados por una línea continua y una punta de flecha simple. Los mensajes asincrónicos no necesitan una respuesta para que el remitente siga adelante. Solo la llamada se debe incluir en el diagrama.
	Símbolo de mensaje de respuesta asincrónico	Representados por una línea discontinua y una punta de flecha simple.
	Símbolo de crear mensaje asincrónico	Representados por una línea discontinua y una punta de flecha simple. Este mensaje crea un nuevo objeto.
	Símbolo de mensaje de respuesta	Están representados con una línea discontinua y una punta de flecha simple. Estos mensajes son las respuestas a las llamadas.
	Símbolo de eliminar mensaje	Están representados por una línea continua y una punta de flecha sólida, seguida de una X. Este mensaje destruye un objeto.

**Tabla 2.3.1:** Principales elementos del diagrama de secuencias

### **2.3.16. *Estimación de costo de un sistema de software***

Todo proyecto de desarrollo de software comienza con una etapa de estimación y planificación, en la cual debemos determinar, a partir de los requerimientos del software, cuánto esfuerzo, personal, recursos materiales, tiempo y en última instancia dinero tomará construir el sistema o producto.<sup>77</sup>

No solo los gerentes de proyectos estiman software, pues a cada desarrollador, analista de sistemas y software tester se le suele asignar un componente y se le pregunta cuánto tiempo le tomará elaborarlo.<sup>78</sup>

Estas estimaciones suelen elaborarse de forma empírica, en base a las experiencias pasadas y conocimiento del trabajo a realizar, sin embargo, suelen ser imprecisas debido a que los requerimientos y sus implicaciones técnicas no se conocen en su totalidad y además ningún desarrollo de software es exactamente igual a uno anterior con el que se pueda comparar.

Frente a esto, se han desarrollado métodos como el de puntos de función, donde estimamos las funcionalidades usando fórmulas matemáticas basadas en parámetros como el tipo de componente, complejidad, factores del entorno, entre otros aspectos.<sup>79</sup>

#### **2.3.16.1. Método punto función**

Es una técnica de estimación de software desarrollada originalmente por Allan Albrecht en 1979 mientras trabajaba para IBM, quien definió conceptos para medir el software a partir de valoraciones de funcionalidades entregadas al usuario y no a partir de aspectos técnicos, con la intención de producir valoraciones independientes de la tecnología y fases del ciclo de vida utilizado.<sup>80</sup>

El trabajo de Albrecht fue continuado por el grupo internacional de usuarios de

---

<sup>77</sup>PMOinformatica,2015,<http://www.pmoinformatica.com/2015/04/estimacion-puntos-funcion-introduccion.html>

<sup>78</sup>Idem

<sup>79</sup>Idem

<sup>80</sup>PMOinformatica, 2015, <http://www.pmoinformatica.com>

puntos de función, quienes plasmaron sus conceptos en el método IFPUG-FPA.

IFPUG-FPA realiza las valoraciones a partir de la funcionalidad del sistema, primero clasificándolas, luego asignando una complejidad y ponderación a cada una según unas tablas predefinidas, determinando así el valor de puntos de función.

Sumando los puntos de todas las funcionalidades se obtiene la valoración de todo el proyecto y finalmente se puede aplicar un factor de ajuste, que puede depender de características generales del sistema como por ejemplo requerimientos no funcionales como el rendimiento, reusabilidad, facilidad de instalación y operación entre otros aspectos.

Los puntos de función permiten traducir el tamaño de funcionalidades de software a un número, a través de la sumas ponderadas de las características que este tiene.

Una vez que tenemos los puntos de función, podemos traducirlos en horas hombre o días de trabajo, según factor de conversión que dependería de mediciones históricas de nuestra productividad. Con las horas hombre, podemos determinar el costo y presupuesto de los proyectos.

#### **2.3.16.2. Utilidades del método de puntos de función**

- **Estimación de proyectos de software**

Una de las principales aplicaciones del método es en la determinación de valoraciones (estimaciones) del producto de software a desarrollar, que es una parte fundamental de todo proceso de ingeniería de software y de la gerencia de estos proyectos. Dentro del ciclo de ingeniería de software, lo normal es que la estimación del proyecto ocurra después que la ingeniería de requerimientos produzca una primera versión de la especificación de requisitos con suficiente información para elaborar la estimación.

- **Validar la calidad de las especificaciones funcionales**

Si asignamos el mismo requisito a dos Analistas de sistemas y la estimación producida por los puntos de función difiere, esto nos puede servir para identificar ambigüedades en las especificaciones funcionales.

- **Seguimiento y control de proyectos**

Otras aplicaciones de los puntos de función incluyen el seguimiento y control de proyectos, donde se puede usar para estimar el impacto de cambios en la funcionalidad solicitados durante el desarrollo y también para medir en puntos de función el avance del proyecto e inclusive la facturación en contratos de servicios.

- **Medir la productividad y calidad de nuestro proyecto o servicio de ingeniería de software**

Por otra parte, los puntos de función tienen aplicaciones en mediciones de productividad, por ejemplo si dividimos las horas hombre empleadas para producir cierta cantidad de puntos de función, obtenemos un indicador de productividad de nuestro equipo y de cómo esta puede mejorar en el tiempo. Con esto adicionalmente podemos calibrar el factor de conversión entre puntos función y horas hombre que utilicemos para nuestras estimaciones. Otros indicadores interesantes son número de incidencias por puntos de función producidos como medición de la calidad del proceso de desarrollo.

### **2.3.16.3. Método de estimación por Puntos de Casos de Uso**

El método de Punto de Caso de Uso (UCP – Use Case Point), está basado en los tradicionales Puntos Función. Es un método originado de la tesis de master de Gustav Karner (Karner, 1993), desarrollada mientras trabajaba en Objectory AB, bajo supervisión de Ivar Jacobson (creador de los casos de uso). La técnica ha sido usada por la empresa Rational (posteriormente adquirida por IBM) durante varios años y con buenos resultados. Además la técnica se ha documentado en varias publicaciones (Carroll, 2005; Clemmons, 2006; Karner, 1993; Nageswaran, 2007 ).<sup>81</sup>

Puntos de caso de uso es un método de estimación de esfuerzo para proyectos de software, a partir de sus casos de uso. Este método utiliza los actores y casos de uso relevados para calcular el esfuerzo que significará desarrollarlos. A los casos de uso se les

---

<sup>81</sup>Javier Garzás, <http://233gradosdeti.com/articulos/metodo-de-estimacion-de-puntos-de-caso-de-uso/>

asigna una complejidad basada en transacciones, entendidas como una interacción entre el usuario y el sistema, mientras que a los actores se les asigna una complejidad basada en su tipo, es decir, si son interfaces con usuarios u otros sistemas. También se utilizan factores de entorno y de complejidad técnica para ajustar el resultado.<sup>82</sup>

**El método de punto de casos de uso consta de cuatro etapas, en las que se desarrollan los siguientes cálculos:**

1. Factor de peso de los actores sin ajustar (UAW)
2. Factor de peso de los casos de uso sin ajustar (UUCW)
3. Puntos de caso de uso ajustados (UCP)
4. Esfuerzo horas-hombre

### **2.3.17. Linode**

Linode Object Storage es un almacenamiento compatible con S3, altamente disponible y fácilmente escalable para cualquiera de sus necesidades de respaldo, big data y archivo de datos. Con Object Storage, no necesita tener ni utilizar una máquina virtual Linode para beneficiarse del almacenamiento de sus archivos o ampliar su capacidad. Linode Object Storage está disponible de inmediato en nuestro centro de datos de Newark y se implementará en las regiones restantes a partir de los próximos meses.

El almacenamiento de objetos Linode comienza en \$ 5 / mes incluye 250GB y 1TB de transferencia saliente, y luego el almacenamiento y la transferencia por encima de eso son \$ 0.02 / GB. Los cubos escalan hasta 10 TB por cliente, por grupo.<sup>83</sup>

Linode es una empresa de Cloud hosting (hosting en la nube) fué fundada en 2003 y durante estos años ha pasado de ser una pequeña compañía con un datacenter en California a tener centros de datos por todo el mundo y ya tiene 400.000 clientes.

Cloud hosting es un VPS (Virtual Private Server) no administrado, esto quiere

---

<sup>82</sup>Revista Avanzada Científica Vol. 13 No.2 Año 2010, pag 2-3

<sup>83</sup>Hillary Wilmoth, <https://www.linode.com/blog/cloud-storage/linode-object-storage-is-here/>, Estados Unidos, 2019

decir que tienes unos recursos disponibles en un servidor y los puedes manejar como tú quieras, en el caso de Linode se escogen los recursos, el sistema operativo que deseas usar (varias distribuciones de Linux) y el lugar de tu servidor (EE.UU, Europa o Asia).

Para usar Linode debes tener experiencia en Linux, si deseas tener una página web pero no conoces Linux te recomiendo un hosting compartido como este ya que obtendrás el panel de administración cPanel, mucho más fácil de usar. <sup>84</sup>

---

<sup>84</sup><https://www.comprarhosting.co/opiniones-linode-com-espanol/>, Opiniones de Linode en español. Cloud hosting

## 2.4. Herramientas para el desarrollo del sistema

### 2.4.1. *Sistema operativo requerido para Servidor Web*

Los sistemas operativos permiten la interacción con el ordenador y darle instrucciones, también es el encargado de administrar los recursos, como el hardware y software su estructura permite manejar los programas instalados, cada ordenador debe tener un sistema que lo opere de lo contrario la utilización de este sería inútil para desempeñar tareas cotidianas del día a día. Entre los sistemas más comunes que existen tenemos Microsoft Windows, Mac OS X y Linux. Donde el recomendado para Servidor web es Debían siendo este una de las muchas distribuciones de GNU/Linux.

Siendo Debían una distribución de Linux proporciona una implementación completa del software de red TCP/IP e incluye un ambiente de programación completo que proporciona todas las librerías estándar, herramientas de programación, compiladores y depuradores que se esperarían de otros sistemas UNIX. Existe una gran variedad de software comercial disponible para Linux. En estos tiempos, se puede adquirir cualquier tipo de aplicación, como Motif, que es una interfaz para el sistema X Window similar a Microsoft Windows, WordPerfect (el popular procesador de palabras) o Maple, que es un paquete que realiza complejas manipulaciones simbólicas, para Linux. Linux soporta consolas virtuales (VC), que son una manera de hacer que una máquina aparezca como múltiples terminales, todos conectados al mismo núcleo Linux.<sup>85</sup>

### 2.4.2. *Sistema operativo requerido para Cliente Web*

El Sistema Informático de Registro, Emisión e Inspección de Credenciales Sanitarias Otorgadas a Los Manipuladores de Alimentos en La Unidad Comunitaria de Salud Familiar de San Francisco Gotera en El Departamento de Morazán, será multiplataforma es decir debe ser soportado para cualquier sistema operativo, excepto para la aplicación móvil que trabajará solo con sistema operativo android.

<sup>85</sup>Anonimo. (s.f.). Sistemas Operativos Para Cliente/Servidor. Extraído de [www.oocities.org/siliconvalley/8195/noscs.html](http://www.oocities.org/siliconvalley/8195/noscs.html)

### 2.4.3. *HTML*

HTML por sus siglas HyperText Markup Language (Lenguaje de marcas de hipertexto) es el lenguaje que lee el navegador, esta estructurado con etiquetas donde cada una es utilizada para contener información y darle cierto significado, dependiendo de la etiqueta con la que se trate. cabe destacar que las etiquetas pueden utilizarse para hacer referencia a todo tipo de información. Las tres características principales de HTML5 es proveer estructura, estilo y funcionalidad. Uno de los principales objetivos es introducir información de forma semántica y no visual, es decir que todos los apartados visuales deben dejarse para el apartado de presentación, que es gestionado con el lenguaje CSS.<sup>86</sup>

### 2.4.4. *CSS*

CSS (Cascading Style Sheets - Hojas de Estilo en Cascada), oficialmente nada tiene que ver con HTML, sino más bien es un complemento que en conjunto con HTML supera las limitaciones de este y la complejidad en la estructura. Inicialmente HTML permitía aplicar estilos esenciales dentro de sus etiquetas, pero a medida que el lenguaje fue evolucionando fue más complejo para los diseñadores la implementación e interpretación del código.

CSS es un lenguaje de hojas de estilos, que fue creado para controlar los aspectos de presentación en documentos HTML, donde separamos la estructura del diseño disminuyendo la complejidad, interpretación, accesibilidad y compatibilidad visual ya que es compatible con distintos dispositivos.<sup>87</sup>

### 2.4.5. *JavaScript*

JavaScript es un lenguaje interpretado orientado a objetos, usado para añadir características interactivas a un documento HTML y usado para crear interactividad dinámica en los sitios web. Este lenguaje fue inventado por Brendan Eich, co-fundador

---

<sup>86</sup>J. D Gauchat. (2012). El gran libro de HTML5, CSS3 y Javascript Primera edición

<sup>87</sup>uniwebsidad. (s.f.). Introducción a CSS. Recuperado de [uniwebsidad.com/libros/css/capitulo-1](http://uniwebsidad.com/libros/css/capitulo-1)

del proyecto Mozilla, Mozilla Foundation y la Corporación Mozilla.

Con JavaScript se puede hacer casi cualquier cosa. como carruseles, galerías de imágenes, diseños fluctuantes, entre otras. Aunque JavaScript es usado para el desarrollo de sitios web, es usado para entornos sin navegador, tales como Adobe Acrobat, node.js, Apache CouchDB entre otros. Es un lenguaje script multi-paradigma, basado en prototipos, dinámico, soporta estilos de programación funcional, orientada a objetos.<sup>88</sup>

#### **2.4.6. PHP**

PHP(Hypertext Preprocessor) es un lenguaje de código abierto utilizado especialmente en el desarrollo Web, se ejecuta en el servidor procesando las peticiones realizadas por el cliente y devolviendo el resultado al navegador.<sup>89</sup>

Tiene la capacidad para: controlar el acceso de usuarios, procesar datos de formulario, conexión a la base de datos (abrir, crear, leer, escribir y cerrar ficheros con el servidor), encriptación de datos, generar páginas dinámicas, hacer cookies entre otras.

Se ha seleccionado este lenguaje de programación gracias a sus características:

- amplia documentación
- soporte de la comunidad
- se puede hacer uso de módulos externos, para mejorar la aplicación
- permite separación de estructura brindando más orden
- permite programación orientada a objetos.

#### **2.4.7. MVC**

El MVC o Modelo-Vista-Controlador es un patrón de arquitectura de software que, hace uso de 3 componentes (Vistas, Modelos y Controladores) separa la lógica de la

---

<sup>88</sup>Mozilla. (2020). JavaScript (JS). Recuperado de <https://developer.mozilla.org/es/docs/Web/JavaScript>

<sup>89</sup>Juan Pavón Mestras. (2012). Aplicaciones Web/Sistemas Web

aplicación de la vista en una aplicación. Es una arquitectura importante puesto que se utiliza tanto en componentes gráficos básicos hasta sistemas empresariales; la mayoría de los frameworks modernos utilizan MVC (o alguna adaptación del mismo).<sup>90</sup>

Se utilizará MVC para mantener ordenado el proyecto ya que separa la lógica del proyecto de la vista y poder acoplar a cada integrante del equipo de desarrollo con la asignación de un módulo en el cual estará trabajando hasta completar todos los componentes correspondientes al MVC.

A continuación, se describen cada uno de los elementos que conforman este framework.

#### ■ **Modelo**

Este será el encargado de proporcionarnos los datos por lo general es el que nos proporciona todas las consultas con la base de datos (actualizaciones, consulta, inserciones, búsquedas y eliminación de registros).

#### ■ **Controlador**

Es el que llevará la lógica recibiendo las órdenes del usuario y manejar la información; También sirve como puente al modelo y transmite los resultados a la vista.

#### ■ **Vistas**

La vista es donde se representa toda la información, proporcionada por el modelo y manejada por el controlador, haciendo de tal manera que los datos sean completamente comprensibles para el usuario que los solicita.

### **2.4.8. *GitHub***

Este será nuestro sistema que gestionará el proyecto, ya que ofrece control de versiones de código y entre sus bondades para los desarrolladores es ser utilizada como una red social, que permite el trabajo en equipo colaborando con personas de distintos

---

<sup>90</sup>Uriel Hernandez. (2015). MVC (Model, View, Controller). Extraído de [códigofacilito.com](http://códigofacilito.com)

lugares, este sistema está especializado para desarrolladores utiliza repositorios que son directorios donde son alojados los proyectos.

Git es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones. Destaca su gestión distribuida. Git le da a cada programador una copia local del historial del desarrollo entero, y los cambios se propagan entre los repositorios locales. Los cambios se importan como ramas adicionales y pueden ser fusionados en la misma manera que se hace con la rama local.

Git fue creado pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente, es decir Git nos proporciona las herramientas para desarrollar un trabajo en equipo de manera inteligente y rápida y por trabajo nos referimos a algún software o página que implique código el cual necesitemos hacerlo con un grupo de personas.<sup>91</sup>

Algunas de las características más importantes de Git son:

- Rápidez en la gestión de ramas, debido a que Git nos dice que un cambio será fusionado mucho más frecuentemente de lo que se escribe originalmente.
- Gestión distribuida; Los cambios se importan como ramas adicionales y pueden ser fusionados de la misma manera como se hace en la rama local.
- Gestión eficiente de proyectos grandes.
- Re-almacenamiento periódico en paquetes.

#### ***2.4.9. Visual Studio Code***

Visual Studio Code es un editor de código fuente ligero y potente fácil de instalar y está disponible para los Sistemas Operativos mas comunes (Windows, macOS y Linux). Viene con soporte incorporado para JavaScript, TypeScript y Node.js y tiene un rico ecosistema de extensiones para otros lenguajes (como C ++, C #, Java, Python, PHP,

---

<sup>91</sup>Andrés. (2015). Qué es Git. Extraído de [codigofacilito.com](http://codigofacilito.com)

Go) y tiempos de ejecución (como .NET y Unity).

Además, cuenta con una amplia variedad de extensiones que facilitan la implementación del código y atajos de teclado; se pretende la utilización de este editor de código por lo liviano que es y será utilizado tanto para el desarrollo del sistema de escritorio como la aplicación móvil, para este último se utilizarán extensiones del lenguaje de programación Dart y Flutter, que facilitan la escritura y eficiencia al momento de escribir el código en dichos lenguajes de programación.

#### **2.4.10. *PHPMYAdmin***

PhpMyAdmin es una herramienta gratuita, que nos brinda el acceso a funcionalidades de la base de datos con MySQL de una forma muy sencilla con su interfaz gráfica sencilla, fácil e intuitiva. Ésta creado en conjunto con lenguaje PHP y su interfaz se muestra mediante un navegador como una página web esto quiere decir que se aloja en nuestro servidor web y seremos capaces de acceder a ella a través de cualquier dispositivo.<sup>92</sup>

Estas son algunas de las características ofrecidas por esta herramienta:

- Esta aplicación nos permitirá realizar las operaciones básicas en base de datos MySQL, como son: crear y eliminar bases de datos, crear, eliminar y alterar tablas, borrar, editar y añadir campos, ejecutar sentencias SQL, administrar claves de campos, administrar privilegios y exportar datos en varios formatos. La función de exportar datos se emplea muchas veces para realizar backups de la base de datos y poder restaurar esta copia de seguridad en el futuro a través de phpMyAdmin mediante la opción “importar”.
- phpMyAdmin es el administrador de bases de datos por defecto en muchos paneles de control comerciales como son cPanel, Plesk o DirectAdmin.
- Otra de las funciones más importantes que nos ofrece es que permite optimizar y

---

<sup>92</sup>Juan Miguel Vergara Pineda. (2016). ¿Qué es? y ¿qué nos ofrece?, la herramienta phpMyAdmin Extraído de [www.coriaweb.hosting](http://www.coriaweb.hosting)

reparar tablas, las cuales son dos tareas de mantenimiento fundamentales.

- Nos da la posibilidad de realizar búsquedas en las base de datos, además de poder escribir nuestras propias consultas SQL de manera directa y ejecutarlas.
- Esta herramienta también es de gran ayuda para desarrolladores de aplicaciones que empleen MySQL, ya que permite depurar consultas y hacer tests de forma rápida y sencilla.

### **Especificaciones de phpMyAdmin**

- Proporciona una interfaz Web y maneja base de datos MySQL, MariaDB, Drizzle.
- Importación de datos desde CSV y SQL.
- Exporta datos a varios formatos: CSV, SQL, XML, PDF (vía la biblioteca TCPDF), ISO/IEC 26300 OpenDocument Text y Spreadsheet, Word, Excel, LaTeX y otros.
- Administración de múltiples servidores.
- Crea gráficos PDF del diseño de la base de datos.
- Crea consultas complejas usando Query-by-Example (QBE).
- Búsqueda global en una base de datos o un subconjunto de esta.
- Transforma datos almacenados a cualquier formato usando un conjunto de funciones predefinidas, tal como BLOB.

## 2.5. Herramientas para el desarrollo de la aplicación móvil

### 2.5.1. *Flutter*

Flutter es un framework especializado en el desarrollo de aplicaciones móviles con sistemas operativos Android e IOS, su versatilidad y eficiencia radica en la fácil creación de aplicaciones desde el mismo código base el cual es compatible con dichos sistemas operativos móviles, sin necesidad de reescribir su estructura o código fuente.

Se utilizará Flutter para la creación de las aplicaciones móviles porque ofrece características y beneficios que facilitarán el desarrollo de la aplicación; entre ellas tenemos:

- Ofrece una interfaz gráfica, donde visualizamos el diseño de nuestra aplicación.
- Las aplicaciones corren a 60fps, permitiendo ver animaciones fluidas y percibir un alto rendimiento.
- Permite el desarrollo de aplicaciones tanto para IOS como Android a partir de una base de código.
- Características de desarrollo ágil como el Hot reload, permitiendo visualizar cada cambio realizado a la aplicación.

### 2.5.2. *Dart*

**Dart** es un lenguaje de programación desarrollado por Google de código abierto, estructurado, flexible, orientado a objetos, basado en clases, con herencias simples, opcionalmente tipado. Actualmente, incluso compilado a JavaScript es superior a código JavaScript nativo en V8. <sup>93</sup>

Estas son algunas de las características de Dart que juntas lo hacen indispensable para Flutter:

---

<sup>93</sup>Moisés Belchín, librodart.es | blogdart, España, 2014

- Dart es AOT (Ahead Of Time) compilado en código nativo rápido y predecible, lo que permite que casi todo Flutter sea escrito en Dart. Esto no sólo hace que Flutter sea rápido, sino que prácticamente todo (incluidos todos los widgets) se puede personalizar.
- Dart también puede ser compilado JIT (Just In Time) para ciclos de desarrollo excepcionalmente rápidos y un flujo de trabajo que cambia el juego (incluyendo la popular sub-segunda Stateful Hot Reload - recarga en caliente con manejo de estado).
- Dart facilita la creación de animaciones y transiciones suaves que se ejecutan a 60fps. Dart puede hacer object allocation y garbage collection sin bloqueos. Y, al igual que JavaScript, Dart evita la programación anticipada y la memoria compartida (y, por lo tanto, los bloqueos). Debido a que las aplicaciones Flutter están compiladas en código nativo, no requieren un puente lento entre dominios (por ejemplo, JavaScript a nativo). También arrancan mucho más rápido.
- Dart permite que Flutter evite la necesidad de un lenguaje de diseño declarativo separado como JSX o XML, o constructores de interfaces visuales separados, porque el diseño declarativo y programático de Dart es fácil de leer y visualizar. Y con todo el diseño en un solo lenguaje y en un solo lugar, es fácil para Flutter proporcionar herramientas avanzadas que hacen que el diseño sea un juego de niños.
- Los desarrolladores han descubierto que Dart es particularmente fácil de aprender porque tiene características que son familiares para los usuarios de lenguajes estáticos y dinámicos.

No todas estas características son exclusivas de Dart, pero la combinación de ellas alcanza un punto óptimo que hace que Dart sea excepcionalmente poderoso para implementar Flutter. Tanto es así, que es difícil imaginar que Flutter sea tan poderoso como lo es sin Dart. <sup>94</sup>

---

<sup>94</sup>Wm Leler, <https://hackernoon.com/>, Estados Unidos, 2019

### 2.5.3. *Android SDK*

Las siglas SDK se refieren a “Software Development Kit” (Kit de Desarrollo de Software). El SDK de Android es una herramienta necesaria para cualquier desarrollador o aficionado que quiera crear su propia aplicación o modificar su dispositivo Android. Estas herramientas pueden ser utilizar tanto en Linux como en Windows o Mac OS X

Este Kit incluye un conjunto de herramientas para facilitar la tarea a los desarrolladores:

- Un depurador de código
- Biblioteca
- Emulador de Android
- Documentación
- Ejemplos de código
- Tutoriales

Las Actualizaciones del SDK están coordinadas con el desarrollo general de Android. El SDK soporta también versiones antiguas de Android, por si los programadores necesitan instalar aplicaciones en dispositivos ya obsoletos o más antiguos. Las herramientas de desarrollo son componentes descargables, de modo que una vez instalada la última versión, pueden instalarse versiones anteriores y hacer pruebas de compatibilidad <sup>95</sup>

### 2.5.4. *Android Studio*

Android Studio es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de apps para Android, basado en IntelliJ IDEA. Además del potente editor de códigos y las herramientas para desarrolladores de IntelliJ, Android Studio ofrece incluso más funciones que aumentan tu productividad cuando desarrollas apps para Android, como

---

<sup>95</sup> Andrés Cruz, <https://www.desarrollolibre.net/blog/android/>, 2015

las siguientes <sup>96</sup>:

- Un sistema de compilación flexible basado en Gradle
- Un emulador rápido y cargado de funciones
- Un entorno unificado donde puedes desarrollar para todos los dispositivos Android
- Aplicación de cambios para insertar cambios de códigos y recursos a la aplicación en ejecución sin reiniciar la aplicación
- Integración con GitHub y plantillas de código para ayudarte a compilar funciones de apps comunes y también importar código de muestra
- Variedad de marcos de trabajo y herramientas de prueba
- Herramientas de Lint para identificar problemas de rendimiento, usabilidad y compatibilidad de la versión, entre otros
- etc.

### 2.5.5. *Firebase*

Firebase se trata de una plataforma móvil creada por Google, cuya principal función es desarrollar y facilitar la creación de apps de elevada calidad de una forma rápida, con el fin de que se pueda aumentar la base de usuarios y ganar más dinero. La plataforma está subida en la nube y está disponible para diferente plataformas como iOS, Android y web. Contiene diversas funciones para que cualquier desarrollador pueda combinar y adaptar la plataforma a medida de sus necesidades.<sup>97</sup>

Firebase se inició cuando Google la compró en 2014, y seguidamente la fue mejorando mediante la compra del equipo de Divshot.<sup>98</sup>

La plataforma de Google para desarrolladores dispone de 3 planes de precio, diversificados por las necesidades que tenga el usuario. el primer plan es gratuito,

<sup>96</sup>DEVELOPERS, Introducción a Android Studio, <https://developer.android.com/studio/intro?hl=es-419>

<sup>97</sup>PÉREZ CARDONA, Manuel, Firebase, qué es y para qué sirve la plataforma de Google, 2016, <https://www.iebschool.com/blog/firebase>

<sup>98</sup>Idem

pero presenta varias limitaciones, sobre todo en el espacio de almacenamiento y las conexiones simultáneas, que puede provocar que en muchos casos no sea suficiente para un desarrollador. Si es así, existen los otros dos planes, el Plan “Frame” que son unos 22 euros al mes, y el Plan “Blaze” que su precio varía en función del consumo que se haga de la plataforma.<sup>99</sup>

### **2.5.6. Patrones de diseño**

#### **1. Patron singleton**

En ingeniería de software, singleton o instancia única es un patrón de diseño que permite restringir la creación de objetos pertenecientes a una clase o el valor de un tipo a un único objeto.

Su intención consiste en garantizar que una clase solo tenga una instancia y proporcionar un punto de acceso global a ella.

El patrón singleton se implementa creando en nuestra clase un método que crea una instancia del objeto solo si todavía no existe alguna. Para asegurar que la clase no puede ser instanciada nuevamente se regula el alcance del constructor (con modificadores de acceso como protegido o privado).

La instrumentación del patrón puede ser delicada en programas con múltiples hilos de ejecución. Si dos hilos de ejecución intentan crear la instancia al mismo tiempo y esta no existe todavía, solo uno de ellos debe lograr crear el objeto. La solución clásica para este problema es utilizar exclusión mutua en el método de creación de la clase que implementa el patrón.

Las situaciones más habituales de aplicación de este patrón son aquellas en las que dicha clase controla el acceso a un recurso físico único (como puede ser el ratón o un archivo abierto en modo exclusivo) o cuando cierto tipo de datos debe estar disponible para todos los demás objetos de la aplicación.

---

<sup>99</sup>Idem

Los críticos consideran al singleton como un anti-patrón utilizando en escenarios donde no es beneficioso, introduce restricciones innecesarias donde una única instancia de una clase no es realmente requerida y agrega un estado global en la aplicación.

El patrón singleton provee una única instancia global gracias a que:

- La propia clase es responsable de crear la única instancia.
- Permite el acceso global a dicha instancia mediante un método de clase.
- Declara el constructor de clase como privado para que no sea instanciable directamente.
- Al estar internamente autoreferenciada, en lenguajes como Java, el recolector de basura no actúa.

## 2. Patron Bloc

BLoC (Business Logic Components) es el patrón más recomendado y usado por la comunidad, ya que nos ofrece flexibilidad tanto para aplicaciones medianas como grandes.

Básicamente, es un sistema de gestión del estado, ayudando a la gestión y acceso de este. Existen patrones muy parecidos en otras tecnologías, como podría ser Redux en React. Consiste en tener un almacén central de estados, al cual podemos acceder desde cualquier punto de la aplicación para consultar, modificar o crear estados, donde dicho almacén es consultado por las interfaces de la aplicación.

Podemos entonces definir las siguientes ventajas:

- No es una librería de terceros, lo cual, podremos seguir utilizando el patrón, ya que no se tiene
- que mantener.
- Lógica, interfaces y control de estado separados.

- Es reactivo, por lo que no tenemos que realizar llamadas para dar a conocer que el estado ha
- cambiado.

En cuanto a las desventajas tenemos las siguientes:

- Conocimientos de `streams`<sup>38</sup> y `rxdart`<sup>39</sup> necesaria para la comprensión del modelo.
- Curva de aprendizaje alta si no se ha trabajado con los elementos anteriores.
- Crea mucha lógica y clases para poder realizar un control exhaustivo de los distintos estados.
- Complicado de utilizar en aplicaciones pequeñas.

## 2.6. Marco legal

### 2.6.1. *Artículos relacionados a la manipulación de alimentos*

#### 2.6.1.1. Educación sanitaria

**Art. 34.-** Las personas que laboran en la manipulación de alimentos deben estar autorizadas para realizar tal actividad, a través de los cursos que imparten los establecimientos de salud del MINSAL.<sup>100</sup>

- La capacitación debe incluir como mínimo los temas siguientes: Buenas Prácticas de Manufactura de Alimentos, en adelante BPM, microbios y parásitos, limpieza y desinfección, manejo y conservación de los alimentos, hábitos higiénicos, enfermedades transmitidas por los mismos.
- La educación sanitaria debe realizarse de manera continua y permanente para todo el personal, y debe estar documentada con medios de verificación.
- El personal nuevo debe recibir la educación sanitaria antes de iniciar labores de manipulación de alimentos.
- El MINSAL debe autorizar a los manipuladores de alimentos capacitados mediante la entrega individual de carné o diploma, previo cumplimiento de asistencia al curso, aprobación del mismo y exámenes de salud.

#### 2.6.1.2. Salud del manipulador

**Art. 35.-** El manipulador de alimentos debe someterse a exámenes generales de heces y de orina, así como a los que el médico indique cada seis meses.<sup>101</sup>

---

<sup>100</sup>Norma técnica de alimentos capítulo 7 (febrero 2013)

<sup>101</sup>Idem

### **2.6.2. Ley de propiedad intelectual**

Según el art 1 de dicha ley lo que se dispone tiene como objeto asegurar una protección suficiente y efectiva de la propiedad intelectual, estableciendo las bases que la promuevan, fomenten y protejan.

Esta ley comprende el derecho de autor, los derechos conexos y la propiedad industrial en lo relativo a invenciones, modelos de utilidad, diseños industriales y secretos industriales o comerciales y datos de prueba.

Según el art. 7 el derecho económico del autor es el derecho exclusivo de autorizar o prohibir el uso de sus obras, así como la facultad de percibir beneficios económicos provenientes de la utilización de las obras.

**Art. 32.-** Programa de ordenador, ya sea programa fuente o programa objeto, es la obra literaria constituida por un conjunto de instrucciones expresadas mediante palabras, códigos, planes o en cualquier otra forma que, al ser incorporadas en un dispositivo de lectura automatizada, es capaz de hacer que un ordenador, o sea, un aparato electrónico o similar capaz de elaborar informaciones, ejecute determinada tarea u obtenga determinado resultado. Se presume que es productor del programa de ordenador, la persona que aparezca indicada como tal en la obra de la manera acostumbrada, salvo prueba en contrario.

**Art. 33.-** El contrato entre los autores del programa de ordenador y el productor, implica la cesión ilimitada y exclusiva a favor de éste de los derechos patrimoniales reconocidos en la presente ley, así como la autorización para decidir sobre su divulgación y la de ejercer los derechos morales sobre la obra, en la medida que ello sea necesario para la explotación de esta, salvo pacto en contrario.

Tomando en cuenta los artículos anteriores los desarrolladores del software del presente trabajo de grado tienen el derecho exclusivo del Sistema Web y Aplicación Móvil y el derecho de autor de dichos software.

En el reglamento de la Administración Académica de la Universidad de El Salvador en el Artículo 215 se estipula que los derechos de autor sobre los trabajos de investigación

elaborados en los procesos de graduación, serán de propiedad exclusiva de la Universidad, la cual podrá disponer de los mismos de conformidad a su marco jurídico interno y legislación aplicable.

### ***2.6.3. Licencia de Software***

Las licencias de software son una forma de dar autorización a los usuarios para la utilización de programas que ofrecen facilitar los procesos de ejecución en las instituciones gubernamentales o empresas privadas las cuales conllevan una serie de términos y cláusulas que básicamente son un contrato entre el desarrollador o autor del programa y el usuario. En la actualidad se encuentran diferentes tipos de licencias de software con que se distribuyen las aplicaciones que usamos diariamente.

#### **Copyleft**

Es un tipo de software libre cuyo término de distribución asegura que todas las copias de todas las versiones son libres. Lo cual significa que dicha licencia no permite a terceros agregar ningún requisito adicional o restricción y exige que el código fuente sea público conservando siempre el derecho del autor en todas sus modificaciones así el usuario podrá modificar y utilizar a su conveniencia.

#### **Fundamentos y libertades**

El sistema será entregado de manera gratuita bajo la licencia de Copyleft, y tendrá la autorización de distribución o modificación mientras se conserve siempre el derecho de autor que desarrollo el software el cual solo será utilizado para fines de ayuda a las instituciones que lo ameriten y dicho software será sin fines de lucro.

### ***2.6.4. Código fuente***

#### **Licencia pública general**

La Licencia Pública General de GNU (GNU GPL) es una licencia de derecho de autor ampliamente usada en el mundo del software libre y código abierto y garantiza a los

usuarios la libertad de usar, estudiar, compartir y modificar el software. De esta manera no tiene que competir con una versión modificada, privativa del sistema será protegida por medio de Copyleft de intentos de apropiación y si se hace una modificación del sistema la GPL exige que se haga público a los usuarios el código fuente modificado, que se encuentra bajo la GPL.<sup>102</sup>

---

<sup>102</sup>Categorías de Software Libre y No Libre Recuperado de [www.analfatecnicos.net](http://www.analfatecnicos.net)

## 2.7. Marco Conceptual

A continuación, se presentan algunos conceptos que le ayudarán a tener una mayor comprensión del contenido del presente trabajo.<sup>103</sup>

### ■ Código Abierto

Código abierto significa que eres libre de usarlo y modificarlo. Cualquiera puede instalar el software. También puedes aprender y personalizar el código fuente para que se adapte mejor a tus necesidades.

### ■ Base de Datos

Una base de datos es un lugar en el que los datos son almacenados y organizados. La palabra «relacional» significa que los datos almacenados en el conjunto de datos son organizados en forma de tablas. Cada tabla se relaciona de alguna manera. Si el software no es compatible con el modelo de datos relacionales, simplemente se llama DBMS.

### ■ Base de Datos Relacional

Una base de datos es relacional cuando esta cumple con el modelo relacional, que se refiere a la relación que existe entre las distintas entidades o tablas de la base. También conocidas como sistemas de gestión de bases de datos relacionales (RDBMS), las cuales nos permiten almacenar y gestionar gran cantidad de datos. Los datos se almacenan en diferentes tablas y las relaciones se establecen usando claves primarias u otras llaves conocidas como claves externas o foráneas.

### ■ Modelo Cliente-Servidor

La arquitectura cliente-servidor es un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Un cliente realiza peticiones a otro programa, el servidor, quien le da respuesta. Esta idea también se puede aplicar a programas que se ejecutan sobre una sola computadora, aunque es más ventajosa en un sistema

---

<sup>103</sup>Terminos y conceptos recuperado de <https://definicion.de/> , <https://deconceptos.com/>

operativo multiusuario distribuido a través de una red de computadoras.

- **SQL**

Es un lenguaje estándar e interactivo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en ellas, gracias a la utilización del álgebra y de cálculos relacionales, el SQL brinda la posibilidad de realizar consultas con el objetivo de recuperar información de las bases de datos de manera sencilla. Las consultas toman la forma de un lenguaje de comandos que permite seleccionar, insertar, actualizar, averiguar la ubicación de los datos, y más.

- **Modelo-Vista-Controlador**

El MVC es un patrón de diseño arquitectónico de software, que sirve para clasificar la información, la lógica del sistema y la interfaz que se le presenta al usuario. En este tipo de arquitectura existe un sistema central o controlador que gestiona las entradas y la salida del sistema, uno o varios modelos que se encargan de buscar los datos e información necesaria y una interfaz que muestra los resultados al usuario final. Es muy usado en el desarrollo web porque al tener que interactuar varios lenguajes para crear un sitio es muy fácil generar confusión entre cada componente si estos no son separados de la forma adecuada. Este patrón permite modificar cada uno de sus componentes si necesidad de afectar a los demás.

- **DNS**

Son las iniciales de Domain Name System (sistema de nombres de dominio) y es una tecnología basada en una base de datos que sirve para resolver nombres en las redes, es decir, para conocer la dirección IP de la máquina donde está alojado el dominio al que queremos acceder.

- **Xampp**

Es un paquete de software libre, que consiste principalmente en el sistema de gestión de bases de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script PHP y Perl. El nombre es en realidad un acrónimo: X, Apache, MariaDB/MySQL, PHP, Perl.

- **MINSAL**

Ministerio de Salud de El Salvador, es la instancia del Estado rectora en materia de salud, que garantiza a los habitantes del país la cobertura de servicios oportunos e integrales, con equidad, calidad y calidez, en corresponsabilidad con la comunidad, incluyendo todos los sectores y actores sociales, para contribuir a lograr una mejor calidad de vida.<sup>104</sup>

- **SIBASI**

Sistema Básico de Salud Integral, Es la estructura básica operativa del Sistema Nacional de Salud, fundamentada en la Atención Primaria de Salud, que mediante la provisión de servicios integrales y articulados de salud del Primer y Segundo Nivel de atención, la participación ciudadana consciente y efectiva, y la corresponsabilidad de otros sectores, contribuye a mejorar el nivel de salud de una población definida.

- **Manipulador de alimentos**

El término se refiere a toda persona que por su actividad laboral tiene contacto directo con los alimentos durante su preparación, fabricación, transformación, elaboración, envasado, almacenamiento, transporte, distribución, venta, suministro y servicio.

- **Infecciones cutáneas**

Son infecciones de la piel causadas por bacterias piógenas, principalmente estafilococos y estreptococos. Representan el diagnóstico dermatológico más frecuente y la forma más común es el impétigo.

- **BPM**

Buenas Prácticas de Manufactura de Alimentos, son una herramienta básica para la obtención de productos seguros para el consumo humano. Estas herramientas se centran en la higiene y forma de manipulación de los alimentos. Son muy útiles en el diseño y funcionamiento de establecimientos, en el desarrollo de procesos y productos relacionados con la alimentación, contribuyendo a la elaboración de alimentos seguros, saludables e inocuos para el consumo humano.

---

<sup>104</sup>Ministerio de Salud de El Salvador recuperado de <https://www.salud.gob.sv/>

- **Caso de uso**

Escenario de negocios o evento respecto del cual el sistema debe proporcionar una respuesta definida. Los casos de uso evolucionaron a partir del análisis orientado a objetos; pero su utilización se ha vuelto común en muchos otros métodos de análisis y diseño de sistemas.

- **Diagrama de casos de uso**

Diagrama que ilustra las interacciones entre el sistema y los sistemas y usuarios externos. En otras palabras, describe gráficamente quién va a usar el sistema y de qué manera el usuario espera interactuar con el sistema.

- **Actor**

Cualquier cosa que necesite interactuar con el sistema para intercambiar información.

- **Muestra**

Muestra es una porción de la totalidad de un fenómeno, producto o actividad que se considera representativa del total también llamada una muestra representativa.

- **Método de muestreo**

Es una herramienta de la investigación científica, cuya función básica es determinar que parte de una población debe examinarse, con la finalidad de hacer inferencias sobre dicha población.

- **Población**

Es el conjunto de elementos (finito o infinito) definido por una o más características, de las que gozan todos los elementos que lo componen.

- **Inhaloterapia**

Consiste en la administración de medicamentos, oxígeno y humedad por la vía respiratoria.

- **Ambulatorios**

Se utiliza en medicina para hacer referencia tanto a un nivel de atención primaria de salud como al carácter de un procedimiento que no requiere que el paciente

permanezca hospitalizado o recluido para llevarlo a cabo.

- **Rehidratación Oral**

Tratamiento que se usa para restituir los líquidos que el cuerpo pierde debido a una cirugía, una lesión, deshidratación, enfermedad u otras afecciones.

- **Inspectores de Saneamiento**

Son los que se encargan de verificar que se cumplan todas las medidas de sanidad reguladas por el código de salud.<sup>105</sup>

- **Credencial**

Documento que se entrega a un funcionario en que se notifica y acredita que ha sido nombrado para un cargo determinado y que puede tomar posesión de él.

- **Interfaz amigable**

Es el conjunto de los controles y canales sensoriales mediante los cuales un usuario puede comunicarse con una máquina, se caracteriza por tener un alto grado de usabilidad, y por ser amigable e intuitiva.

- **Hardware**

Conjunto de elementos físicos o materiales que constituyen una computadora o un sistema informático.

- **Software**

Conjunto de programas y rutinas que permiten a la computadora realizar determinadas tareas.

- **Área de Saneamiento Ambiental**

Es el área en el cual se comprende el conjunto de acciones técnicas y socio-económicas que garantizan la salud pública, lo que conlleva a la salubridad ambiental.

- **Plan de capacitación**

Es un proceso que va desde la detección de necesidades de capacitación hasta la

---

<sup>105</sup>Ministerio de Salud de El Salvador recuperado de <https://w2.salud.gob.sv/novedades/noticias/noticias-ciudadanosas/188-julio-2012/1412-27-07-2012-minsal-celebra-el-dia-del-inspector-tecnico-de-saneamiento-ambiental>

evaluación de los resultados. Estos objetivos deben ser formulados de manera clara, precisa y medible, de tal manera que luego de la aplicación del programa sea posible evaluar los resultados del mismo.<sup>106</sup>

#### ■ **Aplicación móvil**

Son programas diseñados para ser ejecutados en teléfonos, tablets y otros dispositivos móviles, que permiten al usuario realizar actividades profesionales, acceder a servicios, mantenerse informado, entre otro universo de posibilidades.

#### ■ **Promotores de Salud**

Es aquella persona que provee de información sobre una temática, que es capaz de sensibilizar y ayudar a las personas, y propicia modificación de conductas para lograr salud. Un facilitador es una persona que ayuda a aprender las aptitudes que se presentan en el curso.<sup>107</sup>

#### ■ **Modelo en Cascada**

Es un proceso de desarrollo secuencial, en el que el desarrollo de software se concibe como un conjunto de etapas que se ejecutan una tras otra. Se le denomina así por las posiciones que ocupan las diferentes fases que componen el proyecto, colocadas una encima de otra, y siguiendo un flujo de ejecución de arriba hacia abajo, como una cascada.

#### ■ **Datos cualitativos**

Estos datos son los relativos a las *cualidades*, este tipo de información relacionada con los adjetivos.

#### ■ **Datos cuantitativos**

Son datos que miden o calculan un algo para llegar a un punto en su investigación. Estos datos nos dicen a través de números una explicación para alguna tendencia o resultados de algún experimento.

#### ■ **VPS**

---

<sup>106</sup>Plan de capacitación recuperado de <https://www.esan.edu.pe/apuntes-empresariales/2016/07/cinco-pasos-del-plan-de-capacitacion/>

<sup>107</sup>Ministerio de Salud de El Salvador recuperado de <http://asp.salud.gob.sv>

Servidor virtual privado, un servidor virtual privado es un método de particionar un servidor físico en varios servidores virtuales de tal forma que todo funcione como si se estuviese ejecutando en una única máquina. .

- **Lindeo**

Es una plataforma de computación en la nube acelera la innovación, al hacer que las máquinas virtuales Linux sean simples, asequibles y accesibles para todos. Es una empresa estadounidense de alojamiento en la nube de propiedad privada que proporciona servidores privados virtuales.

# Capítulo 3

## Metodología de la Investigación

### 3.1. Metodología de la Investigación

Una metodología es un conjunto de técnicas y procedimientos aplicadas de forma organizada y sistematizada que se deben seguir para efectuar un estudio de investigación. Para los procesos en exploración es muy importante establecer primero la metodología y tener claro todo lo que se realizará en cada una de las etapas en las que se divide, para esto los investigadores deben tomarse el tiempo de deliberar sobre las técnicas y métodos a utilizar.<sup>108</sup>

Uno de los objetivos primordiales para adoptar una metodología de investigación es dar un mayor peso científico a lo que desea estudiar, ya que se desarrolla un proceso de análisis en los resultados obtenidos sobre la investigación, entonces es de suma importancia conocer acerca de la problemática y, así poder realizar un análisis sobre los pasos a seguir.

#### *3.1.1. Investigación descriptiva*

Para el manejo de información en el presente proyecto, se hará utilizando la investigación descriptiva, como su nombre lo dice, describe la realidad, sobre la situación actual en la clínica, así como la problemática pudiendo detallar los eventos o procesos en

---

<sup>108</sup>Fundamentos para la Metodología Recuperado de <https://explorable.com/es/metodologia-de-la-investigacion>

los cuales se necesita mejorar; en la investigación es necesario englobar a personas, grupos y comunidades, a las que se estará abordando para la recopilación de información.

La investigación descriptiva permitirá conocer y analizar las características de la población, lugar o proceso social que conllevan los etapas del estudio, además de plantear relaciones entre los actores, factores y variables involucradas. Los métodos de recopilación de datos que se emplean son la observación, la encuesta y la entrevista para su posterior estudio. A partir de la observación, se suelen extraer datos cualitativos, mientras que la encuesta suele proporcionar datos cuantitativos.

### ***3.1.2. Investigación Tecnológica***

El término *tecnología* es el conjunto de teorías y técnicas que permite llevar a la práctica el conocimiento científico, donde se engloban grupos de técnicas para ser usados con un fin práctico. Si bien la tecnología ha trascendido con respecto al paso del tiempo, su evolución ha sido notable en estos últimos años, cabe destacar que esta siempre ha estado en los orígenes de la humanidad ya que siempre ha existido la necesidad de buscar o indagar el conocimiento con el fin de usarlo a nuestro favor y llevarlo a la práctica.<sup>109</sup>

Para el proceso de desarrollo del presente proyecto es muy necesaria la investigación tecnológica, desde las instancias iniciales de desarrollo del proyecto como las distintas etapas del proceso, para ello será necesaria establecer una tecnología que pueda ser utilizada como instrumento que fomente la innovación, donde se incorpore el conocimiento científico y tecnológico ya sea propio o ajeno, que aporte a las necesidades de los desarrolladores durante todo el proceso de proyecto.

La Investigación Tecnológica procede para todo lo anteriormente citado con un objetivo específico de crear un proceso productivo, tecnológico e innovador para los procesos del desarrollo, donde se refleje su resultado con el fin de suplir las necesidades, minimizando en tiempo de manera eficiente, además de traer beneficios con la implementación de este.

---

<sup>109</sup>Términos de tecnología Recuperado de <https://www.significados.com/tecnologia/>

## 3.2. Metodología de Desarrollo

Para el desarrollo del sistema de inspección de credenciales, se aplicará la metodología de desarrollo de cascada; teniendo en cuenta que son las características de cada proyecto las que definen qué metodología utilizar, se ha identificado una visión clara del producto final que se ve a entregar a la organización, así como también se tiene definido los alcances de cada entregable del proyecto.

La metodología de desarrollo en cascada es el enfoque metodológico que ordena rigurosamente las etapas del ciclo de vida del software, de forma tal que el inicio de cada etapa debe esperar a la finalización de la inmediatamente anterior.<sup>110</sup>

De esta forma, cualquier error de diseño detectado en la etapa de prueba conduce necesariamente al rediseño y nueva programación del código afectado, aumentando los costos del desarrollo.

El modelo en cascada consta de cuatro etapas sucesivas:

1. Análisis
2. Codificación
3. Pruebas
4. Implementación

A continuación se describe la aplicación de las cuatro fases dentro del proyecto en desarrollo:

### 1. Análisis

Para la primera fase se hará un análisis de las necesidades en La Unidad Comunitaria de Salud Familiar principalmente en el área donde se lleva a cabo el registro, la emisión y la inspección de manipuladores de alimentos, para determinar las características del software a desarrollar y llevar a cabo una descripción de los requisitos que deberá

---

<sup>110</sup>Metodología de desarrollo en cascada Recuperado de <https://aspgems.com/metodologia-de-desarrollo-de-software-i-modelo-en-cascada/>

cumplir el software; es decir lo que deberá hacer. Estimar las necesidades del software previo a la segunda fase. Calcular y estimar el costo del software, los riesgos y los plazos.

## 2. Codificación

- Desarrollar la estructura interna del software, y las relaciones entre las entidades que lo componen donde se describirá el comportamiento teórico del software.
- Identificar los módulos y conjunto de funciones que van a estar asociadas entre si.
- Establecer la arquitectura de la información y las tecnologías a implementar, tal como lenguajes de programación, bibliotecas de clases y secuencias de programas.
- Organizar el sistema en elementos que puedan elaborarse por separado, aprovechando las ventajas del desarrollo en equipo.
- Programar los requisitos especificados haciendo uso de las estructuras de datos diseñadas en la fase anterior.
- Analizar las condiciones, creación de algoritmos, y la implementación de éstos en el lenguaje de programación que se haya especificado.

## 3. Pruebas

Verificar que todos los componentes del sistema funcionen correctamente y cumplen con los requisitos.

Seguir el enfoque de las pruebas alfa y beta para evaluar dinámicamente el software, que permitirá comenzar por los componentes mas simples y mas pequeño e ir avanzando progresivamente hasta probar todo el software en su conjunto.

## 4. Implementación

Desarrollo de manuales de usuario y plan de implementación.

Para la implementación del Sistema Informático se ha calculado un crecimiento de la población de usuarios estimados en base a los datos actuales proporcionados por la administración de la Unidad de Saneamiento ambiental; se utilizó la fórmula de Crecimiento Poblacional Compuesto.

La fórmula para encontrar la población después de  $t$  años en un Crecimiento Poblacional Compuesto es:

$$P_t = P_0(1 + r)^t$$

en donde,

$P_t$  Es la población total después de  $t$  años.

$P_0$  Es la población inicial.

$r$  Es la tasa de crecimiento anual expresada en decimales.

$t$  Es el número de años después del año inicial.

En base a los datos proporcionados se tiene que  $P_0 = 1,243$  con  $r = 2\%$  y la estimación se hará para  $t = 3$  años

$$P_t = P_0(1 + r)^t$$

$$P_t = 1,243(1 + 0.02)^3$$

$$P_t = 1,243(1.02)^3$$

$$P_t = 1,243(1.061)$$

$$P_t = 1,319$$

Usuarios se estima para el año 2023

### 3.3. Población y muestra

#### 3.3.1. Población

Una población es el conjunto de todos los casos que concuerdan con una serie de especificaciones.<sup>111</sup>

El universo o población de la investigación estará constituido por todos los comerciantes formales e informales cuyas actividades laborales tengan relación con la manipulación de alimentos en el Municipio de San Francisco Gotera, ya que el Sistema Informático estará enfocado en el control del cumplimiento de buenas prácticas de sanidad alimentaria.

Además, formarán parte del universo los Inspectores Técnicos del Área de Saneamiento Ambiental de La Unidad de Salud de dicho municipio, dado que son ellos los que se encargan del control antes mencionado.

La población en estudio está conformada por mil doscientos cincuenta manipuladores de alimentos, incluyendo tanto el sector formal como informal de vendedores del Municipio de San Francisco Gotera; estos datos fueron brindados por los promotores de salud de la Unidad de Saneamiento Ambiental de la Unidad Comunitaria de Salud Familiar del mismo Municipio.

#### 3.3.2. Método de muestreo

La elección de la muestra se hará mediante un método de muestreo no probabilístico.

En las muestras no probabilísticas, la elección de los elementos no depende de la probabilidad, sino de causas relacionadas con las características de la investigación o propósitos del investigador.

Aquí el procedimiento no es mecánico ni se basa en fórmulas de probabilidad, sino que depende del proceso de toma de decisiones de un investigador o un grupo de

---

<sup>111</sup>Metodología de la investigación, 6TH Edición, Sampieri

investigadores.<sup>112</sup>

### ***3.3.3. Muestra en cadena o por redes***

En este proceso, se identifican participantes clave y se agregan a la muestra, se les pregunta si conocen a otras personas que puedan proporcionar más datos o ampliar la información.

Teniendo en cuenta lo citado en el párrafo anterior, se tomará como muestra a personas relacionadas con la manipulación de alimentos, específicamente a:

- Los jefes de establecimientos formales,
- Las comerciantes informales que tengan a cargo a otras personas
- Los Inspectores Técnicos del Área de Saneamiento Ambiental de La Unidad Comunitaria de Salud Familiar.

---

<sup>112</sup>Ibidem, Pag. 176.

### 3.4. Técnicas e instrumentos de investigación

Según Sampieri, recolectar datos consiste en elaborar un plan detallado de procedimientos que nos conduzcan a reunir datos con un propósito específico, determinando lo siguiente:

- Las fuentes de las que se obtendrán los datos,
- El lugar donde se localizan tales fuentes,
- El medio o el método para recolectar los datos y,
- La forma en la que serán preparados y analizados.

Tomando en cuenta que toda medición o instrumento de recolección de datos debe poseer esencialmente validez, confiabilidad y objetividad.<sup>113</sup>

De acuerdo a los objetivos de la presente investigación se han considerado los siguientes instrumentos:

- **Observación.**

Se realizarán observaciones con el propósito de obtener información relacionada a los procesos que se llevan a cabo en la Unidad de Saneamiento Ambiental de la Unidad Comunitaria de Salud Familiar de San Francisco Gotera, para visualizar directamente los problemas que presentan en los aspectos de procesamiento de información de los manipuladores de alimentos.

- **Entrevistas**

Se realizarán entrevistas para recopilar información, dirigida con un propósito específico para los involucrados en el proyecto de manipuladores de alimentos, en la cual se usa un formato de preguntas y respuestas donde hay que obtener las opiniones del entrevistado y lo que siente sobre el funcionamiento del mecanismo que se lleva a cabo con el estado actual del sistema, los objetivos de la institución y

---

<sup>113</sup>Metodología de la investigación, 6TH Edición, SAMPIERI

los personales.

- **Encuestas**

Teniendo en cuenta que el sistema será complementado por una aplicación móvil, que también será usada por los manipuladores de alimentos; se realizarán encuestas principalmente para conocer el nivel de aceptación que esta obtendrá, entre otros aspectos importantes.

## 3.5. Técnicas e instrumentos para el análisis de los datos

### 3.5.1. *Procedimiento para la validación de los datos*

Identificados los instrumentos que se van a utilizar en la recolección de datos, se procederán con los siguientes pasos:

- Elaboración de las preguntas.
- Las preguntas serán revisadas de forma detallada por el asesor de tesis.
- Se realizará una prueba piloto, la cual consistirá en pasar la encuesta a los manipuladores de alimentos, para verificar si las preguntas son fáciles de comprender.
- Se obtendrá la validación del instrumento una vez los encuestados respondan todas las preguntas sin ningún inconveniente.

### 3.5.2. *Procedimiento de recolección de datos*

El procedimiento de recolección que se seguirá mediante el uso del instrumento de la observación es el siguiente:<sup>114</sup>

- Se asistirá a las capacitaciones que los inspectores de salud imparten a los manipuladores de alimentos.
- Se tomarán apuntes descriptivos de los procesos que llevan a cabo durante la capacitación, tales como: registro de asistencia, inscripción al proceso, información sobre los exámenes, entre otros.
- Se observará también el comportamiento de los manipuladores, el número de asistencia, si hacen preguntas, si prestan atención y el grado de aceptación y recepción.

---

<sup>114</sup>Métodos de Recolección y Análisis de Datos en la Evaluación de Impacto (2014) Greet Peersman Recuperado de [www.unicef-irc.org/publications](http://www.unicef-irc.org/publications)

- Además, se observará el trabajo de campo realizado por los *inspectores técnicos de saneamiento ambiental*, para ello se solicitará asistir a una jornada de inspección a los establecimientos.
- Se realizarán visitas a el área de Saneamiento Ambiental de La Unidad de Salud para observar los procesos que realizan los inspectores técnicos con los manipuladores de alimentos.

Para la obtención de los datos por medio del instrumento de la encuesta, se procederá de la siguiente manera:

- Se seleccionará una muestra de la población de manipuladores de alimentos, que se deberá investigar.
- Para que la muestra sea significativa deberá ser seleccionada cuidadosamente, e incluir a manipuladores de alimentos de todos los sectores.
- Una vez seleccionada la muestra, se deberá pasar el instrumento, brindándoles a los participantes, las indicaciones necesarias y el propósito que conlleva la realización de la encuesta.
- La encuesta se realizara haciendo uso de la herramienta Google Forms, la cual es una aplicación de Google Drive, que facilita enviar la encuesta de una forma fácil y rápida para el procesamiento de datos recopilados.<sup>115</sup>
- Se enviará el enlace de la encuesta por medio de las redes sociales, exclusivamente a los manipuladores de alimentos del Municipio de San Francisco Gotera Departamento de Morazán y se pedirá compartir el enlace a otros manipuladores de alimentos que conozcan.
- Se verificará que todos los encuestados hayan respondido la encuesta de forma correcta.
- Se organizarán los datos obtenidos para su posterior análisis, y darle un enfoque que contribuya con la investigación.

---

<sup>115</sup>Formularios para Encuestas o Entrevistas Recuperado de <https://www.google.com/intl/es/forms/about/>

Mientras que, para la recolección de los datos por medio de la entrevista el procedimiento a seguir es:

- Se identifican los puntos mas relevantes y necesarios que se deban tratar.
- Se hace una lista con los temas de los que se necesita información.
- Se elabora una guía para realizar la entrevista.
- Se realiza la entrevista haciendo uso de Google Forms.
- Se enviará el enlace de la entrevista por medio de contacto telefónico a los inspectores que estarán relacionados con el sistema informático y la aplicación móvil de la Unidad Comunitaria de Salud Familiar de San Francisco Gotera.
- Las respuestas obtenidas de la entrevista serán detalladas en la herramienta de Google Forms.

### ***3.5.3. Procedimientos para procesar e interpretar los datos***

Se recopila toda la información obtenida a través de los instrumentos detallada anteriormente, información que contribuirá con el desarrollo del sistema, al conocer las necesidades y requerimientos principales. Para que la información obtenida sea mas clara y demostrativa, se presentará de la siguiente manera:<sup>116</sup>

- Se detallarán los puntos más relevantes de la observación.
- Se formulará el objetivo de cada pregunta de la encuesta.
- Se presentará cada una de las preguntas que contiene la encuesta.
- Se tabulará los datos obtenidos de las encuestas.
- Se hará una representación de los datos en forma porcentual y gráfica, para una mejor comprensión y análisis de los datos.
- Se simplificarán las respuestas obtenidas de la entrevista.

---

<sup>116</sup>El Proceso de Análisis de Datos (2017) Rescatado de [www.marketing-analitico.com/analitica-web/proceso-analisis-datos](http://www.marketing-analitico.com/analitica-web/proceso-analisis-datos)

- Se obtendrán los requerimientos a través del análisis de la observación y las respuestas obtenidas en la entrevista.
- Al finalizar, se hará un análisis preciso que contribuya con el desarrollo del proyecto.

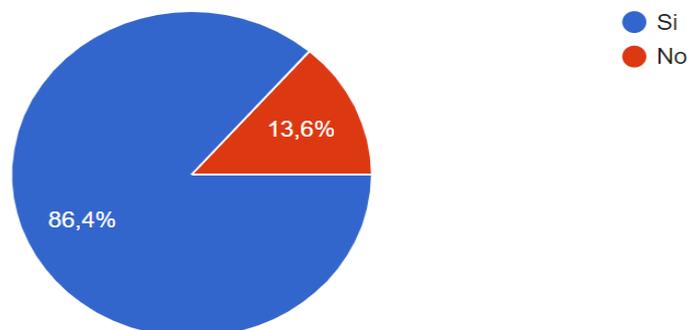
## 3.6. Análisis e interpretación de resultados

### 3.6.1. Tabulación de la encuesta

Encuesta dirigida a los manipuladores de alimentos del Municipio de San Francisco Gotera del Departamento de Morazán.

1. ¿Cuenta con el permiso requerido para laborar en la manipulación de alimentos?

**Objetivo:** Conocer el porcentaje de manipuladores de alimentos que cuentan con sus credenciales autorizadas por el Ministerio de Salud.

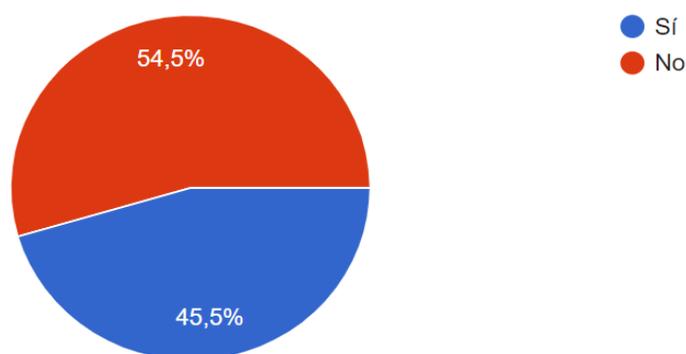


**Figura 3.6.1:** Porcentaje de Manipuladores con permiso

En la gráfica se puede observar un porcentaje de 86,4% cuentan con el permiso para la manipulación de alimentos mientras un 13,6% no son manipuladores autorizados para la realización de alimentos.

2. ¿Alguna vez se le ha olvidado la fecha que le toca renovar su credencial de manipulador de alimentos?

**Objetivo:** Conocer el porcentaje de personas que renuevan sus credenciales de manipuladores de alimentos.

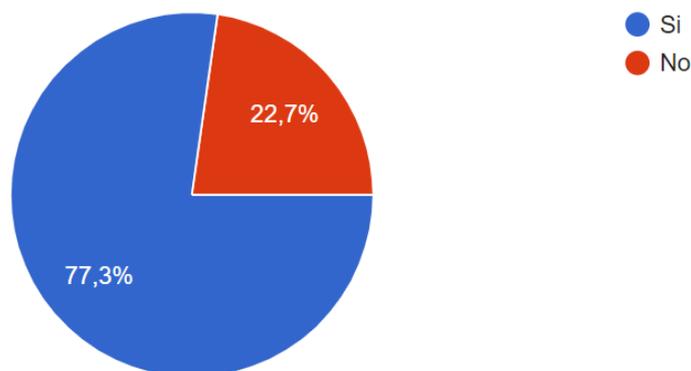


**Figura 3.6.2:** Manipuladores que han olvidado renovar credenciales

En la gráfica se puede observar que un 45,5 % olvidan la fecha de renovación de sus credenciales y un 54,5 % no olvidan la fecha de renovación esto indica que la mayoría de manipuladores renuevan sus credenciales en el periodo de tiempo establecido.

3. ¿Cuenta con un teléfono inteligente (Smartphone)?

**Objetivo:** Conocer el porcentaje de manipuladores de alimentos que cuentan con un teléfono inteligente.

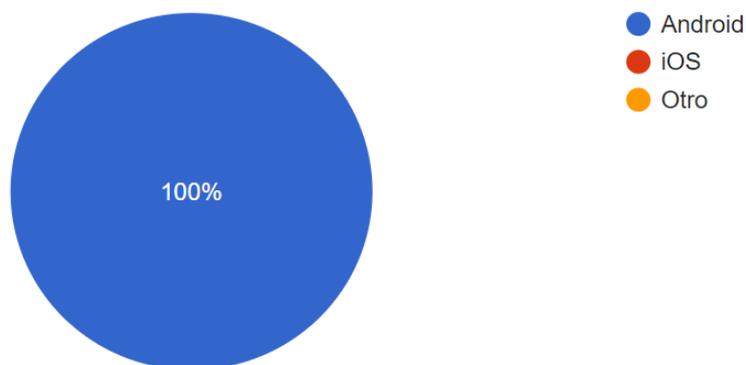


**Figura 3.6.3:** Porcentaje de manipuladores con Smartphone

En la gráfica podemos observar que el 77,3% cuenta con teléfono Smartphone y el 22,7% no cuenta con este móvil esto significa que la mayor parte de los manipuladores de alimentos cuentan con teléfonos inteligentes.

#### 4. ¿Qué sistema operativo tiene su teléfono?

**Objetivo:** Conocer el sistema operativo más utilizado en los teléfonos inteligentes por los manipuladores de alimentos de San Francisco Gotera Morazán.

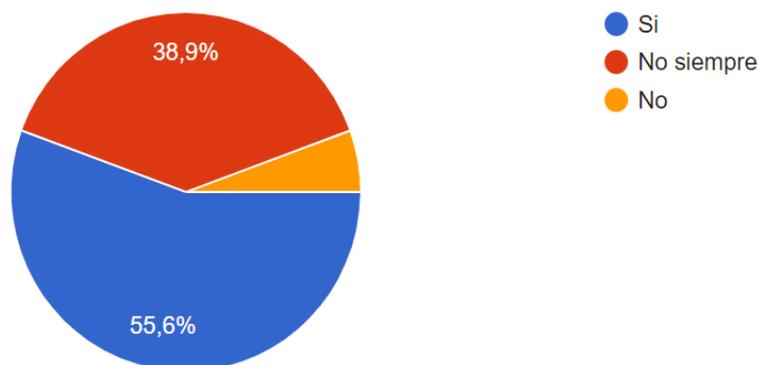


**Figura 3.6.4:** Sistema operativos móviles usado por los manipuladores

Nos podemos dar cuenta que las personas que manipulan alimentos y cuentan con un dispositivo móvil inteligentes son un 100% que prefieren el sistema operativo Android esto quiere decir que no se tendría dificultad en la realización de la aplicación móvil ya sera exclusivamente para dispositivos Android.

## 5. ¿Tiene Acceso a Internet en su Teléfono móvil?

**Objetivo:** Conocer el porcentaje de manipuladores de alimentos que poseen conexión a internet en sus dispositivos móviles.

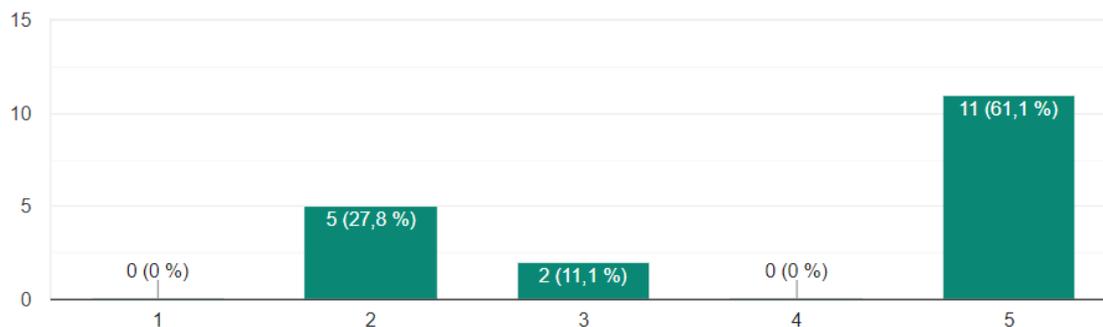


**Figura 3.6.5:** Porcentaje de acceso a internet

En la gráfica se puede observar que un 55,6 % cuenta con internet, un 38,9 % no siempre cuenta con internet y un 5,5 % no tiene internet, lo que significa que la mayoría de manipuladores tienen acceso a internet.

6. ¿Con qué frecuencia se conecta a internet?

**Objetivo:** Conocer la frecuencia con la que los manipuladores de alimentos se conectan a internet.

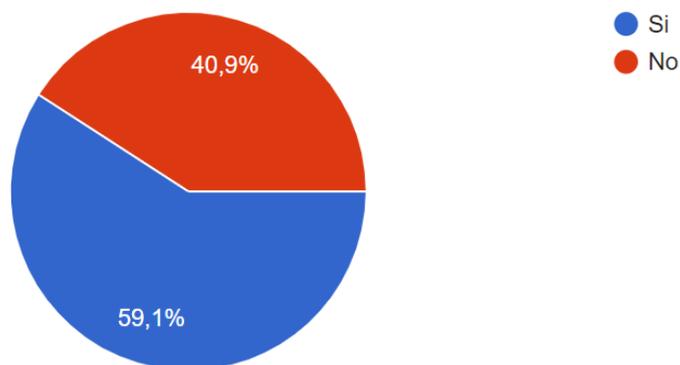


**Figura 3.6.6:** Porcentaje de conexión a internet

En la gráfica podemos conocer la frecuencia con la que más se conectan a internet que es de un 61,1 % que es la frecuencia más alta de la que se conectan y un 27,8 % frecuencia media de conexión y un 11,1 % la frecuencia más baja que se tiene en conectarse a internet esto significa que la mayoría se conecta con frecuencia.

7. ¿Sabe buscar y descargar aplicaciones móviles en su teléfono inteligente?

**Objetivo:** Conocer el grado de conocimiento en búsqueda y descarga de aplicaciones móviles para dicha instalación por parte de los manipuladores de alimentos.

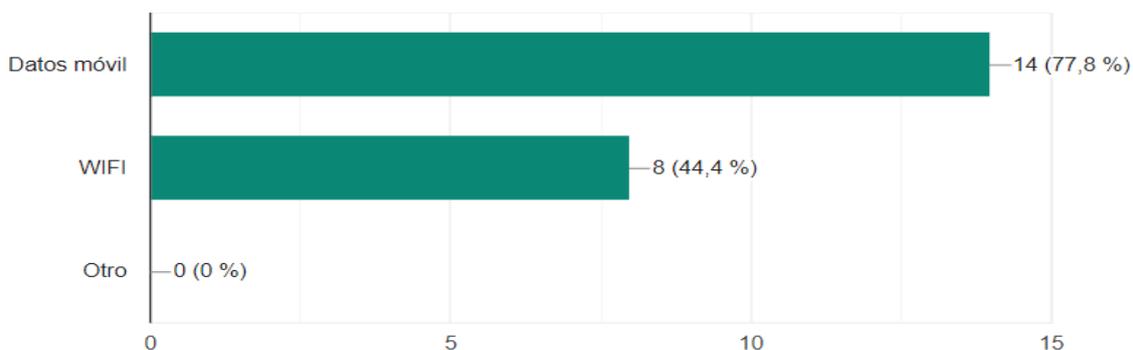


**Figura 3.6.7:** Porcentaje de instalar aplicaciones móviles

En la gráfica se puede observar que un 59,1 % tiene conocimientos de búsqueda y descarga de aplicaciones móviles y el 40,9 % no tiene conocimiento se puede ver que la mayoría tiene el acceso a realizar dicha búsqueda y descarga de aplicaciones móviles.

8. ¿Qué método utiliza para conectarse a Internet?

**Objetivo:** Conocer el método mas usado para conectarse a internet por los manipuladores de alimentos.

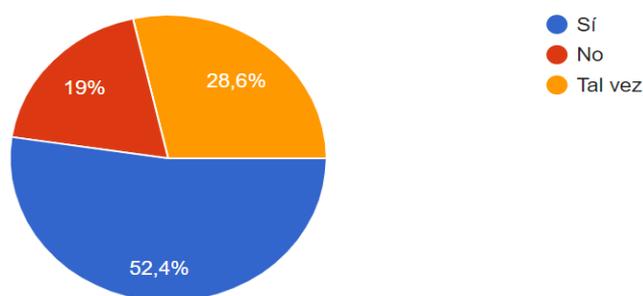


**Figura 3.6.8:** Porcentaje de método para conexión a internet

En la gráfica podemos observar que los manipuladores de alimentos utilizan dos métodos de conexión a internet el cual uno de ellos es datos móviles que tiene el mayor porcentaje de un 77,8 % de utilización y el otro método que se puede ver es el de conexión por medio de WIFI que es de 44,4 % esto significa que la mayoría de personas que manipulan alimentos tienen mayor acceso de conexión a datos móviles en sus dispositivos telefónicos.

9. ¿Consideraría útil una aplicación móvil que le envié notificaciones para recordarles las fechas de presentar exámenes o renovación de su credencial?

**Objetivo:** Conocer la opinión sobre la utilidad de una aplicación móvil que les notifique sobre las fechas de renovación o entrega de exámenes.



**Figura 3.6.9:** Manipuladores que consideran útil una aplicación móvil

Se puede observar acerca de los porcentajes de personas que están de acuerdo con la utilidad de una aplicación móvil donde el 52,4% considera útil, un 28,6% esta desacuerdo y un 19% se mantiene en tal vez es de utilidad lo cual significa que para la mayoría sería de gran utilidad las notificaciones que podrán recibir.

### **3.6.2. Entrevistas**

En la entrevista que se les realizó a los Inspectores Técnicos del Área de Saneamiento Ambiental de La Unidad Comunitaria de Salud Familiar se conocieron los problemas principales que enfrenta los Inspectores con respecto al manejo de los datos de los manipuladores de alimentos de todo el municipio que se encuentran registrados hasta la fecha en tal Unidad, haciendo hincapié en el problema principal de la desorganización que se tiene en los datos de las persona registradas, si bien se expresó que los registros se llevan digitalizados en hojas de cálculos, pero aun en el momento que ellos desean visualizar ciertos datos de una determinada persona o un grupo de personas esta información no es del todo clara, es decir no cuentan con filtros para la búsqueda de información.

Además de los problemas que tienen al realizar trabajos con la información de los manipuladores en las instalaciones de La Unidad de Salud, mencionaron otro problema que les gustaría se solventara con el sistema informático a desarrollar: “Cuando salimos a realizar inspecciones a los establecimientos la información si la tenemos que registrar en formularios físicos, luego tenemos que realizar ciertos cálculos para contabilizar los puntos que obtuvo el establecimiento en la inspección para luego darle una calificación y ver si este cumple con las normas de salud establecidas”; con esas palabras lo expresó Humberto Javier Hernández, Inspector Técnico de Saneamiento Ambiental de La Unidad Comunitaria de Salud Familiar de San Francisco Gotera.

Durante la entrevista también se abordó el tema del desarrollo de una aplicación móvil como parte del sistema, ellos mostraron su apoyo completo, dado que les sería de mucha utilidad para solventar parte del problema que se mencionó en el párrafo anterior, así como también la gran utilidad para los manipuladores de alimentos.

También se pudo conocer que los Inspectores tienen el nivel de conocimiento de manejo de computadoras requerido para el uso del sistema, así como información de las características de los equipos con los que cuentan ya que mencionaron que se tienen dos computadoras de escritorio Marca HP con sistema operativo Windows 10 y Windows 8, las cuales poseen procesadores de Intel Core i5 de sexta generación e Intel Core i3 de séptima

generación dichas computadoras cuentan con almacenamiento de 1 TB, y también con una impresora multifuncional láser y los dispositivos móviles de uso personal los cuales son de Marca SAMSUNG A10 y tienen 64GB de memoria de almacenamiento y para finalizar expresaron agradecimiento por haberles tomado en cuenta para la realización de este proyecto, ya que ellos obtendrán la solución de los problemas con los que han tenido que lidiar.

# Capítulo 4

## Desarrollo del software

### 4.1. Determinación de requerimientos

#### 4.1.1. *Requerimientos funcionales*

<b>Id</b>	<b>Requerimiento</b>	<b>Descripción</b>	<b>Restricción</b>	<b>Prioridad</b>
RF_01	Registro de usuarios	El sistema permitirá ingresar datos de nuevos usuarios	Solo usuarios con credencial de administrador podrán realizar esta tarea.	Alta
RF_02	Gestión de usuarios	El sistema permitirá actualizar datos de los usuarios así como cambiar el estado de estos	Solo usuarios con credencial de administrador podrán realizar esta tarea.	Alta

<b>Id</b>	<b>Requerimiento</b>	<b>Descripción</b>	<b>Restricción</b>	<b>Prioridad</b>
RF_03	Registro de manipuladores	El sistema permitirá registrar datos de los manipuladores de alimentos	Usuario administrador y usuario estándar podrán realizar esta tarea	Alta
RF_04	Gestión de manipuladores	El sistema permitirá modificar y desactivar manipuladores	Usuario administrador y usuario estándar podrán realizar esta tarea	Alta
RF_05	Búsqueda de registros	El sistema permitirá realizar la búsqueda de datos por filtros definidos	Cualquier usuario autorizado puede realizarlo. Los filtros definidos serán: Nombre y apellido, DUI, y id del registro	Alta
RF_06	Registro de establecimientos	El sistema permitirá el registro de datos de nuevos establecimientos	Cualquier usuario autorizado puede realizarlo.	Alta
RF_07	Gestión de establecimientos	El sistema permitirá modificar los datos de los establecimientos registrados	Esta acción estará habilitada para superusuario y administrador.	Alta

<b>Id</b>	<b>Requerimiento</b>	<b>Descripción</b>	<b>Restricción</b>	<b>Prioridad</b>
RF_08	Registro de asistencia	El sistema permitirá el registro de las personas que asistan a las capacitaciones impartidas	El administrador del sistema sera el que definirá los periodos de capacitación	Alta
RF_09	Envío de notificaciones	El sistema estará complementado por una aplicación móvil para el envío de alertas de fechas de renovación de credenciales a los manipuladores que sean parte del sistema	La aplicación solamente estará disponible para el sistema operativo android	Alta
RF_10	Inspecciones de establecimientos	El sistema llevara un registro de cada inspección realizada por los promotores de saneamiento ambiental donde se registrara una forma con cada punto a evaluar y sus respectivo puntaje, en caso de no llegar al puntaje requerido, se programara otra inspección en un determinado lapso de tiempo para permitirle al dueño del establecimiento cambiar los aspectos negativos	Los encargados de esta tarea serán los promotores de saneamiento ambiental	Alta

<b>Id</b>	<b>Requerimiento</b>	<b>Descripción</b>	<b>Restricción</b>	<b>Prioridad</b>
RF_11	Registro de exámenes clínicos	El sistema permitirá llevar un control de los exámenes clínicos que presentan los manipuladores como requisito para obtener su credencial	Usuario administrador y usuario estándar podrán realizar esta tarea	Alta
RF_12	Generación de credencial	Cuando un manipulador haya cumplido con los requisitos el sistema le generara una credencial	Usuario administrador y usuario estándar podrán realizar esta tarea	Alta
RF_13	Generación de bitácora	El sistema registrara cada uno de los eventos generados	N/A	Alta

**Tabla 4.1.1:** Requerimientos Funcionales

### 4.1.2. *Requerimientos no funcionales*

<b>Id</b>	<b>Requerimiento</b>	<b>Descripción</b>	<b>Restricción</b>	<b>Prioridad</b>
RNF_01	Usabilidad	La interfaz gráfica debe de garantizar la fácil navegabilidad para sus usuarios en todos los módulos. El sistema permitirá asignar roles a los usuarios registrados	N/A	Alta
RNF_02	Disponibilidad	El sistema tendrá una disponibilidad de 24 horas al día	N/A	Alta
RNF_03	Multiplataforma	El sistema funcionara en distintos sistemas operativos y plataformas de hardware	La aplicación móvil de complemento estará disponible solamente en sistema android	Alta
RNF_04	Rendimiento	Se debe garantizar que el diseño de las consultas en la base de datos no afecte su desempeño	N/A	Alta

Id	Requerimiento	Descripción	Restricción	Prioridad
RNF_05	Lógica	El sistema estará protegido contra usos o ingresos de información indebida, mediante validaciones en sus formularios. El sistema seguirá la arquitectura de Modelo Vista Controlador	N/A	Alta
RNF_06	Seguridad	El ingreso al sistema estará restringido bajo contraseñas cifradas, para garantizar la integridad de los datos de los usuarios así como de los manipuladores registrados	N/A	Alta

**Tabla 4.1.2:** Requerimientos No Funcionales

## 4.2. Estimación de esfuerzo por medio del método Puntos de Casos de Uso

Para la estimación usando el método de puntos de caso de uso, primero identificamos los actores y caso de uso asociados.

Nombre del actor	Casos de uso asociados
Súper usuario	Registrar exámenes, registrar manipuladores, registrar usuario, gestionar operaciones
Administrador	Gestionar usuarios, gestionar manipuladores, gestionar exámenes, gestionar establecimientos
Manipulador	Recibir notificaciones, ver datos

**Tabla 4.2.1:** Actores y casos de uso asociados

**Determinación del factor de peso de los actores sin ajustar (UAW).**

<b>Tipo de actor</b>	<b>Descripción</b>	<b>Factor de peso</b>	<b>N° de actores</b>	<b>Resultado</b>
Simple	Otro sistema que interactúa con el sistema a desarrollar mediante una interfaz de programación(API, Application Programming Interface)	1	1	1
Promedio	Otro sistema que interactúa con el sistema a desarrollar mediante un protocolo o una interfaz basada en texto.	2	0	0
Complejo	Una persona que interactúa con el sistema mediante una interfaz gráfica.	3	2	6
			<b>Total</b>	<b>7</b>

**Tabla 4.2.2:** Factores de peso de los actores

Ahora vamos a determinar el factor de peso en los casos de uso sin ajustar (UUCW).

Tipo de caso de uso	Descripción	Factor de peso	de Número de casos de uso	Resultado
Simple	De 1 a 3 Transacciones.	5	10	50
Promedio	De 4 a 7 Transacciones.	10	3	30
Complejo	Más de 8 Transacciones.	15	2	30
<b>Total</b>				<b>110</b>

**Tabla 4.2.3:** Factores de peso de los caso de uso

Ahora que ya tenemos el peso de los actores sin ajustar y el peso de los caso de uso sin ajustar, sumamos ambas cantidades

$$\text{UUCP} = \text{UAW} + \text{UUCW}$$

$$\text{UUCP} = 7 + 110$$

$$\text{UUCP} = 117$$

### Cálculo de puntos de caso de uso ajustados

Ahora que ya tenemos el valor de los puntos de caso de uso sin ajustar(UUCP), procedemos a ajustarlos por medio de la siguiente ecuación

$$\text{UCP} = \text{UUCP} \times \text{TCF} \times \text{EF} \text{ donde,}$$

- UCP: Puntos de Casos de Uso ajustados
- UUCP: Puntos de Casos de Uso sin ajustar
- TCF: Factor de complejidad técnica
- EF: Factor de ambiente

### Determinando el factor de complejidad técnica

Cada uno de los factores se cuantifica con un valor de 0 a 5, donde 0 significa un aporte irrelevante y 5 un aporte muy importante.

Número de factor	Descripción	Peso	Valor	Factor
T1	Sistema distribuido	2	3	6
T2	Tiempo de respuesta	1	2	2
T3	Eficiencia por el usuario	1	3	3
T4	Proceso interno complejo	1	3	3
T5	Reusabilidad	1	3	3
T6	Facilidad de instalación	0.5	1	0.5
T7	Facilidad de uso	0.5	5	2.5
T8	Portabilidad	2	1	2
T9	Facilidad de de cambio	1	4	4
T10	Concurrencia	1	4	4
T11	Objetivos especiales de seguridad	1	4	3
T12	Acceso directo a terceras partes	1	4	4
T13	Facilidades especiales de entrenamiento a usuarios finales	1	1	1
			<b>Total</b>	<b>38</b>

**Tabla 4.2.4:** Factores de complejidad técnica

Para obtener El Factor de Complejidad Técnica, desarrollamos la siguiente formula.

$$TCF = 0.6 + 0.01 * \sum(Peso_i * Valorasignado)$$

$$TCF = 0.6 + 0.01 * 38$$

$$TCF = 0.98$$

### Determinando el factor ambiente

Las habilidades y el entrenamiento del grupo involucrado en el desarrollo tienen un gran impacto en las estimaciones de tiempo. Estos factores son los que se contemplan en el cálculo del Factor de ambiente. El valor de cada factor va de 0 a 5, donde 0 es irrelevante y 5 muy importante

Número de factor	Descripción	Peso	Valor	Factor
E1	Familiaridad con el modelo del proyecto usado	1.5	3	4.5
E2	Experiencia en la aplicación	0.5	3	1.5
E3	Experiencia en POO	1	4	4
E4	Capacidad del análisis líder	0.5	4	2
E5	Motivación	1	5	5
E6	Estabilidad de los requerimientos	2	5	10
E7	Personal media jornada	-1	2	-2
E8	Dificultad en el lenguaje de programación	-1	1	-1
			<b>Total</b>	<b>24</b>

**Tabla 4.2.5:** Factores de ambiente

El Factor Ambiente se calcula mediante la siguiente ecuación

$$EF = 1.4 - 0.03 * \sum(Peso_i * Valor_{signado})$$

$$EF = 1.4 - 0.03 * 24$$

$$EF = 0.68$$

Para obtener los Puntos de Casos Uso Ajustados, desarrollamos lo siguiente

$$UCP = UUCP * TCF * EF$$

$$UCP = 117 * 0.98 * 0.68$$

$$UCP = 77.97$$

#### 4.2.1. Cálculo del esfuerzo

El esfuerzo horas-hombre se obtiene de la siguiente manera:

$$E = UCP \times CF \text{ donde,}$$

- E: Esfuerzo estimado en horas-hombre.

- UCP: Puntos de Casos de Uso ajustados.
- CF: Factor de conversión (20 horas-hombre por defecto)

$$E = UCP * CF$$

$$E = 91.76 * 20 \text{ Horas-Hombre}$$

$$E = 1559.38 \text{ Horas-Hombre}$$

Se debe tener en cuenta que éste método proporciona una estimación del esfuerzo en horas-hombre contemplando sólo el desarrollo de la funcionalidad especificada en los casos de uso.

Para la obtención de una estimación de esfuerzo mas exactas se debe incluir las otras actividades que se llevaron acabo en el desarrollo del proyecto; así que la distribución del esfuerzo entre dichas actividades está dada por la siguiente proyección 4.3.15:

- **Distribución genérica del esfuerzo**

Actividad	Porcentaje
Análisis	6.00 %
Levantamiento de requerimiento	4.00 %
Diseño de sistema	8.00 %
Diseño de aplicaciones	6.00 %
Programación	50.00 %
Documentación	16.00 %
Pruebas	2.00 %
Aprendizaje	3.00 %
Elaboración de Manuales	3.00 %
Asesoría y Capacitación	2.00 %

**Tabla 4.2.6:** Distribución genérica del esfuerzo

- **Distribución real del esfuerzo**

A partir de la distribución anterior, podemos obtener esfuerzo horas-hombre en cada una de las actividades del proyecto.

<b>Actividad</b>	<b>Esfuerzo horas-hombre</b>
Análisis	187.13
Levantamiento de requerimiento	124.75
Diseño de sistema	249.50
Diseño de aplicaciones	187.13
Programación	1559.38
Documentación	498.99
Pruebas	62.38
Aprendizaje	93.56
Elaboración de Manuales	93.56
Asesoría y Capacitación	62.38

**Tabla 4.2.7:** Distribución genérica del esfuerzo

### 4.3. Estudio de Factibilidad

La definición de viabilidad va mucho más allá del uso común del término, ya que existen tres formas principales para evaluar la viabilidad de los proyectos de sistemas: en base a su operación, a su capacidad técnica y a su economía.

El estudio de viabilidad no es un estudio detallado de sistemas, sino que se utiliza para recopilar datos más generales para los miembros de la administración, lo cual a su vez les permite tomar una decisión en cuanto a si deben continuar o no con un estudio de sistemas.

Por lo general, el analista de sistemas entrevista a las personas que piden ayuda y a las que están relacionadas en forma directa con el proceso de toma de decisiones, que generalmente son los administradores.<sup>117</sup>

Como ya se han planteado los objetivos razonables para el proyecto, ahora se necesita determinar si es posible que la UCFS y especialmente los miembros de la Unidad de Saneamiento Ambiental puedan ver el proyecto hasta su finalización.

Partiendo de datos obtenidos por medio de entrevistas al personal interesado en el desarrollo del proyecto, se debe mostrar que es viable en los tres aspectos siguientes: técnico, económico y operacional.

#### 4.3.1. *Factibilidad Técnica*

Se llevó a cabo un análisis tecnológico de los factores mas relevantes que determinan la viabilidad del proyecto los cuales, se basan en cuatro ámbitos esenciales para desarrollar un mejor estudio y estos se detallan a continuación:<sup>118</sup>

##### 1. **Recurso Humano**

En el equipo de desarrollo se cuenta con cuatro miembros y se pone a disposición el conocimiento de cada uno para aplicarlos en este proyecto, asignando tareas

---

<sup>117</sup>Kendall Kendall. 2011, Análisis y diseño de sistemas, 8th edición, p62

<sup>118</sup>Estudio de factibilidad Recuperado de <https://virtual.itca.edu.sv/Mediadores/ads/125>

específicas de acuerdo a la etapa del sistema en la que se encuentre.

Los conocimientos requeridos del equipo para el desarrollo del sistema informático son los siguientes:

- Manejo del gestor de Base de Datos MySQL Server,
- Conocimientos de lenguajes de programación: PHP, JavaScript, Dart.
- Conocimiento de HTML5, CSS3, BOOTSTRAP.

Cada miembro del equipo aportara sus conocimientos en las áreas antes mencionadas y de acuerdo a las necesidades que surjan a lo largo de su desarrollo.

## 2. Hardware

Los equipos con los que cuenta la UCSF en el área de Saneamiento Ambiental para la implementación del sistema web son:

Modelo	Especificaciones	
<b>Equipo 1</b> HP ProDesk 400 G3	Sistema Operativo	Windows 8.1 Pro
	Procesador	Intel Core i5 6th Gen
	Almacenamiento	1 TB
	Memoria RAM	4GB
	Tipo de Sistema	64 bits
<b>Equipo 2</b> HP ProDesk 400 G3	Sistema Operativo	Windows 10 Pro
	Procesador	Intel Core i3 7th Gen
	Almacenamiento	1 TB
	Memoria RAM	4GB
	Tipo de Sistema	32 bits
<b>Equipo 3</b> Impresora Multifunción Láser	Marca	Samsung
	Modelo	ProXpress M4580FX

**Tabla 4.3.1:** Especificaciones de Hardware de UCSF para el sistema web

Los Dispositivos móviles con los que cuenta el personal del área de saneamiento

ambiental para la implementación de la aplicación móvil son:

<b>Modelo</b>	<b>Especificaciones</b>	
<b>Dispositivo 1</b> Samsung A50	Sistema Operativo	Android 9
	Procesador	Exynos 9610
	Almacenamiento	64GB
	Memoria RAM	4GB
<b>Dispositivo 2</b> Huawei y6 2019	Sistema Operativo	Android 9
	Procesador	Mediatek Helio A22
	Almacenamiento	32GB
	Memoria RAM	2GB

**Tabla 4.3.2:** Especificaciones de los dispositivos móvil de los inspectores de UCSF para la app

Los equipos informáticos con los que se cuenta para el desarrollo del proyecto son los que se detallan a continuación:

<b>Modelo</b>	<b>Especificaciones</b>	
<b>Equipo 1</b> HP ProBook 6470b	Sistema Operativo	Windows 7 Ultimante
	Procesador	Intel Core i5 3th Gen
	Almacenamiento	512 GB
	Memoria RAM	8GB
	Tipo de Sistema	64 bits
<b>Equipo 2</b> HP1000	Sistema Operativo	Windows 10 Pro
	Procesador	Intel Celeron
	Almacenamiento	500GB
	Memoria RAM	4GB
	Tipo de Sistema	64 bits
<b>Equipo 3</b> PC Desktop Pert BioStar	Sistema Operativo	Windows 10 Pro
	Procesador	Pentium Dual-Core
	Almacenamiento	512 GB
	Memoria RAM	2GB
	Tipo de Sistema	32 bits
<b>Equipo 4</b> HP Laptop 14-dq1xxx	Sistema Operativo	Windows 10 Pro
	Procesador	Intel Core i5 10th Gen
	Almacenamiento	512 GB
	Memoria RAM	8GB
	Tipo de Sistema	64 bits

**Tabla 4.3.3:** Especificaciones de Hardware para el desarrollo

Los dispositivos móviles con los que se cuenta para la ejecución de las pruebas de las app se detallan a continuación:

Modelo	Especificaciones	
<b>Dispositivo 1</b> Redmi note 7	Sistema Operativo	Android 9
	Procesador	Snapdragon 660
	Almacenamiento	64GB
	Memoria RAM	4GB
<b>Dispositivo 2</b> Xiaomi Mi A2	Sistema Operativo	Android 10
	Procesador	Snapdragon 660
	Almacenamiento	64GB
	Memoria RAM	4GB
<b>Dispositivo 3</b> Gtouch Stella Omega	Sistema Operativo	Android 8.1
	Procesador	Quad Core 1.3GHz
	Almacenamiento	16GB
	Memoria RAM	1GB
<b>Dispositivo 4</b> Huawei Eco	Sistema Operativo	Android 5.1
	Procesador	Mediatek 1GHz
	Almacenamiento	8GB
	Memoria RAM	1GB

**Tabla 4.3.4:** Especificaciones de los dispositivos móvil para la ejecución de las pruebas

### 3. Software

El elemento software es de suma importancia para la complejión satisfactoria del proyecto y este debe de cumplir con ciertos requisitos de acuerdo al tipo de sistema a desarrollar; en este sera un sistema web y una aplicación móvil. De acuerdo a los conocimientos por parte del equipo de desarrollo se presentan los software necesarios e indispensables que se usaran:

<b>Nombre</b>	<b>Tipo de Licencia</b>	<b>Descripción</b>
MySQL	GPL	Gestor de base de datos relacional
Apache	Apache License	Servidor web de código abierto
Git (2.25.1)	GPL	Software para gestionar ágilmente los cambios en el código fuente
Google Chrome (83.0)	Freeware	Navegador web
Mozilla Firefox (78.01)	MPL	Navegador Web
Visual Studio Code (1.46.1)	Propietaria	Editor de texto para archivos de código fuente
Windows 10	OEM	Sistema operativo
GNU/Linux	GPL	Sistema operativo

**Tabla 4.3.5:** Herramientas para la construcción del sistema web

<b>Nombre</b>	<b>Tipo de Licencia</b>	<b>Descripción</b>
Flutter (1.12)	Licencia BSD	Kit de desarrollo móvil
Dart	Licencia BSD	Leguaje de Programación
Visual Studio Code (1.46.1)	Propietaria	Editor de código fuente
Android SDK	Gratuita	Herramienta de desarrollo
Google Chrome (83.0)	Freeware	Navegador web
Mozilla Firefox (78.01)	MPL	Navegador Web
Windows 10	OEM	Sistema operativo
GNU/Linux	GPL	Sistema operativo
Android Studio	Apache 2.0	Entorno de desarrollo
Firebase	Gratuita	Plataforma de desarrollo

**Tabla 4.3.6:** Herramientas para la construcción de la Aplicación Móvil

### **Conclusión de Factibilidad Técnica**

Se cuenta con el equipo necesario para el desarrollo del Sistema Informático, tanto en hardware como software, así como también el equipo de desarrollo esta capacitado con los conocimientos necesarios para el desarrollo de cada una de las etapas se cumplan de manera satisfactoria, brindando los resultados esperados. Por lo tanto, se concluye que el desarrollo del sistema informático en estudio es técnicamente factible.

#### ***4.3.2. Factibilidad Financiera***

El estudio de la factibilidad económica permite realizar una evaluación sobre la conveniencia de invertir o no en un proyecto determinado. Dicha factibilidad se establece detallando todos aquellos costos involucrados en el desarrollo, implementación y operación del sistema que se plantea.

Se presentan a continuación los costos de desarrollo del sistema de forma que se detallan los procesos e insumos a necesitar ya que La Unidad Comunitaria de Salud Familiar de San Francisco Gotera Morazán no incurrirá con estos gastos de desarrollo del sistema web.

#### **■ Recursos tecnológicos**

Los recursos tecnológicos son aquellos medios que utilizan la tecnología para que se pueda llevar a cabo el propósito que se desea.<sup>119</sup>

Estos pueden ser físicos, también conocidos como tangibles, y los que son invisibles, conocidos como intangibles o transversales.

---

<sup>119</sup>Recursos tecnológicos Extraído de <https://enciclopediaeconomica.com/recursos-tecnologicos/>

Rubros	Cantidad	Costo Mes(\$)	Meses	Costo Unitario(\$)	Total(\$)
<b>Hardware</b>					
Computadora Laptop	4	-	-	400.00	1,600.00
Depreciación de Hardware por uso (20 %)	-	6.66	9	-	60.00
<b>Software</b>					
Microsoft Office	4	-	-	-	-
Editor de Latex	4	-	-	-	-
Visual Studio Code	4	-	-	-	-
MySQL	4	-	-	-	-
Apache	4	-	-	-	-
Flutter	4	-	-	-	-
Amortización del software (16.8 %)	-	-	-	-	0
<b>Internet</b>					
Internet Residencial 10 megas	4	100.00	9	25.00	900.00
<b>Total \$ 2,560.00</b>					

**Tabla 4.3.7:** Costos de Recursos tecnológicos

- **Recursos de Material Fungible**

Estos recursos son tomados en cuenta para la elaboración y entrega de la documentación; la siguiente tabla detallara cada uno de los costos que estos insumos tendrán durante el periodo de realización del proyecto.

- **Costos de operación**

Detalle	Cantidad	Costo U (\$)	Total(\$)
Papelería	-	-	20.00
Útiles Varios	-	-	25.00
Impresiones	-	-	200.00
Fotocopias	-	-	30.00
<b>Total \$ 275.00</b>			

**Tabla 4.3.8:** Costos de Material Fungible

- **Estimación de esfuerzo por medio del método Puntos de Casos de Uso**

Para la estimación usando el método de puntos de caso de uso, primero identificamos los actores y caso de uso asociados.

Nombre del actor	Casos de uso asociados
Súper usuario	Registrar exámenes, registrar manipuladores, registrar usuario, gestionar operaciones
Administrador	Gestionar usuarios, gestionar manipuladores, gestionar exámenes, gestionar establecimientos
Manipulador	Recibir notificaciones, ver datos

**Tabla 4.3.9:** Actores y casos de uso asociados

- **Determinación del factor de peso de los actores sin ajustar (UAW).**

Tipo de actor	Descripción	Factor de peso	N° de actores	Resultado
Simple	Otro sistema que interactúa con el sistema a desarrollar mediante una interfaz de programación(API, Application Programming Interface)	1	1	1
Promedio	Otro sistema que interactúa con el sistema a desarrollar mediante un protocolo o una interfaz basada en texto.	2	0	0
Complejo	Una persona que interactúa con el sistema mediante una interfaz gráfica.	3	2	6
			<b>Total</b>	<b>7</b>

**Tabla 4.3.10:** Factores de peso de los actores

Ahora vamos a determinar el factor de peso en los casos de uso sin ajustar (UUCW).

Tipo de caso de uso	Descripción	Factor de peso	Número de casos de uso	Resultado
Simple	De 1 a 3 Transacciones.	5	10	50
Promedio	De 4 a 7 Transacciones.	10	3	30
Complejo	Más de 8 Transacciones.	15	2	30
			<b>Total</b>	<b>110</b>

**Tabla 4.3.11:** Factores de peso de los caso de uso

Ahora que ya tenemos el peso de los actores sin ajustar y el peso de los caso de uso sin ajustar, sumamos ambas cantidades

$$UUCP = UAW + UUCW$$

$$\text{UUCP} = 7 + 110$$

$$\text{UUCP} = 117$$

### **Cálculo de puntos de caso de uso ajustados**

Ahora que ya tenemos el valor de los puntos de caso de uso sin ajustar(UUCP), procedemos a ajustarlos por medio de la siguiente ecuación

$$\text{UCP} = \text{UUCP} \times \text{TCF} \times \text{EF} \text{ donde,}$$

- UCP: Puntos de Casos de Uso ajustados
- UUCP: Puntos de Casos de Uso sin ajustar
- TCF: Factor de complejidad técnica
- EF: Factor de ambiente

#### **• Determinando el factor de complejidad técnica**

Cada uno de los factores se cuantifica con un valor de 0 a 5, donde 0 significa un aporte irrelevante y 5 un aporte muy importante.

Número de factor	Descripción	Peso	Valor	Factor
T1	Sistema distribuido	2	3	6
T2	Tiempo de respuesta	1	2	2
T3	Eficiencia por el usuario	1	3	3
T4	Proceso interno complejo	1	3	3
T5	Reusabilidad	1	3	3
T6	Facilidad de instalación	0.5	1	0.5
T7	Facilidad de uso	0.5	5	2.5
T8	Portabilidad	2	1	2
T9	Facilidad de de cambio	1	4	4
T10	Concurrencia	1	4	4
T11	Objetivos especiales de seguridad	1	4	3
T12	Acceso directo a terceras partes	1	4	4
T13	Facilidades especiales de entrenamiento a usuarios finales	1	1	1
			<b>Total</b>	<b>38</b>

**Tabla 4.3.12:** Factores de complejidad técnica

Para obtener El Factor de Complejidad Técnica, desarrollamos la siguiente formula.

$$TCF = 0.6 + 0.01 * \sum(Peso_i * Valor asignado)$$

$$TCF = 0.6 + 0.01 * 38$$

$$TCF = 0.98$$

- **Determinando el factor ambiente**

Las habilidades y el entrenamiento del grupo involucrado en el desarrollo tienen un gran impacto en las estimaciones de tiempo. Estos factores son los que se contemplan en el cálculo del Factor de ambiente. El valor de cada factor va de 0 a 5, donde 0 es irrelevante y 5 muy importante

Número de factor	Descripción	Peso	Valor	Factor
E1	Familiaridad con el modelo del proyecto usado	1.5	3	4.5
E2	Experiencia en la aplicación	0.5	3	1.5
E3	Experiencia en POO	1	4	4
E4	Capacidad del análisis líder	0.5	4	2
E5	Motivación	1	5	5
E6	Estabilidad de los requerimientos	2	5	10
E7	Personal media jornada	-1	2	-2
E8	Dificultad en el lenguaje de programación	-1	1	-1
			<b>Total</b>	<b>24</b>

**Tabla 4.3.13:** Factores de ambiente

El Factor Ambiente se calcula mediante la siguiente ecuación

$$EF = 1.4 - 0.03 * \sum(Peso_i * Valor_{asignado})$$

$$EF = 1.4 - 0.03 * 24$$

$$EF = 0.68$$

Para obtener los Puntos de Casos Uso Ajustados, desarrollamos lo siguiente

$$UCP = UUCP * TCF * EF$$

$$UCP = 117 * 0.98 * 0.68$$

$$UCP = 77.97$$

- **Cálculo del esfuerzo**

El esfuerzo horas-hombre se obtiene de la siguiente manera:

**E = UCP x CF donde,**

- E: Esfuerzo estimado en horas-hombre.

- UCP: Puntos de Casos de Uso ajustados.
- CF: Factor de conversión (20 horas-hombre por defecto)

$$E = UCP * CF$$

$$E = 91.76 * 20 \text{ Horas-Hombre}$$

$$E = 1559.38 \text{ Horas-Hombre}$$

Se debe tener en cuenta que éste método proporciona una estimación del esfuerzo en horas-hombre contemplando sólo el desarrollo de la funcionalidad especificada en los casos de uso.

Para la obtención de una estimación de esfuerzo mas exactas se debe incluir las otras actividades que se llevaron acabo en el desarrollo del proyecto; así que la distribución del esfuerzo entre dichas actividades está dada por la siguiente proyección 4.3.15:

- **Distribución genérica del esfuerzo**

Actividad	Porcentaje
Análisis	6.00 %
Levantamiento de requerimiento	4.00 %
Diseño de sistema	8.00 %
Diseño de aplicaciones	6.00 %
Programación	50.00 %
Documentación	16.00 %
Pruebas	2.00 %
Aprendizaje	3.00 %
Elaboración de Manuales	3.00 %
Asesoría y Capacitación	2.00 %

**Tabla 4.3.14:** Distribución genérica del esfuerzo

- **Distribución real del esfuerzo**

A partir de la distribución anterior, podemos obtener esfuerzo horas-hombre

en cada una de las actividades del proyecto.

<b>Actividad</b>	<b>Esfuerzo horas-hombre</b>
Análisis	187.13
Levantamiento de requerimiento	124.75
Diseño de sistema	249.50
Diseño de aplicaciones	187.13
Programación	1559.38
Documentación	498.99
Pruebas	62.38
Aprendizaje	93.56
Elaboración de Manuales	93.56
Asesoría y Capacitación	62.38

**Tabla 4.3.15:** Distribución genérica del esfuerzo

Para realizar la estimación de los costos, tomamos como base las horas-hombre estimadas de acuerdo al método empleado, Puntos de Casos de Uso. Tomando en cuenta que el salario medio para un desarrollador junior en El Salvador van desde 629.74 hasta 725.70 dólares la mensuales, realizamos la siguiente estimación de costos con un salario de 2.00 dólares para nuestro grupo de desarrollo

Descripción	Horas Trabajadas (h)	Valor (\$)
Costos Directos		
Análisis	187.13	374.26
Levantamiento de requerimientos	124.75	249.50
Diseño de sistema	249.50	499.00
Diseño de aplicación	187.13	324.26
Documentación	498.99	997.98
Programación	1559.38	3 118.76
Pruebas	62.38	124.76
Costos Indirectos		
Aprendizaje	93.56	187.12
Elaboración de manuales	93.56	187.12
Asesorías y capacitaciones	62.38	124.76
<b>Total \$ 6187.52</b>		

**Tabla 4.3.16:** Costo de operación

■ **Gastos varios por servicios**

Detalle de los gastos indirectos que se tendrán en el desarrollo del proyecto de grado.

Servicios	Reunión(\$)	Reunión	Meses	Meses	N° Personas	Total(\$)
Transporte	2.50	4	9	4	360.00	
Comida	6.50	4	9	4	936.00	
Transporte lugar de investigación	2.00	1	9	4	72.00	
<b>Total \$ 1,368.00</b>						

**Tabla 4.3.17:** Costos Varios por Servicios

■ **Imprevistos 20 %**

Son gastos por actividades no estimadas cuando se trata de proyectos innovadores, nuevos en su área o en condiciones que se prevean alzas de precios en el mercado es

recomendable considerar tasas del 20 %. Este proyecto tendrá el 20 % de imprevistos

#### ■ Costo Total del Proyecto

Total de los rubros expuestos previamente 5,043.6 dólares en concepto de desarrollo.

<b>Recursos</b>	<b>Total(\$)</b>
Tecnológicos	2,560.00
Material Fungible	275.00
Gastos Varios por Servicios	1,368.00
Costos de Operación	6187.52
Sub-Total	10,390.52
Imprevistos 20 %	2,078.10
<b>Total \$ 12,468.62</b>	

**Tabla 4.3.18:** Costo Total del Proyecto

#### **4.3.3. Factibilidad Operativa**

La viabilidad operativa depende de los recursos humanos disponibles para el proyecto e implica la acción de pronosticar si el sistema funcionará y se utilizará una vez instalado.

En primera instancia se ha tomado en cuenta la necesidad expresada por parte del personal de la Unidad de Saneamiento ambiental, donde dan a conocer cuan útil les sería un sistema informático para agilizar su trabajo y mejorar su rendimiento, y así cumplir con la capacitación de más personas manipuladoras de alimentos y también llevar un control más estricto de estos; por medio de sus relatos es como se han conocido los requerimientos funcionales del sistema y también el enfoque que debe tener; por lo tanto se han comprometido con brindar toda la información que se requiera para completar el proyecto.

Por otro lado también dicho personal posee conocimientos de computación a un nivel básico, que es lo que se requiere para manejar un sistema de esta magnitud, por lo tanto, solo se les brindaría capacitaciones en aspectos específicos de los módulos que

contendrá para tal proceso se tiene completa disposición por parte del personal. En cuanto a la adquisición del servidor para la puesta en marcha del sistema, se ha dejado claro que se hará mediante el hosting y se ha tenido la aceptación por parte de la administración, ya que ellos se han comprometido con la gestión de los gastos que se incurra.<sup>120</sup>

En cuanto a la aplicación móvil tanto el personal de la UCSF del área de saneamiento ambiental como los manipuladores de alimentos han mostrado total aceptación por los beneficios que esta contendrá, además que no incurrirán con gastos porque será totalmente gratis.

En conclusión, se puede comprobar que el desarrollo del sistema es totalmente factible operativamente, por la aceptación en todos los rubros que se han mencionado en los párrafos anteriores además porque el MINSAL no le había prestado la atención necesaria de sistematizar esa área se ha mostrado agradecimiento por parte de la UCSF.

---

<sup>120</sup>El hosting es un servicio en línea que te permite publicar un sitio o aplicación web en Internet.

## 4.4. Diseño del sistema informático

### 4.4.1. Diagrama de Casos de uso

#### 4.4.1.1. Súper usuario



Figura 4.4.1: Diagrama de Casos de uso de Súper Usuario

4.4.1.2. Administrador



Figura 4.4.2: Diagrama de Casos de uso de administrador

## 4.4.1.3. Manipuladores

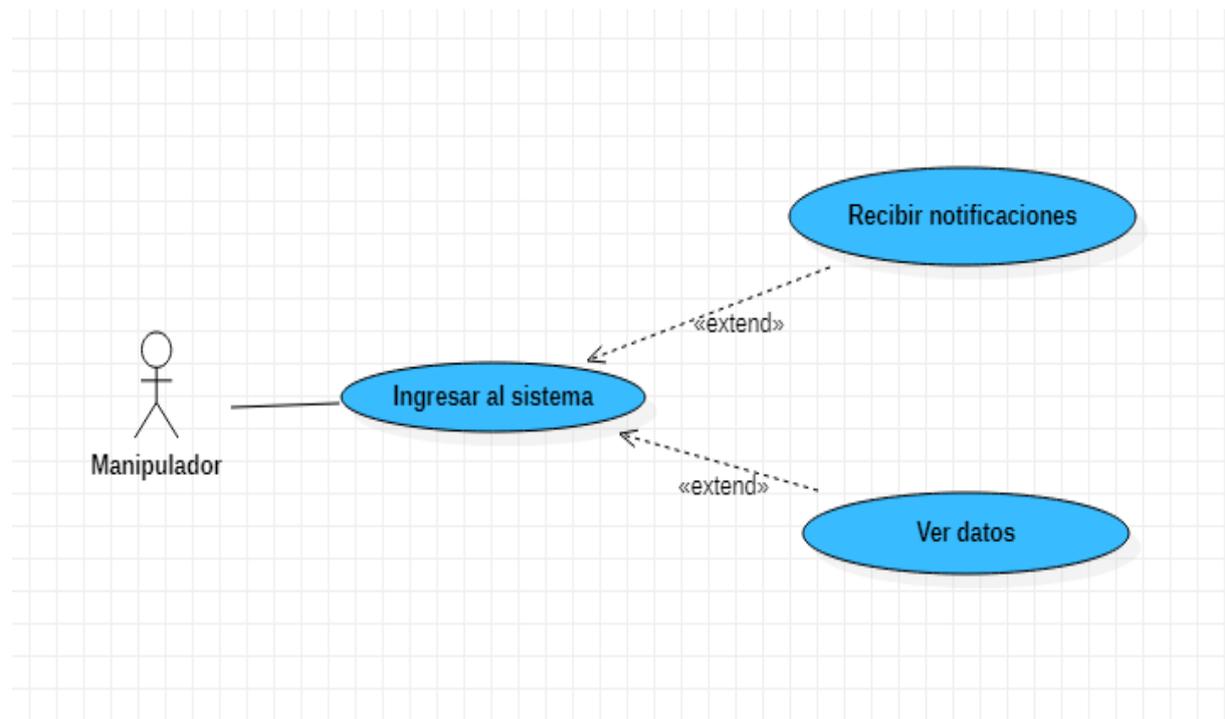


Figura 4.4.3: Diagrama de Casos de uso de manipuladores

4.4.2. *Narración de casos de uso*

<b>Nombre del caso de uso</b>	<b>Ingresar al sistema</b>	
<b>Prioridad</b>	Alta	
<b>Tipo</b>	Esencial	
<b>Actores</b>	Súper usuario	
<b>Descripción</b>	Inicialización de la sistema	
<b>Precondición</b>	Tener acceso a la URL del sistema y contar con un dispositivo con conexión a internet.	
<b>Ocasionador</b>	Se inicia cuando el usuario accede a la URL mediante un dispositivo conectado a internet.	
<b>Curso Típico</b>	<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	<b>Paso 1:</b> Acceder al sistema mediante la URL	<b>Paso 2:</b> Cargar la página de inicio de sesión
<b>Cursos Alternos</b>	<b>Paso Alternativo:</b> La página de inicio del sistema no carga porque el dispositivo no tiene conexión a internet.	
<b>Conclusión</b>	Este caso concluye cuando se carga la página de inicio del sistema.	
<b>Post condición</b>	N/A	
<b>Restricciones</b>	N/A	

Figura 4.4.4: Narración de caso de uso: Ingresar al sistema

<b>Nombre del caso de uso</b>	<b>Registrar Usuarios</b>	
<b>Prioridad</b>	Alta	
<b>Tipo</b>	Esencial	
<b>Actores</b>	Súper usuario	
<b>Descripción</b>	El administrador registra los datos de los usuarios	
<b>Precondición</b>	Contar con credenciales de súper usuario	
<b>Ocasionador</b>	Se inicia cuando el administrador ingresa al módulo usuarios	
<b>Curso Típico</b>	<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	<b>Paso 1:</b> Presionar el botón “Nuevo Usuario”	<b>Paso 2:</b> Muestra el formulario con los campos requeridos para dicho registro
	<b>Paso 3:</b> Completar los campos requeridos y presiona el botón guardar	<b>Paso 4:</b> Guarda la información ingresada por el administrador.
<b>Cursos Alternos</b>	<b>Paso Alternativo:</b> El administrador ingresa datos inválidos y el registro no se guarda	
<b>Conclusión</b>	Este caso concluye cuando se guardan los datos del nuevo usuario.	
<b>Post condición</b>	N/A	
<b>Restricciones</b>	Este caso es de uso exclusivo de los usuarios con credenciales de súper usuario.	

Figura 4.4.5: Narración de caso de uso: Registrar Usuarios

<b>Nombre del caso de uso</b>	<b>Gestionar Usuarios</b>	
<b>Prioridad</b>	Alta	
<b>Tipo</b>	Esencial	
<b>Actores</b>	Súper usuario	
<b>Descripción</b>	El administrador actualiza o desactiva los datos de los usuarios	
<b>Precondición</b>	Contar con credenciales de súper usuario	
<b>Ocasionador</b>	Se inicia cuando el administrador ingresa al módulo usuarios	
<b>Curso Típico</b>	<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	<b>Paso 1:</b> Presionar el botón “Actualizar Usuario”	<b>Paso 2:</b> Carga los datos actuales del usuario.
	<b>Paso 3:</b> Modifica los campos que desea actualizar	<b>Paso 4:</b> Guarda la información con los datos que se han modificado.
<b>Cursos Alternos</b>	<b>Paso Alternativo:</b> El administrador ingresa datos inválidos y el registro no se actualiza.	
<b>Conclusión</b>	Este caso concluye cuando se almacenan los nuevos datos del usuario seleccionado.	
<b>Post condición</b>	N/A	
<b>Restricciones</b>	Este caso es de uso exclusivo de los usuarios con credenciales de administrador.	

Figura 4.4.6: Narración de caso de uso: Gestionar Usuarios

<b>Nombre del caso de uso</b>	<b>Registrar Manipuladores</b>	
<b>Prioridad</b>	Alta	
<b>Tipo</b>	Esencial	
<b>Actores</b>	Súper usuario y administrador	
<b>Descripción</b>	Registra los datos de los nuevos manipuladores.	
<b>Precondición</b>	Que el sistema haya cargado correctamente.	
<b>Ocasionador</b>	Se inicia cuando el usuario ingresa al módulo manipuladores	
<b>Curso Típico</b>	<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	<b>Paso 1:</b> Presionar el botón “Nuevo Manipulador”	<b>Paso 2:</b> Carga los el formulario con los campos requeridos para almacenar el registro.
	<b>Paso 3:</b> Presiona el botón “Guardar”	<b>Paso 4:</b> Guarda la información con los datos que se han ingresado.
<b>Cursos Alternos</b>	<b>Paso Alternativo:</b> El usuario ingresa datos inválidos y el registro no se almacena.	
<b>Conclusión</b>	Este caso concluye cuando se almacenan los datos del nuevo manipulador.	
<b>Post condición</b>	N/A	
<b>Restricciones</b>	N/A	

Figura 4.4.7: Narración de caso de uso: Registrar Manipuladores

<b>Nombre del caso de uso</b>	<b>Gestionar Manipuladores</b>	
<b>Prioridad</b>	Alta	
<b>Tipo</b>	Esencial	
<b>Actores</b>	Súper usuario y administrador	
<b>Descripción</b>	El usuario actualiza o desactiva los datos de los manipuladores.	
<b>Precondición</b>	Que el sistema haya cargador correctamente.	
<b>Ocasionador</b>	Se inicia cuando el administrador ingresa al módulo manipuladores.	
<b>Curso Típico</b>	<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	<b>Paso 1:</b> Presionar el botón “Actualizar Manipulador”	<b>Paso 2:</b> Carga los datos actuales del manipulador.
	<b>Paso 3:</b> Modifica los campos que desea actualizar.	<b>Paso 4:</b> Guarda la información con los datos que se han modificado.
<b>Cursos Alternos</b>	<b>Paso Alternativo:</b> El administrador ingresa datos inválidos y el registro no se actualiza.	
<b>Conclusión</b>	Este caso concluye cuando se almacenan los nuevos datos del manipulador seleccionado.	
<b>Post condición</b>	N/A	
<b>Restricciones</b>	N/A	

Figura 4.4.8: Narración de caso de uso: Gestionar Manipuladores

<b>Nombre del caso de uso</b>	<b>Registrar Establecimientos</b>	
<b>Prioridad</b>	Alta	
<b>Tipo</b>	Esencial	
<b>Actores</b>	Súper usuario y administrador	
<b>Descripción</b>	Registra los datos de los nuevos establecimientos.	
<b>Precondición</b>	Que el sistema haya cargado correctamente.	
<b>Ocasionador</b>	Se inicia cuando el usuario ingresa al módulo Establecimientos.	
<b>Curso Típico</b>	<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	<b>Paso 1:</b> Presionar el botón “Nuevo Establecimiento”	<b>Paso 2:</b> Carga el formulario con los campos requeridos para almacenar el registro.
	<b>Paso 3:</b> Presiona el botón “Guardar”	<b>Paso 4:</b> Guarda la información con los datos que se han ingresado.
<b>Cursos Alternos</b>	<b>Paso Alternativo:</b> El usuario ingresa datos inválidos y el registro no se almacena.	
<b>Conclusión</b>	Este caso concluye cuando se almacenan los datos del nuevo establecimiento.	
<b>Post condición</b>	N/A	
<b>Restricciones</b>	N/A	

Figura 4.4.9: Narración de caso de uso: Registrar Establecimientos

<b>Nombre del caso de uso</b>	<b>Gestionar Establecimientos</b>	
<b>Prioridad</b>	Alta	
<b>Tipo</b>	Esencial	
<b>Actores</b>	Súper usuario y administrador	
<b>Descripción</b>	El usuario actualiza o desactiva los datos de los establecimientos.	
<b>Precondición</b>	Que el sistema haya cargado correctamente.	
<b>Ocasionador</b>	Se inicia cuando el administrador ingresa al módulo establecimientos.	
<b>Curso Típico</b>	<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	<b>Paso 1:</b> Presionar el botón “Actualizar Establecimiento”	<b>Paso 2:</b> Carga los datos actuales del establecimiento.
	<b>Paso 3:</b> Modifica los campos que desea actualizar.	<b>Paso 4:</b> Guarda la información con los datos que se han modificado.
<b>Cursos Alternos</b>	<b>Paso Alternativo:</b> El administrador ingresa datos inválidos y el registro no se actualiza.	
<b>Conclusión</b>	Este caso concluye cuando se almacenan los nuevos datos del establecimiento seleccionado.	
<b>Post condición</b>	N/A	
<b>Restricciones</b>	N/A	

**Figura 4.4.10:** Narración de caso de uso: Gestionar Establecimientos

<b>Nombre del caso de uso</b>	<b>Búsqueda de Registros.</b>	
<b>Prioridad</b>	Alta	
<b>Tipo</b>	Esencial	
<b>Actores</b>	Súper usuario y administrador	
<b>Descripción</b>	Realiza una búsqueda específica de un registro.	
<b>Precondición</b>	Que el sistema haya cargado correctamente.	
<b>Ocasionador</b>	Se inicia cuando el usuario ingresa a un módulo que contenga una barra de búsqueda.	
<b>Curso Típico</b>	<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	<b>Paso 1:</b> Ingresa datos de un registro en la barra de búsqueda.	<b>Paso 2:</b> Muestra los datos que se han encontrado con los datos ingresados.
<b>Cursos Alternos</b>	<b>Paso Alternativo:</b> El administrador ingresa datos que no están definidos en la búsqueda.	
<b>Conclusión</b>	Este caso concluye cuando se muestra el registro ingresado en la búsqueda.	
<b>Post condición</b>	N/A	
<b>Restricciones</b>	N/A	

Figura 4.4.11: Narración de caso de uso: Búsqueda de Registros

<b>Nombre del caso de uso</b>	<b>Generación de Reportes</b>	
<b>Prioridad</b>	Alta	
<b>Tipo</b>	Esencial	
<b>Actores</b>	Súper usuario y administrador	
<b>Descripción</b>	Muestra los reportes requeridos por el usuario.	
<b>Precondición</b>	Que el sistema haya cargado correctamente.	
<b>Ocasionador</b>	Se inicia cuando el administrador ingresa al módulo de reportes.	
<b>Curso Típico</b>	<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	<b>Paso 1:</b> Presionar el botón “Generar reportes” : Usuarios, Establecimientos, Manipuladores	<b>Paso 2:</b> Muestra todos los datos que el usuario solicita.
<b>Cursos Alternos</b>	<b>Paso Alternativo:</b> N/A	
<b>Conclusión</b>	Este caso concluye cuando se muestra el reporte completo que el usuario ha solicitado.	
<b>Post condición</b>	N/A	
<b>Restricciones</b>	N/A	

**Figura 4.4.12:** Narración de caso de uso: Generación de Reportes

<b>Nombre del caso de uso</b>	<b>Inspección de Establecimientos</b>	
<b>Prioridad</b>	Alta	
<b>Tipo</b>	Esencial	
<b>Actores</b>	Súper usuario y administrador	
<b>Descripción</b>	Permite el manejo de las inspecciones que se le realizan a los establecimientos.	
<b>Precondición</b>	Que el sistema haya cargado correctamente.	
<b>Ocasionador</b>	Se inicia cuando el administrador ingresa al módulo Inspecciones.	
<b>Curso Típico</b>	<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	<b>Paso 1:</b> Seleccionar el establecimiento a inspeccionar y presionar el botón <i>inspeccionar</i> .	<b>Paso 2:</b> Carga los datos del establecimiento a inspeccionar.
	<b>Paso 3:</b> Completa los campos necesarios para la inspección	<b>Paso 4:</b> Guarda los registros de la inspección realizada al establecimiento.
<b>Cursos Alternos</b>	<b>Paso Alternativo:</b> N/A	
<b>Conclusión</b>	Este caso concluye cuando guardan los registros de la inspección.	
<b>Post condición</b>	N/A	
<b>Restricciones</b>	N/A	

Figura 4.4.13: Narración de caso de uso: Inspección de establecimientos

<b>Nombre del caso de uso</b>	<b>Resultados de exámenes</b>	
<b>Prioridad</b>	Alta	
<b>Tipo</b>	Esencial	
<b>Actores</b>	Súper usuario y administrador	
<b>Descripción</b>	Permite el almacenar y gestionar los resultados de exámenes que entregan los manipuladores.	
<b>Precondición</b>	Que el sistema haya cargado correctamente.	
<b>Ocasionador</b>	Se inicia cuando el administrador ingresa al módulo Exámenes.	
<b>Curso Típico</b>	<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	<b>Paso 1:</b> Presionar el botón “Inspeccionar” : Usuarios, Establecimientos, Manipuladores	<b>Paso 2:</b> Carga los datos del establecimiento a inspeccionar.
	<b>Paso 3:</b> Completa los campos necesarios para la inspección	<b>Paso 4:</b> Guarda los registros de la inspección realizada al establecimiento.
<b>Cursos Alternos</b>	<b>Paso Alternativo:</b> N/A	
<b>Conclusión</b>	Este caso concluye cuando guardan los registros de la inspección.	
<b>Post condición</b>	N/A	
<b>Restricciones</b>	N/A	

Figura 4.4.14: Narración de caso de uso: Resultados de exámenes

### 4.4.3. *Diagrama de Secuencia*

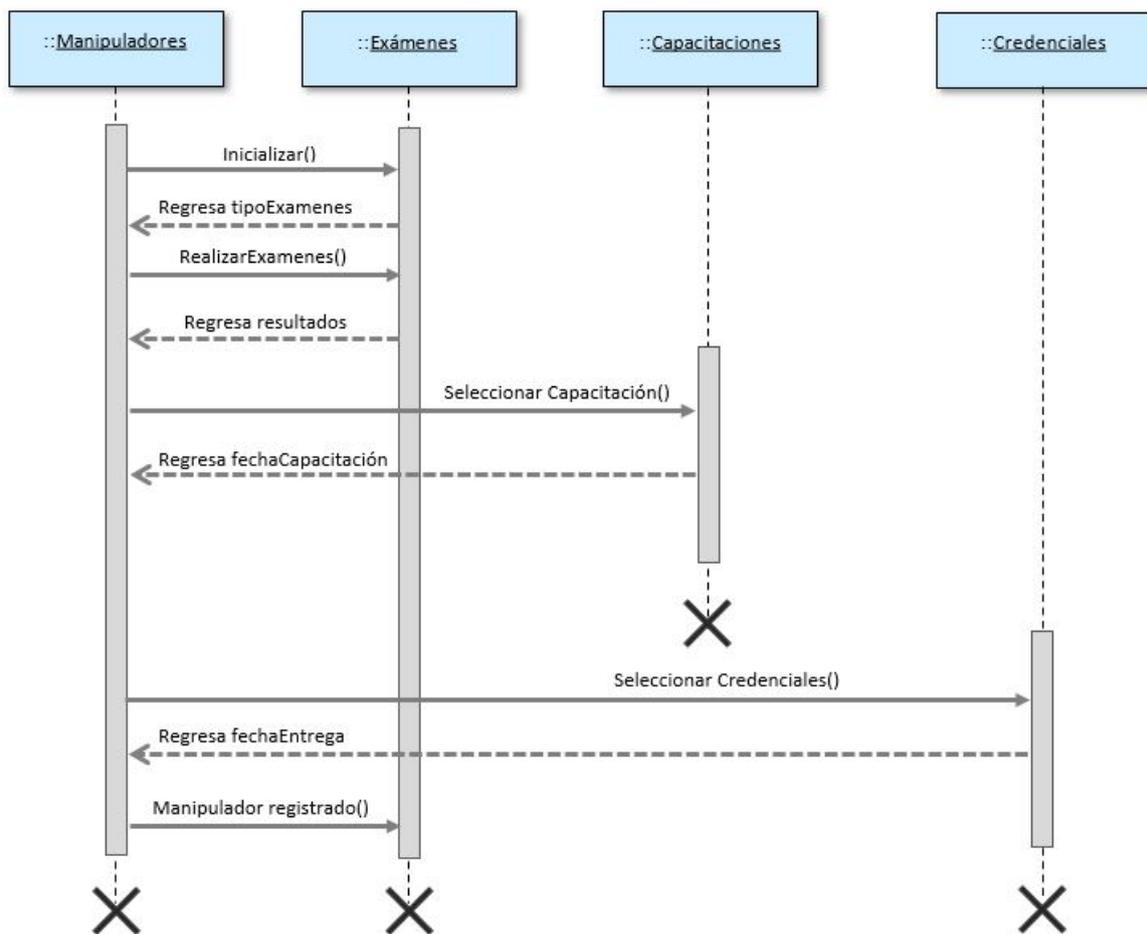
Los diagramas de secuencia pueden ilustrar una sucesión de interacciones entre clases o instancias de objetos a través del tiempo. A menudo, los diagramas de secuencia se utilizan para ilustrar el procesamiento descrito en los escenarios de casos de uso. En la práctica, los diagramas de secuencia se derivan del análisis de casos de uso y se utilizan en el diseño de sistemas para derivar las interacciones, las relaciones y los métodos de los objetos en el sistema. Los diagramas de secuencia se utilizan para mostrar el patrón general de las actividades o interacciones en un caso de uso. Cada escenario de caso de uso puede <sup>121</sup>

El diagrama de secuencia lo usaremos como herramienta que nos permita ilustrar los procedimientos descritos en los escenarios de casos de usos, así, como ilustrar las sucesiones de interacciones de instancias de objetos a través del tiempo.

En el diagrama de secuencia principal vemos la interacción que existe entre las diferentes clases con el autor principal, que en este caso sería el manipulador de alimentos, tales como exámenes, capacitaciones y las credenciales. Figura 4.4.15.

---

<sup>121</sup>Kendall y Kendall, Análisis y Diseño de Sistema, Pearson, Octava Edición (2011)



**Figura 4.4.15:** Diagrama de Secuencia: proceso general de inscripción de un nuevo manipulador de alimentos

4.4.3.1. Diagrama de secuencias inicio de sesión

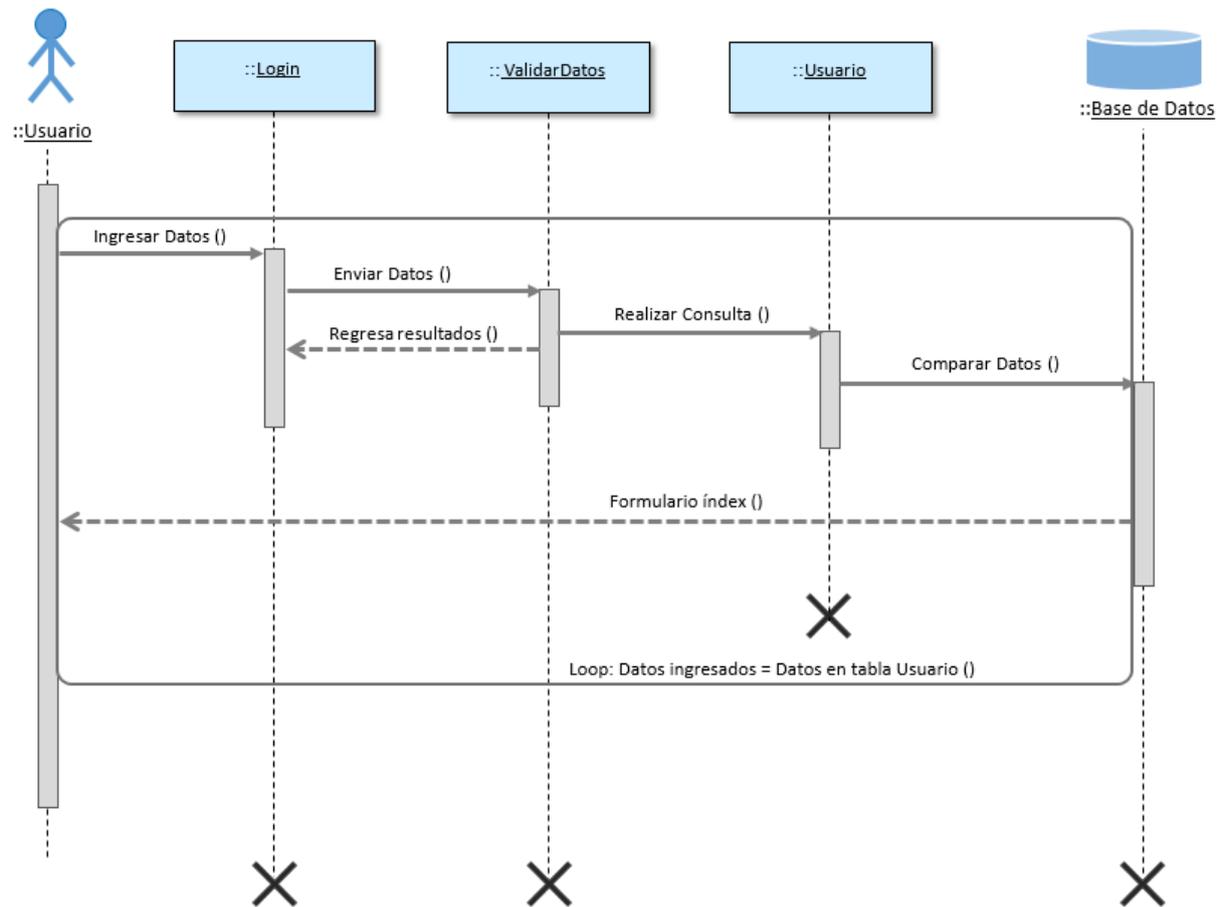


Figura 4.4.16: Diagrama de Secuencia: proceso de inicio de sesión

### 4.4.3.2. Interfaz de Inicio de sesión

Uno de los Procesos principales dentro del sistema es la identificación de usuario, es decir el inicio de sesión del usuario, en el cuál deberá ingresar sus credenciales que fueron autorizadas por el usuario administrador.

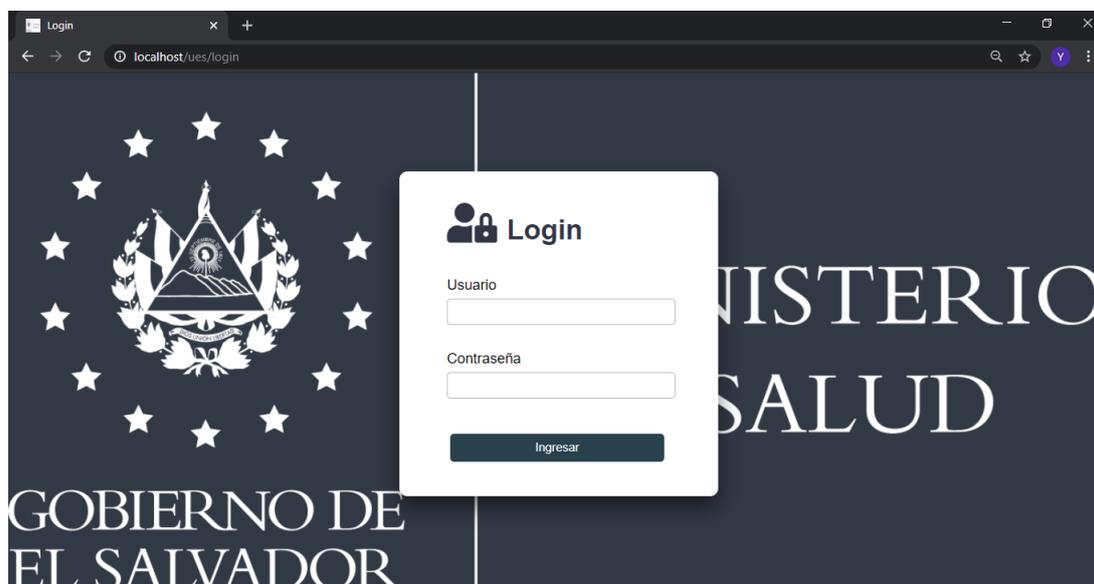


Figura 4.4.17: Interfaz: Inicio de Sesión

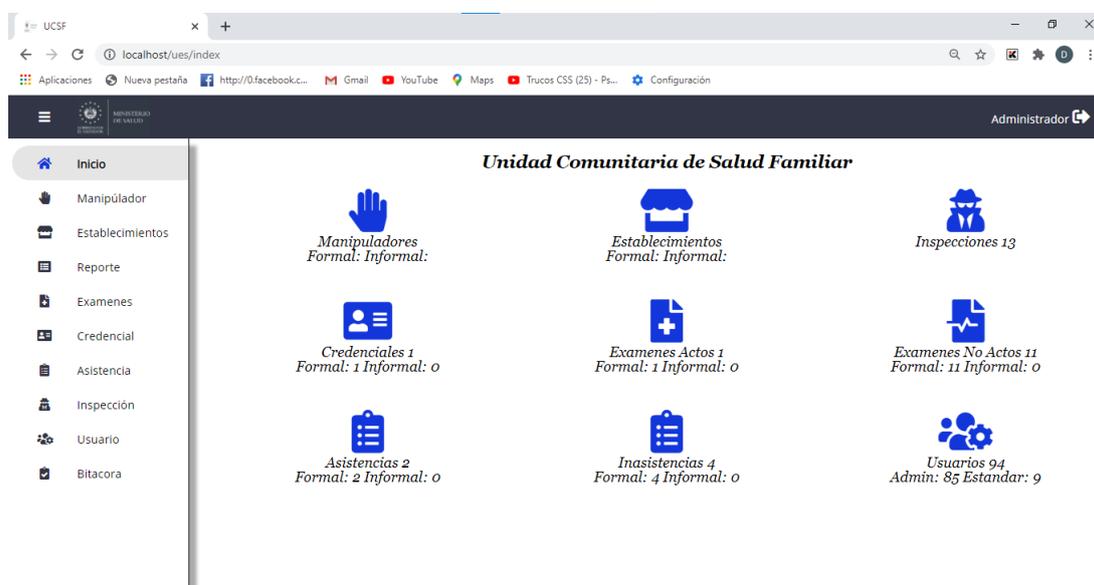


Figura 4.4.18: Interfaz: Pagina principal

4.4.3.3. Diagrama de secuencias procesos de usuario

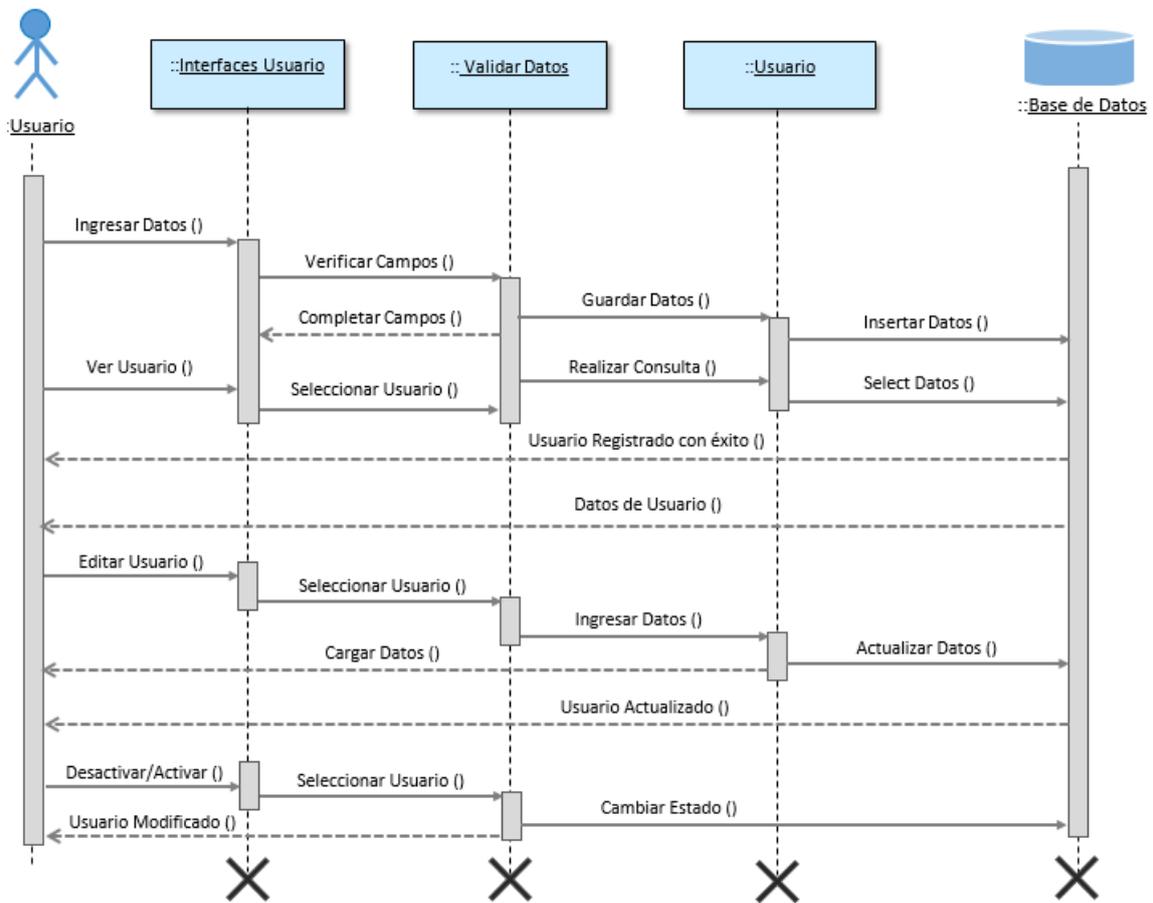


Figura 4.4.19: Diagrama de Secuencia: proceso de usuario

### 4.4.3.4. Interfaz de Usuario

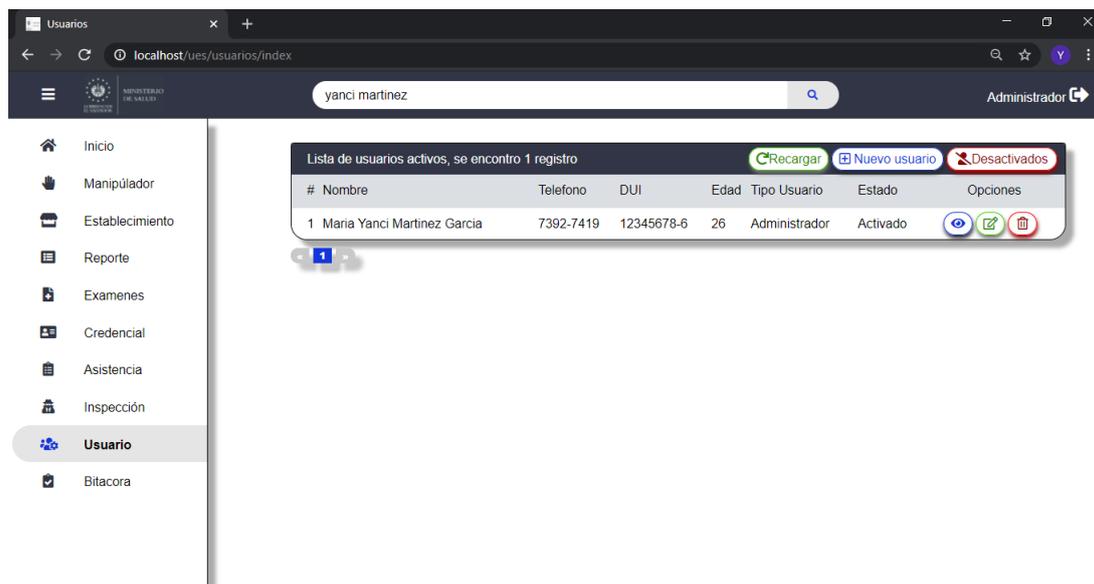


Figura 4.4.20: Interfaz de Usuario búsqueda

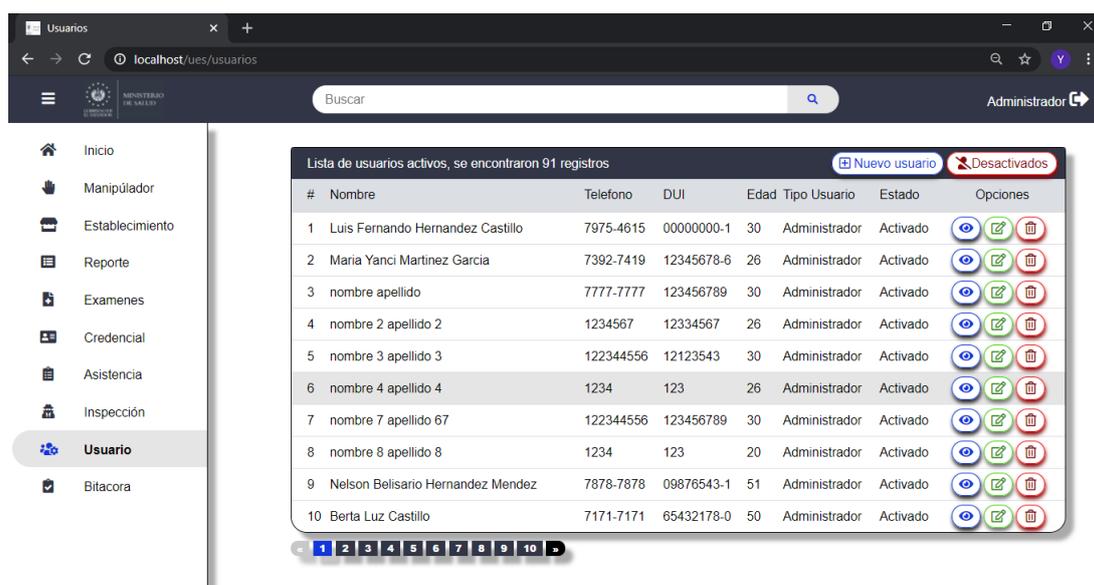


Figura 4.4.21: Interfaz de Usuario index

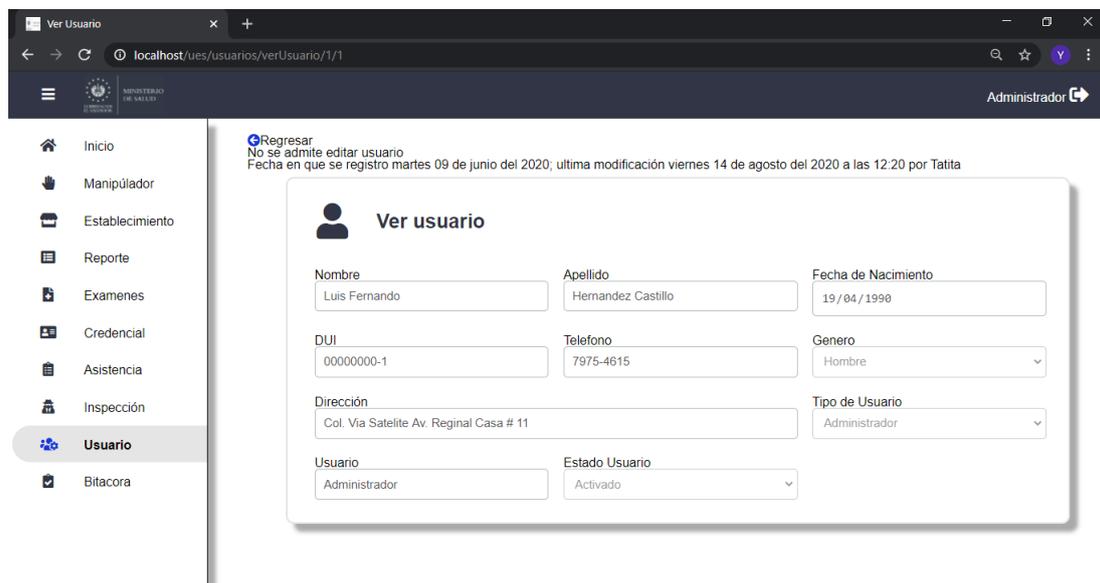


Figura 4.4.22: Interfaz de ver Usuario

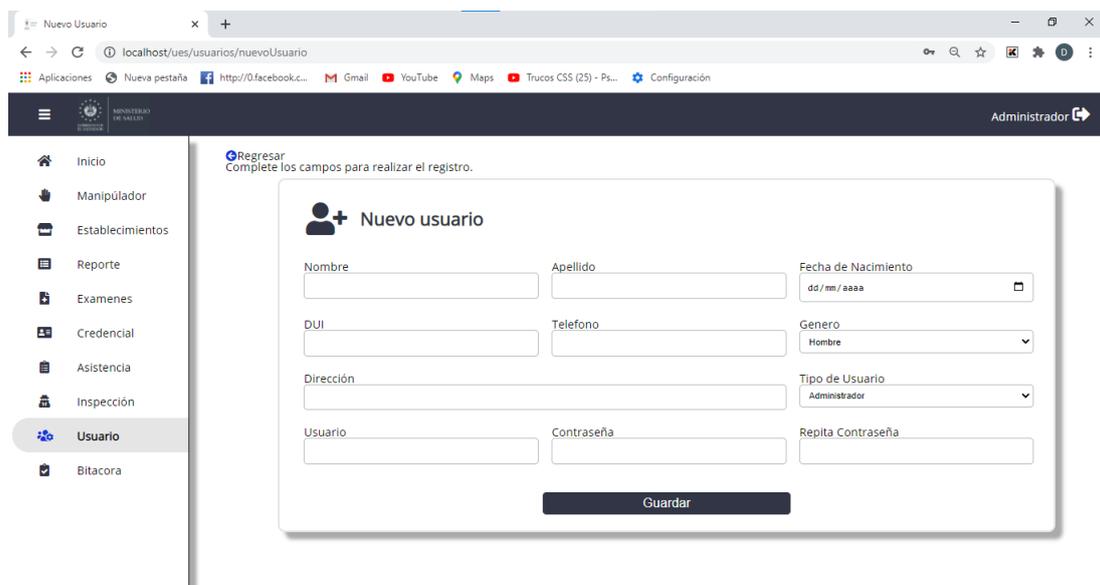


Figura 4.4.23: Interfaz de Registrar Usuario

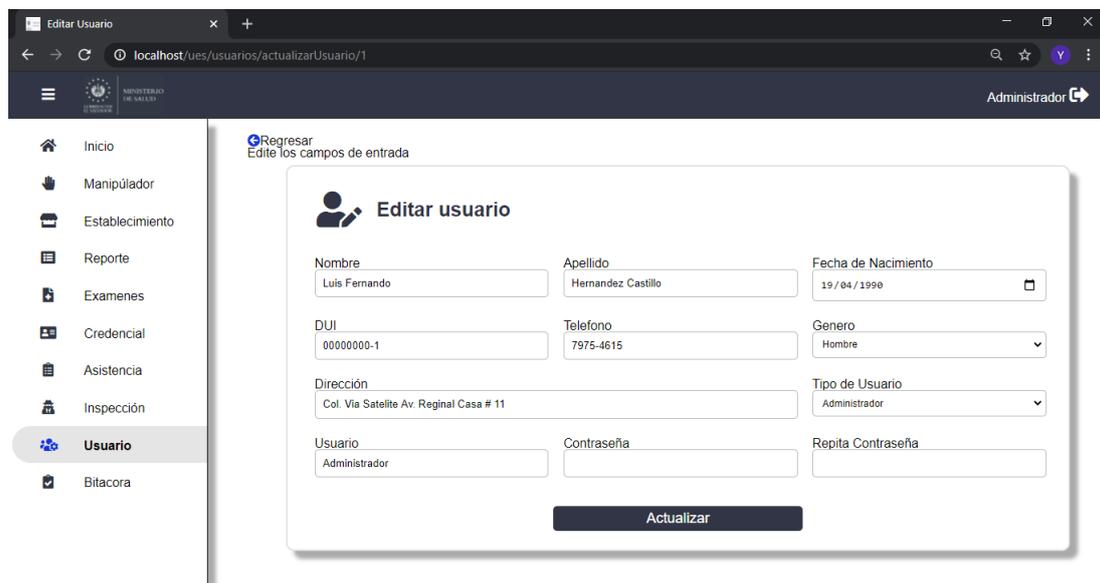


Figura 4.4.24: Interfaz de Editar Usuario

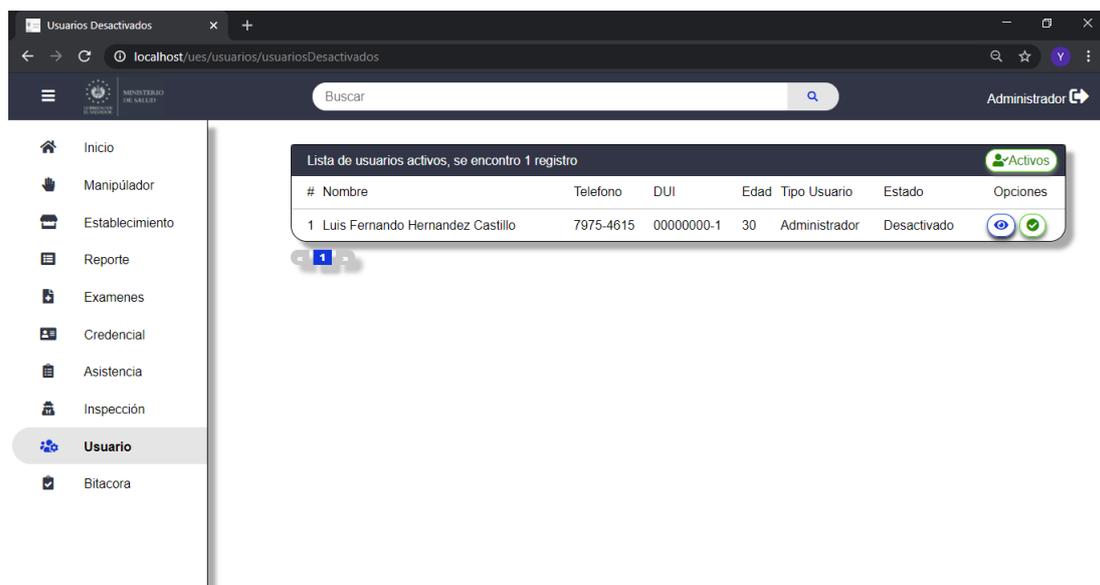


Figura 4.4.25: Interfaz de Desactivar Usuario

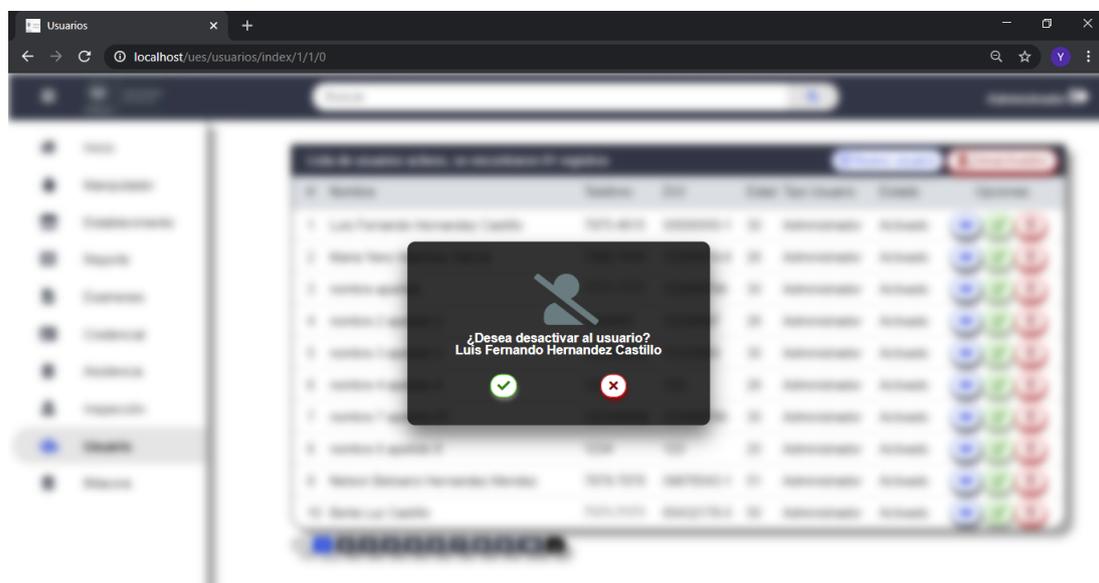


Figura 4.4.26: Interfaz de confirmación para Desactivar Usuario

### 4.4.4. Diagrama de Secuencia Reporte

Otro de los procesos principales que realiza el sistema es la generación de reportes, éstos son los que nos brindan información sobre manipuladores, usuarios y establecimientos. Quedando representado gráficamente en el diagrama de secuencia de la figura 4.4.75.

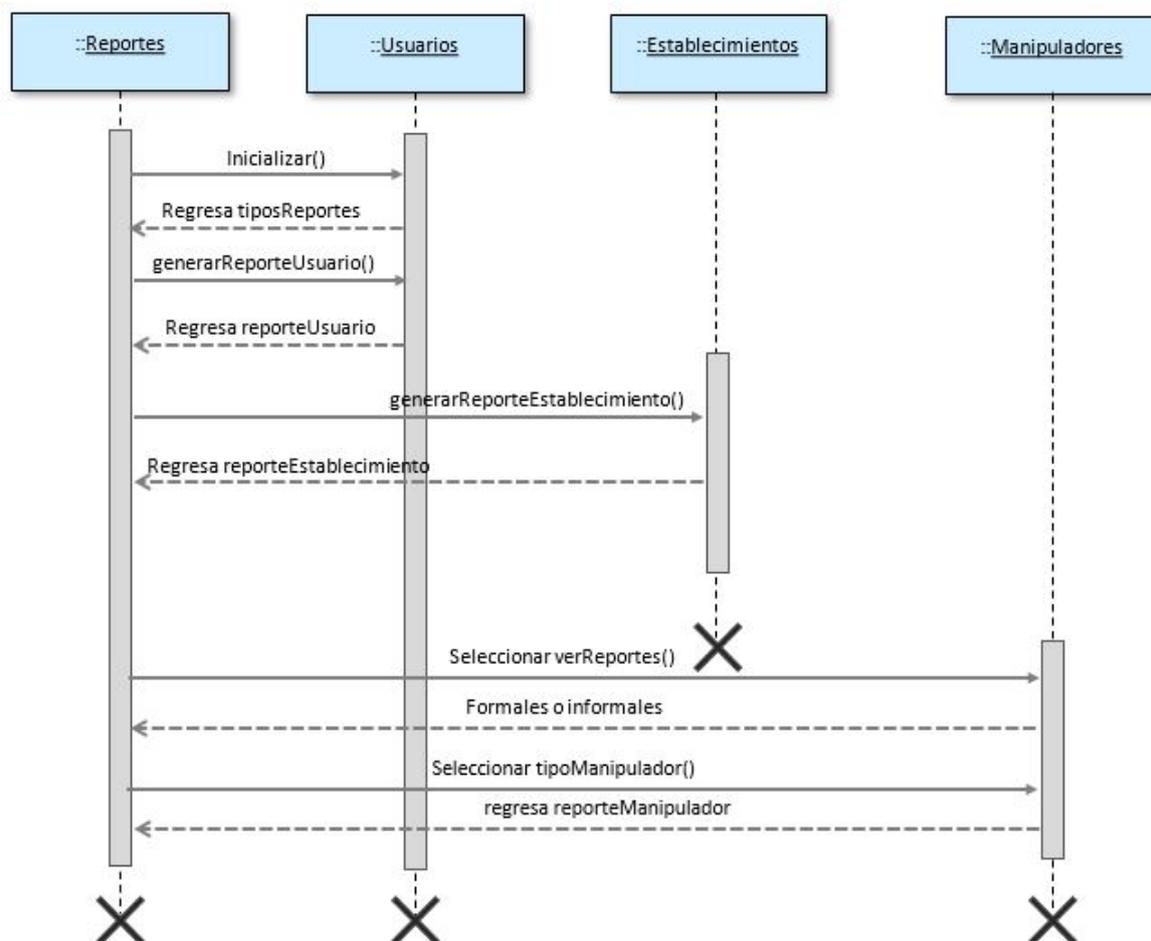


Figura 4.4.27: Diagrama de Secuencia: proceso de reportes

### 4.4.4.1. Interfaz de Reportes

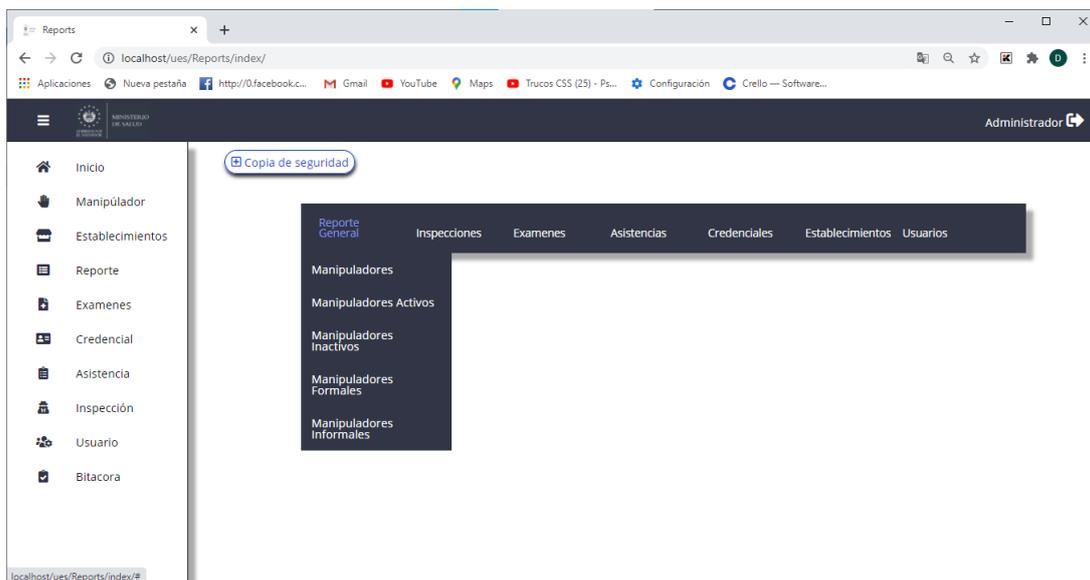


Figura 4.4.28: Interfaz de Reportes

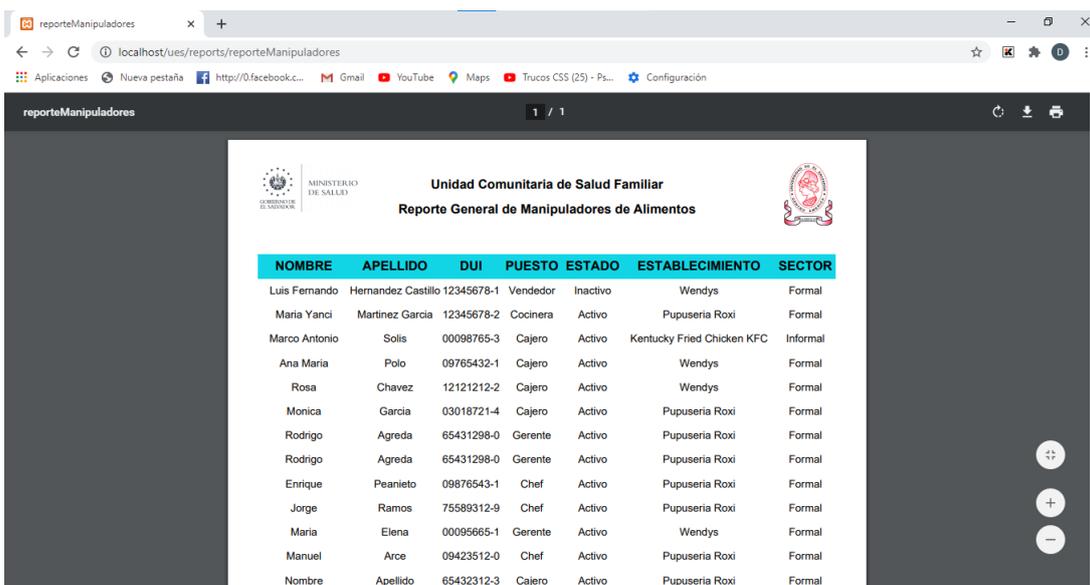


Figura 4.4.29: Interfaz de Reporte generado

4.4.4.2. Diagrama de secuencia proceso de asistencia

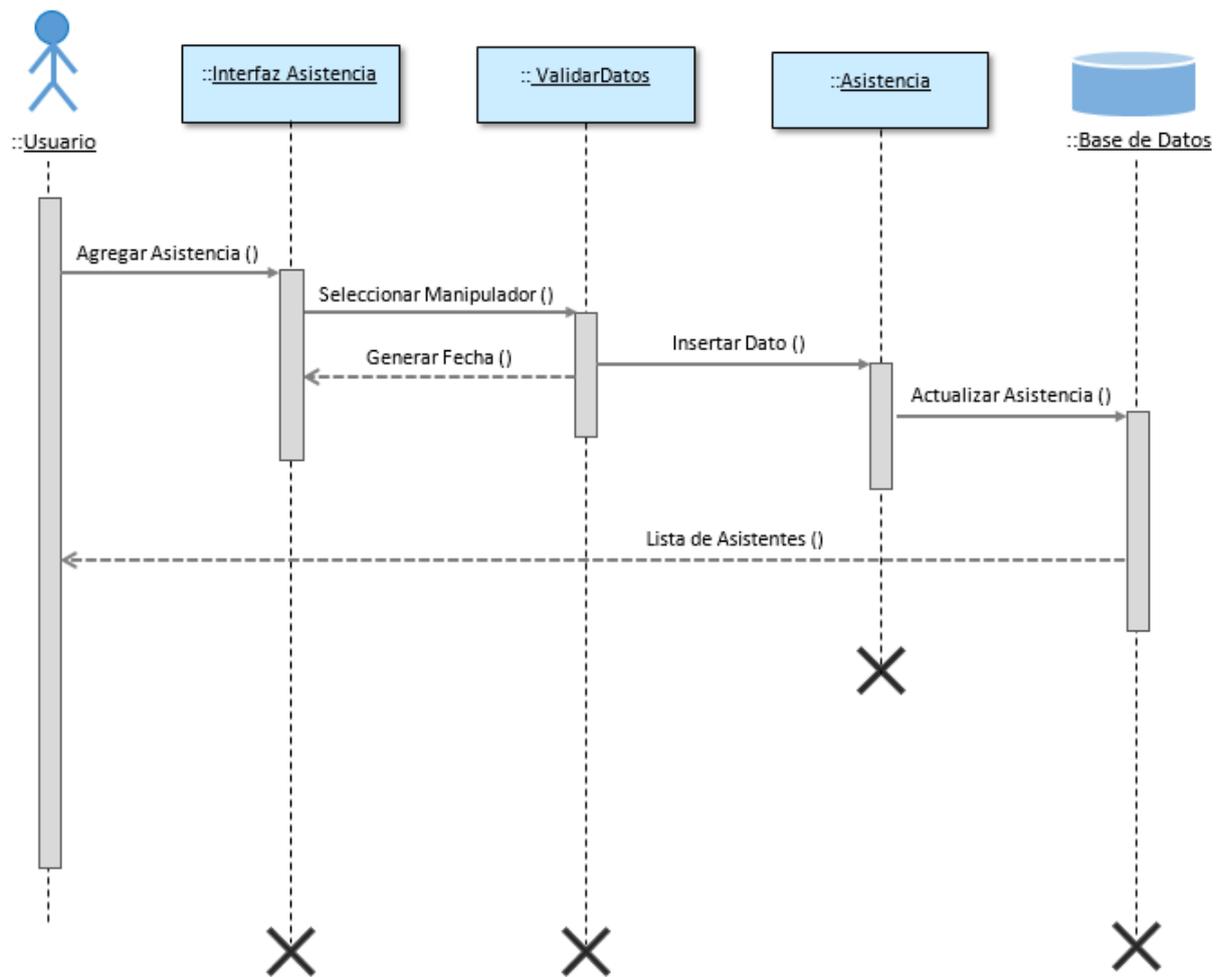


Figura 4.4.30: Diagrama de Secuencia: proceso de asistencia

### 4.4.4.3. Interfaz de Asistencia

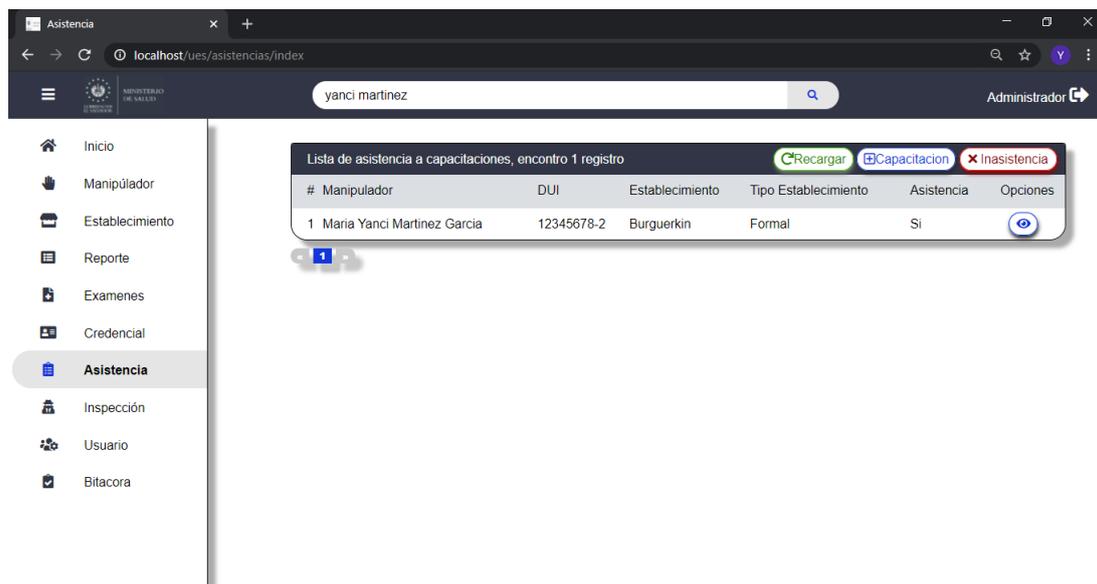


Figura 4.4.31: Interfaz de asistencia búsqueda

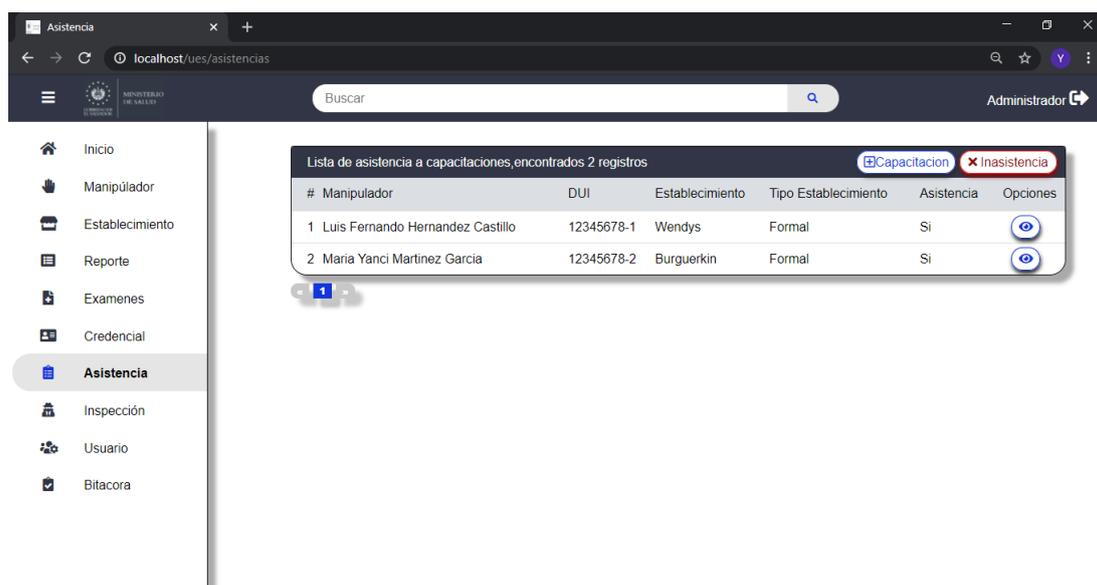


Figura 4.4.32: Interfaz de asistencia

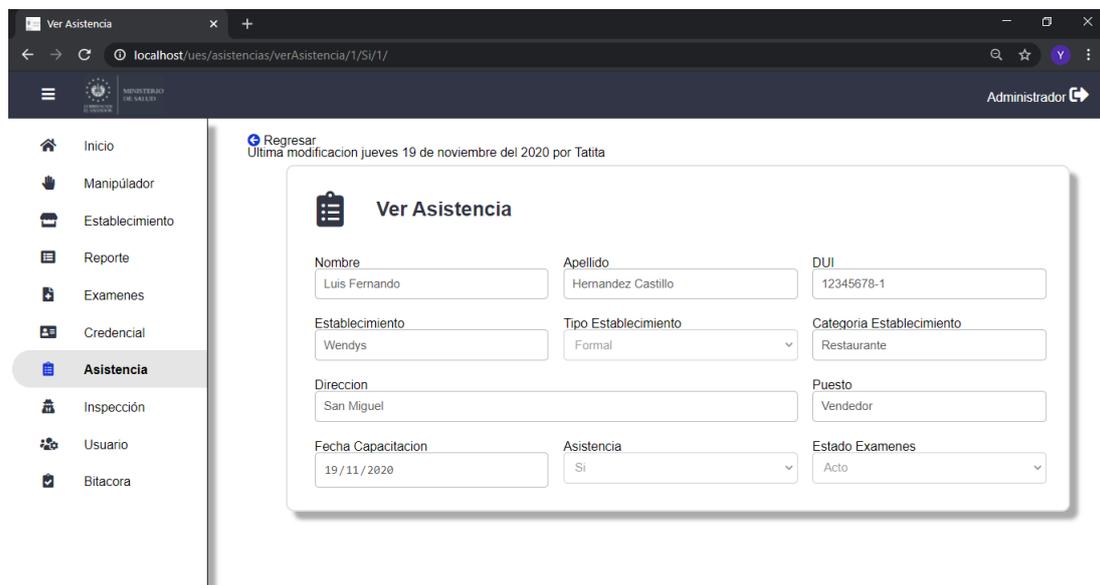


Figura 4.4.33: Interfaz de Ver asistencia

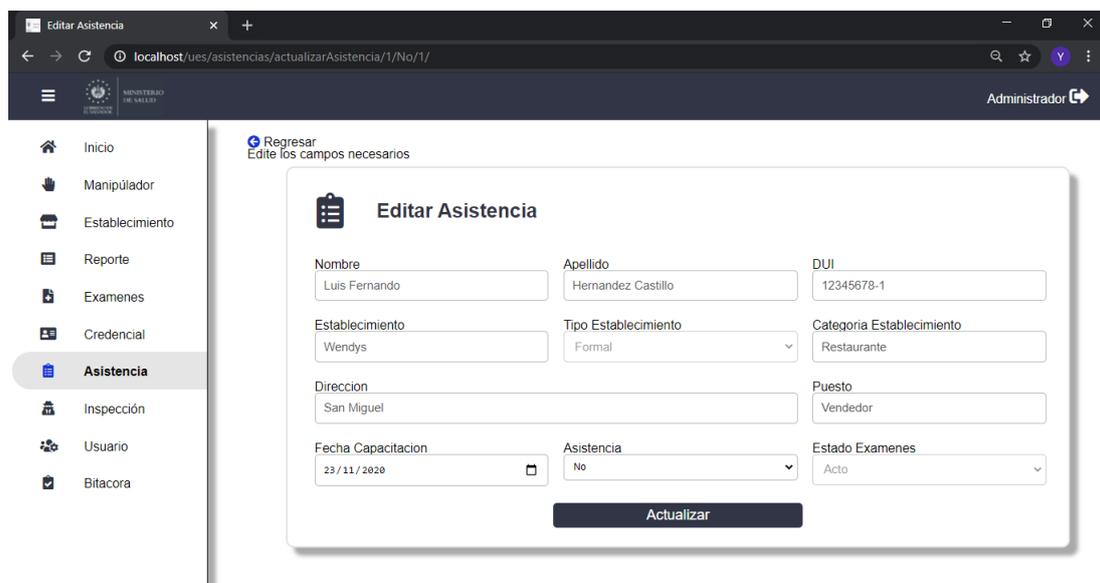


Figura 4.4.34: Interfaz de Editar asistencia

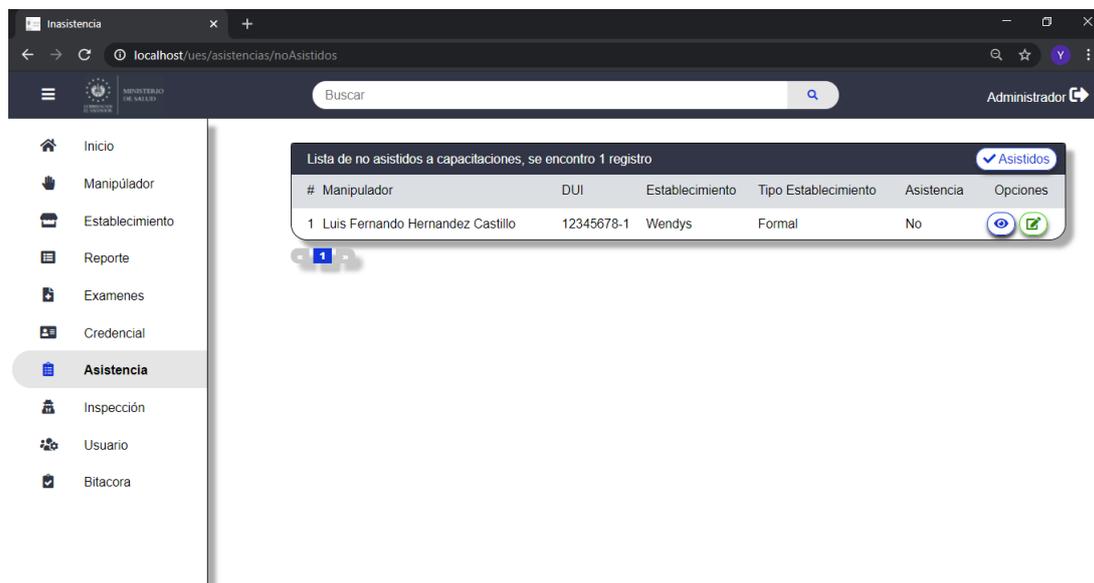


Figura 4.4.35: Interfaz de Inasistencia

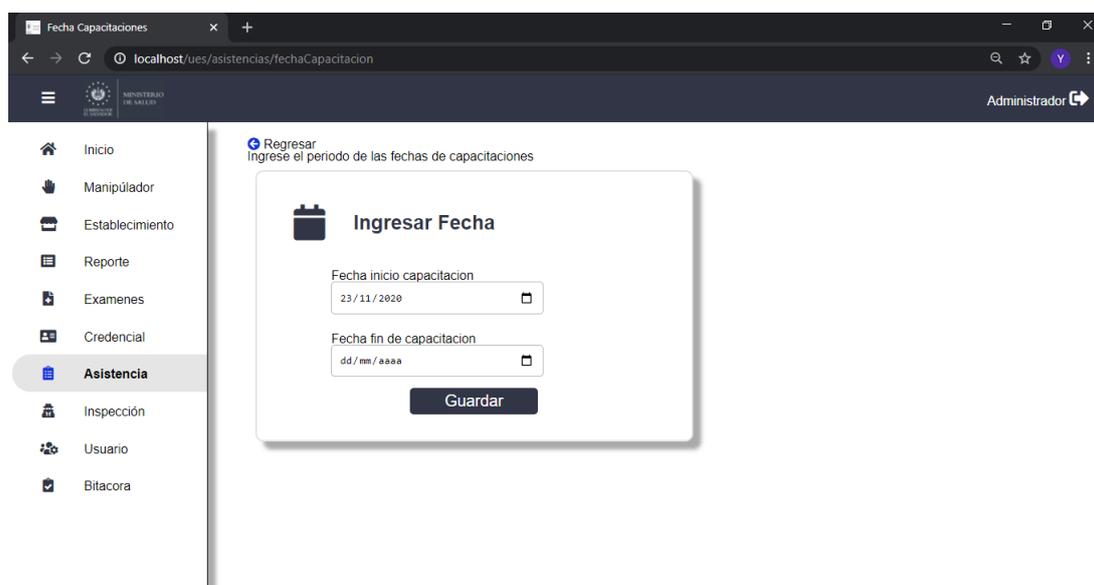


Figura 4.4.36: Interfaz de Fecha capacitaciones

4.4.4.4. Diagrama de secuencia proceso de inspección

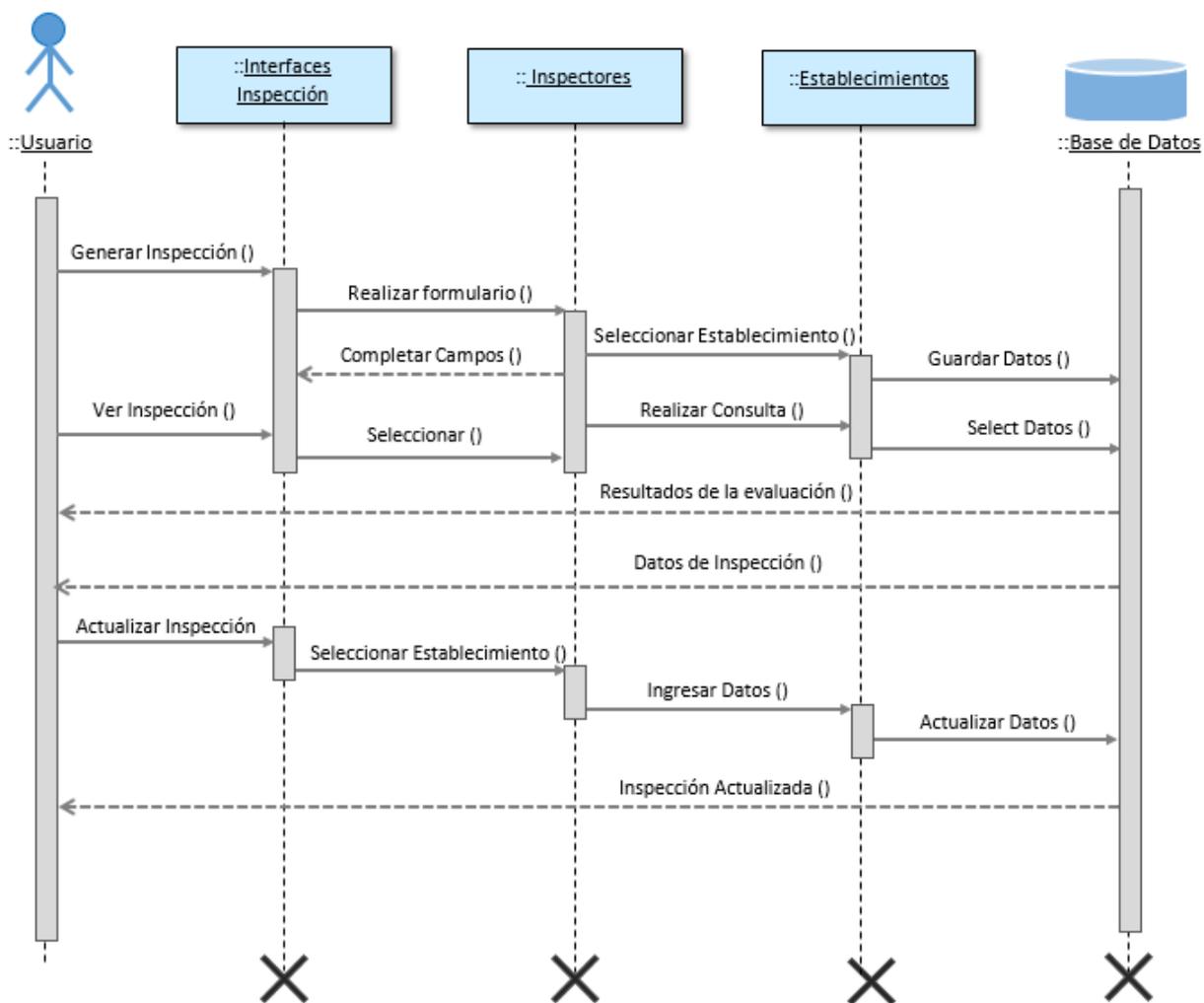


Figura 4.4.37: Diagrama de Secuencia: proceso de inspección

### 4.4.4.5. Interfaz de Inspección

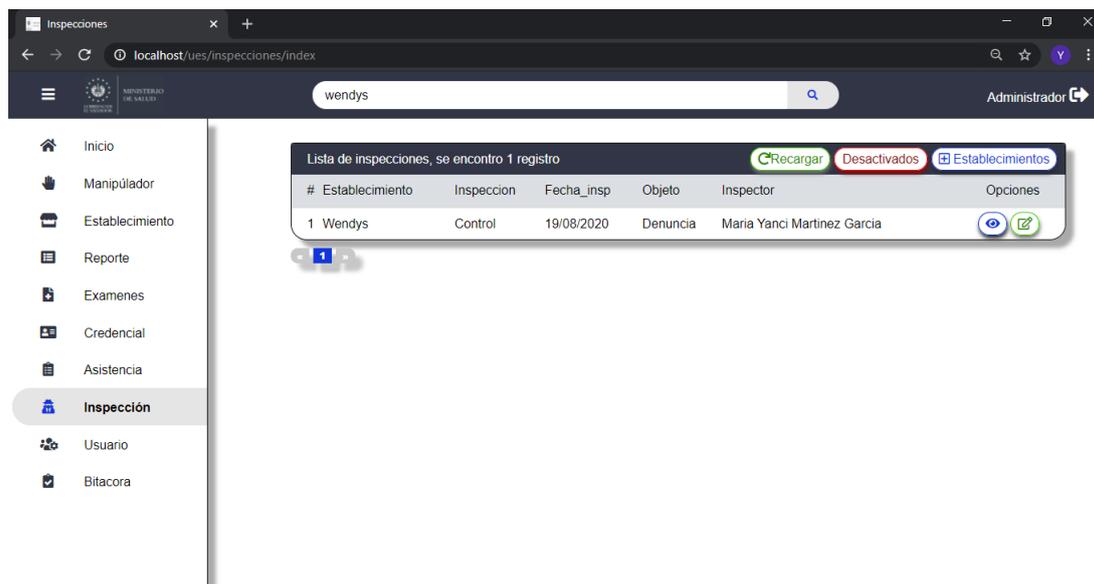


Figura 4.4.38: Interfaz de inspección búsqueda

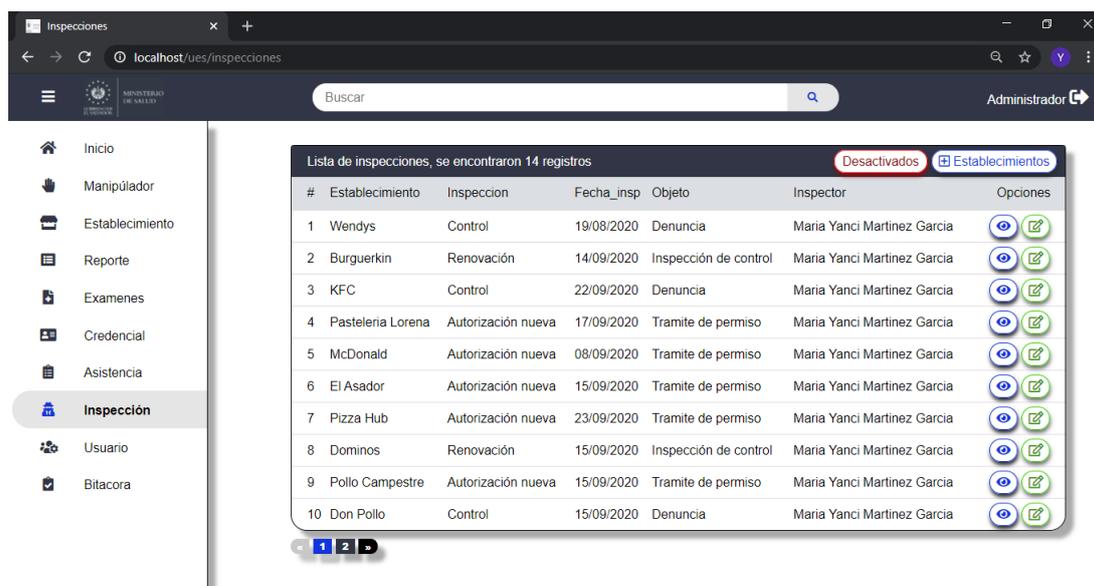


Figura 4.4.39: Interfaz de inspección lista

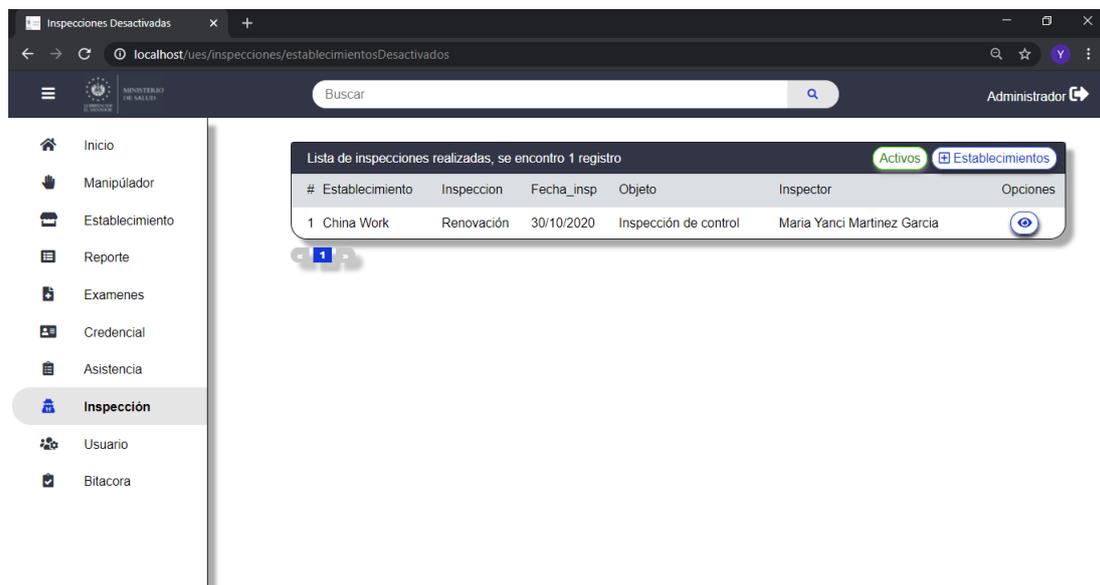


Figura 4.4.40: Interfaz de inspección desactivadas

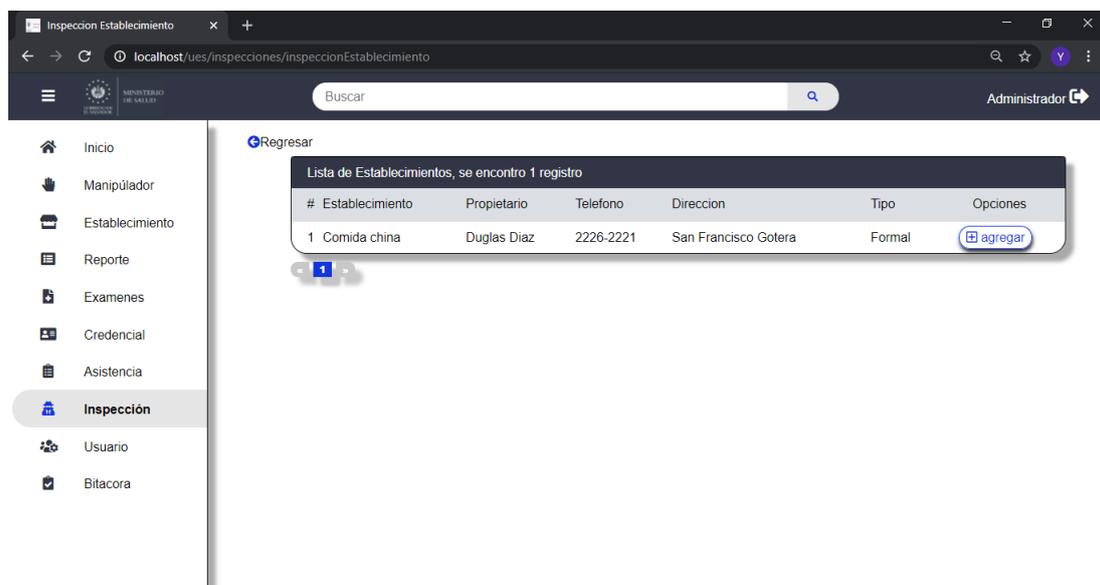


Figura 4.4.41: Interfaz de Establecimientos

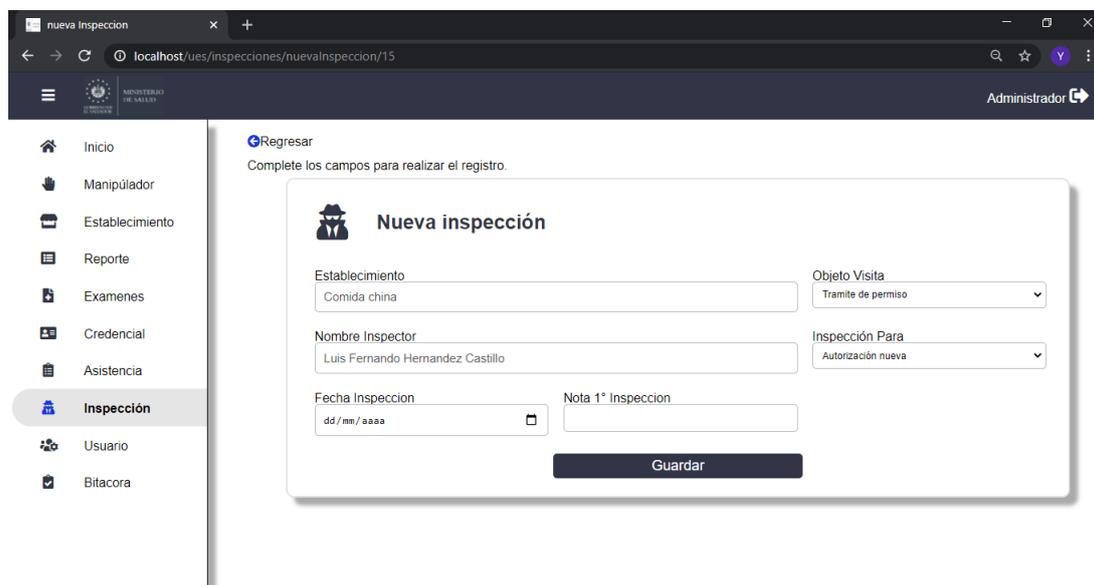


Figura 4.4.42: Interfaz de Agregar nueva inspección

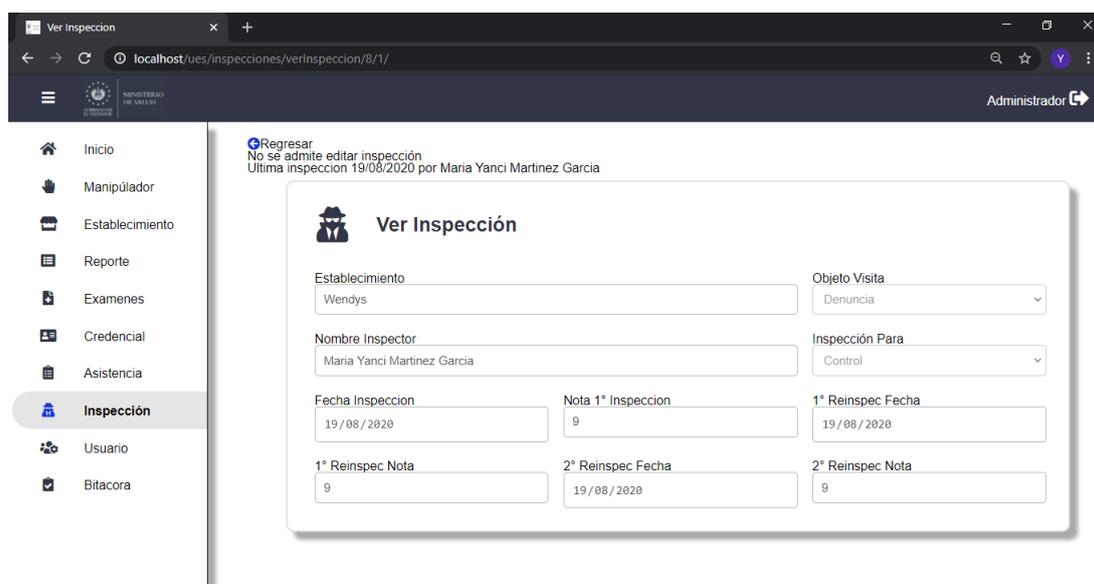


Figura 4.4.43: Interfaz de ver inspección

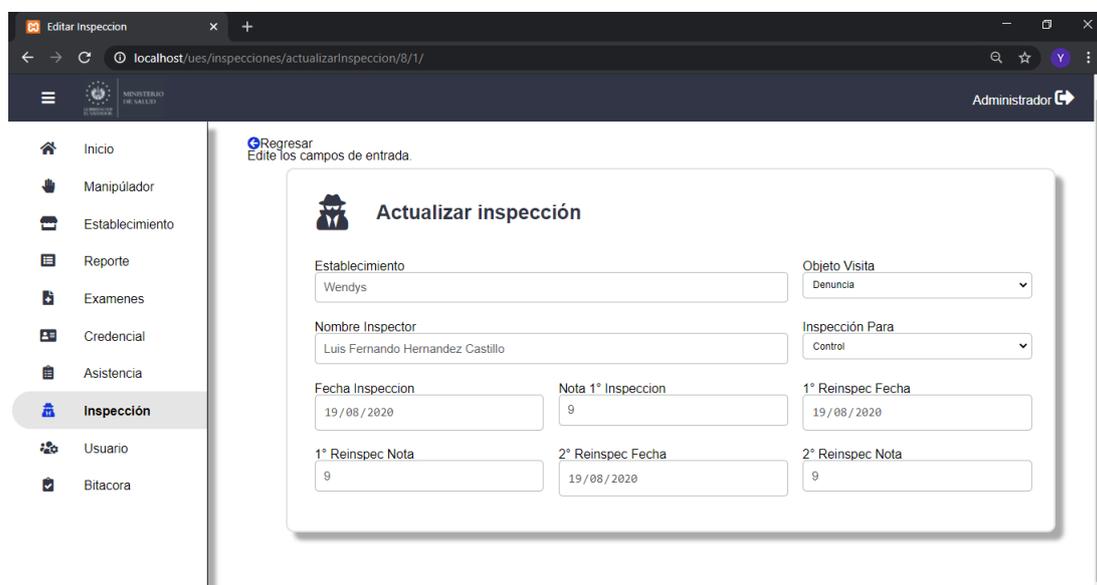


Figura 4.4.44: Interfaz de actualizar

4.4.4.6. Diagrama de secuencia proceso de Bitácora

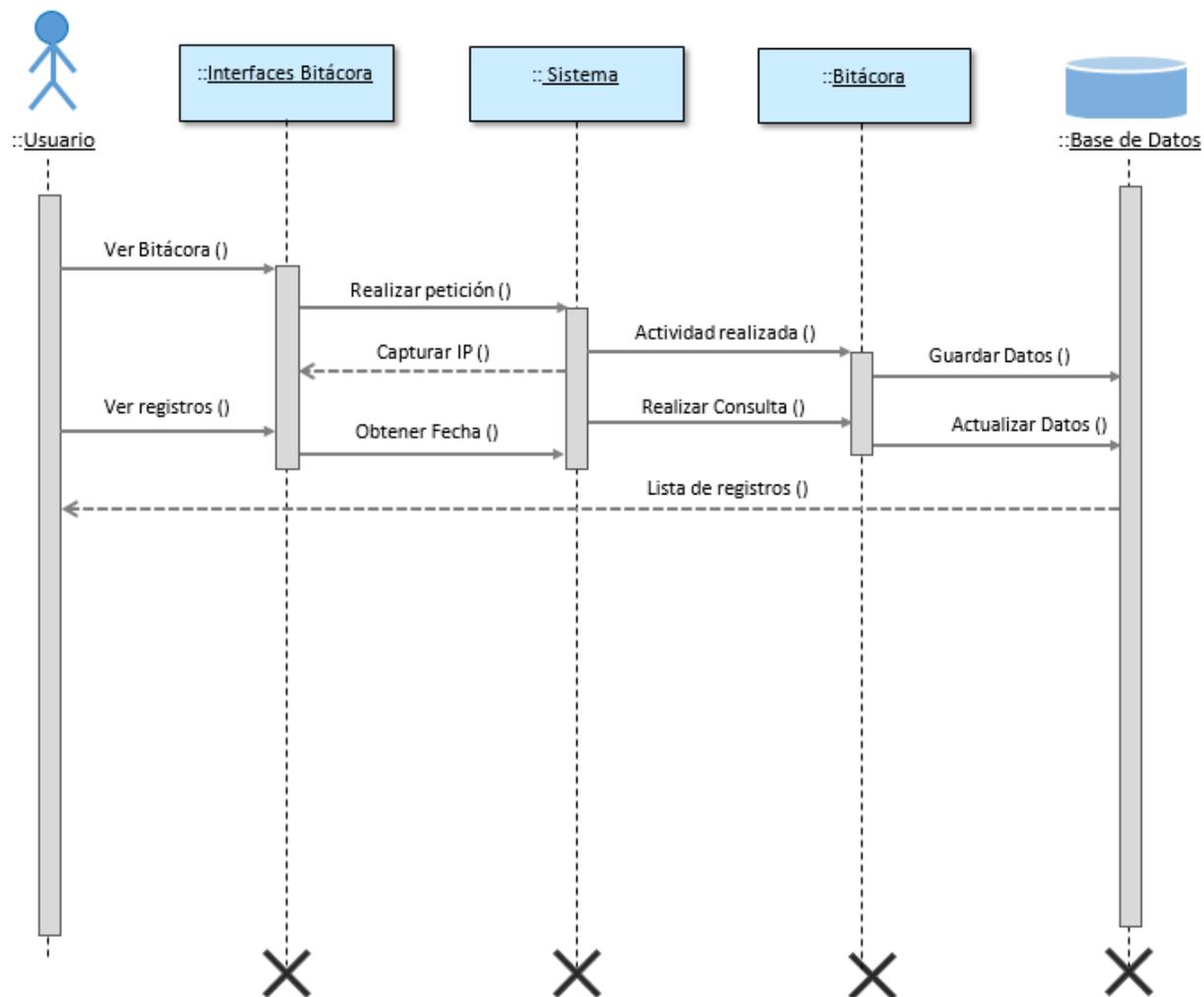


Figura 4.4.45: Diagrama de Secuencia: proceso de bitácora

### 4.4.4.7. Interfaz de bitácora

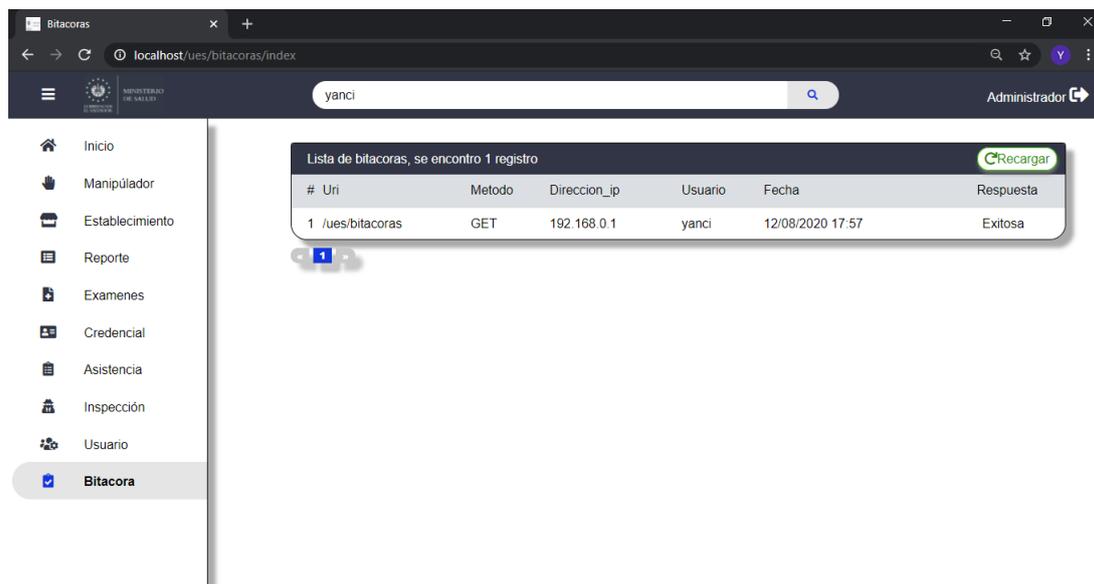


Figura 4.4.46: Interfaz de buscar bitácora

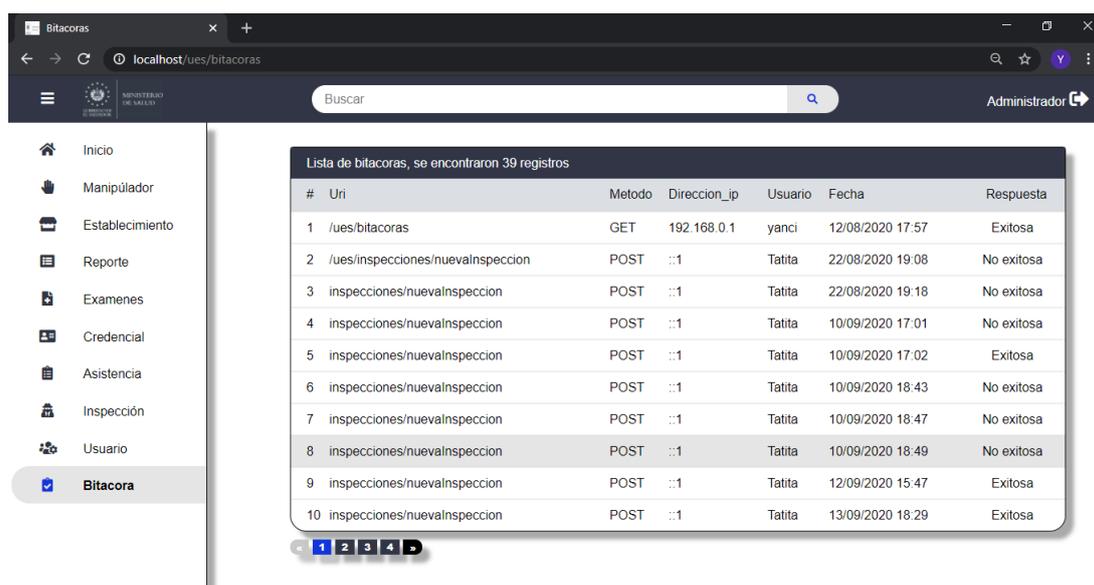


Figura 4.4.47: Interfaz ver registro de bitácora

4.4.4.8. Diagrama de secuencia proceso de Credenciales

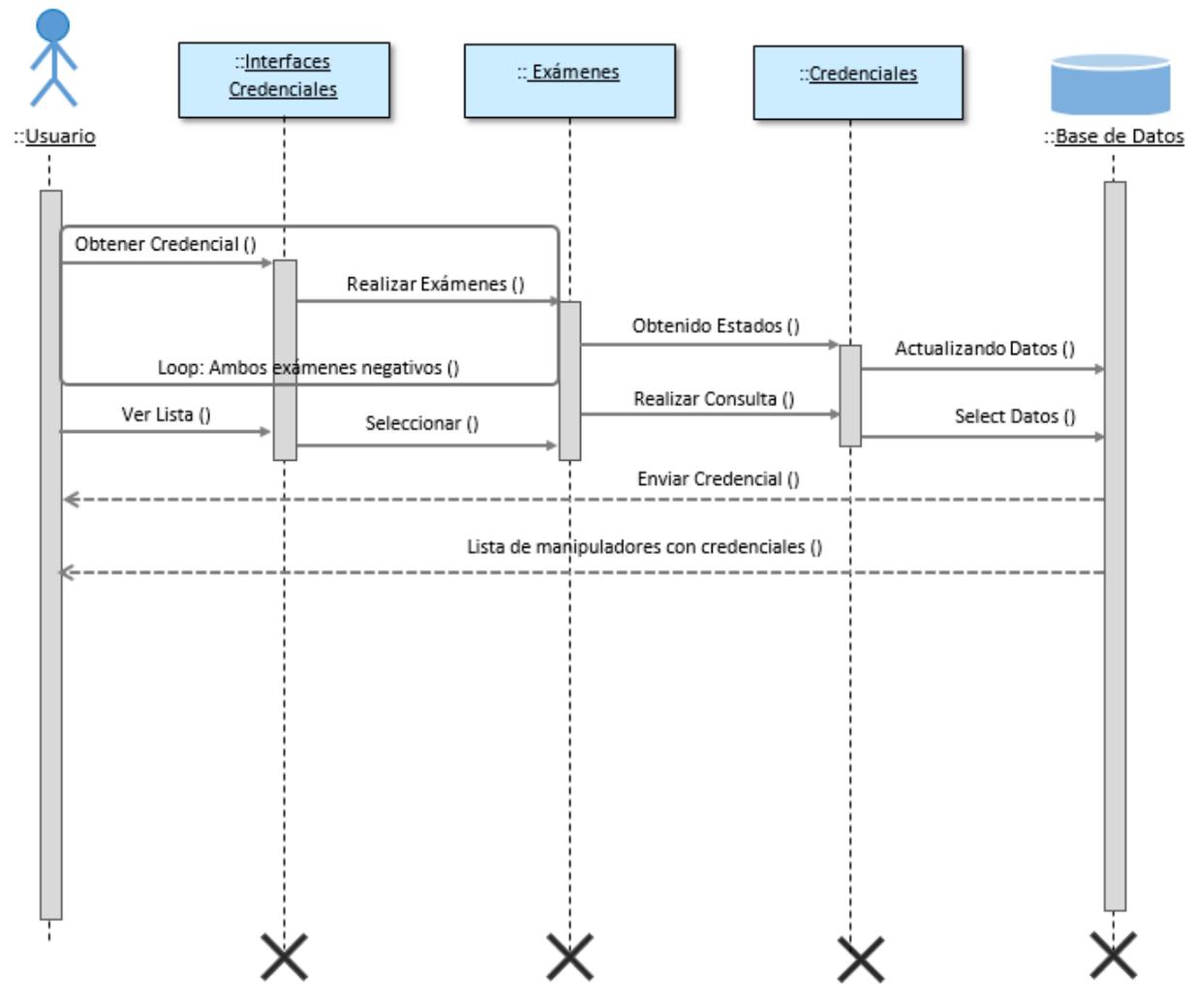


Figura 4.4.48: Diagrama de Secuencia: proceso de credenciales

### 4.4.4.9. Interfaz de Credenciales

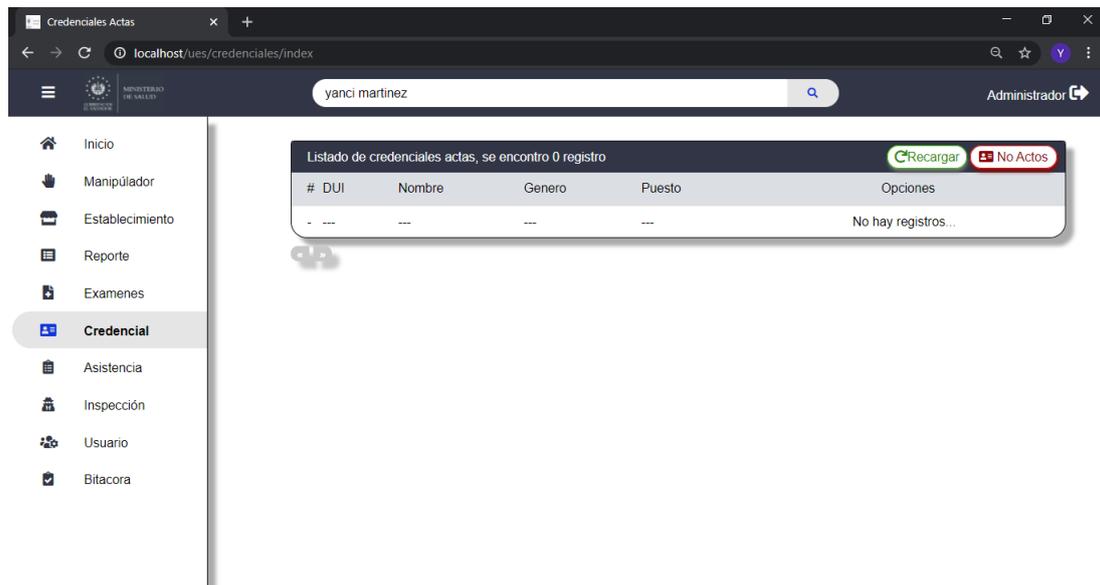


Figura 4.4.49: Interfaz de Búsqueda credenciales

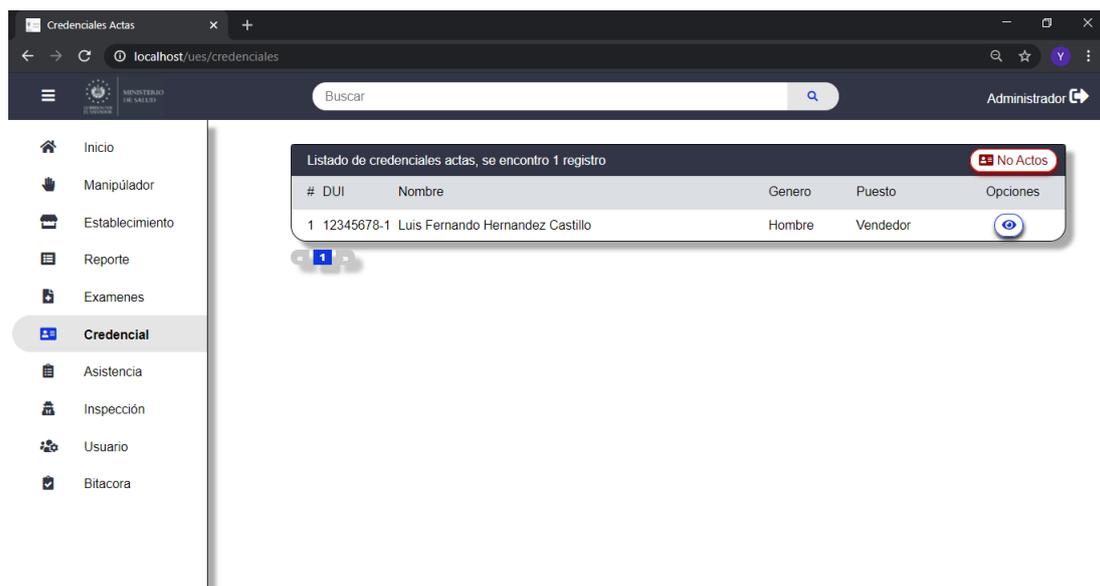


Figura 4.4.50: Interfaz Credenciales aptas

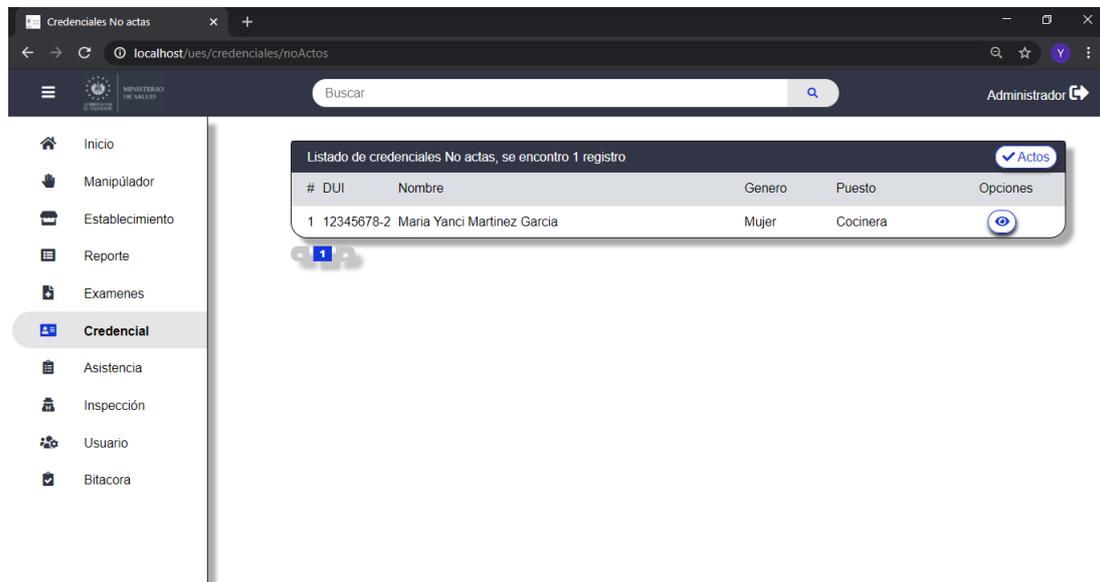


Figura 4.4.51: Interfaz Credenciales no aptas

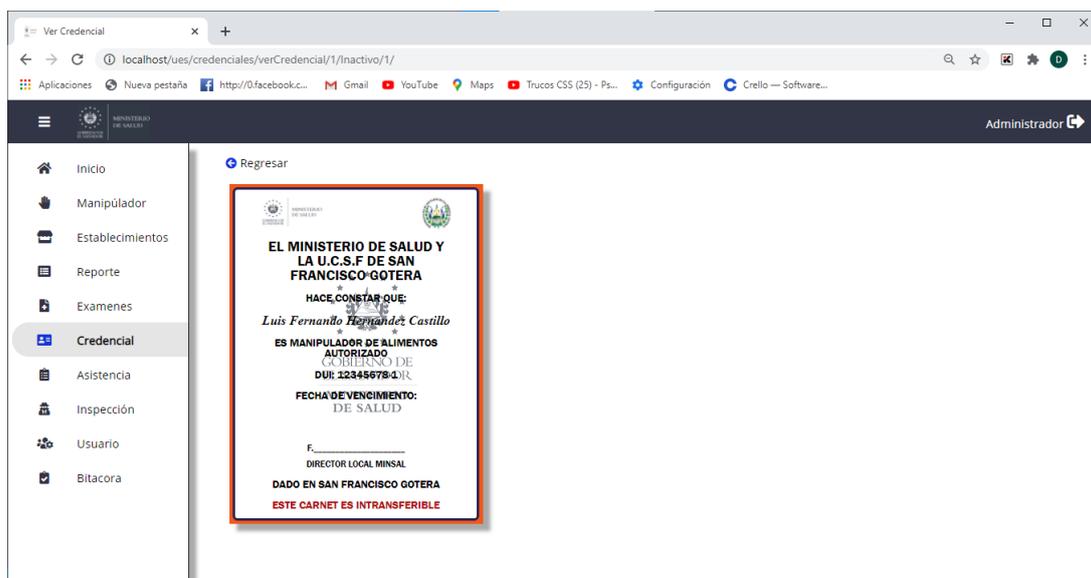


Figura 4.4.52: Interfaz Ver Credenciales

4.4.4.10. Diagrama de secuencia proceso de Manipuladores

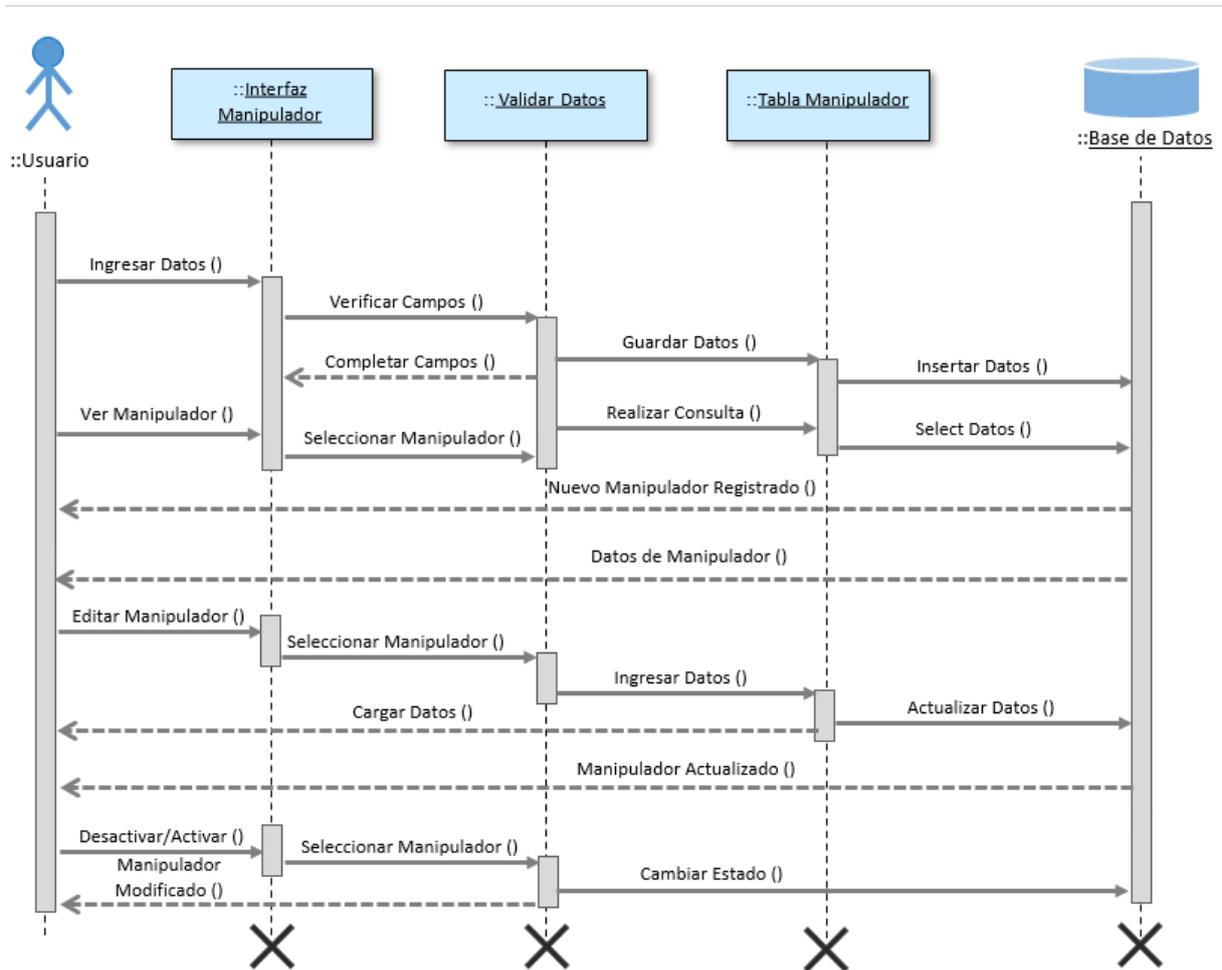


Figura 4.4.53: Diagrama de Secuencia: proceso de Manipulador

### 4.4.4.11. Interfaz de Manipuladores

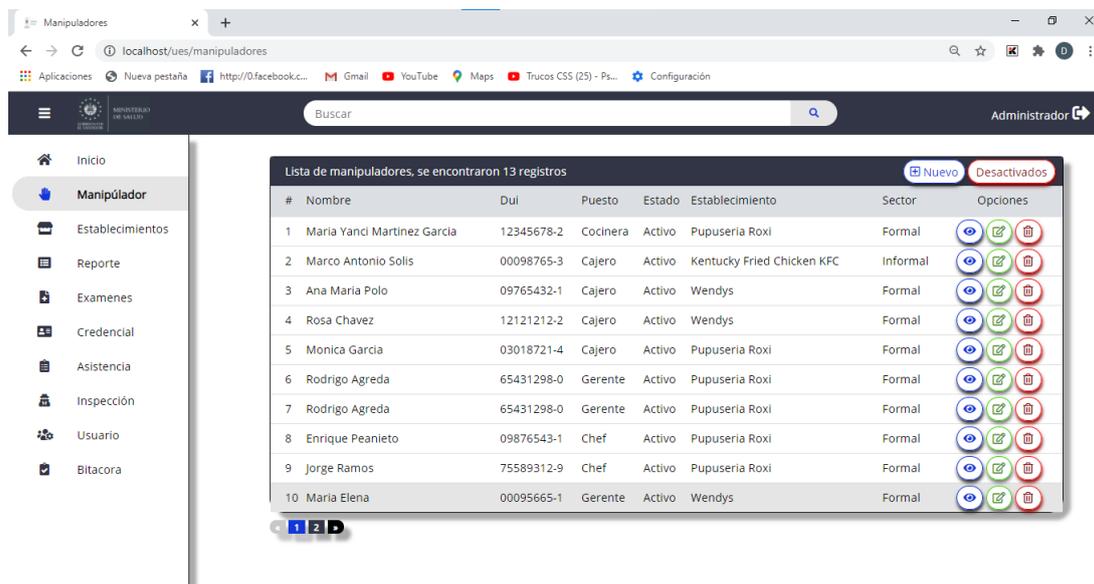


Figura 4.4.54: Interfaz de Manipuladores Activos

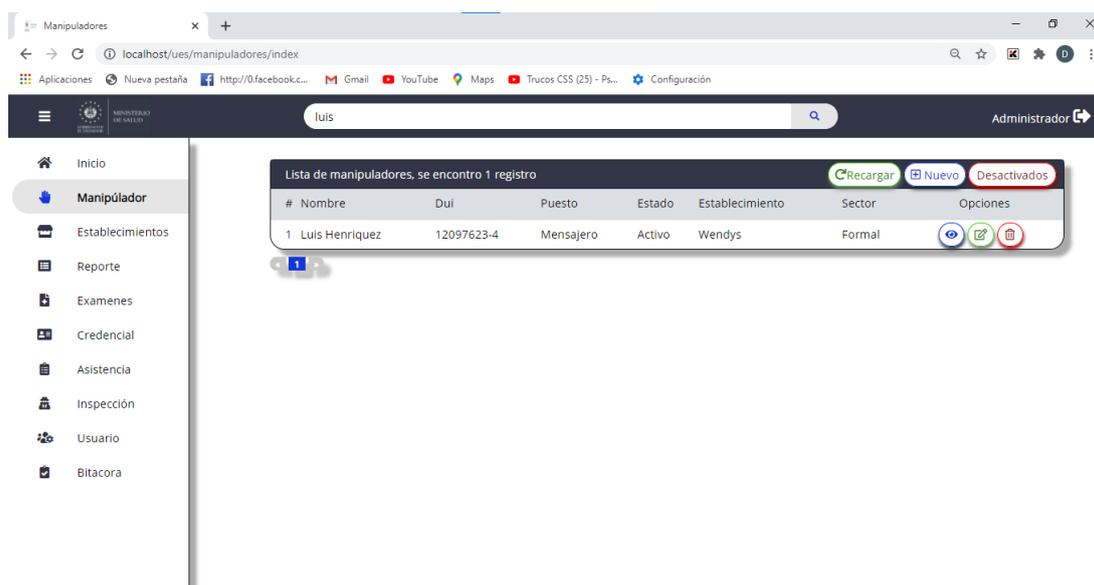


Figura 4.4.55: Interfaz de Búsqueda Manipuladores

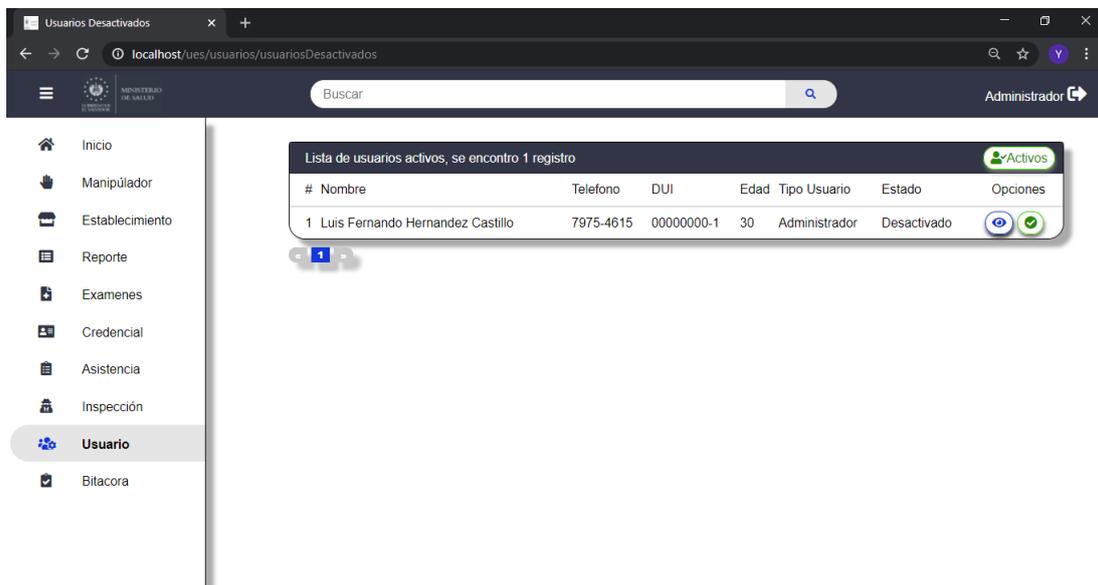


Figura 4.4.56: Interfaz de Lista Manipuladores Desactivados

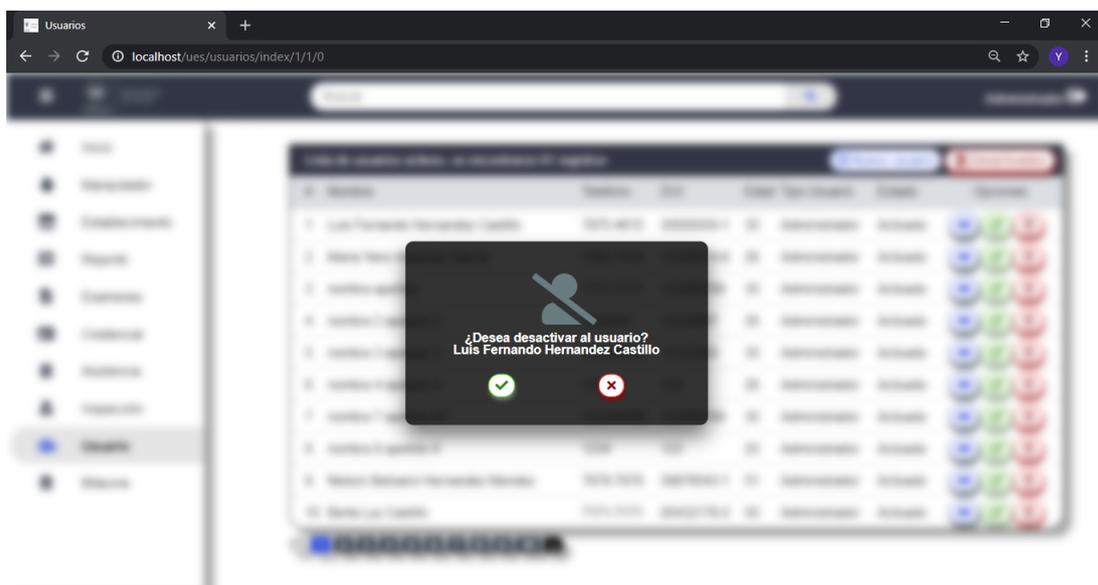


Figura 4.4.57: Interfaz de Manipuladores Desactivados

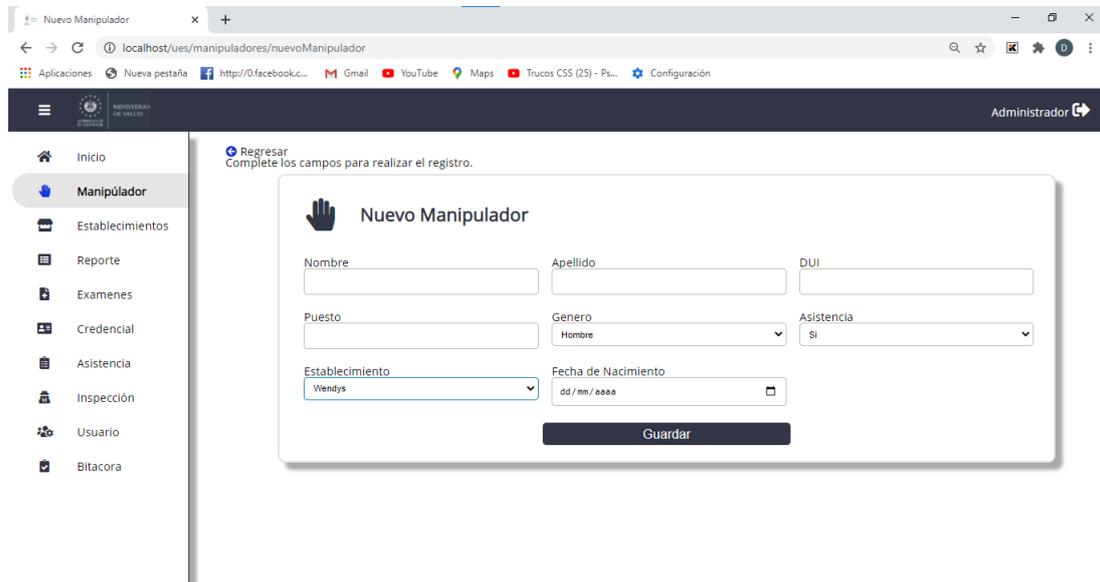


Figura 4.4.58: Interfaz de Nuevo Manipuladores

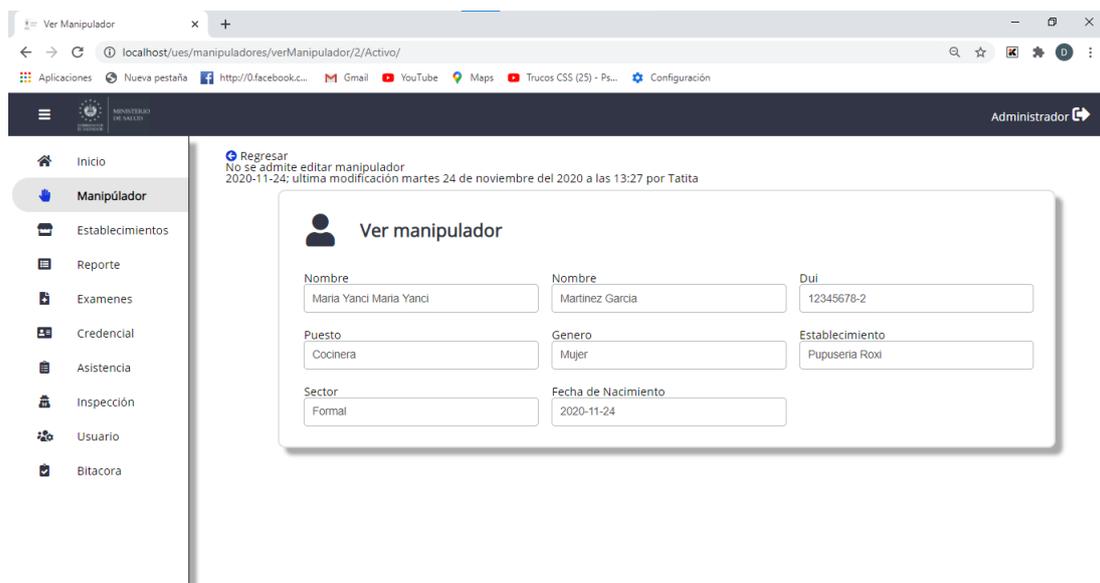


Figura 4.4.59: Interfaz de Ver Manipuladores

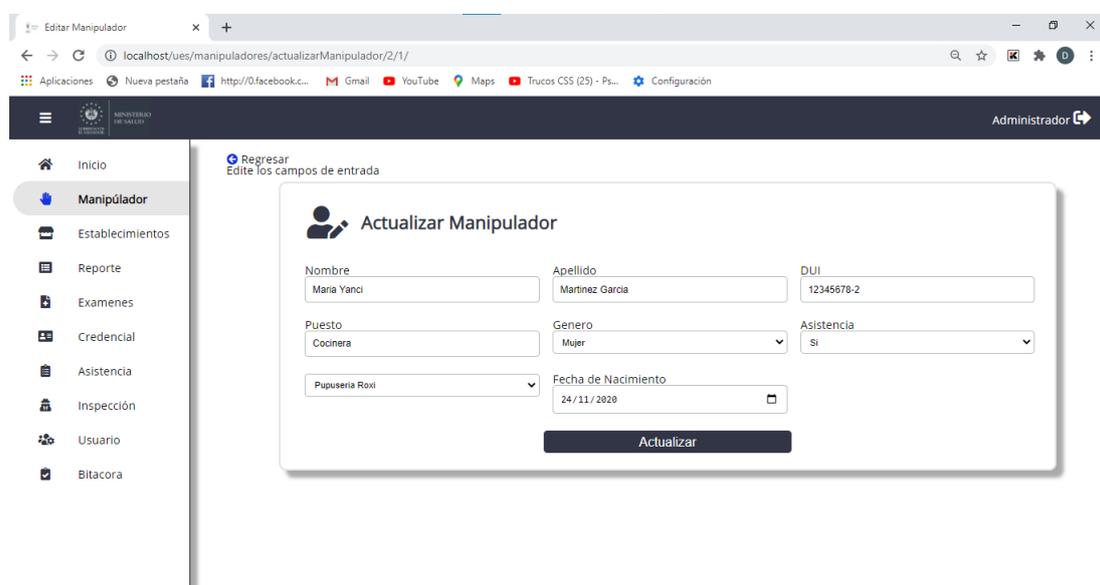


Figura 4.4.60: Interfaz de Actualizar Manipuladores

4.4.4.12. Diagrama de secuencia proceso de Establecimientos

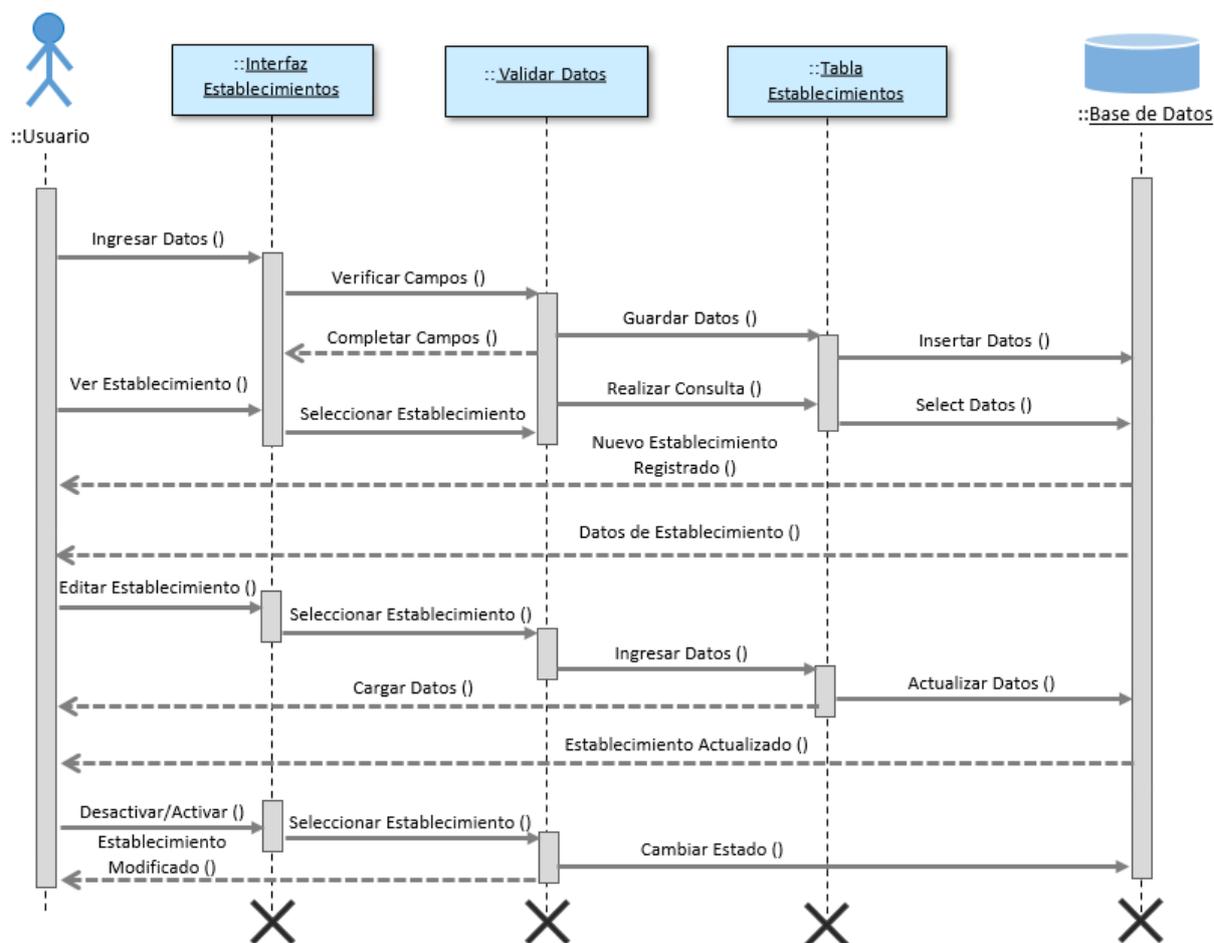


Figura 4.4.61: Diagrama de Secuencia: proceso de Establecimiento

### 4.4.4.13. Interfaz de Establecimientos

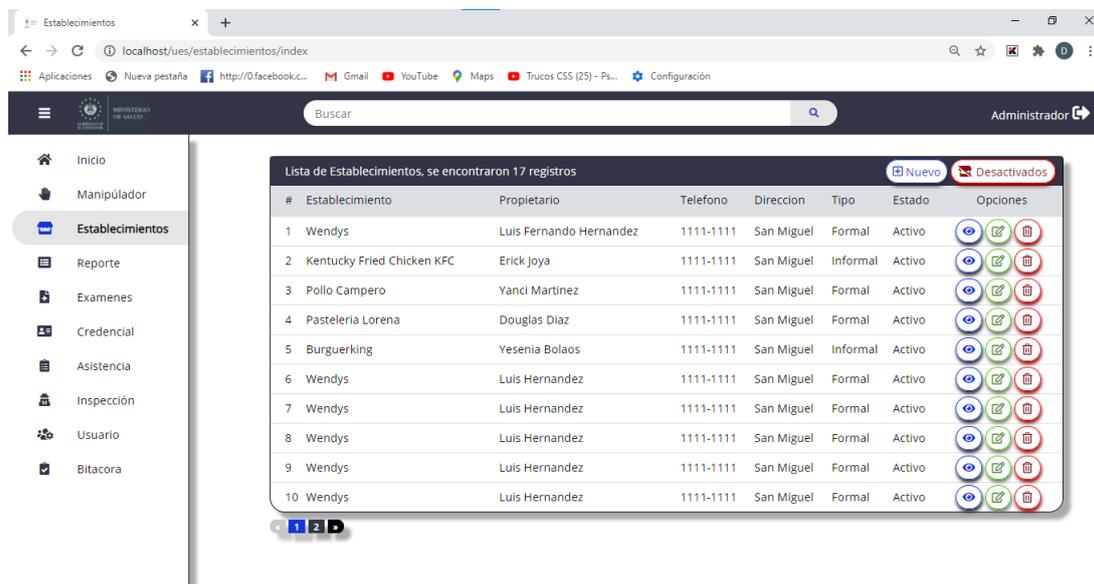


Figura 4.4.62: Interfaz de Establecimientos

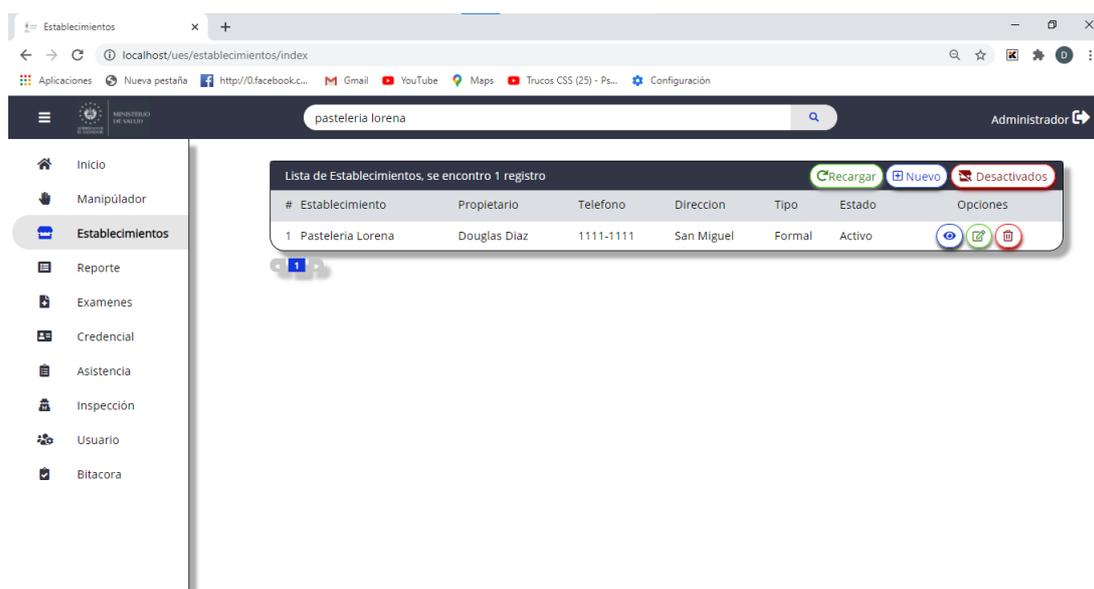


Figura 4.4.63: Interfaz de Búsqueda Establecimientos

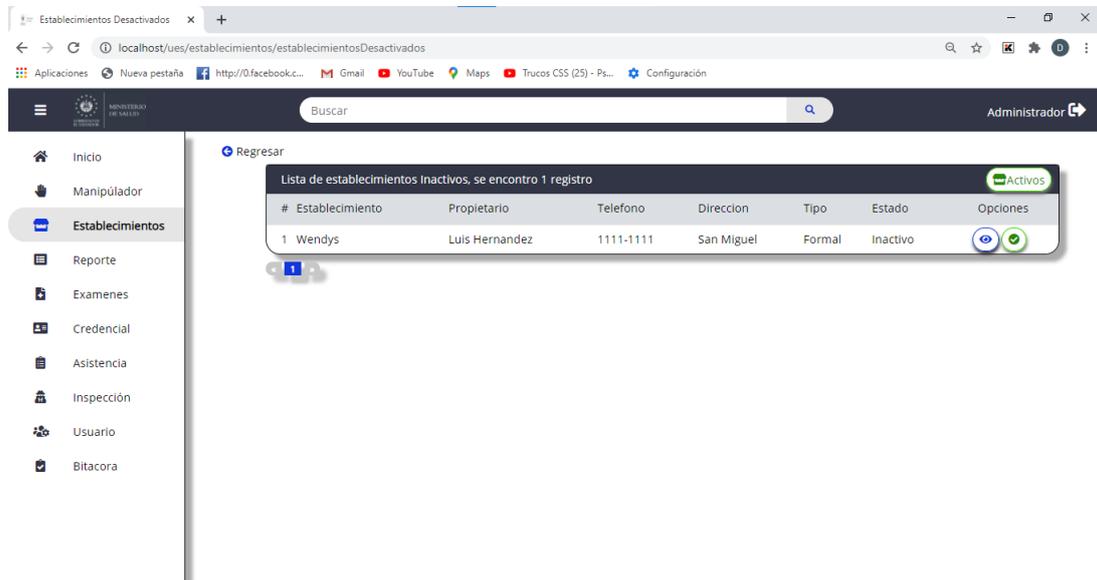


Figura 4.4.64: Interfaz de Lista Establecimientos Desactivados

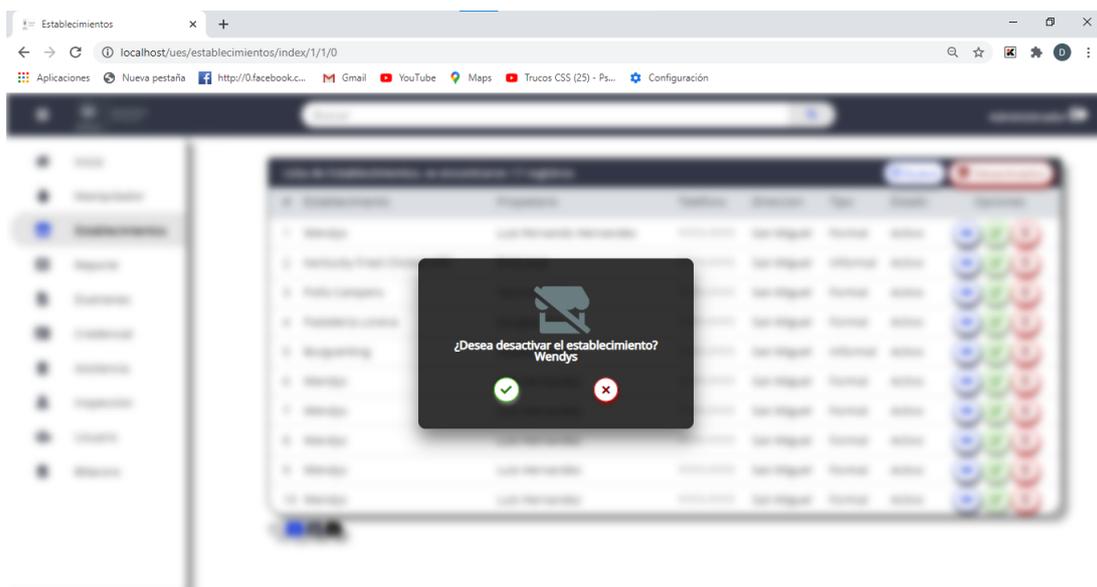


Figura 4.4.65: Interfaz de Establecimientos Desactivados

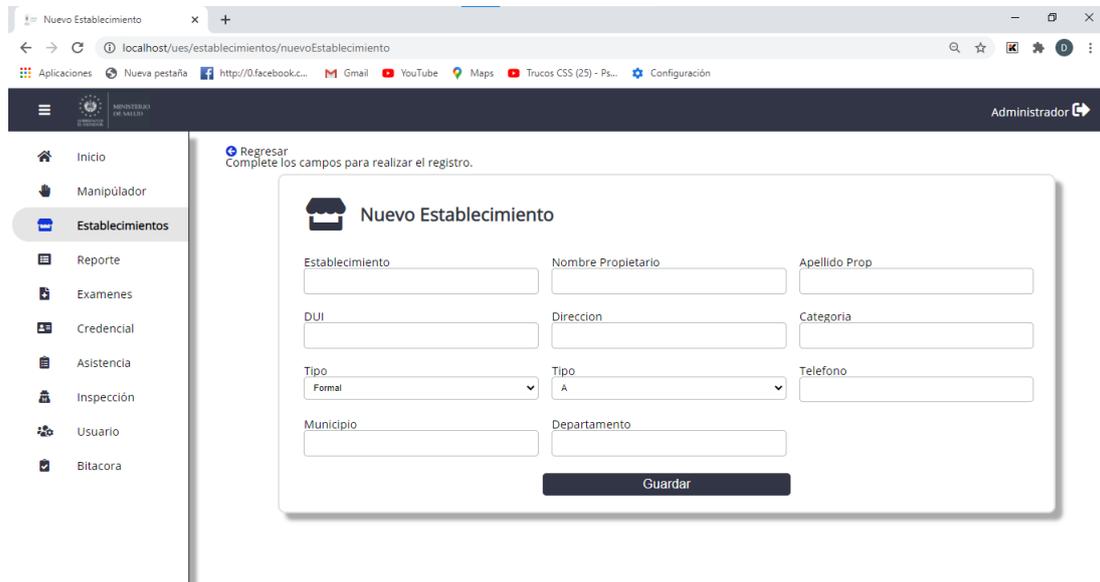


Figura 4.4.66: Interfaz de Nuevo Establecimiento

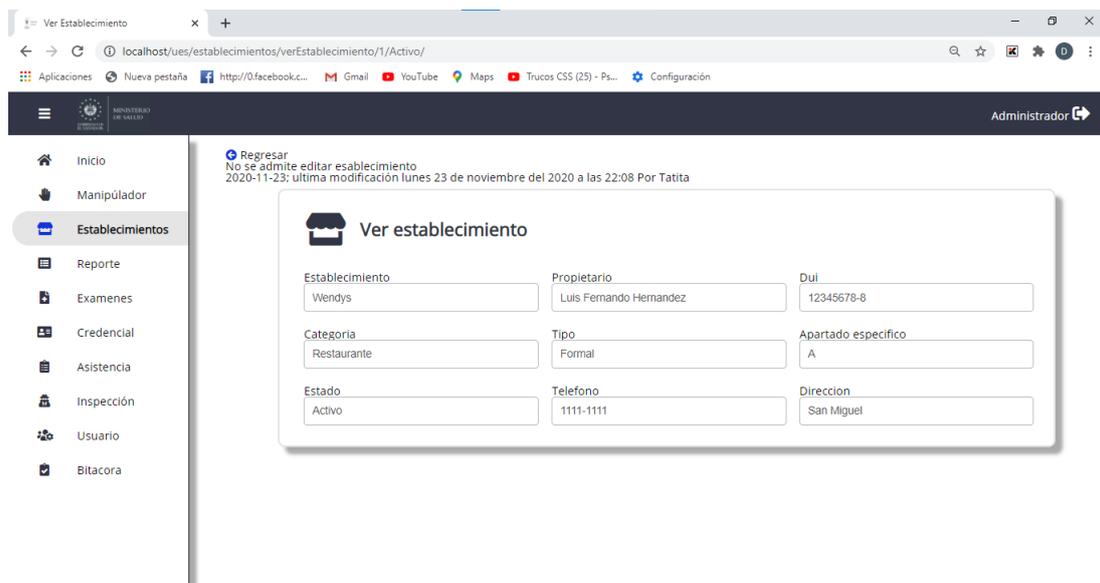


Figura 4.4.67: Interfaz de Ver Establecimiento

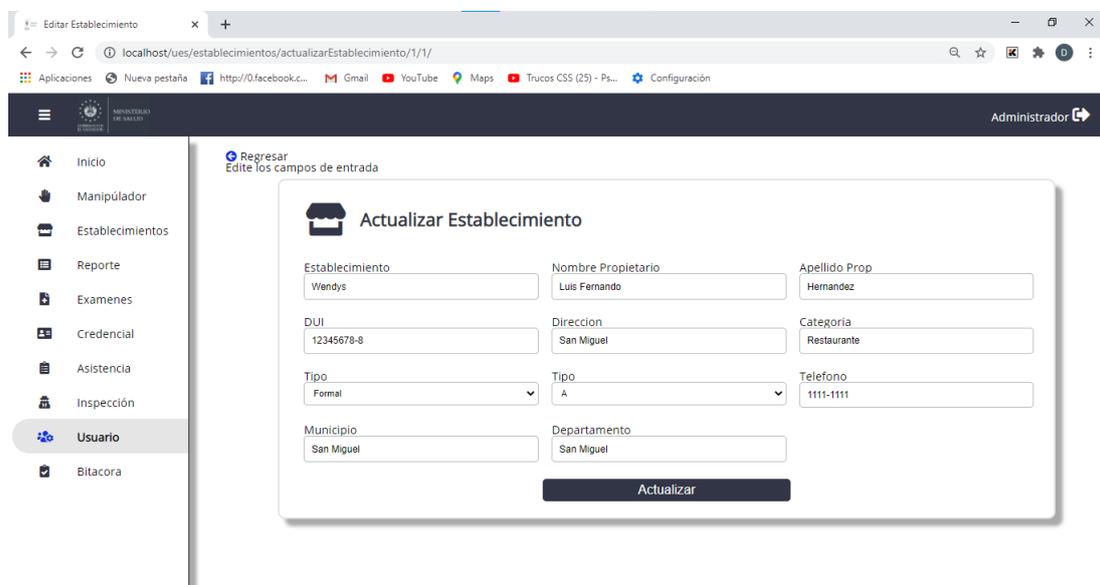


Figura 4.4.68: Interfaz de Actualizar Establecimientos

4.4.4.14. Diagrama de secuencia proceso de Exámenes

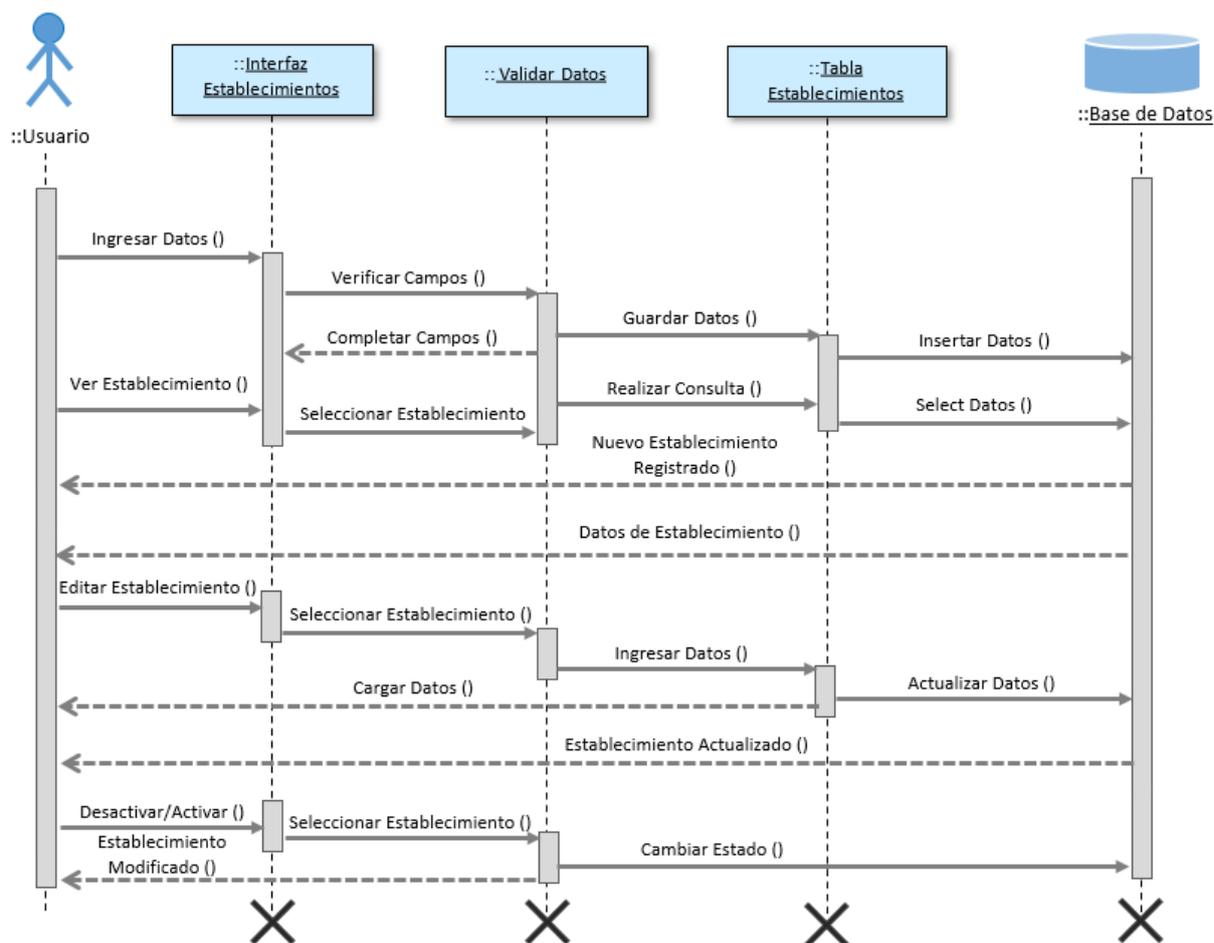


Figura 4.4.69: Diagrama de Secuencia: proceso de Exámenes

4.4.4.15. Interfaz de Exámenes

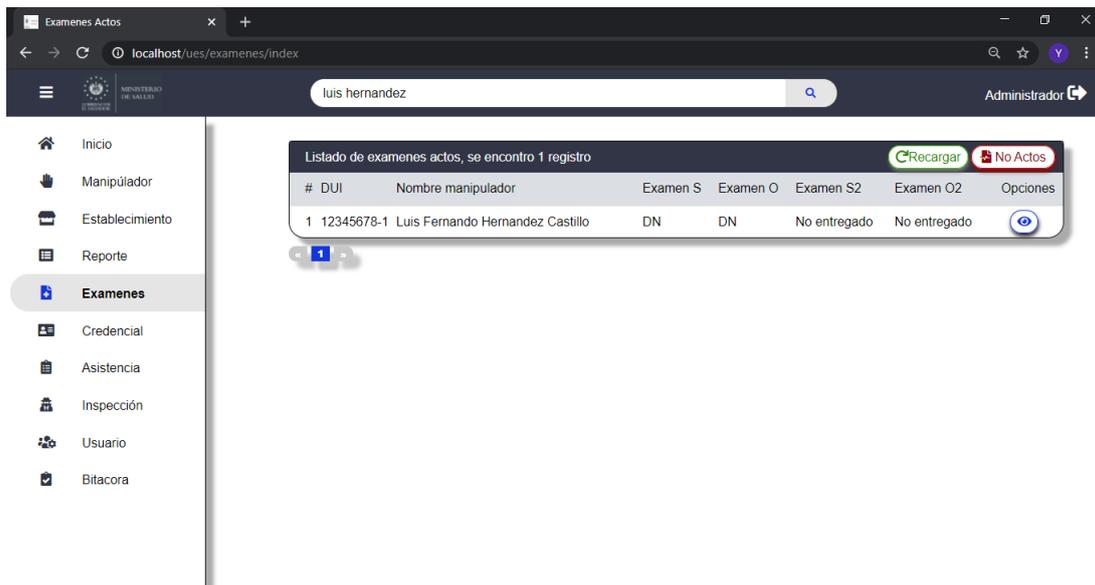


Figura 4.4.70: Interfaz de Búsqueda de Exámenes

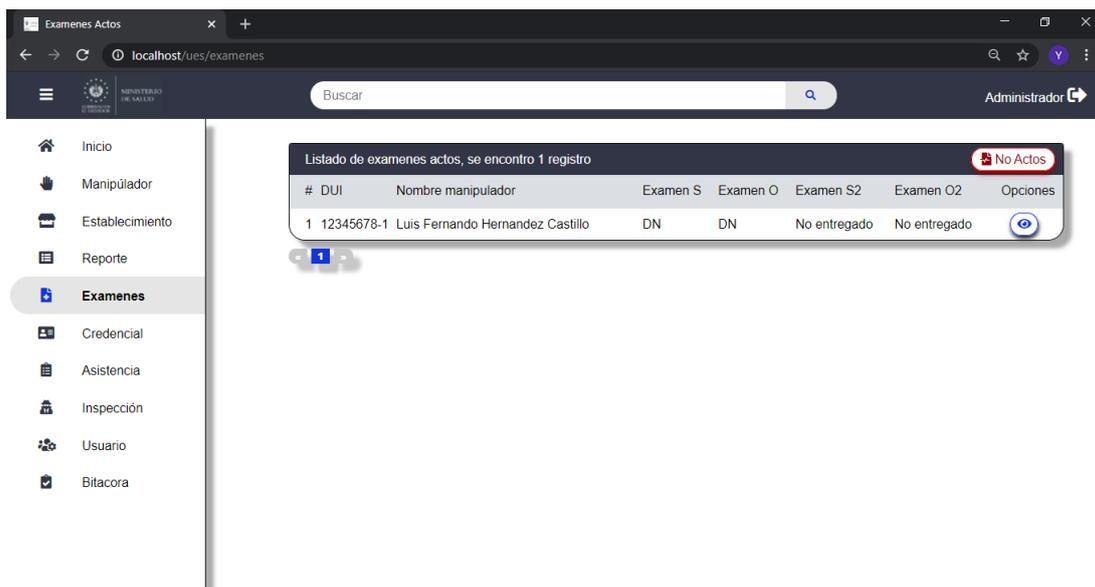


Figura 4.4.71: Interfaz de Exámenes aptos

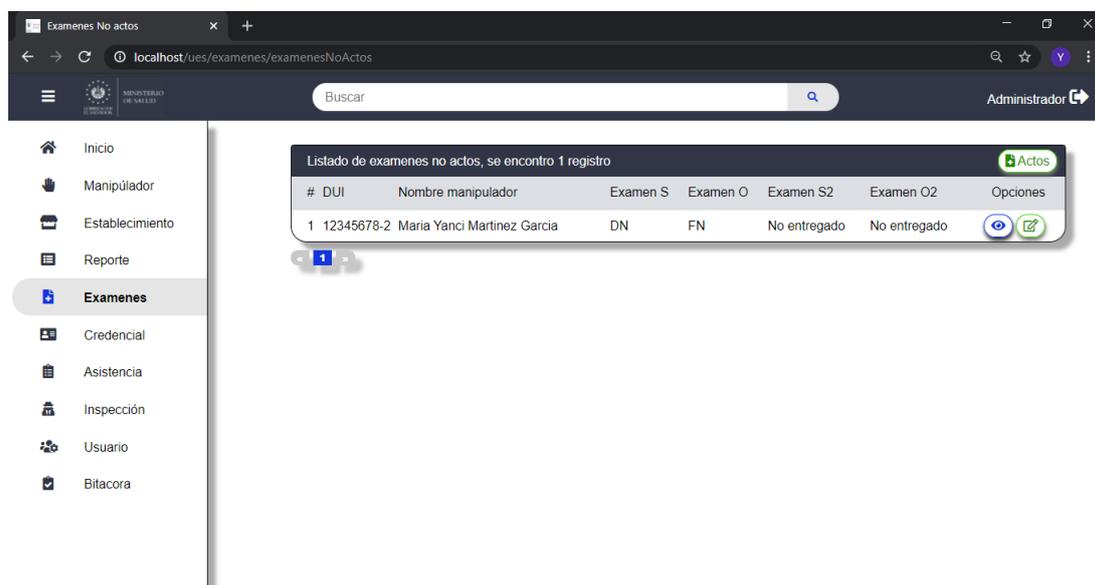


Figura 4.4.72: Interfaz de Exámenes No aptos

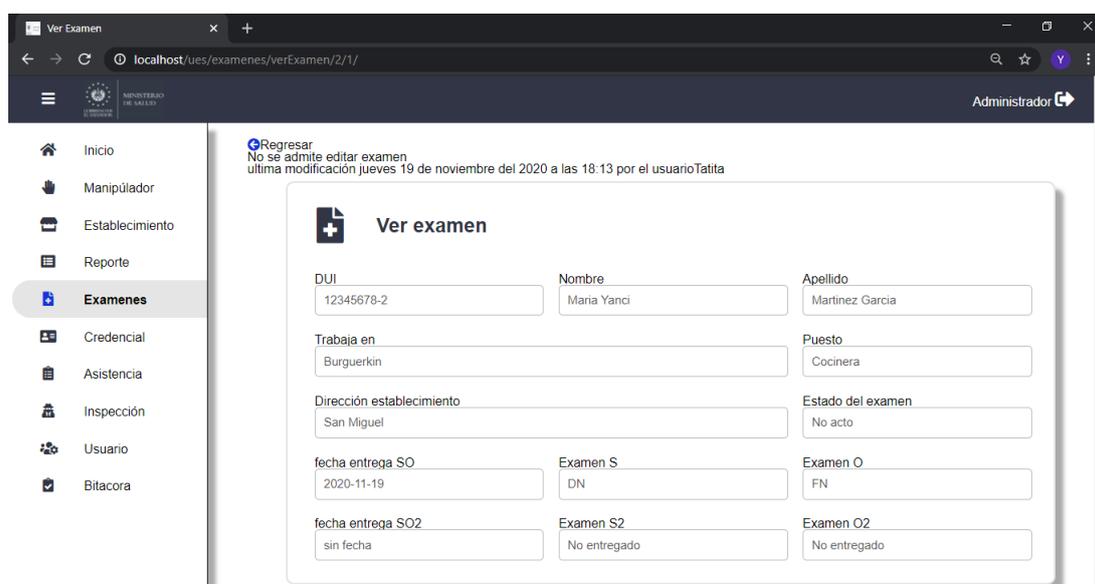


Figura 4.4.73: Interfaz de Ver Exámenes

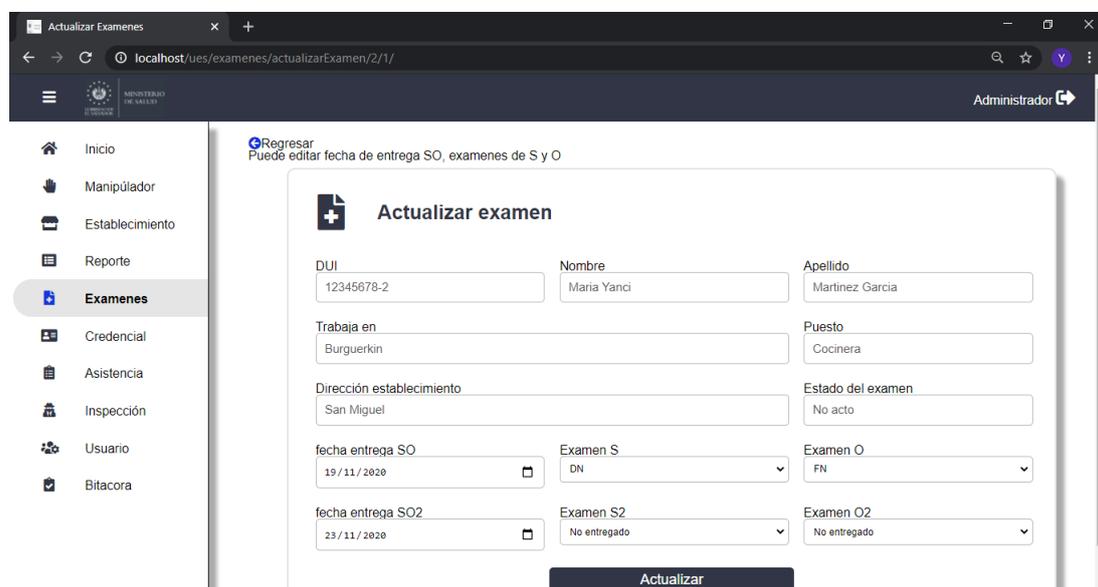


Figura 4.4.74: Interfaz de Actualizar Exámenes

#### 4.4.5. *Diagrama de Clases*

Las metodologías orientadas a objetos trabajan para descubrir las clases, atributos, métodos y relaciones entre las clases. Como la programación ocurre a nivel de clase, definir clases es una de las tareas más importantes del análisis orientado a objetos. Los diagramas de clases muestran las características estáticas del sistema y no representan ningún procesamiento en especial. Un diagrama de clases también muestra la naturaleza de las relaciones entre las clases. En un diagrama de clases, las clases se representan mediante un rectángulo. En el formato más simple, el rectángulo puede incluir sólo el nombre de la clase, pero también puede incluir atributos y métodos. Los atributos son lo que la clase conoce sobre las características de los objetos, y los métodos (también llamados operaciones) son lo que la clase sabe acerca de cómo hacer las cosas. Los métodos son pequeñas secciones de código que trabajan con los atributos. <sup>122</sup>

El diagrama de clases lo usaremos como herramienta que nos permita modelar en forma gráfica la vista de diseño estructural estática de nuestro sistema, también lo usaremos para ilustrar los requerimientos funcionales del sistema recopilados mediante el análisis, así como el diseño del sistema.

---

<sup>122</sup>Kendall y Kendall, Análisis y Diseño de Sistema, Pearson, Octava Edición (2011)

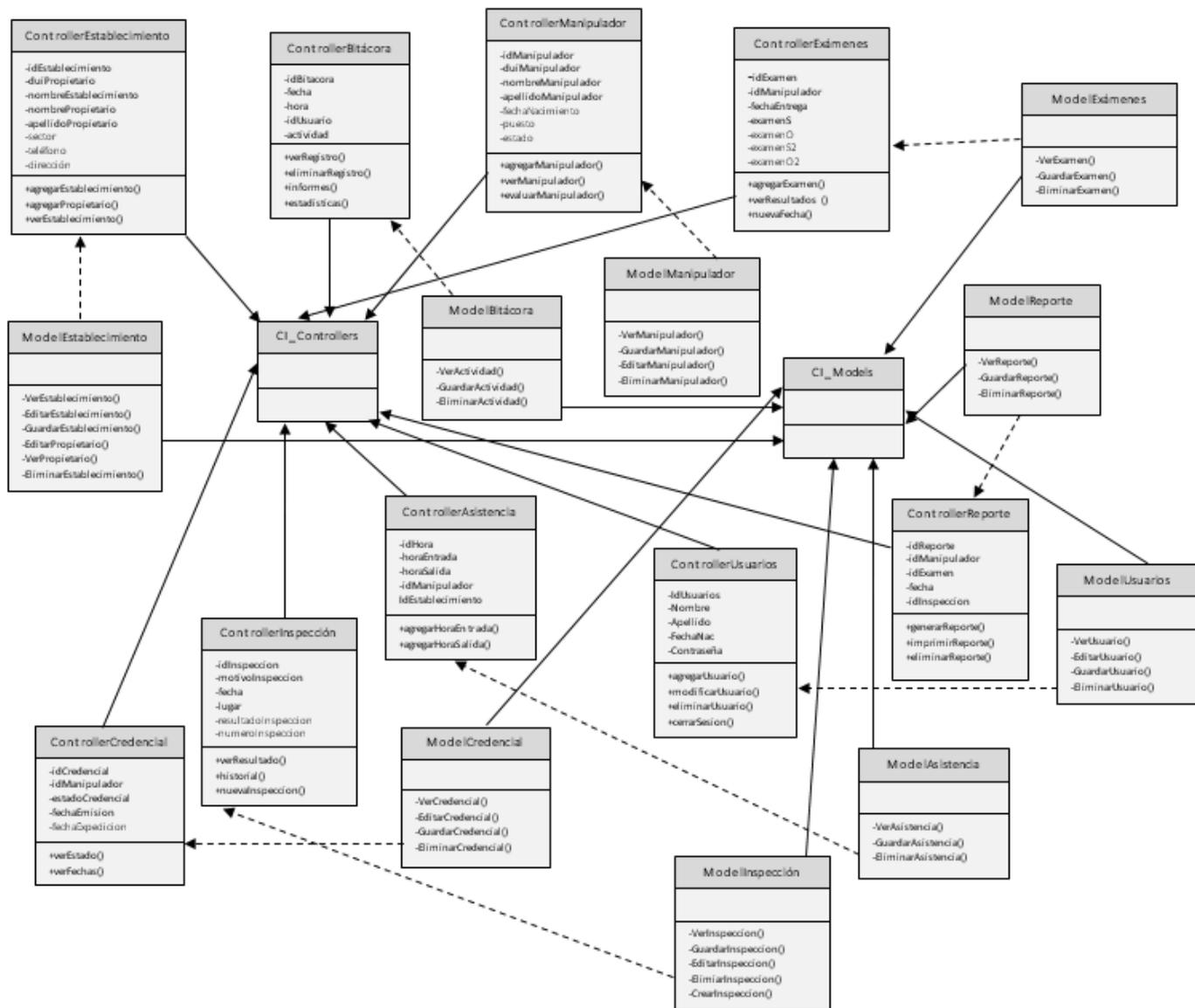


Figura 4.4.75: Diagrama de Clases

#### 4.4.6. Diagramas de la base de datos

El diseño de una base de datos se estructura como un modelo especial llamado esquema de bases de datos. Un esquema de base de datos es el modelo físico o esquema para una base de datos. Representa la implantación técnica del modelo lógico de datos. (*El System Architect lo llama modelo físico de datos.*)<sup>123</sup>

Un esquema relacional de bases de datos define la estructura de la base de datos en términos de tablas, claves, índices y reglas de integridad. Un esquema de la base de

<sup>123</sup>(Jeffrey L. Whitten, 2008, Análisis de sistemas: Diseño y Métodos, p. 430)

datos especifica detalles basándose en las capacidades, terminología y restricciones del sistema de administración de base de datos escogido. Cada DBMS soporta tipos de datos y reglas de integridad, que son diferentes.

### 4.4.6.1. Diagrama Entidad - Relación (ER)

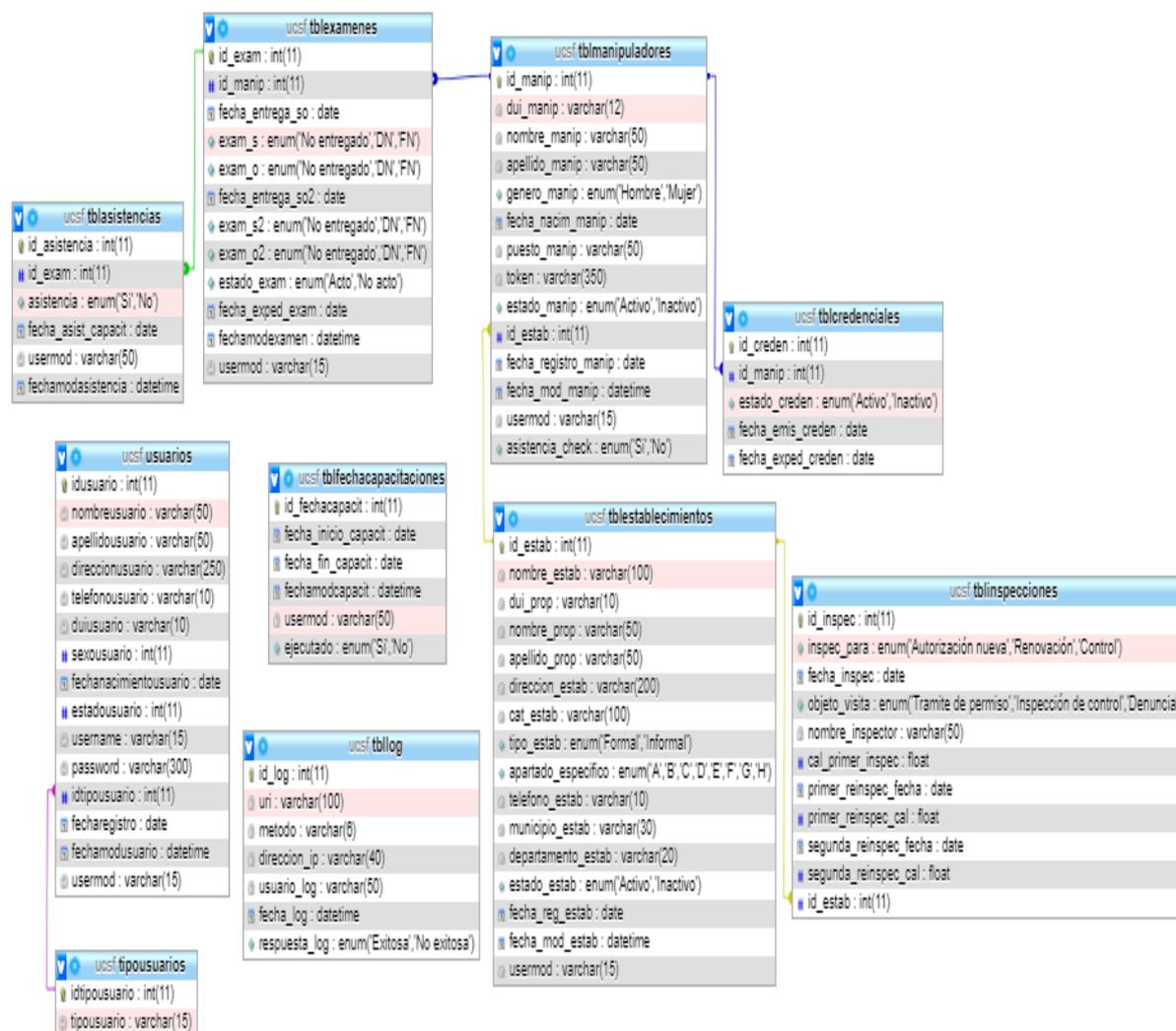


Figura 4.4.76: Diagrama Entidad - Relación

4.4.6.2. Diagrama Físico

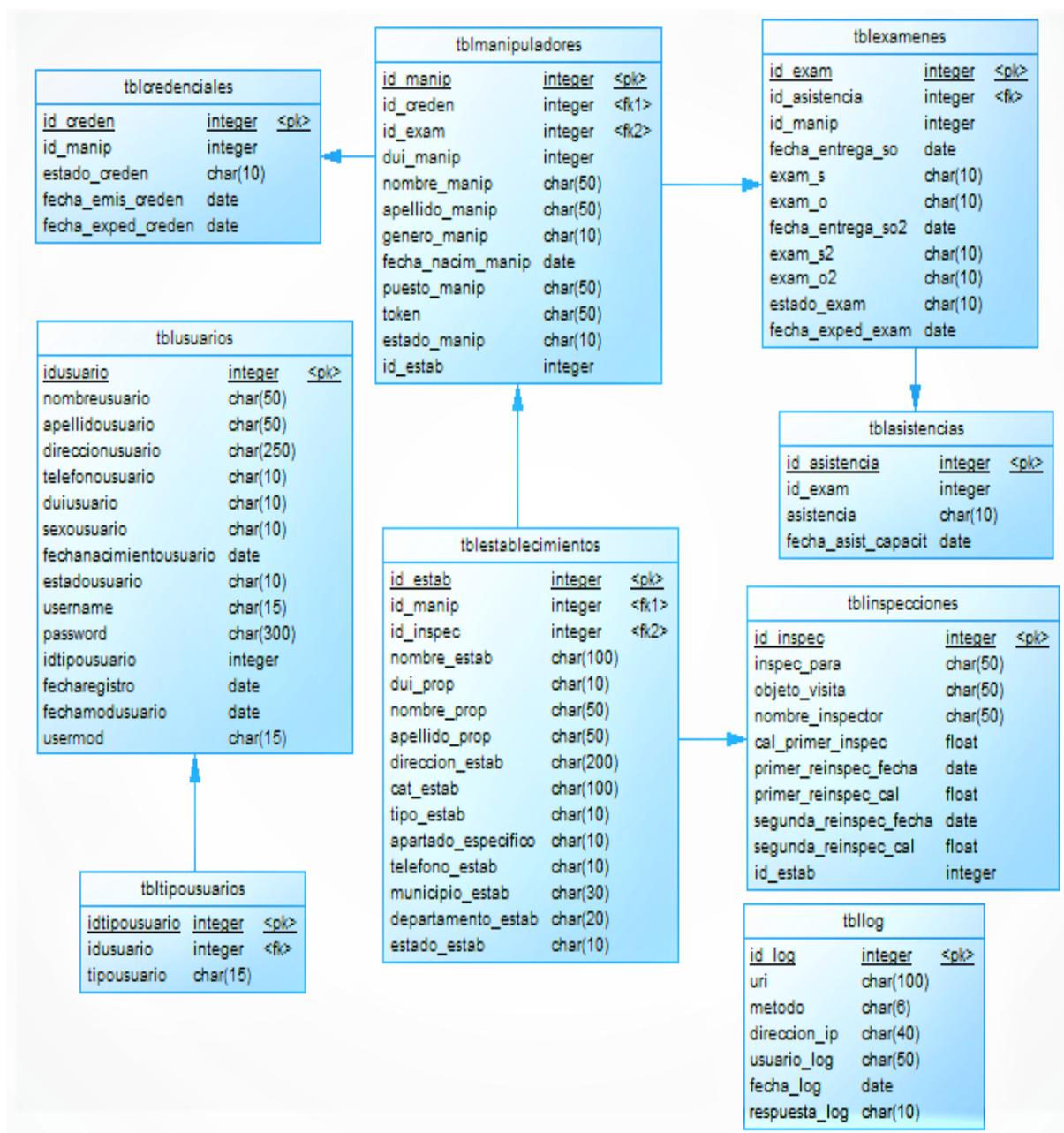


Figura 4.4.77: Diagrama Físico

### 4.4.6.3. Diagrama Lógico

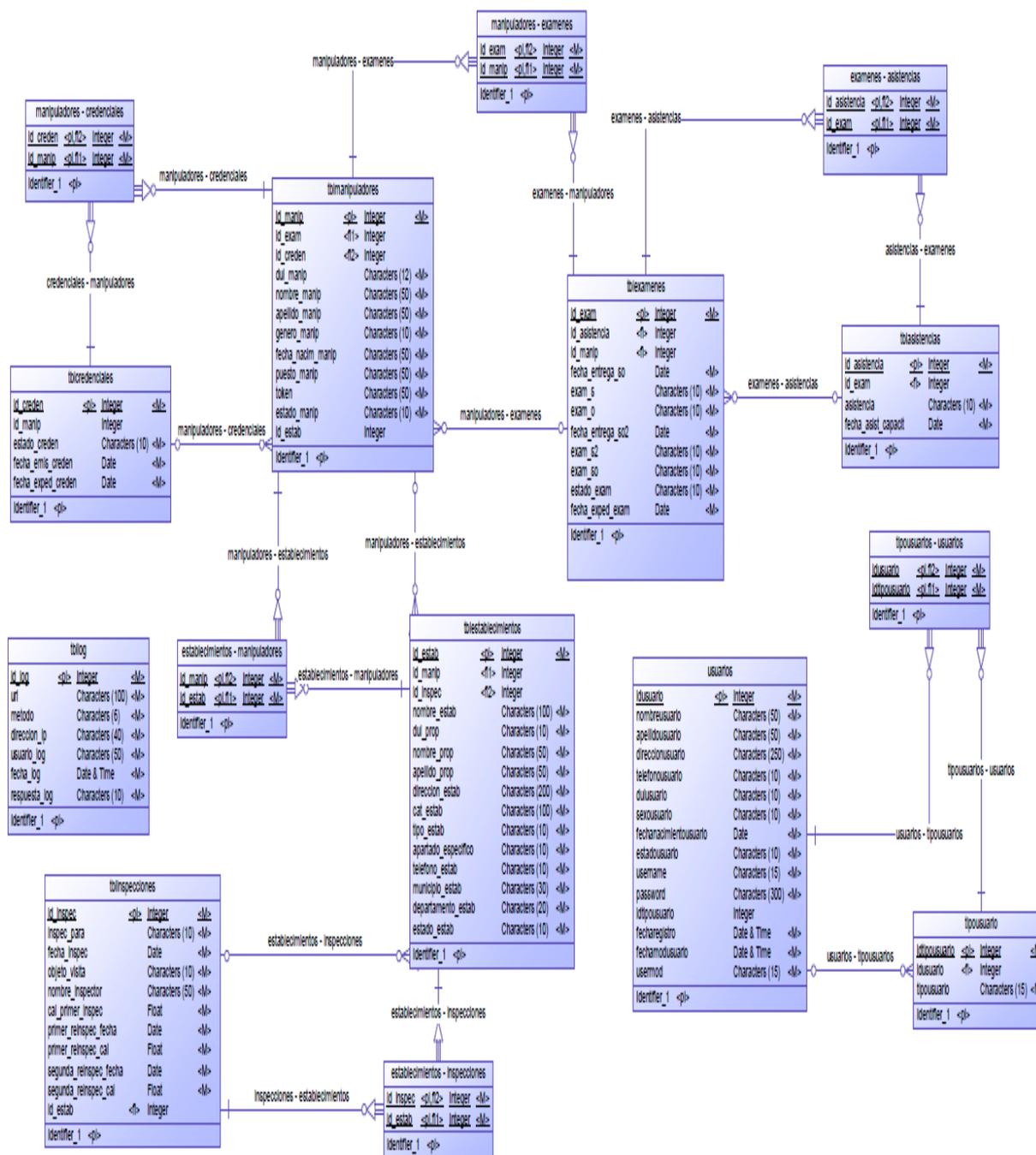


Figura 4.4.78: Diagrama Lógico

### 4.4.6.4. Diagrama Conceptual

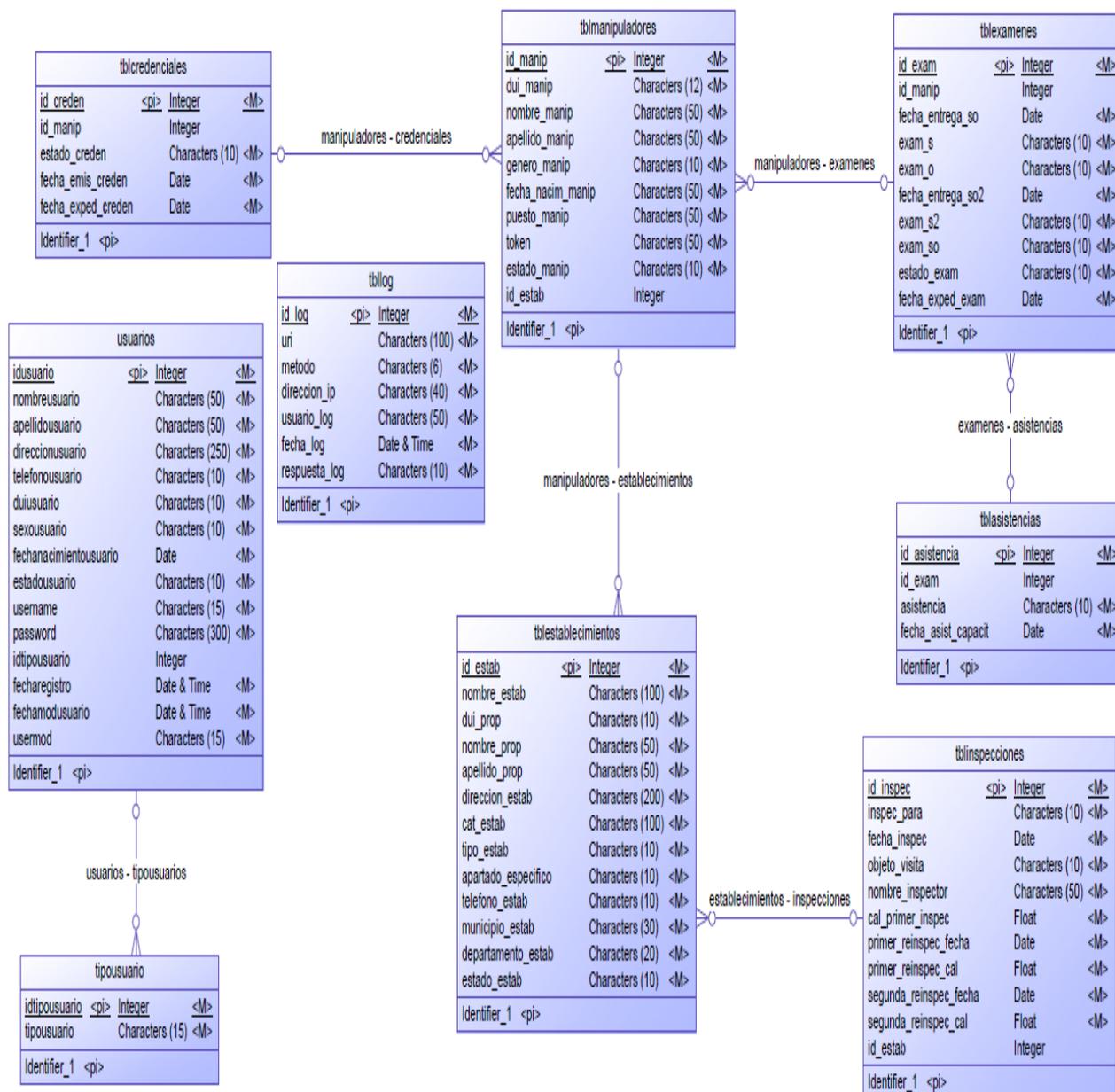


Figura 4.4.79: Diagrama Conceptual

## 4.5. Diccionario de datos

El diccionario de datos es una versión especializada de los diccionarios que se utilizan como referencias en la vida cotidiana. El diccionario de datos es una obra de consulta de información sobre los datos (es decir, *metadatos*); es compilado por los analistas de sistemas para guiarse a través del análisis y diseño. Como documento, el diccionario de datos recopila y coordina términos de datos específicos, además de confirmar lo que significa cada término para distintas personas en la organización. Los diagramas de flujo de datos que vimos en el capítulo 7 son un excelente punto de partida para recolectar entradas para el diccionario de datos.

Una razón importante para tener un diccionario es con el fin de mantener limpios los datos; es decir, para conservarlos consistentes. Si usted almacena datos sobre el sexo de un hombre como “M” en un registro, “Masculino” en un segundo registro y como el número “1” en un tercer registro, los datos no están “limpios”. En este aspecto el diccionario de datos le será muy útil.<sup>124</sup>

### 4.5.1. *Uso del diccionario de datos*

El diccionario de datos ideal es automatizado, interactivo, en línea y evolutivo. A medida que el analista de sistemas aprende sobre los sistemas de la organización, se agregan elementos de datos al diccionario de datos. Por otra parte, el diccionario de datos no es un fin en sí mismo y nunca deberá serlo. Para evitar desviarse con la construcción de un diccionario de datos completo, el analista de sistemas debe considerarlo como una actividad paralela al análisis y diseño de sistemas.

Para que tenga el máximo poder, el diccionario de datos debe enlazarse en varios programas de sistemas, de manera que cuando se actualice o elimine un elemento del diccionario de datos, se actualice o elimine automáticamente de la base de datos. El diccionario de datos se convierte simplemente en una curiosidad histórica si no se mantiene

---

<sup>124</sup>Kendall Kendall. 2011, Análisis y diseño de sistemas, 8th edición, p228

actualizado.<sup>125</sup>

#### ***4.5.2. Diccionario de la base de datos UCSF***

A continuación se muestra el tamaño, descripción, nombre de las columnas, tipo y otros atributos que conforman la base de datos para el Desarrollo de un Sistema Informático de Registro, Emisión e Inspección de Credenciales Sanitarias Otorgadas a los Manipuladores de Alimentos en la Unidad Comunitaria de Salud Familiar de San Francisco Gotera en el Departamento de Morazán.

---

<sup>125</sup>Kendall Kendall. 2011, Análisis y diseño de sistemas, 8th edición, p242

Tabla	Columna	Tipo	Tamaño KB	Descripción
tblasistencias	6	InnoDB	48.0	Almacena la asistencia de los manipuladores a las capacitaciones
tblcredenciales	5	InnoDB	32.0	Almacena las credenciales de los manipuladores
tbl establecimientos	16	InnoDB	16.0	Almacena los datos de los establecimientos formales e informales de los manipuladores
tbl examenes	12	InnoDB	32.0	Almacena los exámenes dentro o fuera de norma de los manipuladores de alimentos
tbl fechacapitaciones	6	InnoDB	16.0	Almacena las fechas de inicio y fin de las capacitaciones
tbl inspecciones	11	InnoDB	32.0	Almacena las inspecciones de los establecimientos formales
tbllog	7	InnoDB	16.0	Almacena la bitácora del sistema
tblmanipuladores	14	InnoDB	32.0	Almacena los datos de los manipuladores de alimentos
tipousuarios	2	InnoDB	16.0	Almacena el tipo de usuario
usuarios	15	InnoDB	64.0	Almacena los datos de los manipuladores

**Tabla 4.5.1:** Tablas de la base de datos

**tblasistencias**

Columna	Tipo	Null	Llave	Llave	Descripción
			P	F	
id_asistencia	int(11)	No	Si		Almacena identificar de asistencia
id_exam	int(11)	No		No	Almacena el id de los exámenes
asistencia	enum('Si', 'No')	No			Almacena la asistencia de los manipuladores
fecha_asist_capacit	date	Si			Almacena la fecha de las asistencias para las capacitaciones
usermod	varchar(50)	Si			Almacena usuario que modifica el registro
fechamodasistencia	datetime	Si			Almacena la fecha de modificación

**Tabla 4.5.2:** Tabla asistencias de BD UCSF**tblcredenciales**

Columna	Tipo	Null	Llave	Llave	Descripción
			P	F	
id_creden	int(11)	No	Si		Almacena identificador de las credenciales
id_manip	int(11)	No		No	Almacena el id de los manipuladores
estado_creden	enum('Activo', 'Inactivo')	No			Almacena el estado de las credenciales
fecha_emis_creden	date	Sí			Almacena la fecha de emision de las credenciales
fecha_exped_creden	date	Sí			Almacena la fecha de expedición de las credenciales

**Tabla 4.5.3:** Tabla Credenciales de BD UCSF

**tblestablecimientos**

<b>Columna</b>	<b>Tipo</b>	<b>Null</b>	<b>Llave</b>	<b>Llave</b>	<b>Descripción</b>
			<b>P</b>	<b>F</b>	
id_estab	int(11)	No	Si		Almacena el identificador de establecimientos
nombre_estab	varchar(100)	No			Almacena el nombre del establecimiento
dui_prop	varchar(10)	No			Almacena el dui del manipulador
nombre_prop	varchar(50)	No			Almacena el nombre del propietario del almacenamiento
apellido_prop	varchar(50)	No			Almacena el apellido del propietario del almacenamiento
direccion_estab	varchar(200)	No			Almacena la dirección del establecimiento
cat_estab	varchar(100)	No			Almacena la categoría del establecimiento
tipo_estab	enum('Formal', 'Informal')	No			Almacena el tipo formal o informal del establecimiento
apartado_especifico	enum('A', 'B', 'C', 'D', 'E', 'F', 'G', 'H')	No			Almacena el apartado específico del establecimiento
telefono_estab	varchar(10)	No			Almacena el número de teléfono del establecimiento
municipio_estab	varchar(30)	No			Almacena el municipio donde se encuentra el establecimiento

Columna	Tipo	Null	Llave P	Llave F	Descripción
departamento_estab	varchar(20)	No			Almacena el departamento donde se encuentra el establecimiento
estado_estab	enum('Activo', 'Inactivo')	No			Almacena el tipo de estado activo o inactivo del establecimiento
fecha_reg_estab	date	No			Almacena la fecha de registro del almacenamiento
fecha_mod_estab	datetime	No			Fecha de modificación del establecimiento
usermod	varchar(15)	No			almacena el usuario que realiza modificación

**Tabla 4.5.4:** Tabla establecimientos de BD UCSF

tblexamenes

Columna	Tipo	Null	Llave P	Llave F	Descripción
id_exam	int(11)	No	Si		Almacena id de exámenes
id_manip	int(11)	No		Si	Almacena el id de manipulador
fecha_entrega <sub>so</sub>	date	Si			Fecha de primera entrega de exámenes de SO
exam_s	enum('No entregado', 'DN', 'FN')	No			Almacena la primera entrega de exámenes de s
exam_o	enum('No entregado', 'DN', 'FN')	No			Almacena la primera entrega de examen de Orina
fecha_entrega_so2	date	Si			Almacena la fecha de segunda entrega
exam_s2	enum('No entregado', 'DN', 'FN')	No			Almacena la segunda entrega de exámenes de s
exam_o2	enum('No entregado', 'DN', 'FN')	No			Almacena segunda entrega de examens de orina
estado_exam	enum('Acto', 'No acto')	No			Almacena el estado de examen apto o no apto
fecha_exped_exam	date	Si			almacena fecha de expedición del examen
fechamodexamen	datetime	Si			Almacena la fecha de modificación del examen
usermod	varchar(15)	Si			Almacena usuario que modifico examenes

Tabla 4.5.5: Tabla examenes de BD UCSF

**tblfechacapacitaciones**

Columna	Tipo	Null	Llave		Descripción
			P	F	
id_fechacapacit	int(11)	No	Si		Almacena el identificador de fecha capacitaciones
fecha_inicio_capacit	date	Si			Almacena la fecha que inician las capacitaciones
fecha_fin_capacit	date	Si			Almacena la fecha en la que finalizara la capacitación
fechamodcapacit	datetime	Si			Almacena la fecha de modificación
usermod	varchar(50)	Si			Almacena usuario que realizo modificaciones
ejecutado	enum('Si', 'No')	No			Almacena la ejecucion de la fecha de capacitación

**Tabla 4.5.6:** Tabla fecha capacitaciones de BD UCSF

**tblinspecciones**

Columna	Tipo	Null	Llave P	Llave F	Descripción
id_inspec	int(11)	No	Si		Almacena el identificador de las inspecciones
inspec_para	enum('Autorización nueva', 'Renovación', 'Control')	No			Almacena las inspecciones segun proceso a realizar
fecha_inspec	date	Si			Almacena la fecha de inspeccion
objeto_visita	enum('Tramite de permiso', 'Inspección de control')	No			Almacena el objetivo de la inspección
nombre_inspector	varchar(50)	No			Almacena el nombre del inspector que realizo la inspección
cal_primer_inspec	float	Si			Primera calificación de la inspección
primer_reinspec_fecha	date	Si			Primera fecha de reinspeccion
primer_reinspec_cal	float	Si			primera calificación de la reinspeccion
segunda_reinspec_fecha	date	Si			Segunda fecha de reinspeccion
segunda_reinspec_cal	float	Si			Segunda calificación de reinspeccion
id_estab	int(11)	No		Si	Almacena el id del establecimiento

**Tabla 4.5.7:** Tabla inspecciones de BD UCSF

## tbllog

Columna	Tipo	Null	Llave P	Llave F	Descripción
id_log	int(11)	No	Si		
uri	varchar(100)	No			Almacena la ruta de donde se han realizado las modificaciones
metodo	varchar(6)	No			Almacena el metodo que se envian los datos
direccion_ip	varchar(40)	No			Almacena la dirección ip
usuario_log	varchar(50)	No			Almacena el usuario que realizo las modificaciones
fecha_log	datetime	No			Almacena la fecha que fue modificado
respuesta_log	enum('Exitosa', 'No exitosa')	No			Almacena la respuesta si fue exitosa o no exitosa los cambios realizados

**Tabla 4.5.8:** Tabla log de BD UCSF

**tblmanipuladores**

Columna	Tipo	Null	Llave		Descripción
			P	F	
id_manip	int(11)	No	Si		Almacena el identificador de los manipuladores
dui_manip	varchar(12)	No			Almacena el dui de los manipuladores
nombre_manip	varchar(50)	No			Almacena el nombre de manipuladores
apellido_manip	varchar(50)	No			Almacena el apellido de los manipuladores
genero_manip	enum('Hombre', 'Mujer')	No			Almacena el genero de los manipuladores
fecha_nacim_manip	date	No			Almacena la fecha de nacimiento de los manipuladores
puesto_manip	varchar(50)	No			Almacena el puesto de los manipuladores
token	varchar(350)	Si			Almacena el caracteres identificadores al cliente
estado_manip	enum('Activo', 'Inactivo')	No			Almacena el estado de los manipuladores activos y inactivos
id_estab	int(11)	No		Si	Almacena id de establecimiento
fecha_registro_manip	date	No			Almacena fecha que fue registrado el manipulador

Columna	Tipo	Null	Llave	Llave	Descripción
			P	F	
fecha_mod_manip	datetime	No			Almacena la fecha que se modifico un registro
usermod	varchar(15)	No			Almacena usuario que modifico un registro
asistencia_check	enum('Si', 'No')	No			Almacena el estado de las capacitaciones asistencia

**Tabla 4.5.9:** Tabla manipuladores de BD UCSF

**tbltipousuarios**

Columna	Tipo	Null	Llave	Llave	Descripción
			P	F	
idtipousuario	int(11)	No	Si		Almacena el identificador de tipo usuario
tipousuario	varchar(15)	No			Almacena el tipo usuario

**Tabla 4.5.10:** Tabla tipo usuarios de BD UCSF

**tblusuarios**

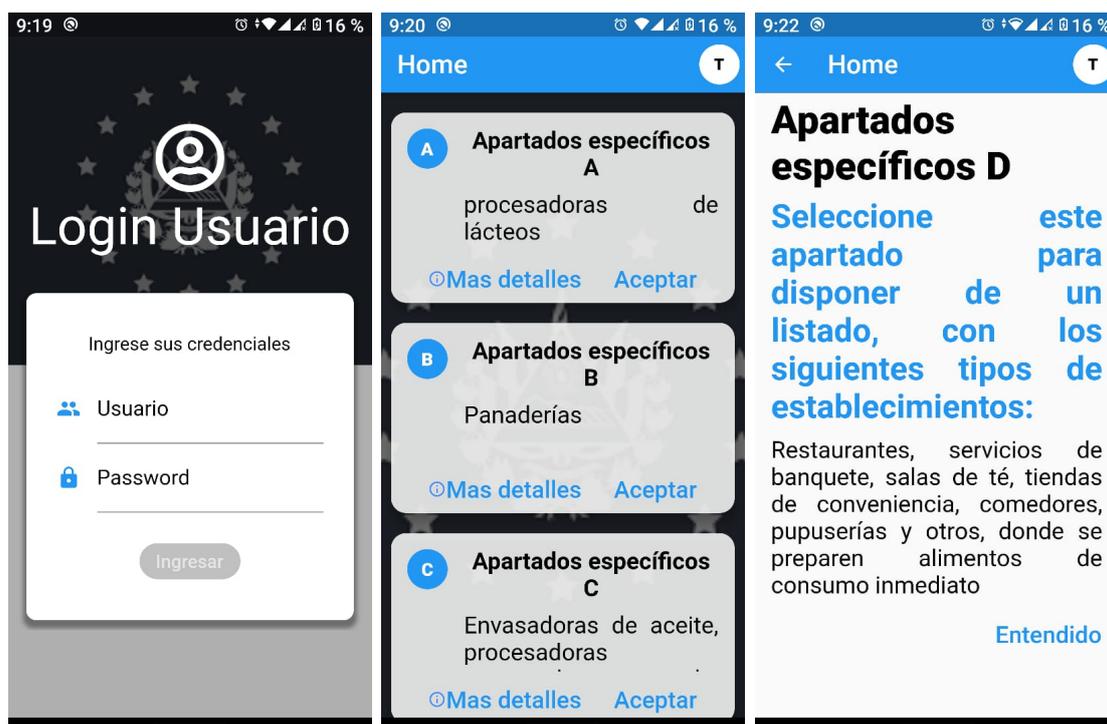
<b>Columna</b>	<b>Tipo</b>	<b>Null</b>	<b>Llave P</b>	<b>Llave F</b>	<b>Descripción</b>
idusuario	int(11)	No	Si		Almacena el identificador de usuarios
nombreusuario	int(11)	No	Si		Almacena nombre de usuario
apellidousuario	varchar(50)	No			Almacena el apellido del usuario
direccionusuario	varchar(250)	No			Almacena la dirección del usuario
telefonousuario	varchar(10)	No			Almacena el teléfono de usuario
duiusuario	varchar(10)	No			Almacena el dui del usuario
sexousuario	int(11)	No			Almacena el sexo del usuario
fechanacimientousuario	date	No			Almacena la fecha de nacimiento del usuario
estadousuario	int(11)	No			Almacena el estado del usuario activo o inactivo
username	varchar(15)	No			Almacena el nombre de usuario
password	varchar(300)	No			Almacena la contraseña del usuario
idtipousuario	int(11)	No		Si	Almacena el id de tipo de usuario

<b>Columna</b>	<b>Tipo</b>	<b>Null</b>	<b>Llave</b>	<b>Llave</b>	<b>Descripción</b>
			<b>P</b>	<b>F</b>	
fecharegistro	date	No			Almacena la fecha que registro el usuario
fechamodusuario	datetime	No			Almacena la fecha de modificación
usermod	varchar(15)	No			Almacena el usuario que hizo una modificación

**Tabla 4.5.11:** Tabla usuarios de BD UCSF

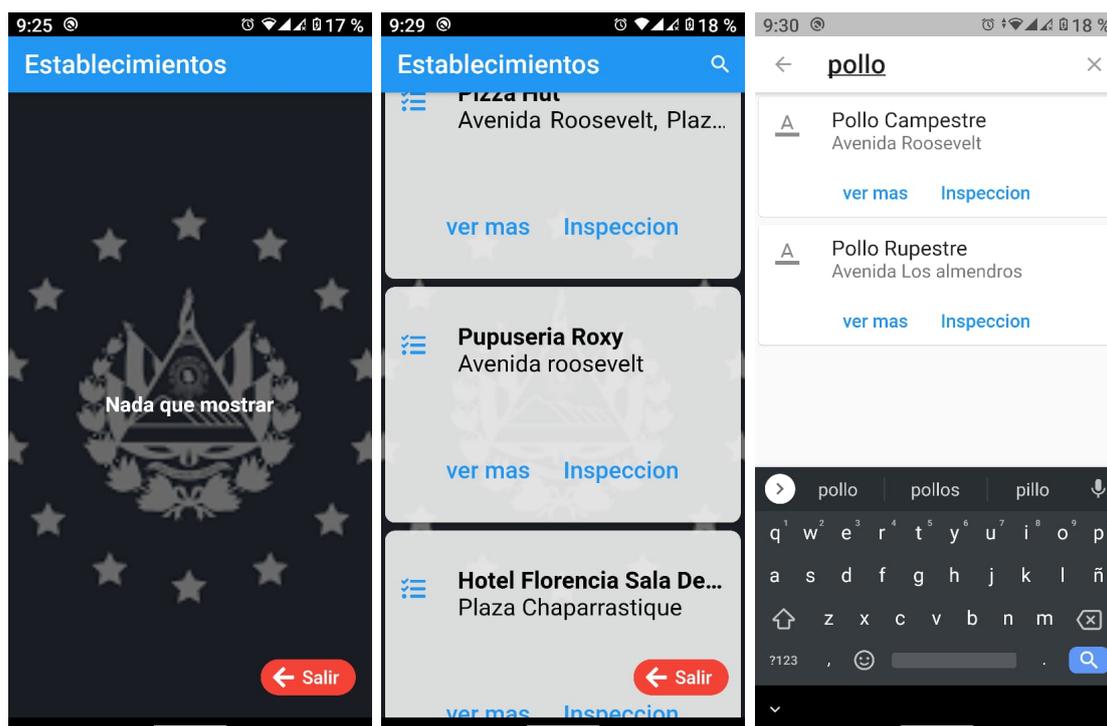
## 4.6. Diseño de la aplicación Móvil

### 4.6.1. Aplicación móvil de los Inspectores de Saneamiento Ambiental de la UCSF.



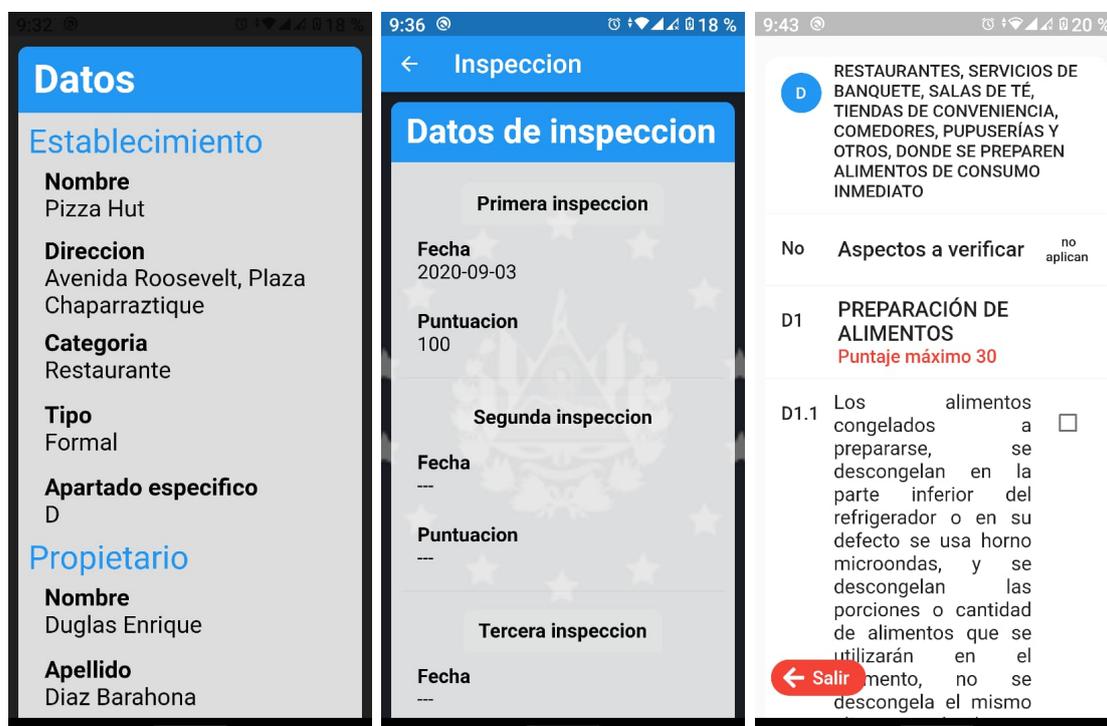
(a) Inicio de sesión inspectores (b) Home de app inspectores (c) Home de app inspectores

**Figura 4.6.1:** Diseño de la aplicación móvil de los inspectores



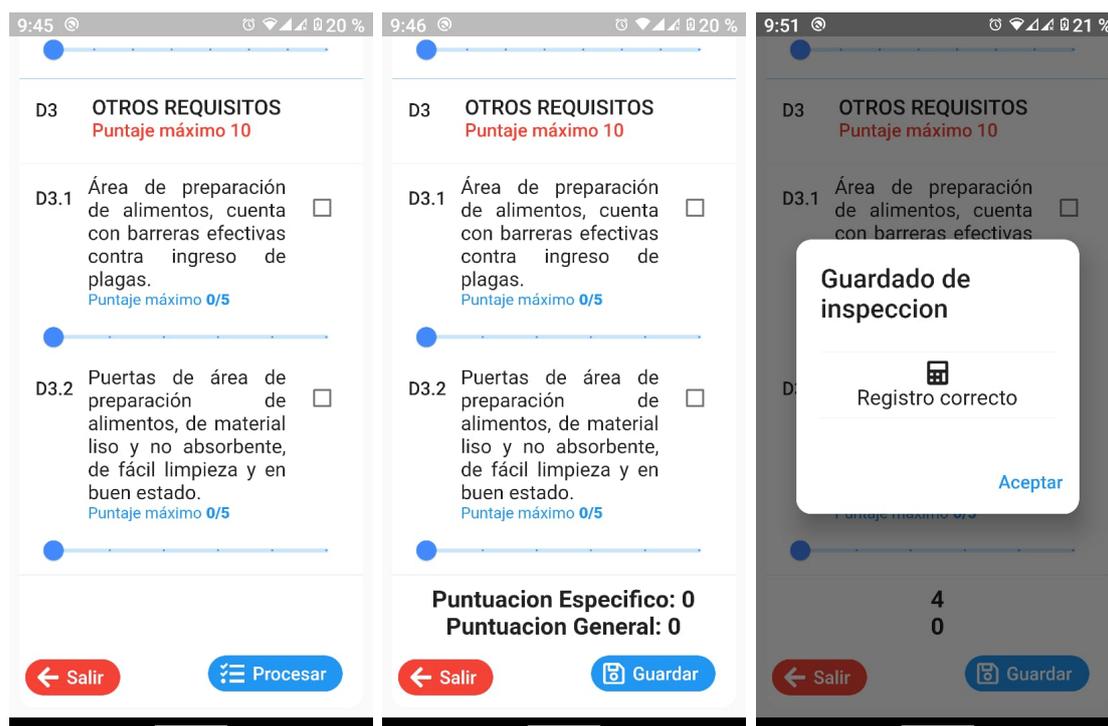
(a) Establecimiento (b) Listado establecimientos (c) Búsqueda establecimientos

Figura 4.6.2: Diseño de la aplicación móvil de los inspectores



(a) Datos Establecimiento (b) Datos de inspección (c) Test de evaluación

Figura 4.6.3: Diseño de la aplicación móvil de los inspectores



(a) Procesar datos

(b) Guardar datos

(c) Registro guardado correctamente

Figura 4.6.4: Diseño de la aplicación móvil de los inspectores

4.6.1.1. Diagrama de secuencias de la aplicación móvil para los Inspectores

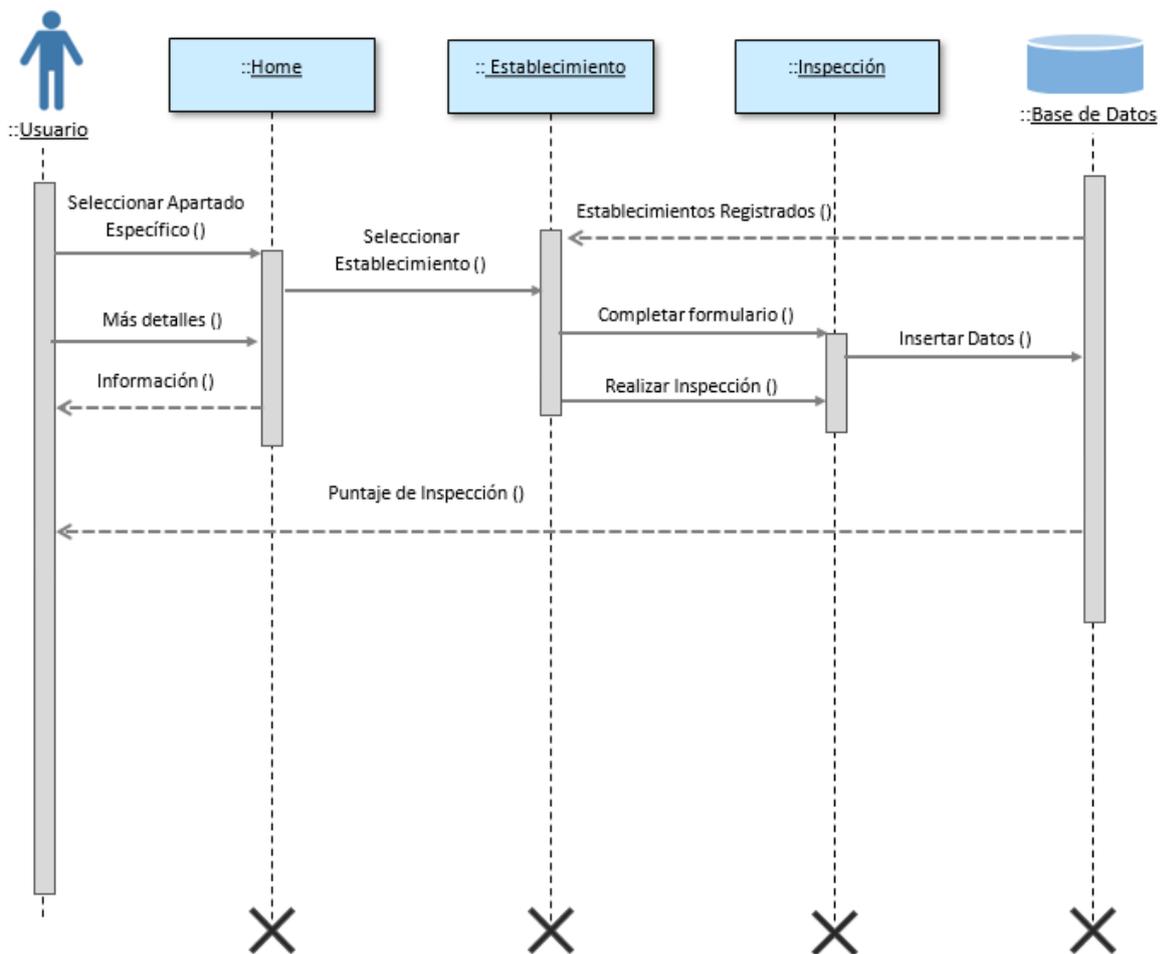


Figura 4.6.5: Diagrama de Secuencia: App Inspectores

### 4.6.2. Aplicación Móvil de los Manipuladores de Alimentos de San Francisco Gotera Morazán

#### 4.6.2.1. Diagrama de secuencias de la aplicación móvil para los manipuladores

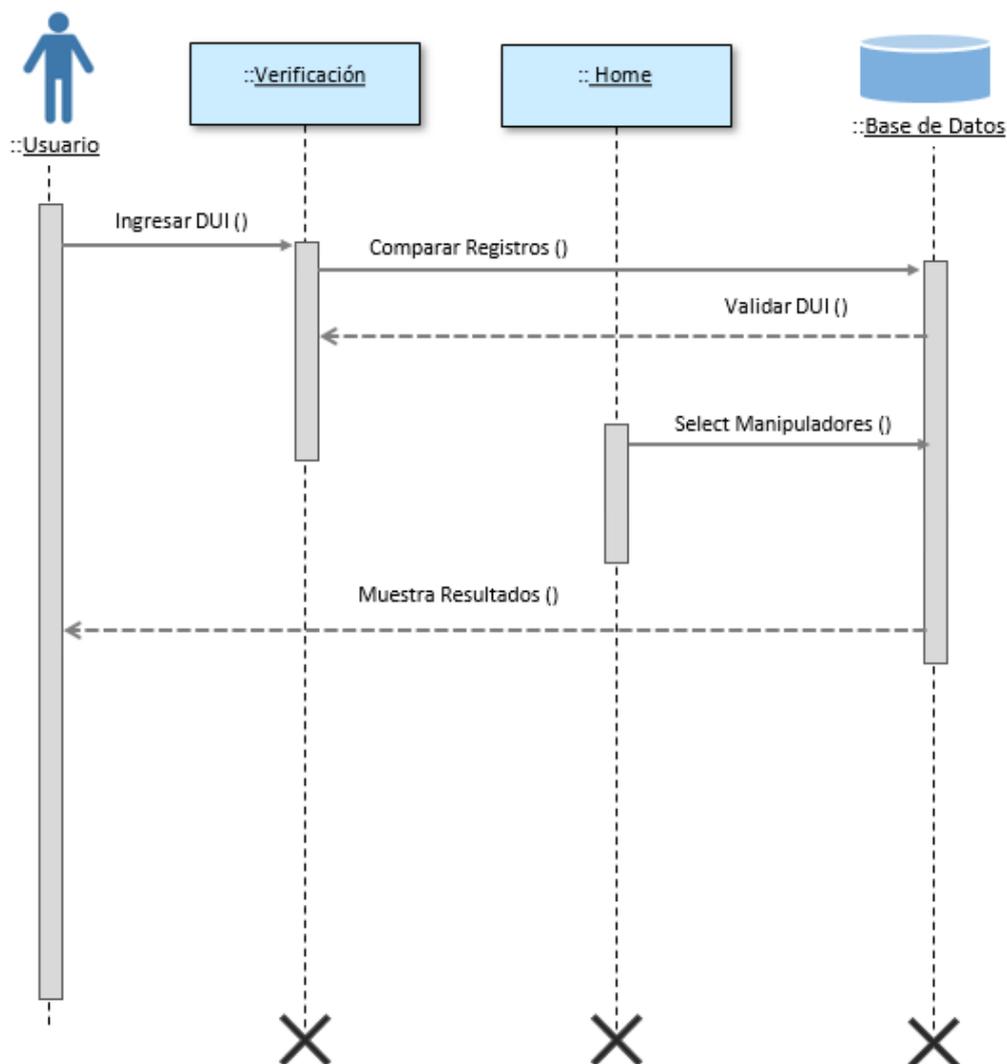
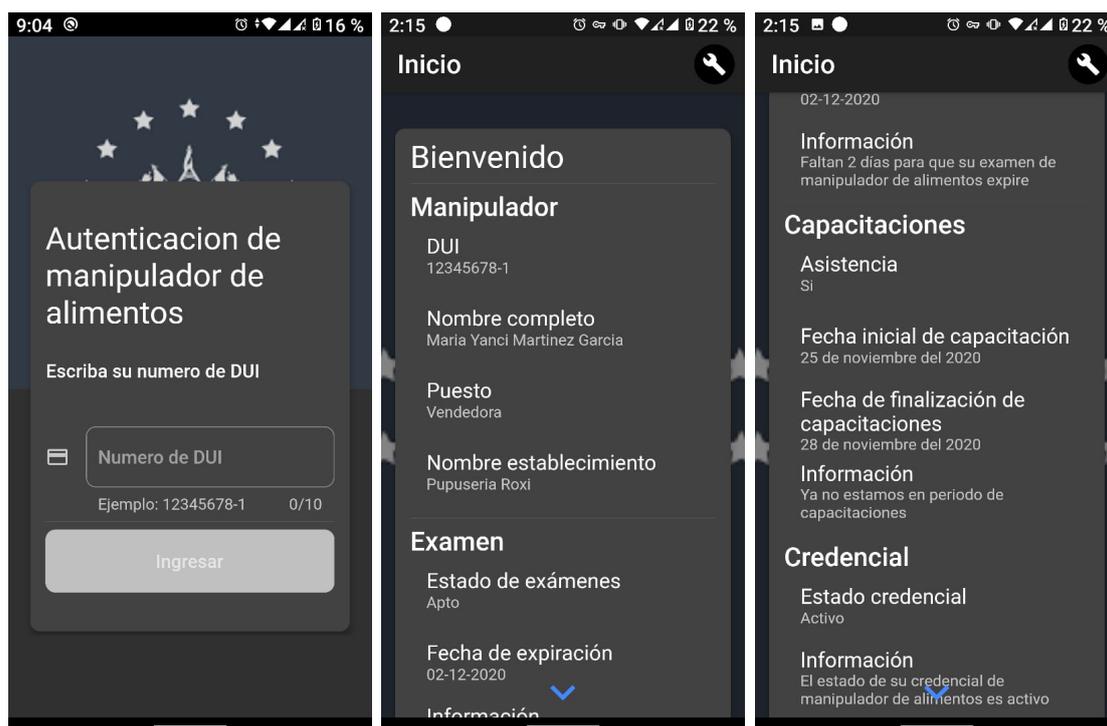


Figura 4.6.6: Diagrama de Secuencia: App Manipuladores

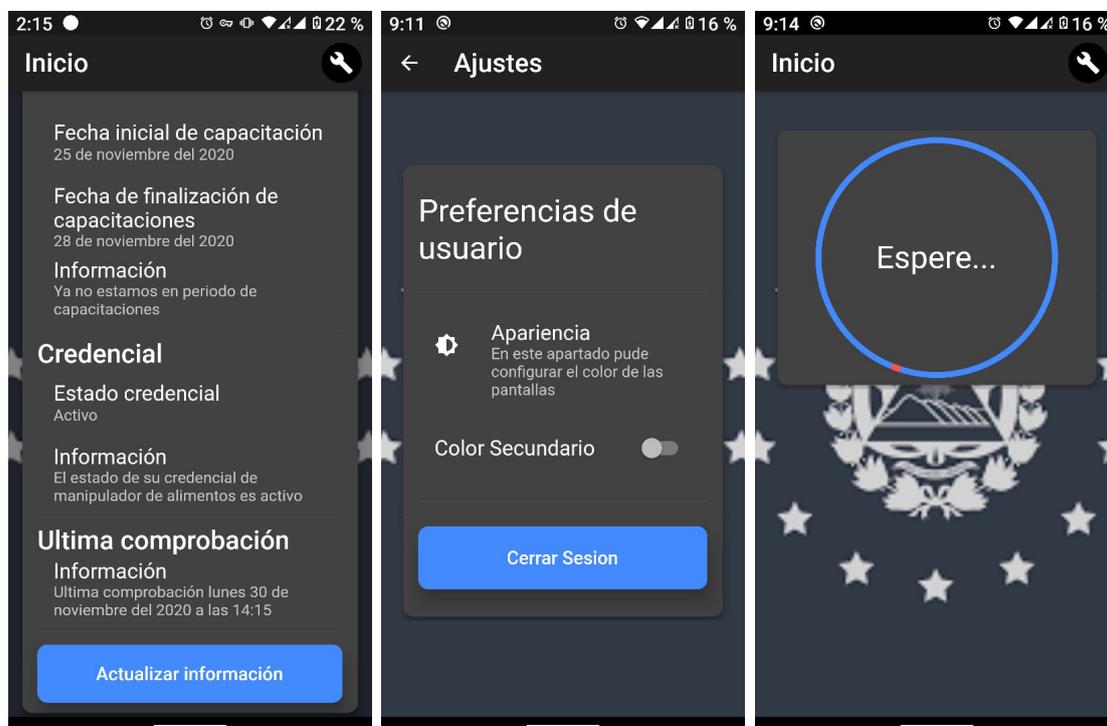


(a) Ingreso de usuario

(b) Inicio

(c) Inicio

Figura 4.6.7: Diseño de la aplicación móvil de los manipuladores



(a) Comprobar

(b) Ajustes

(c) Cargando configuración

Figura 4.6.8: Diseño de la aplicación móvil de los manipuladores

## 4.7. Metodología para la programación

Una metodología de programación es un conjunto o sistema de métodos, principios y reglas que permiten enfrentar de manera sistemática el desarrollo de un programa que resuelve un problema algorítmico. Estas metodologías generalmente se estructuran como una secuencia de pasos que parten de la definición del problema y culminan con un programa que lo resuelve.

### 4.7.1. Programación orientada a objetos

Programación orientada a objetos (POO). Es un paradigma de programación que usa objetos y sus interacciones, para diseñar aplicaciones y programas informáticos. Está basado en varias técnicas, incluyendo herencia, abstracción, polimorfismo y encapsulamiento. Su uso se popularizó a principios de la década de los años 1990. En la actualidad, existe variedad de lenguajes de programación que soportan la orientación a objetos.<sup>126</sup>

#### Propiedades fundamentales de POO<sup>127</sup>

- **Abstracción:** Una abstracción puede definirse como las características específicas de un objeto, aquellas que lo distinguen de los demás tipos de objetos y que logran definir límites conceptuales respecto a quien está haciendo dicha abstracción del objeto.
- **Herencia:** La herencia es específica de la programación orientada a objetos, donde una clase nueva se crea a partir de una clase existente. La herencia (a la que habitualmente se denomina subclase) proviene del hecho de que la subclase (la nueva clase creada) contiene los atributos y métodos de la clase primaria. La principal ventaja de la herencia es la capacidad para definir atributos y métodos nuevos para la subclase, que luego se aplican a los atributos y métodos heredados. Esta

---

<sup>126</sup>[https://www.ecured.cu/ProgramaciónOrientada\\_a\\_Objetos](https://www.ecured.cu/ProgramaciónOrientada_a_Objetos)

<sup>127</sup>Propiedades fundamentales de POO Recuperado de <https://sites.google.com/site/2bvpgalvarezpmariana/segundo-parcial/metodologia-poo>

particularidad permite crear una estructura jerárquica de clases cada vez más especializada. La gran ventaja es que uno ya no debe comenzar desde cero cuando desea especializar una clase existente. Como resultado, se pueden adquirir bibliotecas de clases que ofrecen una base que puede especializarse a voluntad (la compañía que vende estas clases tiende a proteger los datos, miembros usando la encapsulación).

- **Polimorfismo:** El polimorfismo se refiere a la propiedad por la que es posible enviar mensajes sintácticamente iguales a objetos de tipos distintos. El único requisito que deben cumplir los objetos que se utilizan de manera polimórfica es saber responder al mensaje que se les envía. La apariencia del código puede ser muy diferente dependiendo del lenguaje que se utilice, más allá de las obvias diferencias sintácticas. Por ejemplo, en un lenguaje de programación que cuenta con un sistema de tipos dinámico (en los que las variables pueden contener datos de cualquier tipo u objetos de cualquier clase) como Smalltalk se requiere que los objetos que se utilizan de modo polimórfico sean parte de una jerarquía de clases.
- **Encapsulamiento:** La encapsulación es un mecanismo que consiste en organizar datos y métodos de una estructura, conciliando el modo en que el objeto se implementa, es decir, evitando el acceso a datos por cualquier otro medio distinto a los especificados. Por lo tanto, la encapsulación garantiza la integridad de los datos que contiene un objeto.

### Características base

La Programación Orientada a Objetos supone un cambio en la concepción del mundo de desarrollo de software, introduciendo un mayor nivel de abstracción que permite mejorar las características del código final. De manera muy básica, las aportaciones de este paradigma se pueden resumir en: Conceptos de clase y objeto, que proporcionan una abstracción del mundo centrada en los seres y no en los verbos.

Los datos aparecen encapsulados dentro del concepto de clase. El acceso a los datos se produce de manera controlada e independiente de la representación final de los mismos. Como consecuencia, se facilita el mantenimiento y la evolución de los sistemas, al

desaparecer las dependencias entre distintas partes del sistema.

Mediante conceptos como la composición, herencia y polimorfismo se consigue simplificar el desarrollo de sistemas. La composición y la herencia nos permiten construir clases a partir de otras clases, aumentando en gran medida la reutilización.<sup>128</sup>

---

<sup>128</sup>Rodríguez Echeverría, Roberto / Prieto Ramos, Álvaro / Sosa Sánchez, Encarna. 2004, Programación Orientada a Objetos, p8.

## 4.8. Estándar de desarrollo

### 4.8.1. *Estándar de programación*

Un estándar de programación, es una forma de normalizar la programación, de forma tal que al trabajar en un proyecto, cualquier persona involucrada tenga acceso y comprenda el código; además nos permite definir la escritura y organización del código fuente de un programa, facilita la modificación del código fuente, define la forma en que deben ser declaradas las variables, las clases, los comentarios, permite especificar qué datos deben incluirse acerca del programador y de los cambios que se le han realizado al código fuente.

#### 4.8.1.1. **PSR-1.Codificación estándar básica**

La norma comprende lo que debería considerarse el estándar de elementos de codificación necesarios para garantizar un alto nivel de interoperabilidad entre código PHP compartido.

##### **Etiquetas de PHP**

El código PHP debe usar las etiquetas `<? php?>` Largas o las etiquetas short-echo `<? =?>`; eso no debe usar las otras variaciones de etiquetas.

##### **Codificación de caracteres**

El código PHP debe usar solo UTF-8

##### **Espacios de nombres y nombres de las Clases**

Cada clase está en un archivo por sí misma, y está en un espacio de nombres de al menos un nivel: un nombre de proveedor de primer nivel. Los nombres de clase deben declararse en StudlyCaps. Es una forma de notación de texto que sigue el patrón de palabras en minúscula sin espacios y con la primera letra de cada palabra en mayúscula.

##### **Constantes de clase y métodos**

El término clase se refiere a todas las clases, interfaces y rasgos.

### Constantes

Las constantes de clase deben declararse en mayúsculas separados con guion bajo.

```
const VERSION = '1.0' ; const DATE_APPROVED = '2012-06-01' ;
```

### Métodos

Los nombres de los métodos deben declararse en camelCase. Es una forma de notación de texto que sigue el patrón de palabras en minúscula sin espacios y con la primera letra de cada palabra en mayúsculas exceptuando la primera palabra.

#### 4.8.1.2. PSR-2 y PSR-12. Guía de estilos de codificación

##### General

El código debe seguir todas las reglas delineadas en PSR-1.

##### Archivos

- Todos los archivos PHP deben usar el final de línea Unix LF (avance de línea).
- Todos los archivos PHP deben terminar con una sola línea en blanco.
- La etiqueta de cierre debe dejarse omitida de los archivos que solo contengan PHP.

##### Líneas

- No debe haber un límite estricto en la longitud de la línea.
- El límite suave en la longitud de línea debe ser de 120 caracteres; inspectores de estilo automatizado debe advertir, pero no debe error en el límite suave.
- Las líneas no deberían tener más de 80 caracteres; las líneas más largas que eso deberían dividirse en múltiples líneas subsiguientes de no más de 80 caracteres cada una.
- No debe estar rastreando espacios en blanco al final de líneas no en blanco.

- Se pueden agregar líneas en blanco para mejorar la legibilidad e indicar bloques de código.
- No debe haber más de una declaración por línea.

#### **Palabras clave y verdadero / falso / nulo**

- Las palabras clave de PHP deben estar en minúsculas.
- Las constantes de PHP true, false y null deben estar en minúsculas.

#### **Espacio de nombres y declaraciones de uso**

- Debe haber una línea en blanco después de la declaración del namespace.
- Todas las declaraciones de use deben ir después de la declaración namespace.
- Debe haber una palabra clave de use por declaración.
- Debe haber una línea en blanco después del bloque de use.

Como por ejemplo: use FooClass ; use BarClass as Bar ; use OtherVendor;

#### **Clases, propiedades y métodos**

- El término "clase" se refiere a todas las clases, interfaces y rasgos.

#### **Extiende e Implementa**

- Las palabras clave extends e implements DEBEN declararse en la misma línea que el nombre de la clase.
- La llave de apertura para la clase debe ir en su propia línea; el llave de cierre para la clase debe ir en la siguiente línea después del cuerpo.

Por ejemplo: use FooClass ; use BarClass as Bar ; use OtherVendor; class ClassName extends ParentClass implements , //constants, properties, methods

- Las listas de implements pueden dividirse en varias líneas, donde cada la línea subsiguiente se sangra una vez. Al hacerlo, el primer elemento en la lista debe estar en la línea siguiente, y debe haber solo una interfaz por línea.

Ejemplo: use FooClass ; use BarClass as Bar ; use OtherVendor; class ClassName extends ParentClass implements , , // constants, properties, methods

### Propiedades

- La visibilidad debe declararse en todas las propiedades.
- La palabra clave var no debe usarse para declarar una propiedad.
- No debe haber más de una propiedad declarada por estado.
- Los nombres de propiedad no deben tener el prefijo único para indicar visibilidad protegida o privada.
- Una declaración de propiedad se parece a lo siguiente. class ClassName public \$foo = null ;

### Métodos

- La visibilidad debe declararse en todos los métodos.
- Los nombres de los métodos no deberían estar precedidos de un solo guión bajo para indicar visibilidad protegida o privada.
- Los nombres de los métodos no deben declararse con un espacio después del nombre del método.
- Las llaves de apertura deben ir en su propia línea, y la llave de cierre debe ir en la siguiente línea después del cuerpo. No debe haber un espacio después de la apertura paréntesis, y no debe haber un espacio antes del paréntesis de cierre.
- Una declaración de método tiene el siguiente aspecto. Tenga en cuenta la ubicación de paréntesis, comas, espacios y llaves:

```
<?php namespace Vendor; class ClassName public function fooBarBaz ( $arg1 ,  
$arg2 , $arg3 = []) { // method body }
```

### Argumentos del método

- En la lista de argumentos, no debe haber un espacio antes de cada coma, y debe ser un espacio después de cada coma.
- Los argumentos de método con valores predeterminados deben ir al final del argumento lista.
- Las listas de argumentos pueden dividirse en varias líneas, donde cada línea subsiguiente está sangrado una vez al hacerlo, el primer elemento de la lista debe estar en la siguiente línea, y debe haber solo un argumento por línea.
- Cuando la lista de argumentos se divide en varias líneas, el paréntesis de cierre y la llave de apertura deben colocarse juntas en su propia línea con un espacio entre ellos.
- Cuando está presente, las declaraciones `abstract` y `final` deben preceder a la declaración de visibilidad.
- Cuando está presente, la declaración `static` debe venir después de la declaración visibilidad.

### **Método y función de llamadas**

- Al hacer un método o llamar a una función, no debe haber un espacio entre el método o nombre de la función y el paréntesis de apertura, no debe ser un espacio después del paréntesis de apertura, y no debe haber un espacio antes del paréntesis de cierre. en la lista de argumentos, no debe haber un espacio antes cada coma, y debe haber un espacio después de cada coma.

```
<?php namespace Vendor; class ClassName public function foo ( arg1,arg2 ,  
arg3 = [])//methodbody
```

- Las listas de argumentos pueden dividirse en varias líneas, donde cada línea subsiguiente está con sangría una vez al hacerlo, el primer elemento de la lista debe estar en la siguiente línea, y debe haber solo un argumento por línea.
- Cuando la lista de argumentos se divide en varias líneas, el paréntesis de cierre y

la llave de apertura deben colocarse juntas en su propia línea con un espacio entre ellos.

```
<?php namespace Vendor; class ClassName public function aVeryLongMethodName (
ClassTypeHint arg1,arg2 , array arg3 = [])//methodbody
```

### **Estructuras de control**

Las reglas de estilo generales para las estructuras de control son las siguientes:

- Debe haber un espacio después de la palabra clave de la estructura de control.
- No debe haber un espacio después del paréntesis de apertura.
- No debe haber un espacio antes del paréntesis de cierre.
- Debe haber un espacio entre el paréntesis de cierre y la apertura de llave.
- El cuerpo de la estructura debe estar sangrado una vez.
- La llave de cierre debe estar en la siguiente línea después del cuerpo.
- El cuerpo de cada estructura debe estar rodeado por llaves. Esto estandariza cómo las estructuras se ven y reduce la probabilidad de introducir errores como nuevos las líneas se agregan al cuerpo.

#### **if, elseif, else**

- Una estructura if se tiene en cuenta la colocación de paréntesis, espacios y llaves; y else y elseif están en la misma línea que la llave de cierre del cuerpo anterior.
- La palabra clave elseif debe ser utilizada en lugar de else if así para que todo el control las palabras clave parecen palabras sueltas.

#### **switch, case switch , case**

- Una estructura de switch se debe tener en cuenta la ubicación de paréntesis, espacios y llaves. La declaración de case debe ir con sangría una vez desde el switch, y la palabra clave break (u otra palabra clave de terminación) debe ir con sangría al

mismo nivel que el cuerpo del case, debe haber un comentario como `// no break` cuando la caída es intencional en un cuerpo de case no vacío.

### **while, do while**

- Una instrucción `while` se debe tener en cuenta la ubicación de paréntesis, espacios y llaves.

```
<?php while ( $expr ) // structure body
```

- Del mismo modo, una declaración `do while` parece a lo siguiente. Tener en cuenta la ubicación de paréntesis, espacios y llaves.

```
<?php do // structure body; while ( $expr );
```

### **for**

- Una declaración `for` tener en cuenta la colocación de paréntesis, espacios y llaves.

```
<?php for ( $i = 0 ; $i <10 ; $i ++ ) // for body
```

### **foreach**

- Una declaración `foreach`. Tener en cuenta la ubicación de paréntesis, espacios y llaves.

```
<?php foreach ( $iterable as $key =>$value ) // foreach body
```

### **try , catch**

- Un bloque `try catch`. Tener en cuenta la ubicación de paréntesis, espacios y llaves.

```
<?php try // try body catch ( FirstExceptionType $e ) // catch body catch (
OtherExceptionType $e ) // catch body
```

### **Cierres**

- Los cierres deben declararse con un espacio después de la palabra clave de function, y un espacio antes y después de la palabra clave `use`.

- La llave de apertura debe seguir en la misma línea, y la llave de cierre debe continuar la siguiente línea sigue al cuerpo.
- No debe haber un espacio después del paréntesis de apertura de la lista de argumentos o lista de variables, y no debe haber un espacio antes del paréntesis de cierre de la lista de argumentos o lista de variables.
- En la lista de argumentos y la lista de variables, no debe haber un espacio antes de cada coma, y debe haber un espacio después de cada coma.
- Los argumentos de cierre con valores predeterminados deben ir al final del argumento lista.
- Una declaración de cierre se parece a lo siguiente. Tenga en cuenta la ubicación de paréntesis, comas, espacios y llaves:

```
<?php $closureWithArgs = function ( $arg1 , $arg2 ) // body ;  
$closureWithArgsAndVars = function ( $arg1 , $arg2 ) use ( $var1 , $var2 ) // body ;
```

- Las listas de argumentos y las listas de variables pueden dividirse en varias líneas, donde cada línea subsiguiente se sangra una vez. al hacerlo, el primer elemento en la lista debe estar en la siguiente línea, y debe haber solo un argumento o variable por línea.
- Cuando la lista final (ya sea de argumentos o variables) se divide en múltiples líneas, el paréntesis de cierre y la llave de apertura deben colocarse juntos en su propia línea con un espacio entre ellos.

## HTML

Los códigos en HTML deben cumplir con unas reglas adicionales:

- Los tags tienen que estar escritos en minúsculas. Por ejemplo <td> en vez de <TD>
- Los objetos html deben tener id y name, y ambos deben ser iguales. Cuando solo se pone solamente el name y en IE se usa getElementById funciona (incorrectamente), pero en FireFox no. Para evitar esto se deben usar ambos. Ejp:

```
<input id="nombre"pre= name="nombre"type="text"/>.
```

- Cuando hay que escribir variables de PHP dentro de un código HTML se debe tener el HTML dentro de PHP y no al revés.
- Cuando se construye un código de php, después del tag de mayor-interrogación, debe venir la palabra php.
- Finalmente, el código HTML generado por un programa también debe ser legible, bien estructurado e indentado. Por lo que se recomienda el uso de `/n` `/t`"

## CSS

- Existen reglas adicionales para la creación de hojas de estilo.
- Usar una línea para cada atributo, ya que hace más fácil saber qué línea editar para modificar cierta etiqueta.

### 4.8.2. *Estructura de ficheros utilizando modelo, vista controlador*

Como sabemos el Modelo Vista Controlador es un patrón de diseño de software que se encarga de separar la lógica de negocio de la interfaz de usuario, y es el patrón de diseño que estamos siguiendo para el desarrollo del sistema para la Administración de Credenciales de la Unidad comunitaria de Salud de San Francisco Gotera, ya que facilita la funcionalidad, mantenibilidad, y escalabilidad del sistema, de forma cómoda y sencilla.

Para seguir este patrón de diseño es indispensable crear una estructura de ficheros el cuál garantice el cumplimiento de los principios de organización del MVC.

Contar con un sistema bien organizado a nivel de estructura de directorios y ficheros en el modelo vista controlador es importante, ya que nos permite respetar los principios de portabilidad, encapsulamiento e independencia, que son la base de una aplicación modular.<sup>129</sup>

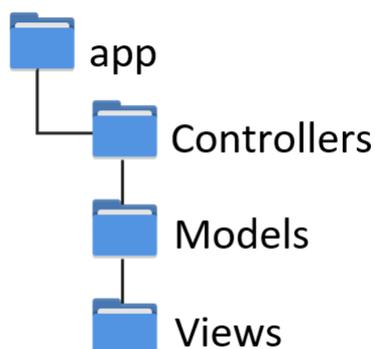
Un sistema modular es aquel que se compone de un número desconocido de

---

<sup>129</sup>Eugenia Bahit, Ingeniería de software y estructuras de directorios definitivas en aplicaciones MVC modulares, 2013, pág. 7

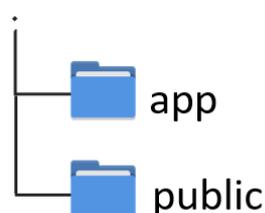
aplicaciones denominadas **módulos** y que permite la incorporación o eliminación de un módulo sin alterar en absoluto las aplicaciones restantes.<sup>130</sup>

Para empezar entenderemos que existen tres directorios indispensable bajo el patrón de diseño MVC,<sup>131</sup> que es donde se alojan los módulos de nuestro sistema como se puede apreciar en la figura 4.8.1



**Figura 4.8.1:** directorios principales en MVC

Pero nuestra estructura de ficheros no es tan simple como eso, ya que contiene una serie de directorios y archivos adicionales a esto e igualmente necesarios para el funcionamiento del sistema. Ya que en el directorio raíz contamos con dos carpetas que son las que contienen todos los archivos del sistema como se puede ver en la figura 4.8.2



**Figura 4.8.2:** carpetas base

El directorio **app** tiene por objetivo el almacenamiento de los módulos del sistema. Mientras que el directorio **public** debe ser una carpeta fácilmente trasladable (portable)

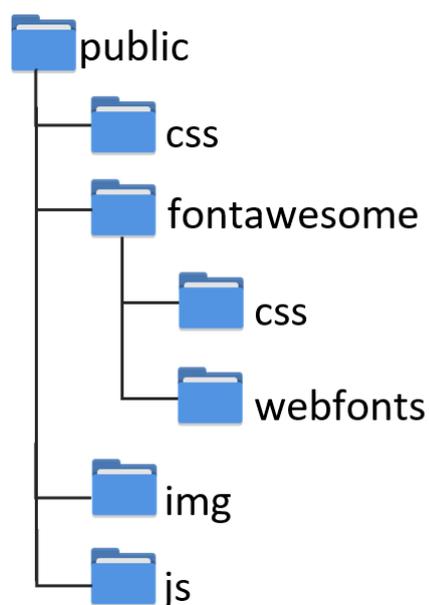
---

<sup>130</sup>IBIDEM, pág. 8

<sup>131</sup>Victor Robles, MVC (Modelo Vista Controlador) en PHP nativo, 2014, victorroblesweb.es

cuyo único objetivo sea el almacenamiento de archivos estáticos inherentes a la interfaz gráfica del usuario del sistema.<sup>132</sup>

La estructura interna de la carpeta *public*, está compuesta por otras carpetas que contienen archivos para diseño de la interfaces de usuario, tales como archivos css, imágenes, archivos de javascript, etc. Figura 4.8.3



**Figura 4.8.3:** estructura interna del directorio public

La estructura del directorio *app*, comprende esencialmente las carpetas que ya se mencionaron anteriormente, controllers, models y views, pero además, de esas carpetas también tiene otras adicionales como se puede observar en la figura 4.8.4

Cada uno de estos directorios cumple con funciones específicas por ejemplo:  
**config/** contiene un archivo php con la configuración del sistema.

**controllers/** contiene los controladores del módulo.

**helpers/** almacena las clases y/o funciones ayudantes propias de los módulos.

**library/** contiene todas las librerías del sistema.

**models/** almacena todos los modelos de aplicación (módulos del sistema).

**views/** contiene la parte lógica de todas las vistas de los módulos.

<sup>132</sup>IBIDEM, pág. 10

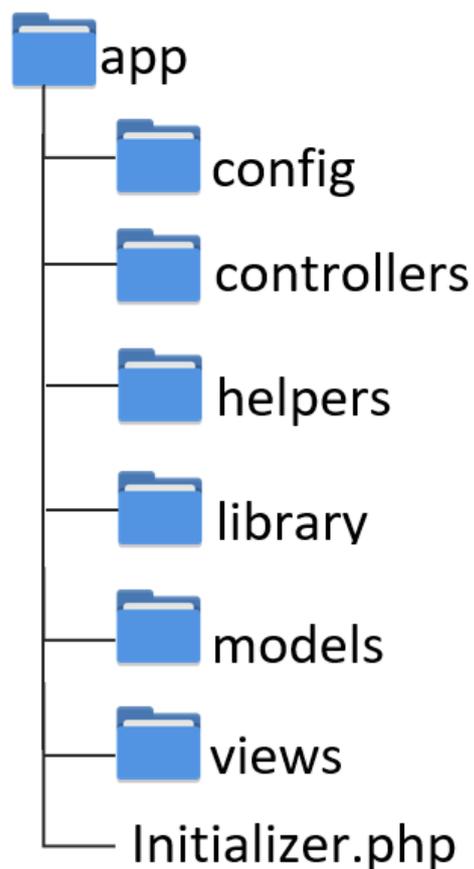


Figura 4.8.4: estructura interna del directorio app

**initializer.php/** es un archivo que su función es cargar las librerías y funciones.

Además, de todos estos directorios y archivos, tenemos en un nuestra estructura, para ser más específico en el directorio raíz un archivo que nunca puede faltar, el archivo **README**. De verdad, este es el archivo más importante. Puede ser un simple archivo de texto con instrucciones cortas. Cada vez que el desarrollo demande un cambio de configuración en la plataforma, la instalación de un software, librería o herramienta de terceros, dicha información se agregará en tiempo real en este archivo, bajo el título «requerimientos e instalación»<sup>133</sup>

Los nombres en inglés representan una sugerencia universal que puede ser bien entendida por cualquier programador/a independientemente de su nacionalidad o procedencia.<sup>134</sup>

---

<sup>133</sup>IBIDEM, pág. 12

<sup>134</sup>IBIDEM, pág. 9

### 4.8.3. REST API

Crear usuario	POST	controlador/usuarios
Obtener usuario	GET 2	controlador/usuarios/(id)
Obtener usuarios	GET	controlador/usuarios
Actualizar usuario	PUT	controlador/usuarios/(id)
Eliminar	DELETE	controlador/usuarios/(id)

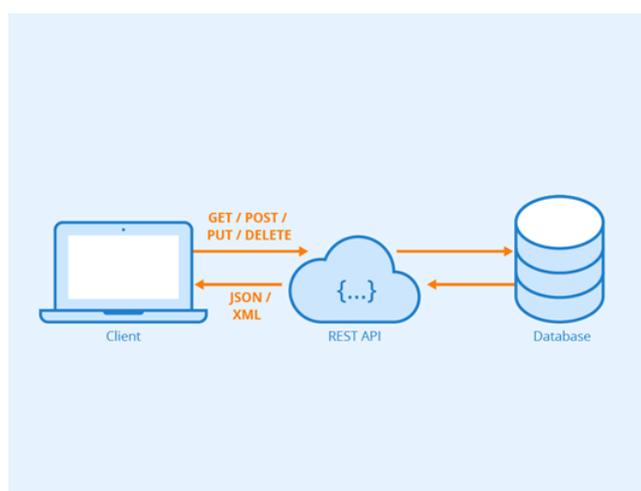
**Tabla 4.8.1:** Rest API

REST: Representational State Transfer (Transferencia de Estado Representacional)

- Es la interfaz entre sistemas que use HTTP para obtener datos o generar operaciones sobre esos datos en todos los formatos posibles, como XML y JSON.
- Protocolo Cliente – Servidor
- Realiza un conjunto de operaciones bien definidas

API : Application programming interface (Interfaz de Programación de Aplicaciones)

Servicios web



**Figura 4.8.5:** Servicios web

#### 4.8.4. *Estándares de base de datos*

El uso de estándares de base de datos es gran importancia ya que asegura la legibilidad del modelo de datos, incluso para personas que no estén relacionadas con el desarrollo del sistema, en etapas de análisis y diseño. Además, estos estándares facilitan la portabilidad entre motores de base de datos, plataformas y aplicaciones; además, facilita la tarea en el momento de programar para los desarrolladores del sistema.<sup>135</sup> (?).

Algunos de los estándares que se usaron al momento de diseñar y construir nuestra base de datos son:

- Únicamente se utilizaron caracteres alfabéticos, salvo que por la naturaleza del nombre se necesiten dígitos numéricos. Se prohíbe el uso de caracteres de puntuación o símbolos.
- Las letras acentuadas se reemplazarán por sus equivalentes no acentuadas.
- El nombre elegido debe de ser lo más descriptivo posible para evitar que se preste a distintas interpretaciones.
- Los nombres de las tablas deben especificarse en plural y de acuerdo a las reglas anteriores.
- Toda relación entre tablas debe implementarse mediante **constraints** (claves foráneas) con integridad referencial, de acuerdo al motor de base de datos utilizado.
- Los nombres de las tablas deben anteponer el prefijo "tbl"seguido del nombre específico de cada tabla.
- Toda tabla debe poseer uno o más campos clave.
- El nombre del campo clave debe estar compuesto por "id" + nombre de la tabla en singular (para claves no compuestas). Dependiendo de la naturaleza de la entidad, el nombre de la tabla a usar es el de la misma tabla, o el de la relacionada.

---

<sup>135</sup>Dirección general de gobierno digital, Estándares de diseños de Base de Datos, <http://static.aytoroquetas.org/>

- Los campos clave deben ubicarse al inicio de la definición de la tabla (deben ser los primeros).

# Capítulo 5

## Plan de implementación

Para llevar a cabo la implementación exitosa del sistema se requiere de un proceso adecuado que cumpla con los requerimientos tanto en tecnología como son software y hardware, como también el conocimiento técnico por parte de los usuarios y administradores.

El plan consiste en la elaboración de ciertos manuales claves y puntuales que permitan mostrar a los usuarios y personal de informática de la Unidad de Salud Familiar de San Francisco Gotera cada uno de los elementos fundamentales para una correcta instalación y que el sistema pueda quedar en perfecto funcionamiento.

Cada uno de los factores necesarios para la puesta en marcha del proyecto de forma funcional se describen perfectamente en la siguiente documentación.

Cabe señalar que el proyecto está disponible para que la implementación sea inmediata, pero queda a disposición del personal de la Unidad de Salud y de los administradores de la misma el periodo o fecha para realizar la implementación según estimen conveniente.

## 5.1. Documentación

La documentación del sistema de registro, inspección y emisión de credenciales sanitarias otorgadas a los manipuladores de alimentos en la Unidad de Salud Familiar de San Francisco Gotera consiste en la recopilación de la información necesaria en donde se explican las características técnicas y funcionalidades del sistema a través de una serie de manuales que sirven como guías para los usuario y personal encargado de la configuración y mantenimiento del sistema.

Todo lo necesario para la instalación, uso, configuración o modificación del sistema se incluye en los 3 diferentes manuales que son:

- Manual de Instalación
- Manual de Usuario
- Manual de programación

### 5.1.1. *Manual de usuario*

El manual de usuario se ha elaborado con el propósito de contribuir como una herramienta para que guíe al usuario del sistema de Registro, Inspección y emisión de credenciales sanitarias otorgadas a los manipuladores de alimentos en la Unidad Comunitaria de Salud Familiar de San Francisco Gotera, ya sea a los Inspectores de Salud o colaboradores para el uso correcto y donde se muestra cómo realizar cada tarea específica que los usuarios requieran.

Es necesario que antes de que el sistema sea utilizado, se lea y ponga en práctica el contenido de este manual.

En el manual de usuario se puede encontrar paso a paso todas las tareas que la plataforma realiza con sus respectivas ilustraciones que permiten tener mayor comprensión de cada tarea. Además, ayudará a los futuros usuarios a tener un mayor conocimiento del uso del sistema.

### ***5.1.2. Manual de programación***

El manual de programación va dirigido más que todo al personal del área de informática de la Unidad de Salud Familiar de San Francisco Gotera o a quién corresponda la instalación, soporte y mantenimiento del sistema dentro de la Unidad.

El personal encargado de la administración y mantenimiento del sistema tiene que tener conocimientos técnicos sobre las siguientes tecnologías, para mayor comprensión del manual:

- PHP
- JavaScript
- HTML5
- CSS3
- MYSQL

Este manual le permitirá al personal de informática conocer la estructura de cómo se elaboró el sistema y los pasos necesario para realizar cualquier cambio o modificación en el mismo.

### ***5.1.3. Manual de instalación***

En el manual de instalación se muestran todos los pasos a seguir para la instalación y configuración del sistema de registro, inspección y emisión de credenciales a los manipuladores de alimentos en la Unidad de Salud Familiar de San Francisco Gotera.

El contenido de este manual consiste en instrucciones básicas para llevar a cabo la instalación y configuración de forma exitosa en los ordenadores para que funciones de forma óptima.

El manual también contiene ilustraciones para que sea más comprensible su contenido. Este manual es de uso obligatorio la omisión del uso de este manual puede

ocasionar una mala instalación o configuración y por lo tanto un mal funcionamiento del sistema.

#### ***5.1.4. Plan de implementación***

El plan de implementación de nuestro sistema desarrollado de registro inspección y emisión de credenciales sanitarias otorgadas a los manipuladores de alimentos en la Unidad Comunitaria de Salud Familiar de San Francisco Gotera, consiste en el proceso que asegura la operatividad del sistema de información y que permite al usuario obtener beneficios por su operación.

El plan de implementación también servirá como un instrumento de programación y control de la ejecución del proyecto y todas las actividades que se lleven a cabo para dar cumplimiento a la puesta en marcha del proyecto.

De acuerdo a los alcances y limitaciones del proyecto, el desarrollo del sistema llega hasta la etapa de elaboración de un plan de implementación.

### 5.1.5. *Estructura Organizativa*

La siguiente ilustración muestra las áreas de la institución que estarán involucradas en la implementación del sistema que permitirá mejorar el proyecto de capacitación para la higiene en la Unidad Comunitaria de Salud Familiar de San Francisco Gotera Morazán.



**Figura 5.1.1:** Estructura organizativa de las personas involucradas en la implementación del sistema web

### 5.1.6. *Descripción de la estructura organizativa*

A continuación se describe específicamente el rol que realizara cada una de las personas involucradas en el plan de implementación

- **Dirección de La Unidad Comunitaria de Salud Familiar** Es la entidad encargada de supervisar la implementación de todos los proyectos que se llevan acabo en la institución y la directora de UCSF de San Francisco Gotera Morazan Dr.Sandra de la Paz Laínez tendrá la ficción de facilitadora de los recursos y el personal requerido para las respectivas capacitaciones que se deberán llevar acabo en la sede.
- **Área de Saneamiento Ambiental** Son las personas que estarán a cargó del sistemas web y las asignadas a recibir las capacitaciones necesarias para el uso y la administración de este para luego poder transmitir este conocimiento a usuarios que ellos otorguen autorización del sistema informático.

### 5.1.7. Recursos a utilizar

#### ■ Financieros

Las capacitaciones no tendrán ningún costo ya que serán impartidas por los bachilleres María Yanci Martínez García, Luis Fernando Hernández Castillo, Duglas Enrique Díaz Barahona y Erick Adalberto López Joya, creadores del sistema web.5.1.1.

Cantidad	Equipo	Detalle
2	Computadoras	La primera de ellas, será la que se conecte a un proyector, para que los capacitadores puedan exponer el funcionamiento del sistema web. La segunda servirá para que el personal del área de saneamiento ambiental pueda hacer uso y pruebas con el sistema web.
2	Smartphone	Este dispositivo será utilizado por los inspectores en capacitación, para la visualización del sistema. Ya que deberán estar capacitados en su uso tanto en computadora como en dispositivos móviles.
1	Proyector	Para proyectar la capacitación del sistema web y visualizar lo que se va a impartir en las capacitaciones.
-	Conexión a internet	Para poder desarrollar las capacitaciones

**Tabla 5.1.1:** Recursos necesarios para la realización de la capacitación

#### ■ Tiempo

El tiempo estimado para desarrollar el plan de implementación del sistema informático por personal autorizado se visualiza mediante el siguiente cronograma.



**CRONOGRAMA DE ACTIVIDADES PARA LLEVAR A CABO LA IMPLEMENTACIÓN DEL SISTEMA WEB PARA MEJORAR EL DESARROLLO DE UN SISTEMA INFORMÁTICO DE REGISTRO, EMISIÓN E INSPECCIÓN DE CREDENCIALES SANITARIAS OTORGADAS A LOS MANIPULADORES DE ALIMENTOS EN LA UNIDAD COMUNITARIA DE SALUD FAMILIAR DE SAN FRANCISCO GOTERA EN EL DEPARTAMENTO DE MORAZÁN.**



Actividades a desarrollar	Tiempo estimado														
	Día 1	Día 2	Día 3	Día 4	Día 5	Día 6	Día 7	Día 8	Día 9	Día 10	Día 11	Día 12	Día 13	Día 14	Día 15
Instalar el sistema web en un host de prueba.	x														
Presentar el sistema web a las autoridades respectivas de la UCSF de San Francisco Gotera, Morazán, área de saneamiento ambiental.		x													
Realizar pruebas supervisadas por personas designadas por los inspectores técnicos.			x	x											
Realizar las capacitaciones a los inspectores técnicos del área de saneamiento ambiental de la UCSF.					x	x	x	x	x						
Instalación del sistema informático en servidor designado por la UCSF.										x	x	x	x	x	
Plan de implementación finalizado con el lanzamiento del sistema informático.															x

Tabla 5.1.2: Cronograma de implementación

- **Infraestructura**

El lugar donde se realizarán las capacitaciones serán las instalaciones de La Unidad Comunitaria de Salud Familiar de San Francisco Gotera, Morazán. en donde la directora de la institución sera quien asigne el espacio físico adecuado para la realización de la actividad.

- **Material de apoyo o Documentación**

Para llevar a cabo las capacitaciones por parte del personal del Área de saneamiento ambiental de la UCSF de San Francisco Gotera (Inspector técnico Javier Hernández y Humberto Sánchez) se cuenta con el apoyo de los manuales de la Plataforma, en donde se explican los pasos o procesos de cómo se utiliza y como se realizan las configuraciones previas para poder hacer uso de ella.

# Capítulo 6

## Conclusiones y Recomendaciones

### 6.1. Conclusiones

Mediante el estudio preliminar se pudo conocer la situación problemática que atraviesa la Unidad Comunitaria de Salud Familiar de San Francisco Gotera y se determinó la necesidad del desarrollo de un Sistema Informático para llevar un registro preciso de los manipuladores de alimentos y tener una clasificación de los que están aptos o no aptos para la obtención del carné correspondiente.

El marco referencial brindo un panorama general de la institución para comenzar la maquetación del software.

En el marco teórico se plantearon los conceptos y definiciones relacionadas con la elaboración del Sistema Informático para así, brindar una perspectiva de las herramientas y procedimientos a seguir a lo largo del desarrollo.

La metodología de la investigación nos brindó un enfoque para llegar a la solución de la problemática planteada. Los instrumentos de recolección de datos permitieron conocer la información más relevante en cuanto a las necesidades de cada módulo a desarrollar así también los recursos con los que cuenta la Unidad Comunitaria de Salud Familiar de San Francisco Gotera, para implementar un Sistema Informático que satisfaga tales necesidades.

Lo datos recopilados mediante los instrumentos se analizaron para proveer soluciones a las necesidades planteadas, dando el surgimiento a los requerimientos funcionales y no funcionales del Sistema Informático, para dar lugar al diseño lógico y la estructura de la propuesta planteada previamente.

A lo largo de la etapa de programación se le realizaron pruebas al software para darle solución a fallas potenciales, así también una vez culminado, se le realizaron pruebas finales para exponerlo a diferentes escenarios que podrían ocasionar problemas futuros.

Con la automatización del registro de los manipuladores de alimentos y el control de las inspecciones realizadas a los establecimientos en San Francisco Gotera, se solventará la problemática actual del manejo de la información en el área de saneamiento ambiental evitando procesos engorrosos y agilizando tramites.

## 6.2. Recomendaciones

Se recomienda hacer uso de la metodología de desarrollo en cascada para el futuro actualizaciones y modificaciones del Sistema Informático.

Se recomienda seguir la documentación establecida (manuales) si se requiere realizar una modificación o cambios en la funcionalidad del Sistema.

Realizar pruebas al momento de implementar cualquier tipo de cambio en funcionamiento del Sistema Informático.

Documentar cualquier tipo de cambio en el Sistema para facilitar el soporte y mantenimiento del mismo.

Hacer uso de las opciones de copias de respaldo de la base de datos como del código fuente del sistema, además de restringir el acceso a éstas de usuarios no autorizados.

# Bibliografía

- [1] **Antonio Camacho** Herramienta para el análisis de requerimientos dentro de la pequeña empresa desarrolladora de software en Bogotá, Colombia, Proyecto de grado, Pontificia Universidad Javeriana, 2005.
- [2] **Arthur Pointeau** Diagrama de casos de uso, Openclassrooms, 2018.
- [3] **Carlos Mario Zapata Gilma Liliana Garces**, Generación del diagrama de secuencias de UML2.1.1 desde esquemas preconceptuales, 2008.
- [4] **Eugenia Bahit** Ingeniería de software y estructuras de directorios definitivas en aplicaciones MVC modulares, 2013.
- [5] **Greet Peersman** Métodos de Recolección y Análisis de Datos en la Evaluación de Impacto, 2014.
- [6] **Ivar Jacobson** Ian Spence, Kurt Bittner. Casos de uso 2.0, la guía para ser exitoso con los casos de uso, Estados Unidos, IVAR JACOBSON INTERNATIONAL SA, 2013.
- [7] **J. D Gauchat** El gran libro de HTML5, CSS3 y Javascript Primera edición, 2012.
- [8] **Jeffrey L. Whitten** Análisis de sistemas: Diseño y Métodos, 2008.
- [9] **Juan Garrido Cobo**. TFC Desarrollo de Aplicaciones Móviles, 2013.
- [10] **Juan Miguel Vergara Pineda** ¿Qué es? y ¿qué nos ofrece?, la herramienta phpMyAdmin Extraído de [www.coriaweb.hosting](http://www.coriaweb.hosting), 2016.
- [11] **Juan Pavón Mestras**. Aplicaciones Web/Sistemas Web, 2012.

- [12] **Karla Ceballos** UML: Diagrama de Secuencia, Portafolio Digital, 2015.
- [13] **Kendall & Kendall** Analisis y diseño de sistemas, 8th Edicion, 2011, P.4
- [14] **Manuel Perez Cardona** irebase, qué es y para qué sirve la plataforma de Google, 2016, <https://www.iebschool.com/blog/firebase>.
- [15] **Martin Fowler** UML distilled, Reino Unido, Addison Wesley, 1997.
- [16] **Moisés Belchín** logdart, España, 2014, librodart.es.
- [17] **Roger S. Pressman** Ingeniería de software : un enfoque practico, 7th edición, 2010.
- [18] **Rodríguez Echeverría** Programación Orientada a Objetos, 2004.
- [19] **Sampieri** Metodología de la investigación, 6TH Edición.
- [20] **Victor Robles** MVC (Modelo Vista Controlador) en PHP nativo, 2014, [victorroblesweb.es](http://victorroblesweb.es).

# ANEXOS

# Anexos A

## Entrevista

La entrevista se realizó a los dos Inspectores de Salud de la Unidad de Salud Familiar de San Francisco Gotera, de forma individual, por medio de Google Forms, esto debido a la emergencias de Covid-19 durante el año 2020.

1. **¿El sistema con el que se cuenta actualmente cubre todas las necesidades que se demandan en la Unidad de Salud?**

Ambos inspectores respondieron "No".

2. **¿Que carencias o dificultades le encuentra al sistema que Utilizan actualmente?**

Respuesta 1: Se necesita mucho tiempo, para acceder a la información individual de las personas que se encuentran en el registro de manipulación de alimentos que han recibido capacitación y cumplido otros requisitos.

Respuesta 2: Cuesta que trabaje.

3. **¿Cuáles son los procesos que le toma más tiempo realizar?**

Respuesta 1: Encontrar en una gran lista de nombres de personas participantes, un nombre en particular y sus requisitos de cumplimiento.

Respuesta 2: Los trabajos en excel.

4. **¿Qué tipo de información le gustaría filtrar para mejorar el sistema de búsqueda?**

Respuesta 1: Nombre individual del participante, clasificación si es manipulador formal o informal, nombre del establecimiento, dónde labora, tiene capacitación si o no, que exámenes de laboratorio tiene, y cuáles cumplen con la norma cuales no cumplen la norma.

Respuesta 2: El diagnóstico de manipuladores de alimentos.

5. **¿Cuál es su nivel de conocimiento de computación que posee?**

Respuesta 1: Medio.

Respuesta 2: Bajo.

6. **¿Cuentan con una conexión estable a internet?**

Respuesta 1: Si.

Respuesta 2: No.

7. **¿Cuál es el sistema operativo que manejan actualmente?** Ambos respondieron Windows 10.

8. **¿Cuál es el número de personas que acuden a la Unidad en los periodos de renovación de credenciales?**

Respuesta 1: 1500 o más

Respuesta 2: 2000

9. **¿Quiénes tienen autorización para realizar los registros en el sistema?**

Respuesta 1: Los 2 inspectores de la UCSFI.

Respuesta 2: Mi compañero y yo.

10. **¿En cuantos equipos tienen instalado el sistema actual( de registro de personas del sector de alimentos?**

Respuesta 1: 4 Equipos.

Respuesta 2: 2 Equipos.

# Anexos B

## Encuestas

Encuesta dirigida a los manipuladores de alimentos del municipio de San Francisco Gotera.

1. ¿Qué tipo de alimentos prepara o comercializa?

---

---

---

2. ¿A que sector pertenece como vendedor de alimentos?

*\*Marca sólo un círculo*

Formal.

Informal.

3. ¿Trabaja por su cuenta(negocio propio)?

*\*Marca sólo un círculo*

Si.

No.

4. ¿Cuenta con el permiso para la manipulación de alimentos?

*\*Marca sólo un círculo*

Si.

No.

5. ¿Ha asistido a capacitaciones brindadas por personal de la Unidad de Salud Familiar de San Francisco Gotera?

*\*Marca sólo un círculo*

Varias veces.

Una vez.

Nunca.

6. ¿Se ha realizado alguna verificación de las condiciones de salud e higiene en su negocio (o lugar de trabajo)?

*\*Marca sólo un círculo*

Si.

No.

7. ¿Cuenta con un teléfono inteligente (Smartphone)?

*\*Marca sólo un círculo*

Si.

No.

**Si su respuesta anterior fue Si continuar con la siguiente sección, de lo contrario no continuar y terminar la encuesta.**

8. ¿Que sistema operativo tiene su teléfono?

*\*Marca sólo un círculo*

Android.

iOS.

Otro.

9. ¿Tiene acceso a internet en su teléfono móvil?

*\*Marca sólo un círculo*

Si.

No siempre.

No.

10. ¿Con que frecuencia se conecta a internet?

*\*Marca sólo un círculo*

Menos de 5 veces por semana.

Entre 5 y 10 veces por semana.

Entre 10 y 20 veces por semana.

Más de 20 veces por semana.

11. ¿Alguna vez se le ha olvidado la fecha que le toca renovar su carné de manipulador de alimentos?

*\*Marca sólo un círculo*

Si.

No.

12. ¿Consideraría útil una aplicación móvil que le envié notificaciones para recordarle las fechas de presentar exámenes o renovación de su credencial?

*\*Marca sólo un círculo*

Si.

No.

Tal vez.

13. ¿Qué método utiliza para conectarse a Internet?

*\*Selecciona todos los que correspondan*

Datos móvil.

WIFI.

Otro.

## Anexos C

# Formato de los permisos para manipuladores



REPUBLICA DE EL SALVADOR  
EL SALVADOR  
ENCAMARADO PARA UNIRSE

EL MINISTERIO DE SALUD Y  
LA ALCALDIA MUNICIPAL DE  
SAN FRANCISCO GOTERA  
HACE CONSTAR QUE:

ES MANIPULADOR DE ALIMENTO AUTORIZADO

No. \_\_\_\_\_

FECHA DE VENCIMIENTO: \_\_\_\_\_

DIRECTOR LOCAL MINSAL ALCALDE MUNICIPAL

DADO EN SAN FRANCISCO GOTERA  
ESTE CARNET ES INTRANSFERIBLE

Figura C.0.1: Ejemplo de formato de permiso otorgado a los manipuladores

# Anexos D

## Presentación de sistema y cartas de aceptación U.C.S.F

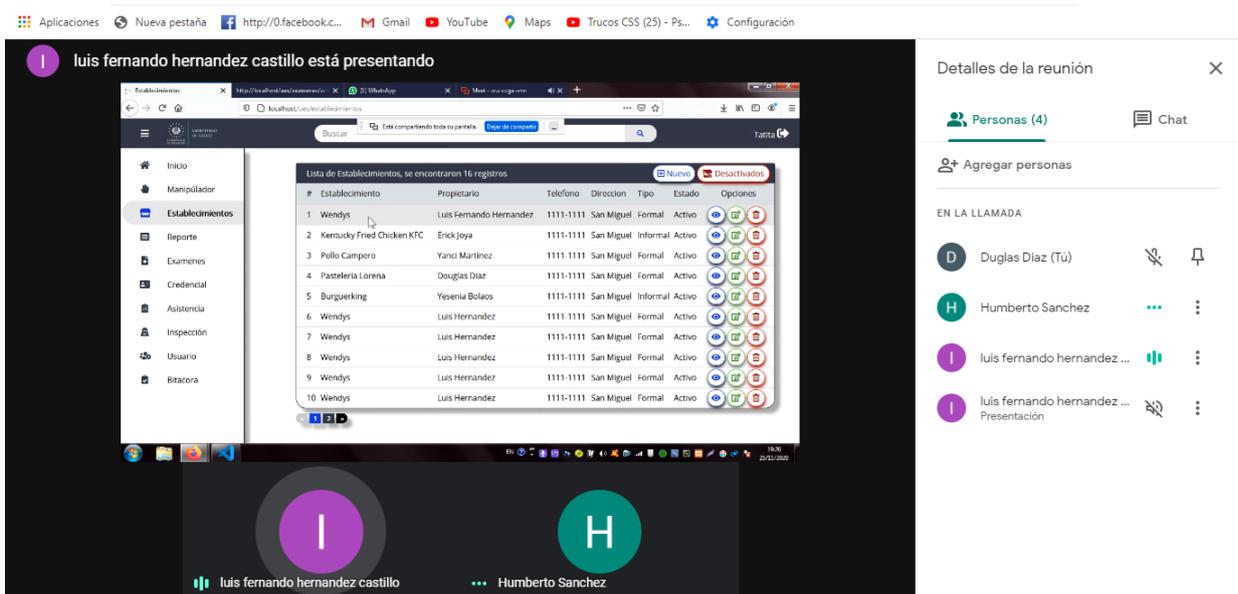


Figura D.0.1: Reunión con inspectores de UCSF



MINISTERIO  
DE SALUD

**Ministerio de Salud de El Salvador  
U.C.S.F de San Francisco Gotera**

San Francisco Gotera, 7 de Febrero de 2020

Señores.

**Universidad de El Salvador  
Facultad Multidisciplinaria Oriental  
Departamento de Ingeniería y Arquitectura**

Reciban un cordial saludo en nombre de la dirección de la U.C.S.F de San Francisco Gotera y del personal Técnico en Saneamiento Ambiental que integra el proyecto de “Capacitación sobre higiene de los alimentos” deseándole éxitos en sus labores.

Tenemos el agrado de dirigirnos a ustedes con la finalidad que permitan a los estudiantes de la carrera de Ingeniería de Sistemas Informáticos, **Luis Fernando Hernández Castillo, Duglas Enrique Díaz Barahona, Erick Adalberto López Joya y María Yanci Martínez García**, para el Desarrollo de un Sistema Informático de Registro, Emisión e Inspección de Credenciales Sanitarias Otorgadas a los Manipuladores de Alimentos en La Unidad Comunitaria de Salud Familiar de San Francisco Gotera en el Departamento de Morazán, ya que mejoraría de gran manera el desarrollo de las actividades e integrar nuestros procesos a tecnologías de vanguardia y por esta razón nos comprometemos a brindar toda la información necesaria que sea requerida por parte de los estudiantes para la realización del sistema.

A continuación, les describimos los principales problemas que tenemos en el manejo de la información de los vendedores por no tener un sistema automatizado que agilice los procesos:

- Pérdida de información.
- Procesos de búsqueda o ejecución lentos.
- Duplicación de información
- Seguridad de los datos es vulnerable.
- A todo esto, le agregamos que se cuente con una cantidad considerable de vendedores que deben ser controlados y el escaso personal no logra suplir tanto el registro como el control de estos.

Agradecemos de antemano su atención y esperando que comprendan y nos tomen en cuenta por la necesidad que sufre la institución y el sector de vendedores de alimentos de San Francisco Gotera con la aprobación de nuestra petición ya que será de gran beneficio y ayuda a la U.C.S.F

Atentamente:

Dra. Sandra de La Paz Laínez Rivera  
DOCTORA EN MEDICINA  
J. V. P. M. N. 10817



**Dra. Sandra de la Paz Laínez de Canales**  
Directora de U.C.S.F de San Francisco Gotera.

**Figura D.0.2:** Carta de solicitud de la Directora



MINISTERIO  
DE SALUD

**Ministerio de Salud de El Salvador  
U.C.S.F de San Francisco Gotera**

San Francisco Gotera, 1 de Diciembre de 2020

**Señores.**

**Universidad de El Salvador  
Facultad Multidisciplinaria Oriental  
Departamento de Ingeniería y Arquitectura**

Reciban un cordial saludo en nombre de nuestra institución, deseándoles éxitos en sus labores cotidianas.

Por medio de la presente, nos dirigimos a ustedes para hacer constar que la **U.C.S.F de San Francisco Gotera** avala el funcionamiento en su totalidad del proyecto: **Desarrollo de un Sistema Informático de Registro, Emisión e Inspección de Credenciales Sanitarias Otorgadas a los Manipuladores de Alimentos en La Unidad Comunitaria de Salud Familiar de San Francisco Gotera en El Departamento de Morazán.**

Presentado por los estudiantes:

María Yanci Martínez García.  
Luis Fernando Hernández Castillo.  
Erick Adalberto López Joya.  
Duglas Enrique Díaz Barahona.

Hemos comprobado el funcionamiento del sistema desarrollado, por lo tanto; aceptamos su funcionamiento y, estando conformes con este, manifestamos su aval para la contribución a las actividades desarrolladas en el área de Saneamiento Ambiental de esta Unidad de Salud.

Atentamente:

  
  
**Dr. Sandra de la Paz Lainez de Canales**  
Directora de U.C.S.F de San Francisco Gotera



**Figura D.0.3:** Carta de aceptación del sistema Directora



MINISTERIO  
DE SALUD

**Ministerio de Salud de El Salvador  
U.C.S.F de San Francisco Gotera**

San Francisco Gotera, 1 de Diciembre de 2020

**Señores.**

**Universidad de El Salvador  
Facultad Multidisciplinaria Oriental  
Departamento de Ingeniería y Arquitectura**

Reciban un cordial saludo en nombre de nuestra institución, deseándoles éxitos en sus labores cotidianas.

Por medio de la presente, nos dirigimos a ustedes para hacer constar que la **U.C.S.F de San Francisco Gotera** avala el funcionamiento en su totalidad del proyecto: **Desarrollo de un Sistema Informático de Registro, Emisión e Inspección de Credenciales Sanitarias Otorgadas a los Manipuladores de Alimentos en La Unidad Comunitaria de Salud Familiar de San Francisco Gotera en El Departamento de Morazán.**

Presentado por los estudiantes:

María Yanci Martínez García.  
Luis Fernando Hernández Castillo.  
Erick Adalberto López Joya.  
Duglas Enrique Díaz Barahona.

Hemos comprobado el funcionamiento del sistema desarrollado, por lo tanto; aceptamos su funcionamiento y, estando conformes con este, manifestamos su aval para la contribución a las actividades desarrolladas en el área de Saneamiento Ambiental de esta Unidad de Salud.

Atentamente:

José Humberto Sánchez  
Inspector Técnico de Saneamiento Ambiental

**Figura D.0.4:** Carta de aceptación del sistema Inspector Técnico