

**UNIVERSIDAD DE EL SALVADOR
FACULTAD MULTIDISCIPLINARIA ORIENTAL
DEPARTAMENTO DE INGENIERÍA Y ARQUITECTURA**



TRABAJO DE GRADO:

IMPACTO DEL SOFTWARE LIBRE EN LAS INSTITUCIONES DE EDUCACIÓN MEDIA DEL MUNICIPIO DE SAN MIGUEL DURANTE EL AÑO 2019 Y CREACIÓN DE PLATAFORMA VIRTUAL PARA EL REGISTRO DE DICHA INFORMACIÓN.

PARA OPTAR AL TÍTULO DE:

INGENIERO DE SISTEMAS INFORMÁTICOS

PRESENTADO POR:

EVER FERNANDO ARGUETA CONTRERAS.
ROBERTO CARLOS CÁRDENAS RAMÍREZ.
GERSON ALEXANDER SANDOVAL GUERRERO.

DOCENTE ASESOR:

INGENIERO LUIS JOVANNI AGUILAR

CIUDAD UNIVERSITARIA ORIENTAL, 11 DE SEPTIEMBRE DE 2020 SAN MIGUEL,
EL SALVADOR, CENTRO AMÉRICA

UNIVERSIDAD DE EL SALVADOR
AUTORIDADES

Msc. ROGER ARMANDO ARIAS

RECTOR

PhD. RAÚL ERNESTO AZCÚNAGA LÓPEZ

VICERECTOR ACADÉMICO

INGENIERO JUAN ROSA QUINTANILLA

VICERECTOR ADMINISTRATIVO

INGENIERO FRANCISCO ALARCÓN

SECRETARIO GENERAL

LICENCIADO RAFAEL HUMBERTO PEÑA MARÍN

FISCAL GENERAL

LICENCIADO LUIS ANTONIO MEJÍA LIPE

DEFENSOR DE LOS DERECHOS UNIVERSITARIOS

FACULTAD MULTIDISCIPLINARIA ORIENTAL

AUTORIDADES

LICENCIADO CRISTÓBAL HERNÁN RÍOS BENÍTEZ

DECANO

LICENCIADO OSCAR VILLALOBOS

VICEDECANO

LICENCIADO ISRAEL LÓPEZ MIRANDA

SECRETARIO INTERINO

LICENCIADO JORGE PASTOR FUENTES CABRERA

DIRECTOR GENERAL DE PROCESOS DE GRADUACIÓN

DEPARTAMENTO DE INGENIERIA Y ARQUITECTURA

AUTORIDADES

INGENIERO JUAN ANTONIO GRANILLO COREAS.

JEFE DEL DEPARTAMENTO.

INGENIERA LIGIA ASTRID HERNANDEZ BONILLA

**COORDINADORA DE LA CARRERA DE INGENIERIA EN SISTEMAS
INFORMATICOS**

INGENIERA MILAGRO DE MARÍA ROMERO DE GARCÍA

COORDINADORA DE PROCESOS DE GRADUACIÓN

TRIBUNAL EVALUADOR

INGENIERO LUIS JOVANNI AGUILAR

JURADO ASESOR

INGENIERO LUDWIN ALDUVÍ HERNÁNDEZ VÁSQUEZ

DOCENTE JURADO CALIFICADOR

INGENIERA LIGIA ASTRID HERNANDEZ BONILLA

DOCENTE JURADO CALIFICADOR

AGRADECIMIENTOS

A DIOS: Por darme la oportunidad de vivir y por haberme dado la sabiduría para poder culminar mis estudios y por fortalecer mi corazón e iluminar mi mente, por haber puesto en mi camino a aquellas personas que han sido mi soporte y compañía durante todo el periodo de estudio.

A MIS PADRES: Elda Marisol Guerrero de Sandoval y José Ángel Sandoval Gómez, quienes son la fuerza y estímulo más grande que la vida me dio. Quienes que con tanto sacrificio han logrado darme la educación y me han cultivado los valores espirituales y morales y por llevarme por el sendero del bien y por todos sus consejos, porque sin su apoyo no hubiese podido concluir mi carrera, no tengo palabras para agradecerles.

A MIS HERMANAS: Iris Leydi Sandoval Guerrero y Keyri Xiomara Sandoval Guerrero, por toda su comprensión y apoyo incondicional en el transcurso de mi superación, gracias por ayudarme a cumplir mi meta.

A MIS COMPAÑEROS DE TESIS: Roberto Carlos Cárdenas Ramírez y Ever Fernando Argueta, por su paciencia y comprensión y el apoyo que me brindaron durante el proceso de tesis y por la confianza que depositaron en mí. Por tomar en cuenta mis opiniones y aportar de la suyas para que el trabajo de investigación sea una realidad. Además a todos mis amigos y amigas que siempre estuvieron apoyándome en las buenas y malas y que siempre me dieron consejos para seguir luchando en el proceso de mi carrera.

Gerson Alexander Sandoval Guerrero.

Doy gracias primeramente a **Dios** por permitirme terminar mis estudios universitarios, por ser mi guía y apoyo en todo momento, por darme la fortaleza para continuar a pesar de las adversidades que se me presentaron en el camino, las palabras no alcanzan para agradecerle, pues él hizo todo posible durante este proceso, por tanto ofrezco a Él este triunfo, para honrar su poderoso nombre siempre. ¡A Dios sea la Gloria!.

Mis agradecimientos a mi **Hijo** Ángel Eliel Argueta Nolasco, el mejor regalo que haya podido recibir de parte de Dios, El mi mayor tesoro y también la fuente más pura de mi inspiración, aunque aun es un bebito pero es mi gran motivación para seguir adelante en todas mis metas y poder brindarle lo mejor.

Expreso mis agradecimientos a una persona muy especial ella es mi **Esposa** Norma Saraí Nolasco de Argueta por brindarme su apoyo incondicional en las diferentes áreas, no fue fácil culminar con éxito este proceso, sin embargo siempre fuiste muy motivadora y esperanzadora, me decías que lo lograría perfectamente. Me ayudaste hasta donde te era posible. Muchas gracias.

Expreso mis agradecimientos especialmente a mi **Padre** Ever Eli Argueta y a mi **Madre** Carmen Contreras por ser un pilar importante en este proceso de formación académica, motivándome a salir adelante. Por su apoyo económico, espiritual, moral y afectivo ante cualquier circunstancia, porque siempre han creído en mí y sobre todo porque me enseñaron que cualquier meta puede lograrse con esfuerzo y sacrificio.

Mis agradecimientos a mis dos **Hermanas** Idalia Marisol Argueta y Yasmin Soraida Argueta, Por todo el apoyo y comprensión que me brindaron en esta etapa, sus consejos, palabras de ánimo y sobre todo saber que en ellas siempre hay una palabra que me motiva a seguir adelante.

Gracias a mis **Compañeros de trabajo de graduación**, **Roberto Carlos Cárdenas** y **Gerson Alexander Sandoval** por su comprensión, paciencia, amistad, apoyo y por permitirme compartir con ellos momentos agradables y difíciles.

Agradezco a mi **Asesor Ing. Luis Jovani Aguilar**, por su paciencia, comprensión, sabiduría y todo el tiempo que nos brindaron para culminar satisfactoriamente nuestro trabajo de graduación.

Ever Fernando Argueta Contreras.

Quiero comenzar agradeciendo a Dios Todopoderoso por el don de la vida, la salud, la sabiduría para conducirme por el camino del bien y por permitirme finalizar este proyecto y acercarme más a la meta de titularme como Ingeniero de Sistemas Informáticos.

A mi madre **Sonia Margarita Ramírez de Cárdenas**, por todo el amor y el apoyo incondicional que me ha brindado a lo largo de mi vida; por su orientación, sus consejos y por estar ahí cuando más lo he necesitado, sin ella, este logro no sería posible.

A mi padre **Humberto Cárdenas Soto**, Por todo el apoyo que me ha brindado, por sus consejos y por motivarme a seguir hacia adelante.

A mi abuela **María Eloísa Ramírez González**, por su amor, su apoyo incondicional, por ser un pilar importante durante todos estos años de estudio.

A mis hermanas **Estefani Margarita** y **Karla Vanessa** por todo su amor y por sus palabras de aliento. A mis tíos y tías **Julio César**, **Berta del Carmen**, **Delmy Aracely**, **Dora Antonia**, **José Nicolás** por todo su afecto y por estar pendientes de mí en cada momento.

Agradecer también a mis compañeros y amigos **Gerson Guerrero** y **Fernando Argueta** por su paciencia, esmero, dedicación y entrega demostrados durante la realización de este proyecto.

A mis amigos **Ing. Jonathan Centeno**, **Wiber Méndez**, **Ing. Javier Paiz**, **Ing. Melvin Flores**, **Ing. Diego Herrera**, **Ing. Jacob Alvarenga**, **Ing. Roxana Blanco**, **Ing. Javier Machuca**, **Rebeca Díaz**, **Juan José**, **Mauricio Vásquez**, **Ana Ruth** por compartir sus conocimientos, por luchar juntos cuando las cosas se ponían difíciles, por su gran apoyo durante toda la carrera y por cada momento bueno y malo que pasamos juntos.

A mi asesor **Ing. Luis Jovani Aguilar** por acompañarnos en este difícil proceso y brindarnos su conocimiento y orientación. A los miembros del tribunal calificador **Inga. Ligia Astrid Hernández** y **Ludwin Alduvi Hernández** por sus observaciones puntuales y la aportación de sus ideas.

Agradecer a todos los docentes que me han brindado su conocimiento durante mi estadía en la universidad, gracias por la paciencia y la dedicación para ayudarme a ser mejor persona, a comprender de mejor manera lo relacionado al amplio campo del saber que implica nuestra carrera. Y en general a toda aquella persona que indirectamente ha estado al pendiente de mí, motivándome a seguir luchando. No me queda más que desear que Dios les bendiga infinitamente dónde sea que se encuentren y les aparte de todo mal. ¡Miles de gracias!

Roberto Carlos Cárdenas Ramírez.

Índice

| | |
|--|----------|
| 1. Planteamiento del problema | 3 |
| 1.1. Definición del tema | 3 |
| 1.2. Situación problemática. | 3 |
| 1.3. Enunciado del problema | 4 |
| 1.4. Justificación | 4 |
| 1.5. Objetivos | 5 |
| 1.5.1. Objetivo General: | 5 |
| 1.5.2. Objetivos Específicos: | 5 |
| 1.6. Alcances | 6 |
| 1.7. Limitaciones | 6 |
| 1.8. Delimitación espacial. | 7 |
| 1.9. Delimitación temporal. | 7 |
| 1.10. Tipo de Investigación | 7 |
| 2. Marco Teórico | 9 |
| 2.1. Software Libre | 9 |
| 2.2. Motivaciones del software libre | 10 |
| 2.3. Historia del software libre y de código abierto | 11 |
| 2.3.1. Antes de 1957 | 11 |
| 2.3.2. Lo que persiste de la época | 15 |
| 2.3.3. Inicios del proyecto GNU y la FSF | 15 |
| 2.3.4. Linux | 16 |
| 2.3.5. Los sistemas BSD libres | 17 |
| 2.3.6. Los años del punto com (finales de los años 1990) | 18 |
| 2.3.7. Nace el código abierto | 19 |
| 2.3.8. Escritorio | 20 |

| | | |
|---------|---|----|
| 2.3.9. | Desarrollo reciente | 22 |
| 2.3.10. | Las cuatro libertades del software libre | 24 |
| 2.3.11. | Formatos abiertos | 26 |
| 2.4. | Comparación con el software de código abierto | 26 |
| 2.4.1. | Implicaciones económico-políticas | 28 |
| 2.4.2. | Modelo de negocio | 29 |
| 2.4.3. | Seguridad relativa | 29 |
| 2.4.4. | Blobs binarios y otros propietarios del software | 30 |
| 2.4.5. | Software libre en la Administración Pública | 30 |
| 2.4.6. | Software libre y gestión sanitaria | 32 |
| 2.5. | El software en Educación | 33 |
| 2.5.1. | Ventajas y desventajas del software libre en la educación | 34 |
| 2.6. | Historia De Gnu/Linux En El Salvador. | 36 |
| 2.7. | Situación actual del software libre en el MINSAL | 37 |
| 2.8. | Antecedentes del software libre en el MINED. | 37 |
| 2.8.1. | Software libre en el Ministerio de Educación | 38 |
| 2.9. | Software Libre como factor de desarrollo en El Salvador | 41 |
| 2.9.1. | Casos de éxito en migración al software libre en educación. | 45 |
| 2.10. | Marco Legal | 46 |
| 2.10.1. | Tipos de licencias | 46 |
| 2.10.2. | Licencias para software | 53 |
| 2.11. | Herramientas para la creación de la plataforma | 94 |
| 2.11.1. | PHP (Hypertext Preprocessor) | 94 |
| 2.11.2. | CodeIgniter | 96 |
| 2.11.3. | Atom | 98 |
| 2.11.4. | MariaDB | 98 |

| | |
|---|------------|
| 3. Formulación del sistema de hipótesis | 101 |
| 3.1. Hipótesis de la investigación | 101 |
| 3.1.1. Hipótesis general. | 101 |
| 3.1.2. Hipótesis Específicas. | 101 |
| 3.2. Operacionalización de Hipótesis. | 102 |
| | |
| 4. Metodología de la investigación | 104 |
| 4.1. Tipo de investigación | 104 |
| 4.1.1. Método de Investigación: cuantitativo | 104 |
| 4.1.2. Investigación descriptiva | 104 |
| 4.1.3. Investigación Tecnológica | 106 |
| 4.2. Técnicas e Instrumentos de Investigación | 107 |
| 4.2.1. Instrumentos Primarios | 107 |
| 4.2.2. Instrumentos Secundarios | 108 |
| 4.3. Universo. | 109 |
| 4.4. Poblacion. | 110 |
| 4.5. Muestra. | 110 |
| 4.5.1. Tamaño de la Muestra. | 111 |
| 4.5.2. Calculo de la Muestra. | 111 |
| 4.5.3. Tipo de muestreo. | 112 |
| 4.6. Técnicas e instrumentos para el análisis de los datos. | 114 |
| 4.6.1. Procedimiento de recolección de datos. | 114 |
| 4.6.2. Procedimiento para procesar e interpretar los datos. | 114 |
| | |
| 5. Tabulación de resultados y comprobación de hipótesis | 117 |
| 5.1. Tabulacion de Encuestas | 117 |
| 5.1.1. Encuesta dirigida a Estudiantes | 117 |
| 5.1.2. Encuesta dirigida a docentes | 126 |

| | | |
|-----------|--|------------|
| 5.1.3. | Encuesta dirigida a encargados de CRA | 137 |
| 5.2. | Comprobación de Hipótesis | 151 |
| 5.2.1. | Hipótesis general. | 151 |
| 5.2.2. | Hipótesis Específicas. | 152 |
| 6. | Conclusiones y Recomendaciones | 155 |
| 6.1. | Conclusiones | 155 |
| 6.2. | Recomendaciones | 156 |
| 6.2.1. | Poblacion estudiantil. | 156 |
| 6.2.2. | Poblacion docente. | 156 |
| 6.2.3. | Coordinadores de aulas informáticas. | 158 |
| 7. | Documentación de la plataforma | 159 |
| 7.1. | Documentación plataforma | 159 |
| 7.1.1. | Introducción | 159 |
| 7.1.2. | Objetivo | 159 |
| 7.2. | Diseño y desarrollo de la plataforma | 159 |
| 7.2.1. | Requerimientos de usuario. | 163 |
| 7.2.2. | Requerimientos del sistema. | 163 |
| 7.2.3. | Composición y función de la plataforma | 165 |
| 7.2.4. | Diagrama de Base de Datos. | 165 |
| 7.2.5. | Diagrama de Clases. | 166 |
| 7.3. | Diccionario de Base de Datos. | 166 |
| 7.3.1. | Tabla detalle de Software | 166 |
| 7.3.2. | Tabla Institución | 167 |
| 7.3.3. | Tabla Registros | 167 |
| 7.3.4. | Tabla Rol | 168 |
| 7.3.5. | Tabla de Software | 168 |

| | |
|---|------------|
| 7.3.6. Tabla Tipo de Institución | 169 |
| 7.3.7. Tabla Tipo de Software | 169 |
| 7.3.8. Tabla Usuario | 170 |
| 7.3.9. Caso de Uso | 171 |
| 8. Bibliografía | 172 |
| 9. Anexos | 174 |
| 9.1. Encuestas | 174 |
| 9.1.1. Encuestas dirigidas a encargados aulas CRA | 174 |
| 9.1.2. Encuestas dirigidas a docentes | 176 |
| 9.1.3. Encuestas dirigidas a estudiantes | 178 |
| 9.2. Manuales | 180 |
| 9.2.1. Manual de instalación | 180 |
| 9.2.2. Manual De Administrador | 189 |
| 9.2.3. Manual De Usuario | 196 |
| 9.3. Glosario | 203 |

Índice de figuras

| | |
|--|-----|
| 1. Tabla de Operacionalización de Hipótesis. | 102 |
| 2. Pregunta 1 | 117 |
| 3. Pregunta 2 | 118 |
| 4. Pregunta 3 | 119 |
| 5. Pregunta 4 | 120 |
| 6. Pregunta 5 | 121 |
| 7. Pregunta 6 | 122 |
| 8. Pregunta 6a | 123 |

| | | |
|-----|----------------------------------|-----|
| 9. | Pregunta 7 | 124 |
| 10. | Pregunta 7 | 125 |
| 11. | Pregunta 1 | 126 |
| 12. | Pregunta 2 | 127 |
| 13. | Pregunta 2.1 | 128 |
| 14. | Pregunta 3 | 129 |
| 15. | Pregunta 4 | 130 |
| 16. | Pregunta 4A | 131 |
| 17. | Pregunta 5 | 132 |
| 18. | Pregunta 6 | 133 |
| 19. | Pregunta 6A | 134 |
| 20. | Pregunta 7 | 135 |
| 21. | Pregunta 7A | 136 |
| 22. | Pregunta 1 | 137 |
| 23. | Pregunta 2 | 138 |
| 24. | Pregunta 2.1 | 139 |
| 25. | Pregunta 2.1 | 140 |
| 26. | Pregunta 3 | 141 |
| 27. | Pregunta 4 | 142 |
| 28. | Pregunta 5 | 143 |
| 29. | Pregunta 6 | 144 |
| 30. | Pregunta 7 | 145 |
| 31. | Pregunta 8 | 146 |
| 32. | Pregunta 9 | 147 |
| 33. | Pregunta 9 | 148 |
| 34. | Diagrama Base de Datos | 165 |
| 35. | Diagrama de Clases | 166 |

| | | |
|-----|---|-----|
| 36. | Tabla detalle de Software | 166 |
| 37. | Tabla Institución | 167 |
| 38. | Tabla Registros | 167 |
| 39. | Tabla Rol | 168 |
| 40. | Tabla Software | 168 |
| 41. | Tabla Tipo de Institución | 169 |
| 42. | Tabla Tipo de Software | 169 |
| 43. | Tabla Usuario | 170 |
| 44. | Caso de uso | 171 |
| 45. | Encuesta a encargados aulas CRA | 174 |
| 46. | Encuesta a encargados aulas CRA | 175 |
| 47. | Encuesta a docentes | 176 |
| 48. | Encuesta a docentes | 177 |
| 49. | Encuesta a estudiantes | 178 |
| 50. | Encuesta a estudiantes | 179 |
| 51. | Pagina de descarga de la xampp | 180 |
| 52. | Descarga para linux | 181 |
| 53. | Guardando la descarga | 181 |
| 54. | Descarga de la aplicación | 182 |
| 55. | Ventana de terminal | 182 |
| 56. | Ventana de terminal | 182 |
| 57. | Ventana de instalación | 183 |
| 58. | Finalizar instalación | 183 |
| 59. | Ejecutando xampp | 184 |
| 60. | localhost | 184 |
| 61. | Sitio oficial | 185 |
| 62. | carpeta de descarga | 185 |

| | | |
|-----|-------------------------------------|-----|
| 63. | sitio oficial atom | 187 |
| 64. | Descarga atom | 187 |
| 65. | Terminal | 188 |
| 66. | Bienvenido | 189 |
| 67. | Login | 190 |
| 68. | Menu | 190 |
| 69. | Registro | 191 |
| 70. | Software registrado | 191 |
| 71. | Usuarios registrados | 192 |
| 72. | Instituciones Registradas | 192 |
| 73. | Acciones | 192 |
| 74. | Agregar Información | 193 |
| 75. | Agregar Información | 193 |
| 76. | Agregar Institución | 194 |
| 77. | Agregar Software | 194 |
| 78. | Perfil | 195 |
| 79. | Cerrar Sesión | 195 |
| 80. | Bienvenido | 196 |
| 81. | Iniciar Sesión | 197 |
| 82. | Registrarse | 197 |
| 83. | Inicio Usuario | 198 |
| 84. | Mis Registros | 198 |
| 85. | Instituciones | 199 |
| 86. | Software registrados | 199 |
| 87. | Acciones | 200 |
| 88. | Registra Información | 200 |
| 89. | Registrar Información | 201 |

| | |
|----------------------------------|-----|
| 90. Registrar Software | 201 |
| 91. Cerrar Sesión | 202 |

Índice de cuadros

| | |
|--|-----|
| 1. Instituciones Educativas del Municipio de San Miguel | 109 |
| 2. Instituciones de Educación Media | 110 |
| 3. Instituciones Educativas en estudio del Municipio de San Miguel | 112 |
| 4. Total de Encuestas por Institución | 113 |
| 5. Pregunta 1 | 117 |
| 6. Pregunta 2 | 118 |
| 7. Pregunta 3 | 119 |
| 8. Pregunta 4 | 120 |
| 9. Pregunta 5 | 121 |
| 10. Pregunta 6 | 122 |
| 11. Pregunta 6a | 123 |
| 12. Pregunta 7 | 124 |
| 13. Pregunta 7a | 125 |
| 14. Pregunta 1 | 126 |
| 15. Pregunta 2 | 127 |
| 16. Pregunta 2.1 | 128 |
| 17. Pregunta 3 | 129 |
| 18. Pregunta 4 | 130 |
| 19. Pregunta 4A | 131 |
| 20. Pregunta 5 | 132 |
| 21. Pregunta 6 | 133 |
| 22. Pregunta 6A | 134 |

| | | |
|-----|--------------------------------------|-----|
| 23. | Pregunta 7 | 135 |
| 24. | Pregunta 7A | 136 |
| 25. | Pregunta 1 | 137 |
| 26. | Pregunta 2 | 138 |
| 27. | Pregunta 2.1 | 139 |
| 28. | Pregunta 2.1 | 140 |
| 29. | Pregunta 3 | 141 |
| 30. | Pregunta 4 | 142 |
| 31. | Pregunta 5 | 143 |
| 32. | Pregunta 6 | 144 |
| 33. | Pregunta 7 | 145 |
| 34. | Pregunta 8 | 146 |
| 35. | Pregunta 9 | 147 |
| 36. | Pregunta 9a | 148 |
| 37. | Entorno de Ejecución | 160 |
| 38. | Requerimientos de hardware | 163 |
| 39. | Requerimientos de software | 163 |

INTRODUCCIÓN

El software libre se refiere al término “free software”, este término en inglés es ambiguo y puede referirse tanto a libertad como a gratitud. En español no existe tal ambigüedad, distinguiéndose claramente el software libre.

Este documento trata sobre el software libre en educación. Está dirigido a docentes en activo de todos los niveles de educación media, a gestores educativos y estudiantes, para conocer los beneficios que el Software libre proporciona a las instituciones educativas del municipio de San Miguel.

La investigación llevará algunos objetivos, el primero es introducir los conceptos clave del software libre, su definición, su origen, implicaciones económicas y ventajas y desventajas del software libre en la educación.

El segundo objetivo es motivar el uso del software libre tanto en las instituciones educativas como en los hogares de los estudiantes de educación media del municipio de San Miguel.

Finalmente, nos gustaría analizar si las instituciones de nivel medio utilizan software libre y conocen las ideas sobre libertad y cooperación que lo sustentan. Los educadores crean materiales formativos y actividades didácticas para los alumnos, que puedan elaborar utilizando herramientas de software libre.

CAPITULO I
PLANTEAMIENTO DEL PROBLEMA

1. Planteamiento del problema

1.1. Definición del tema

Impacto del Software libre en las instituciones de educación media del municipio de San Miguel durante el año 2019 y creación de plataforma virtual para el registro de dicha información.

1.2. Situación problemática.

Actualmente a nivel mundial las tecnologías de la información son elementos importantes para la integración del ser humano en el desarrollo social, económico, cultural y educativo. En el ámbito de la educación, la tecnología ha generado importantes cambios en la aplicación de modelos pedagógicos y en la didáctica en particular. Estas variaciones se presentan tanto en la gestión administrativa, en el salón de clases y en todos los espacios: materiales educativos, laboratorios, acceso a los recursos bibliográficos, etc.

El Software es imprescindible para cualquier sistema informático, puesto que es quien permite la comunicación entre el usuario y la máquina. Podemos notar dos grandes tipos, basándose en la forma de desarrollo del mismo:

Software privativo: Se define como el software que restringe el acceso a su código fuente, el cuál solo se encuentra a disposición del desarrollador y no se permite la libre modificación, adaptación o incluso lectura por parte de terceros.

Software libre: Es todo programa informático cuyo código fuente puede ser estudiado, modificado, y utilizado libremente con cualquier fin y redistribuido sin o con cambios y/o mejoras.

El software privativo genera grandes costos a las organizaciones, por tal razón, las instituciones gubernamentales de El Salvador han optado por Software libre, que aparte de un menor costo económico, ofrece un mejor control, capacidad de adaptar el software a las necesidades, etc.

Las instituciones de educación media de El Salvador no son la excepción a estos cambios pues el Ministerio de educación (MINED) ha iniciado con el proceso de migración hacia el software libre comenzando con la sustitución de paquetes de ofimática privativos, en este caso Microsoft Office a LibreOffice.

1.3. Enunciado del problema

¿Cuál es el impacto de la migración del software privativo al software libre en las instituciones de Educación Media en el municipio de San Miguel?

1.4. Justificación

La siguiente investigación surge de la problemática para conocer cuál es el impacto de plataformas de software libre en los Centros de Recursos de Aprendizaje (CRA) de las instituciones educativas de educación media en el municipio de San Miguel, la cual a su vez involucra aspectos como el acceso a las tecnologías educativas y la inversión del Ministerio de Educación (MINED) en hardware y software educativo.

La implantación de software libre en instituciones educativas hacen posible no solo el ahorro por concepto de licenciamiento de software, sino también permite el ahorro de costos de administración de los recursos informáticos, toda vez que sus bondades técnicas disminuyen la cantidad de horas de trabajo requeridas para mantener operativos dichos recursos informáticos.

GNU-Linux es un sistema operativo de libre licenciamiento que es conocido no solo por su alto nivel de seguridad, sino también por su flexibilidad; lo que permite optimizar el uso del hardware; pudiéndose instalar en equipos con modestos recursos.

Microsoft Office ha sido la principal plataforma en la cual muchos estudiantes de las

instituciones educativas públicos y privados han sido formados a través de las aulas CRA. En vista de la demanda de dichas aulas, que han sido aperturadas en diferentes instituciones educativas del país, ha acrecentado la inversión que se debe realizar en concepto de equipo y licencias educativas.

Según los estudios del MINED señalan que los ahorros en concepto de pago de licencias educativas ascienden a 700 mil dólares anuales (Joma, 2014). Por lo cual el MINED dejó de cancelar en julio de 2014 dicha inversión ya que se ha vuelto insostenible.

El MINED, el 08 de agosto del 2014 emitió una normativa en las unidades administrativas, departamentales de educación e instituciones educativas, dónde dió a conocer la plataforma oficial del sistema operativo y ofimática que se implementaría en dichas instituciones. En donde recomendaba la utilización de Sistema Operativo de escritorio libre para el entorno educativo Linux Max, la cual se deriva de la distribución Debian.

La población a involucrar en el estudio será la educación media de las instituciones educativas públicas y privadas del municipio de San Miguel.

1.5. Objetivos

1.5.1. Objetivo General:

- Identificar el Impacto del uso del Software Libre en las instituciones de educación media, en el municipio de San Miguel.

1.5.2. Objetivos Específicos:

- Determinar las necesidades que llevan a las instituciones de educación media a utilizar software libre en el municipio de San Miguel.

- Conocer las ventajas que surgen en las instituciones de educación media al utilizar software libre en el Municipio de San Miguel.
- Determinar el nivel de impacto social, educativo y económico que inciden en el uso de las tecnologías de software libre en las instituciones de educación media en el municipio de San Miguel.
- Crear una plataforma para el registro de información de software libre.

1.6. Alcances

- Se pretende investigar el impacto que genera el software libre en las instituciones de educación media en el Municipio de San Miguel.
- Elaboración de plataforma que sirva para la administración de la información referente al uso del software libre educación media en el Municipio de San Miguel, para un mayor conocimiento acerca de dicho tema.

1.7. Limitaciones

- El poco conocimiento de los usuarios que en su mayoría están muy familiarizados con los soportes de Microsoft, lo que hace elevar el costo de aprendizaje a la hora implementar software libre.
- Para su configuración se requieren conocimientos previos de funcionamiento del sistema operativo. Es necesario capacitar a los usuarios técnicos para poder configurar este tipo de tecnologías.
- El conocimiento básico del software libre por parte de los alumnos, ya que muchas personas que estudian en las instituciones no conocen estas tecnologías ni sus beneficios.
- Problemas conocidos en compatibilidad de hardware, esto debido al poco uso de estas tecnologías en las instituciones.

- Dificultad de desplazarse para recolectar información en las instituciones.
- Difícil comunicación con los directores, subdirectores y docentes de las instituciones.
- Dificultad para comunicarse con el personal encargado de IT en el MINED central.

1.8. Delimitación espacial.

Esta investigación se llevará a cabo en las instituciones de Educación Media del municipio de San Miguel, del departamento de San Miguel, El Salvador.

1.9. Delimitación temporal.

La investigación y creación de la plataforma virtual tiene un plazo de 10 meses, el cual inicia el primero de febrero hasta el treinta de noviembre del dos mil diecinueve.

1.10. Tipo de Investigación

La investigación descriptiva es la que se utiliza, tal como el nombre lo dice, para describir la realidad de situaciones, eventos, personas, grupos o comunidades que se estén abordando y que se pretenda analizar.

En este tipo de investigación la cuestión no va mucho más allá del nivel descriptivo; ya que consiste en plantear lo más relevante de un hecho o situación concreta.

De todas formas, la investigación descriptiva no consiste únicamente en acumular y procesar datos. El investigador debe definir su análisis y los procesos que involucrará el mismo.

A grandes rasgos, las principales etapas a seguir en una investigación descriptiva son: examinar las características del tema a investigar, definirlo y formular hipótesis, seleccionar la técnica para la recolección de datos y las fuentes a consultar.

CAPITULO II
MARCO TEÓRICO

2. Marco Teórico

2.1. Software Libre

El software libre es todo programa informático cuyo código fuente puede ser estudiado, modificado, y utilizado libremente con cualquier fin y redistribuido con o sin cambios¹. Su definición está asociada al nacimiento del movimiento de software libre, encabezado por Richard Stallman y la consecuente fundación en 1985 de la Free Software Foundation, que coloca la libertad del usuario informático como propósito ético fundamental.²

Proviene del término en inglés *free software*, que presenta ambigüedad (en inglés) entre los significados «libre» y «gratis» asociados a la palabra *free*. Por esto es que suele ser considerado, de manera errónea³, como software gratuito y no en su acepción más precisa como software que puede ser modificado y compartido sin infringir la licencia. El software libre suele estar disponible gratuitamente, o al precio de coste de la distribución a través de otros medios; sin embargo no es obligatorio que sea así, por lo tanto no hay que asociar «software libre» a «software gratuito» (denominado usualmente *freeware*), ya que, conservando su carácter de libre, puede ser distribuido comercialmente. Análogamente, el software gratis o gratuito incluye en ocasiones el código fuente; no obstante, este tipo de software no es «libre» en el mismo sentido que el software libre, a menos que se garanticen los derechos de modificación y redistribución de dichas versiones modificadas del programa. En este sentido, es importante conocer las implicaciones jurídicas que emanan del uso del software libre⁴.

Tampoco debe confundirse software libre con «software de dominio público». Este últi-

¹Free Software Foundation Staff. Consultado el 24 de agosto de 2015.

²Williams, Sam (2 de octubre de 2010). *Free as in Freedom (2.0): Richard Stallman and the Free Software Revolutoin* (2 edición). GNU Press. p.71.

³Fundación por el Software Libre. Archivado desde el original el 19 de noviembre de 2018. Consultado el 21 de noviembre de 2018.

⁴Rodríguez, G. (2008). «El software libre y sus implicaciones jurídicas». *Revista de Derecho*. Scielo. Consultado el 8 de abril de 2018

mo es aquel que no requiere de licencia, pues sus derechos de explotación son para toda la humanidad, porque permite el acceso a todos por igual. Cualquiera puede hacer uso de él, consignando su autoría original. Este software sería aquel cuyo autor lo dona a la humanidad o cuyos derechos de autor han expirado. Si un autor condiciona su uso bajo una licencia, por muy débil que sea, ya no es del dominio público.

Un programa informático es software libre si otorga a los usuarios de manera adecuada las denominadas cuatro libertades (libertad de usar, estudiar, distribuir y mejorar). De lo contrario no se trata de software libre. Existen diversos esquemas de distribución que no son libres, y si bien podemos distinguirlos sobre la base de cuánto les falta para llegar a ser libres, su uso bien puede ser considerado contrario a la ética en todos los casos por igual.

2.2. Motivaciones del software libre

Entre las causas que motivan el software libre se encuentran⁵

- La motivación ética, esgrimida la Free Software Foundation⁶, la cual es un ente que argumenta que el software es conocimiento y debe poderse difundir sin trabas. Su ocultación es una actitud antisocial y la posibilidad de modificar programas es una forma de libertad de expresión, aunque sin olvidar una estructura jerarquizada por la meritocracia.
- La motivación pragmática, defendida por la Open Source Initiative⁷, que argumenta ventajas técnicas y económicas, con respecto a evitar una tragedia de los anticomunes mejorando los incentivos.

Aparte de estas motivaciones, quienes trabajan con software libre suelen hacerlo por muchas otras razones, que van desde la diversión a la mera retribución económica, que es posible

⁵<http://softwarelibre20.blogspot.com/2013/04/motivaciones-del-software-libre.html>

⁶<https://www.fsf.org/es>

⁷<https://opensource.org/>

debido a modelos de negocio sustentables.

La mayoría de los desarrolladores de software libre provienen de países industrializados. El mapa de desarrolladores del Proyecto Debian muestra que la mayoría de desarrolladores se encuentran en Europa y Estados Unidos.

2.3. Historia del software libre y de código abierto

La historia del software libre y de código abierto como lo conocemos actualmente, se remonta a inicios de los años 1980, época en la que la mayoría de software era privativo y surgió la necesidad, por parte de algunos programadores, de crear proyectos que impulsaran la creación de software libre. Cabe mencionar que antes, cuando las primeras computadoras nacieron (y por ende los primeros programas informáticos), el software tenía un modelo de desarrollo cooperativo, similar al de otras ciencias como la física; esto empezó a cambiar en los años 1960 y los años 1970, cuando nacieron las primeras compañías que «privatizaron» su código.

Es importante señalar que el software libre y de código abierto, no debe ser confundido con el llamado "freeware"; el software libre y de código abierto suele ser gratuito, lo que puede llevar a confusión. El FOSS (acrónimo en inglés para free and open source software) también puede ser comprado y vendido. La confusión es aún mayor en países de habla inglesa por la ambigüedad de la palabra free que significa tanto libertad, como gratuidad. Esta es una línea del tiempo acerca de cómo el software libre y de código abierto ha existido y evolucionado desde su concepción.

2.3.1. Antes de 1957

Los conceptos de «código abierto» (fuente abierta) y de «compartición libre de información» existen desde mucho antes de la existencia de computadoras y del software. En los primeros años del desarrollo automotriz, la ALAM (siglas en inglés para Association of

Licensed Automobile Manufacturers en español «Asociación de fabricantes de automóviles autorizados»), un grupo de monopolizadores de capital, eran propietarios de los derechos de la patente de un motor de gasolina de dos tiempos archivada por George B. Selden. Con el control de esta patente, fueron capaces de controlar la industria e intentaron forzar a los fabricantes de automóviles a construir vehículos de acuerdo con sus especificaciones (automóviles pesados, lujosos y caros; diseñados para la gente adinerada). Los fabricantes a quienes otorgaban una licencia, tenían que pagarles una comisión, así como un porcentaje de las ganancias brutas. En el marco del Auto Show de Chicago de 1905, un grupo de fabricantes de automóviles independientes que incluía a Ford Motor Company, REO Motor Car Company y Maxwell-Briscoe, formaron una nueva organización: la «Asociación de Fabricantes de Carros de Motor de Estados Unidos» (AMCMA por sus siglas en inglés) para pelear de manera colectiva contra la patente de Selden y el monopolio de la ALAM. Tras algunos contratiempos, incluyendo la pérdida de una demanda legal y la disolución de la AMCMA en 1909, Henry Ford apeló la decisión, ganándola en 1911. La resolución fue que la patente de Selden, misma que era para motores de 2 tiempos, no cubría los motores que estaban siendo utilizados por la mayoría de fabricantes de aquel tiempo (que eran motores de cuatro tiempos, basados en el diseño de Nikolaus Otto). Resultando en que la patente de Selden quedó virtualmente sin ningún valor y en la disolución de la ALAM en una nueva asociación que posteriormente se convertiría en la Asociación de Fabricantes de Vehículos de Motor. La nueva sociedad entabló un nuevo acuerdo de licencia múltiple entre todos los fabricantes de automóviles de Estados Unidos: aunque cada compañía desarrollaría tecnologías y presentaría patentes, estas patentes serían compartidas abiertamente y sin intercambio monetario alguno entre todos los fabricantes. Hasta el momento de la participación de EE.UU. en la segunda guerra mundial, 92 patentes de Ford fueron utilizadas libremente por otros fabricantes y a su vez fueron usadas 515 patentes de otras compañías, todo sin demandas legales o intercambio monetario de ningún tipo.

Algunos de los principios fundamentales del software libre, surgieron de filosofías de cooperación y sinceridad, ampliamente establecidas en los campos académicos y de la investigación científica (en este caso, ciencia computacional).

Algunas comunidades de software, que pueden compararse con la comunidad del software libre actual, existieron por mucho tiempo antes que el movimiento del software libre y el término “software libre“ existieran. Según Richard Stallman, la comunidad que compartía software en el MIT existió por muchos años antes de involucrarse (con el movimiento) en 1971.

En los años 1958 y 1959 y en los años 1960 casi todo el software era producido en masa por académicos e investigadores corporativos en colaboración y no era visto así mismo como un producto. Los sistemas operativos eran ampliamente distribuidos y mantenidos por comunidades de usuarios. El código fuente, la versión legible para humanos del software, era distribuido con el software porque los usuarios modificaban el software ellos mismos para arreglar errores de programación o agregar nuevas funcionalidades. El código fuente del sistema operativo de un mainframe⁸ de IBM, el ACP (acrónimo de Airline Control Program), de 1967 era distribuido de manera muy parecida al del software libre.

Se formaron grupos de usuarios como los de IBM 701, llamados «SHARE», y los de Digital Equipment Corporation, llamados "DECUS"; formados para facilitar el intercambio de software. Así en esa era, el software era libre en cierto sentido, no por algún esfuerzo realizado por usuarios o desarrolladores de software, sino porque el software era desarrollado por la comunidad de usuarios. Para finales de los años 1960 se empezó a producir un cambio: al evolucionar los sistemas operativos y los compiladores de lenguaje de programación, los costos del software de las compañías se incrementaron dramáticamente. Una naciente,

⁸<http://www.carlospes.com/minidiccionario/mainframe.php>

aunque después creciente industria del software estaba compitiendo con el software incluido al hardware de los fabricantes (el costo de los programas precargados estaba incluido en el precio del hardware), las máquinas arrendadas requerían soporte del software al tiempo que no se proveía la renta del software, y algunos clientes capaces de satisfacer de mejor manera sus propias necesidades, no quería que el costo del software del fabricante fuera incluido con el costo del producto. En la demanda, realizada el 17 de enero de 1969, por parte del “Estado Unido contra IBM“ por violaciones al derecho de competencia (antitrust) el gobierno de los Estados Unidos dictaminó que el software atado (incluido o precargado) era anti-competitivo. Mientras algún software continuaba publicándose de manera gratuita, existía una creciente cantidad de software que era vendido únicamente.

En los años 1970 AT&T distribuía las primeras versiones de UNIX⁹ sin costo al gobierno y a investigadores académicos, pero estas versiones no permitían su distribución ni la distribución de versiones modificadas, y por ende no eran software libre en la acepción moderna de la frase.

A finales de los años 1970 y principios de los años 1980, los vendedores de computadoras y compañías de software empezaron a cobrar por licencias de software de manera rutinaria, comercializándolas como “Productos Informáticos“ e imponiendo restricciones legales a los nuevos desarrollos de software, ahora vistos como activos, a través de derechos de autor, marcas registradas y contratos de arrendamiento. En 1976, Bill Gates marcó el gran cambio de era cuándo escribió, su ahora famosa Carta abierta a los aficionados, mandando el mensaje de que lo que los hackers llaman «compartir» era, en sus palabras, «robar». Por ejemplo, en 1979, AT&T empezó a hacer cumplir sus licencias restrictivas cuando la compañía decidió que podrían generar utilidades vendiendo el sistema Unix.

⁹Es un sistema operativo portable, multitarea y multiusuario; desarrollado en 1969 por un grupo de empleados de los laboratorios Bell de AT&T,

La llegada de Usenet¹⁰ al principio de los años 1980 conectó, aún más, a la comunidad de programadores y facilitó a los programadores, una forma más simple de compartir su software y de contribuir al software que otros hayan escrito.

2.3.2. Lo que persiste de la época

Mucho software libre que fue desarrollado en los años 1970 y en los años 1980 aún continúa siendo utilizado, incluyendo el simulador de circuitos integrados SPICE, TeX (desarrollado por Donald Knuth) y el sistema de ventanas X. El sistema de ventanas W sentó las bases para el sistema de ventanas X, pero tenía muchas diferencias fundamentales. El desarrollo del sistema de ventanas X fue simultáneo al del proyecto GNU, sin embargo GNU de ninguna manera fue responsable del desarrollo del sistema de ventanas X.

2.3.3. Inicios del proyecto GNU y la FSF

En 1983, Richard Stallman lanzó el proyecto GNU para escribir un sistema operativo completo libre de restricciones para el uso, modificación y distribuirlo con o sin mejoras. Uno de los incidentes particulares que lo motivaron a esto fue el caso de una molesta impresora que no podía ser arreglada porque el código fuente no era revelado. Otro posible evento de inspiración para el proyecto GNU y su manifiesto fue el desacuerdo entre Stallman y Symbolics, Inc. sobre el acceso a las actualizaciones, por parte del MIT, que Symbolics había realizado a su máquina Lisp, la cual estaba basada en código del MIT. Poco tiempo después de su lanzamiento, acuñó el término "software libre" para promover el concepto y fundó la Free Software Foundation. Una definición de software libre fue publicada en febrero de 1986.

En 1989, fue publicada la primera versión de la Licencia Pública General de GNU. En 1991 se publicó la ligeramente actualizada la versión 2 de la licencia. En 1989, algunos desarrolladores de GNU crearon la compañía Cygnus Solutions.

¹⁰Usenet es el acrónimo de Users Network (Red de usuarios), consistente en un sistema global de discusión en Internet

El núcleo (kernel) del proyecto GNU, posteriormente llamado GNU Hurd, fue retrasado continuamente, pero la mayor parte de los demás componentes fueron completados para 1991. Algunos de éstos, especialmente la Colección de Compiladores de GNU, se han convertido en líderes del mercado por méritos propios. El Depurador de GNU y GNU Emacs también fueron éxitos notables.

2.3.4. Linux

El núcleo Linux iniciado por Linus Torvalds, fue liberado para poder ser modificado libremente en 1991. La licencia inicial, no fue exactamente una licencia de software libre, sin embargo la versión 0.12 lanzada en febrero de 1992, fue licenciada nuevamente por Torvalds bajo los términos de la Licencia Pública General de GNU. Así como Unix en su tiempo, el núcleo de Torvalds atrajo la atención de programadores voluntarios.

Hasta este punto, la falta de núcleo del proyecto GNU significaba la no existencia de un sistema operativo libre completo. El desarrollo del núcleo de Linus Torvalds llenó ese último hueco. La combinación del casi terminado sistema operativo de GNU y el núcleo Linux resultó en el primer sistema operativo completo de software libre.

Entre las distribuciones Linux, Debian GNU/Linux, iniciado por Ian Murdock en 1993, es notorio por estar comprometido explícitamente con los principios de GNU y la FSF del software libre. Los principios de los desarrolladores de Debian están expresados en el Contrato Social de Debian. Desde su inicio, el proyecto Debian ha estado íntimamente ligado con la FSF, y de hecho fue patrocinado por la FSF durante un año, de 1994 a 1995. En 1997, el antiguo líder del proyecto Debian, Bruce Perens ayudó también a fundar Software in the Public Interest, una fundación y organización de soporte sin ánimo de lucro para varios proyectos de software libre. Sin embargo Debian ya no se considera software libre por la FSF y el proyecto GNU ya que el núcleo Linux incluido con Debian contiene partes privativas

además de ofrecer repositorios con software no libre.

GNU/Linux (o simplemente Linux) continúa siendo software libre desarrollado por programadores voluntarios, pero también muchas compañías ofrecen productos personalizados basados en el núcleo Linux así como distribuciones con soporte comercial.

El nombre del sistema operativo continúa generando controversia dentro de la comunidad del software libre, por un lado el proyecto GNU y otros grupos de usuarios piden que el sistema sea llamado GNU/Linux o GNU+Linux argumentando que la mayoría de sistemas basados en el núcleo Linux son derivaciones del sistema operativo GNU, que empezaron a desarrollarlo 7 años antes que Linus Torvalds publicara su núcleo y que "sobre la base de lo justo, deberían tener por lo menos una mención equivalente". Por el otro lado, los grupos que apoyan el nombre "Linux" para referirse al sistema operativo completo y no solo a su núcleo, argumentan que el nombre "Linux" es más reconocido, mejor recibido y más práctico; también argumentan que probablemente Richard Stallman se siente frustrado porque Torvalds "ha recibido toda la gloria" y que por esta razón gasta mucha energía en nomenclaturas. Es importante señalar que la mayoría de versiones del sistema operativo basado en Linux, contienen muchas otras partes además de las desarrolladas por el proyecto GNU y el núcleo Linux, el ejemplo más representativo de estos componentes ajenos a GNU y Linux es el sistema de ventanas X.

2.3.5. Los sistemas BSD libres

BSD son las iniciales de Berkeley Software Distribution (en español, Distribución de Software Berkeley) y se utiliza para identificar un sistema operativo derivado del sistema Unix nacido a partir de los aportes realizados a ese sistema por la Universidad de California en Berkeley.

El primer sistema Unix en Berkeley fue el PDP-11, que fue instalado en 1974, después en

1977 Bill Joy, entonces un estudiante de grado en Berkeley, grabó y envió cintas del primer Berkeley Software Distribution (BSD). En 1983 fue publicado BSD 2.9 que incluye código de BSD 4.1c y fue la primera distribución considerada como un sistema operativo completo (Una modificación de Unix 7).

Hasta este punto, todas las versiones de BSD incluían código privativo de AT&T, lo que requería el pago de licencias haciendo al software muy caro y generó la exigencia por un sistema libre por parte de entidades externas. Esto se consiguió con Network Tape 1 (Net/1), lanzada en 1989 y creada sin código propietario de AT&T que fue libremente distribuida sobre los términos de la permisiva licencia BSD. Más adelante, otros proyectos adoptaron la filosofía de software libre y adoptaron la licencia BSD, así en junio de 1991 nació Network Tape 2 (Net/2).

Poco tiempo después, BSDi (Berkeley Software Design Inc.) tuvo un problema legal con AT&T, propietarios de los derechos de System V y la marca Unix. El pleito fue archivado en 1992, bajo la prescripción de no distribuir Net/2 hasta que la validez de las demandas pudiera ser determinada.

Cuando se resolvió la demanda legal que USL levantó en contra de BSDi por parte de la corte en 1993, las distribuciones derivadas de 386BSD, FreeBSD y NetBSD fueron liberadas como software libre. Más tarde en 1995, OpenBSD fue publicada, una distribución derivada de NetBSD. También existen otras versiones derivadas en la actualidad, entre las que destacan SunOS y Mac OS X.

2.3.6. Los años del punto com (finales de los años 1990)

En la segunda mitad de los años 1990, cuándo muchas compañías basadas en web empezaban a operar; el software libre se volvió la opción popular de los servidores web. El servidor

HTTP Apache se volvió en el software de servidor web más utilizado, título que aún conserva a finales de la primera década del siglo XXI. Estos servidores web, se han llegado a conocer como "sistemas LAMP" son sistemas que vienen con un "paquete" de software inicial que incluye al núcleo Linux como base, a Apache como proveedor de servicios web, la base de datos MySQL para archivo de datos y al lenguaje de programación PHP uniéndolos a todos.

2.3.7. Nace el código abierto

En 1997, Eric Raymond publicó *La catedral y el bazar*, un ensayo dónde analiza a la cultura hacker y los principios del software libre. El escrito recibió particular atención al principio de 1998 y fue un factor estimulante para que Netscape Communications Corporation liberara su popular suite de internet Netscape Communicator como software libre. Este código, es mejor conocido en la actualidad como Firefox y Thunderbird.

Este acto por parte de Netscape, impulsó a Raymond y otros a buscar cómo llevar los principios del software libre a la industria del software comercial. Concluyeron que el activismo social de la FSF no era atractivo para las compañías como Netscape y buscaron maneras de volver a inventar el movimiento del software libre para enfatizar el potencial comercial de compartir código fuente. El nuevo nombre que escogieron fue «código abierto» (open source en inglés) y rápidamente Bruce Perens, el editor Tim O' Reilly, Linus Torvalds y otros firmaron a favor del nuevo nombre. La Open Source Initiative (Iniciativa por el Código Abierto) fue fundada en febrero de 1998 para fomentar el nuevo término y para difundir los principios del «código abierto».

Sin embargo, Richard Stallman y la FSF objetaron enérgicamente al enfoque de la nueva organización. Sintieron que, con la estrecha visión del código fuente, la OSI estaba sepultando los valores filosóficos y sociales del software libre y que estaba escondiendo el problema de la libertad de los usuarios de computadoras. Aun así, Stallman seguía manteniendo la

postura de que los usuarios de ambos términos eran aliados en contra del software privativo.

En septiembre de 2000, Sun Microsystems lanzó la suite ofimática StarOffice 5.2 como software libre bajo los términos de la Licencia Pública General Reducida de GNU. La versión libre de la suite fue renombrada a OpenOffice.org (comúnmente abreviado ".oOo") y coexiste con StarOffice, la versión 1.0 de OpenOffice.org fue lanzada el 30 de abril de 2002; el producto y el código fuente abierto coexisten desde el 13 de octubre de 2000.

2.3.8. Escritorio

X se ha convertido en el sistema de ventanas defecto del software libre.

KDE fue fundado en 1996 por Matthias Ettrich. En aquel tiempo, encontraba problemático el número de inconsistencias en las aplicaciones Unix. Propuso un nuevo entorno de escritorio pero también quería que fuera un escritorio fácil de usar. Su primer mensaje en Usenet generó mucho interés.

Ettrich decidió usar la biblioteca Qt para el proyecto KDE. En ese tiempo, Qt no gozaba de una licencia de software libre. Miembros del proyecto GNU sintieron preocupación por tal uso de una biblioteca para construir un entorno de escritorio libre. En agosto de 1997, iniciaron dos proyectos en respuesta a KDE: la biblioteca Harmony (un replazo libre de las bibliotecas Qt) y GNOME (un entorno de escritorio diferente, sin Qt y construido enteramente de software libre). Se escogió a GTK+ como la base para GNOME en lugar de las bibliotecas de Qt.

En noviembre de 1998, la biblioteca Qt fue publicado bajo la licencia libre/código abierto Licencia Pública Q (QPL). Pero el debate acerca de la compatibilidad con la GPL continuó. En septiembre de 2000, Trolltech creo la versión Unix de las bibliotecas Qt bajo los términos de la GPL, en adición de la QPL, lo cual ha eliminado las preocupaciones de la Free Software

Foundation (Fundación para el Software Libre).

Ambos entornos de escritorio, KDE y GNOME ahora participan en freedesktop.org, un esfuerzo por estandarizar la interoperabilidad de escritorios Unix, no obstante aún existe cierta competencia entre ellos.

GNOME

GNOME es un entorno de escritorio e infraestructura de desarrollo para sistemas operativos GNU/Linux, Unix y derivados Unix como BSD o Solaris; compuesto enteramente de software libre.

El proyecto fue iniciado por los programadores mexicanos Miguel de Icaza y Federico Mena en agosto de 1997 y forma parte oficial del proyecto GNU. Nació como una alternativa a KDE bajo el nombre de GNU Network Object Model Environment (Entorno de Modelo de Objeto de Red GNU). Actualmente, incluyendo al español, se encuentra disponible en 166 idiomas.

GNOME está disponible en las principales distribuciones GNU/Linux, incluyendo Fedora, Debian, Ubuntu, Red Hat Linux, CentOS, Oracle Linux, Arch Linux y Gentoo¹¹.

GNOME provee un gestor de ventanas «intuitivo y atractivo» y una plataforma de desarrollo para crear aplicaciones que se integran con el escritorio. El Proyecto pone énfasis en la simplicidad, facilidad de uso y eficiencia. Tiene como objetivo la libertad para crear un entorno de escritorio que siempre tendrá el código fuente disponible para reutilizarse bajo una licencia de software libre.

Desde GNOME 2, el enfoque fue puesto en la productividad. Con este fin, se crearon las Pautas de Interfaz Humana de GNOME (Human Interface Guidelines, HIG). Todos los programas de GNOME comparten un estilo coherente de interfaz gráfica de usuario (GUI), pero no están limitados a los mismos widgets de GUI. Por el contrario, el diseño de la GUI de

¹¹www.gnome.org (en inglés). Consultado el 1 de abril de 2018.

GNOME está guiado por conceptos que se describen en GNOME HIG, que a su vez depende de la ergonomía cognitiva. Después de HIG, los desarrolladores pueden crear programas de GUI de alta calidad, consistentes y utilizables, ya que abordan todo, desde el diseño de la GUI hasta el diseño recomendado de widgets basado en píxeles.

2.3.9. Desarrollo reciente

El 8 de mayo de 2007, Sun Microsystems liberó el Java Development Kit con el nombre OpenJDK bajo la Licencia Pública General de GNU. Una parte de las clases de la biblioteca (el 4%) no pudieron ser publicadas como código abierto por estar publicadas bajo licencia de otros organismos y fueron incluidos como complementos binarios. Por esta razón, en junio de 2007, Red Hat lanzó IcedTea para resolver el problema de impedimento de los componentes con la implementación de los equivalentes de GNU Classpath. Desde su lanzamiento, la mayoría de impedimentos han sido resueltos, dejando pendiente el motor de audio y el sistema de gestión del color únicamente.

En los años 2000, se ha buscado expandir la libertad de usar, compartir, modificar y publicar más allá el código fuente y el software, implementando los principios del software libre a la documentación que acompaña al software así como otro tipo de información: libros, manuales, fotografías, vídeo, audio, etcétera. Para lograrlo, se han creado licencias y movimientos que buscan la libertad de estudiar y compartir virtualmente cualquier tipo de contenido.

En marzo de 2000 se publicó, por parte de la FSF, la versión 1.1 de la Licencia de Documentación Libre de GNU (GFDL por sus siglas en inglés) con el propósito de hacer que los manuales, libros de texto o cualquier documento escrito sean libres.

En 2001, se fundó Creative Commons, una corporación sin ánimo de lucro dedicada a

hacer el proceso de compartir y construir a partir del trabajo de otros más fácil, de manera consistente con las reglas del copyright. Facilitan licencias libres y otras herramientas para publicar trabajos con la libertad que el autor quiere otorgar a otros, para que puedan compartir, modificar y usar comercialmente su obra; o cualquier combinación de estas libertades.

El 15 de enero de 2001 es lanzada Wikipedia, un proyecto de la Fundación Wikimedia (una organización sin ánimo de lucro) para construir una enciclopedia libre y políglota. Wikipedia en un inicio publicó todo su contenido bajo la Licencia de Documentación Libre de GNU únicamente, pero a partir del 1 de agosto de 2009 cambió a «Creative Commons Atribución Compartir Igual 3.0». Para que esto fuera posible, la FSF actualizó la GFDL en noviembre de 2008, para permitir la migración de licencia por parte de Wikimedia. Aunque es necesario mencionar que algunas versiones de Wikipedia, como la inglesa, tiene contenido no libre.

Linux cumplió 25 años en 2016 y su difusión en la industria ha sido impensada. Según datos de la Linux Foundation, el kernel se utiliza:

- en los sistemas de control de tráfico ferroviario y aéreo.
- en más de 1,3 millones de teléfonos inteligentes Android activados cada día.
- en submarinos nucleares y sistemas de defensa.
- en la mayoría de los 700.000 televisores vendidos cada día.
- en el software de secuenciación del ADN.
- en la iniciativa One Laptop Per Child.
- en la Estación Espacial Internacional.
- en el Gran Colisionador de Hadrones.

- en 8 de cada 10 de las transacciones financieras globales.
- en el 98 % de los súper ordenadores alrededor del mundo.
- en los servidores de Google, Facebook y Amazon.

2.3.10. Las cuatro libertades del software libre

De acuerdo con la definición establecida por Richard Stallman, un software es «libre» cuando garantiza las siguientes libertades:

| Libertad | Descripción |
|---|--|
| 0 | la libertad de usar el programa, con cualquier propósito (uso). |
| 1 | la libertad de estudiar cómo funciona el programa y modificarlo, adaptándolo a las propias necesidades (estudio). |
| 2 | la libertad de distribuir copias del programa, con lo cual se puede ayudar a otros usuarios (distribución). |
| 3 | la libertad de mejorar el programa y hacer públicas esas mejoras a los demás, de modo que toda la comunidad se beneficie (mejora). |
| Las libertades 1 y 3 requieren acceso al código fuente, porque estudiar y modificar software sin su código fuente es muy poco viable. | |

Ciertos teóricos usan este cuarto punto (libertad 3) para justificar parcialmente las limitaciones impuestas por la licencia GNU GPL frente a otras licencias de software libre (ver Licencias GPL)¹². Sin embargo el sentido original es más libre, abierto y menos restrictivo que el que le otorga la propia situación de incompatibilidad, que ha sido resuelta a cierto grado en versión 3 de la licencia GNU GPL.

Tanto la Open Source Initiative¹³ como la Free Software Foundation¹⁴, mantienen en sus webs oficiales, listados de las licencias de software libre que aprueban.

¹²European Parliament. «Legal aspects of free and open source software,». Inglés. Consultado el 30 de marzo de 2018.

¹³Open Source Initiative. «OSI licenses» (en inglés).

¹⁴Free Software Foundation. «Licencias de Software Libre».

El término software no libre se emplea para referirse al software distribuido bajo una licencia de software más restrictiva que no garantiza estas cuatro libertades. Las leyes de la propiedad intelectual reservan la mayoría de los derechos de modificación, duplicación, y redistribución, para el dueño del copyright; el software dispuesto bajo una licencia de software libre rescinde específicamente la mayoría de estos derechos reservados.

La definición de software libre no contempla la cuestión del precio; un eslogan frecuentemente usado es «libre como en libertad, no como en cerveza gratis» o en inglés «Free as in freedom, not as in free beer» (aludiendo a la ambigüedad del término inglés free), y es habitual ver a la venta CD de software libre como distribuciones Linux. Sin embargo, en esta situación, el comprador del CD tiene el derecho de copiarlo y redistribuirlo. El software gratis puede incluir restricciones que no se adaptan a la definición de software libre —por ejemplo, puede no incluir el código fuente, puede prohibir explícitamente a los distribuidores recibir una compensación a cambio, etc.

Para evitar la confusión, algunas personas utilizan los términos «libre» (software libre) y «gratis» (software gratis) para evitar la ambigüedad de la palabra inglesa free. Sin embargo, estos términos alternativos son usados únicamente dentro del movimiento del software libre, aunque están extendiéndose lentamente hacia el resto del mundo. Otros defienden el uso del término open source software (software de código abierto). La principal diferencia entre los términos «open source» y «free software» es que este último tiene en cuenta los aspectos éticos y filosóficos de la libertad, mientras que el open source se basa únicamente en los aspectos técnicos.

En un intento por unir los mencionados términos que se refieren a conceptos semejantes, se está extendiendo el uso de la palabra «FLOSS» con el significado de free/libre and open source software e, indirectamente, también a la comunidad que lo produce y apoya.

2.3.11. Formatos abiertos

Los formatos abiertos permiten al software libre mantener sus cuatro libertades y la libre difusión de todo el código y formatos utilizados, su distribución y estudio, debido a esto, los creadores de software libre desarrollan a la vez de programas libres, formatos libres para estos programas o utilizan formatos libres ya creados anteriormente.

Los formatos libres permiten a los usuarios poder trabajar con programas libres; aunque al ser libres pueden ser implementados y utilizados en cualquier programa sea cerrado o no. Algunas compañías, como Microsoft, no suelen utilizar formatos libres en sus programas, no por impedimento si no por falta de voluntad de implementar formatos abiertos en sus programas, aun así los usuarios pueden instalar software libre en sus sistemas para trabajar con estos formatos.

2.4. Comparación con el software de código abierto

Aunque en la práctica el software de código abierto y el software libre comparten muchas de sus licencias, la Free Software Foundation opina que el movimiento del software de código abierto es filosóficamente diferente del movimiento del software libre¹⁵. Apareció en 1998 con un grupo de personas, entre los que cabe destacar a Eric S. Raymond y Bruce Perens, que formaron la Open Source Initiative (OSI). Ellos buscaban darle mayor relevancia a los beneficios prácticos del compartir el código fuente, e interesar a las principales casas de software y otras empresas de la industria de la alta tecnología en el concepto. Por otro lado, la Free Software Foundation y Richard Stallman prefieren plantear el asunto en términos éticos empleando el término "software libre"¹⁶.

Los defensores del término «código abierto» afirman que este evita la ambigüedad del término en ese idioma que es free en free software. El término «código abierto» fue acuñado

¹⁵«Open Source» (código fuente abierto) en el libro Software libre para una sociedad libre de Stallman

¹⁶www.gnu.org. Consultado el 21 de diciembre de 2016.

por Christine Peterson del instituto de investigación Foresight Institute, y se registró para actuar como marca registrada el término en inglés, para los productos de software libre.

Mucha gente reconoce el beneficio cualitativo del proceso de desarrollo de software cuando los desarrolladores pueden usar, modificar y redistribuir el código fuente de un programa. El movimiento del software libre hace especial énfasis en los aspectos morales o éticos del software, viendo la excelencia técnica como un producto secundario de su estándar ético. El movimiento de código abierto ve la excelencia técnica como el objetivo prioritario, siendo la compartición del código fuente un medio para dicho fin. Por dicho motivo, la Fundación del Software Libre se distancia tanto del movimiento de código abierto como del término «Código Abierto».

Puesto que la OSI solo aprueba las licencias que se ajustan a la Open Source Definition (definición de código abierto), la mayoría de la gente lo interpreta como un esquema de distribución, e intercambia libremente los términos «código abierto» y «software libre». Aunque existen importantes diferencias filosóficas entre ambos términos, especialmente en términos de las motivaciones para el desarrollo y el uso de tal software, raramente suelen tener impacto en el proceso de colaboración.

Aunque el término «código abierto» elimina la ambigüedad de libertad frente a precio (en el caso del inglés), introduce una nueva: entre los programas que se ajustan a la definición de código abierto, que dan a los usuarios la libertad de mejorarlos, y los programas que simplemente tiene el código fuente disponible, posiblemente con fuertes restricciones sobre el uso de dicho código fuente. Mucha gente cree que cualquier software que tenga el código fuente disponible es de código abierto, puesto que lo pueden manipular (un ejemplo de este tipo de software sería el popular paquete de software gratuito Graphviz, inicialmente no libre pero que incluía el código fuente, aunque luego AT&T le cambió la licencia). Sin embargo,

mucho de este software no da a sus usuarios la libertad de distribuir sus modificaciones, restringe el uso comercial, o en general restringe los derechos de los usuarios.

2.4.1. Implicaciones económico-políticas

Una vez que un producto de software libre ha empezado a circular, rápidamente está disponible a un costo muy bajo. Al mismo tiempo, su utilidad no decrece. El software, en general, podría ser considerado un bien de uso inagotable, tomando en cuenta que su costo marginal es pequeñísimo y que no es un bien sujeto a rivalidad (la posesión del bien por un agente económico no impide que otro lo posea).

Puesto que el software libre permite el libre uso, modificación y redistribución, a menudo encuentra un hogar entre usuarios para los cuales el coste del software no libre es a veces prohibitivo. También es sencillo modificarlo localmente, lo que permite que sean posibles los esfuerzos de traducción a idiomas que no son necesariamente rentables comercialmente.

La mayoría del software libre se produce por equipos internacionales que cooperan a través de la libre asociación. Los equipos están típicamente compuestos por individuos con una amplia variedad de motivaciones, y pueden provenir tanto del sector privado, del sector voluntario o del sector público. Existen muchas posturas acerca de la relación entre el software libre y el actual sistema político-económico:

- Algunos consideran el software libre como un competidor contra el centralismo en empresas y gobiernos, una forma de orden espontáneo o de anarquismo práctico.
- Algunos consideran el software libre como una forma de trabajo colaborativo en un modelo de mercado, tal como se había planteado el cooperativismo.
- Algunos comparan el software libre a una economía del regalo, donde el valor de una persona está basado en lo que esta da a los demás, sin que incurra valor monetario formal de por medio.

- Grupos como Oekonux¹⁷ e Hipatia¹⁸ consideran que todo debería producirse de esta forma y que este modelo de producción no se limita a reemplazar el modelo no libre de desarrollo del software. La cooperación basada en la libre asociación puede usarse y se usa para otros propósitos (tales como escribir enciclopedias, por ejemplo).
- Hay proyectos de desarrollo con impulso gubernamental que utilizan software libre, así como en proyectos de voluntariado en países en vías de desarrollo.

Las implicaciones políticas y económicas del software libre, o su afinidad con el antiautoritarismo, es discutida. Mientras para unos estas implicaciones son notorias y representan un factor importante a tomarse en cuenta, para otros si bien podría existir una leve relación, no tiene suficiente relevancia.

2.4.2. Modelo de negocio

El negocio detrás del software libre se caracteriza por la oferta de servicios adicionales al software como: la personalización y/o instalación del mismo, soporte técnico, donaciones, patrocinios o como un elemento de responsabilidad social corporativa¹⁹ en contraposición al modelo de negocio basado en licencias predominante en el software de código cerrado.

2.4.3. Seguridad relativa

Existe una cierta controversia sobre la seguridad del software libre frente al software no libre (siendo uno de los mayores asuntos la seguridad por oscuridad). Un método usado de forma habitual para determinar la seguridad relativa de los productos es determinar cuántos fallos de seguridad no resueltos existen en cada uno de los productos involucrados. Por lo general, los usuarios de este método recomiendan que no sean usados productos que no suministren un método de solucionar los fallos de seguridad, al menos hasta que no esté disponible un arreglo.

¹⁷<http://www.oekonux.org/>

¹⁸<https://docs.hipatia.net/dsl/>

¹⁹Modelos open source y responsabilidad social

2.4.4. Blobs binarios y otros propietarios del software

En 2006, OpenBSD²⁰ comenzó la primera campaña contra el uso de blobs binarios en Kernel. Los Blobs son generalmente controladores de dispositivos libremente distribuibles para el hardware de los proveedores pero que no revelan el código fuente del controlador a los usuarios o desarrolladores. Esto restringe la libertad del usuario para modificar el software y distribuir versiones modificadas. Además, dado que los blobs no están documentados y pueden tener errores de software, suponen un riesgo para la seguridad de cualquier sistema operativo cuyo kernel los incluya. El objetivo de la campaña sobre los blobs es recopilar documentación de hardware que permita a los desarrolladores escribir controladores de software gratuitos para ese hardware, lo que finalmente permite que todos los sistemas operativos gratuitos se conviertan o permanezcan libres de blobs.

El problema de los blobs binarios en el kernel de Linux y en otros controladores de dispositivos motivó a algunos desarrolladores en Irlanda a lanzar gNewSense, una distribución basada en Linux con todos los blobs binarios eliminados. El proyecto recibió el apoyo de la Free Software Foundation y estimuló la creación, encabezada por la Free Software Foundation Latin America, del kernel Linux-libre.

2.4.5. Software libre en la Administración Pública

Entre los argumentos que sustentan la utilización de software libre en las administraciones públicas encontramos los siguientes:

- Defensa del interés general. El uso de software libre beneficia a la sociedad en su conjunto. Fomenta el crecimiento de la industria nacional de servicios y permite ahorrar divisas. La Administración Pública tiene cierta función de guía de la industria que la hace tener un gran impacto, que se considera que debe dirigirse a la creación de un ámbito tecnológico generador de riqueza nacional.

²⁰<https://www.openbsd.org/>

- **Transparencia.** El uso de software libre permite que cualquier ciudadano pueda comprobar por sí mismo la integridad y seguridad de los programas que usa su administración. Esto es especialmente importante en la Administración electrónica y en los sistemas que manipulan datos sensibles.
- **Independencia tecnológica.** El software libre, al garantizar el acceso al código fuente, permite elegir los proveedores de desarrollo, mantenimiento y soporte en un mercado de libre competencia. El software libre garantiza la libertad de elección de productos gracias a su modularidad, su disponibilidad en un gran número de plataformas y al uso de formatos de archivo abiertos, que permitan interoperar con independencia de estrategias comerciales.
- **Accesibilidad.** El Estado debe garantizar la accesibilidad a los servicios desde una amplia variedad de plataformas, sin discriminar a los ciudadanos por usar un software determinado ni forzarlos a adquirir determinados productos. Para ello es fundamental la utilización de formatos y estándares abiertos. El software libre puede ser traducido por la comunidad de usuarios de una lengua o por la propia Administración.
- **Seguridad.** El modelo de seguridad del software libre se basa en la transparencia. La amplia base de usuarios y desarrolladores de estos programas garantizan además un tiempo de respuesta rápido ante incidencias de seguridad. La transparencia permite verificar mediante auditorías la ausencia de código maligno que pueda abrir puertas traseras en los programas.
- **Protección de la inversión.** La continuidad de los productos de software libre está garantizada por la disponibilidad del código fuente, que permite mantener los productos durante tiempo indefinido. También se protege la inversión en hardware al evitarse las actualizaciones forzosas, prolongando el ciclo de vida de los mismos. Los servicios sobre el software libre se ofrecen en un mercado de libre competencia, por lo que sus precios son menores a los ofrecidos por el mantenimiento y soporte de programas propietarios.

- Bajo costo de adquisición de este tipo de software: Los productos de software libre no suelen tener un costo de licencia asociado, que si existe en los programas propietarios, además el costo de instalación y configuración es menor que el software privativo.
- Innovación: ya que un usuario curioso y con conocimientos técnicos puede aportar su experiencia para crear un nuevo software con características adicionales o adecuadas a su necesidad; basado en el software original.

Existen países en los cuales, sus administraciones públicas, han mostrado apoyo al software libre, sea migrando total o parcialmente sus servidores y sistemas de escritorio, o subvencionándolo. Este es el caso de Alemania, Cuba, Chile, China, Ecuador, España, Francia, México, República Dominicana, Venezuela, Argentina ,entre otros.

2.4.6. Software libre y gestión sanitaria

La inclusión de la gestión digital en el ámbito sanitario ha permitido transformar la relación entre los sistemas de salud y los usuarios, proporcionando una mayor agilidad, rentabilidad y accesibilidad. Su uso radica en la aplicación de las nuevas tecnologías para la mejora de los servicios dados a los pacientes y sus comunidades, a través de estrategias que facilitan la información y la divulgación; por ejemplo, la teleasistencia para minimizar los costes de la atención domiciliaria y facilitar el acceso a comunidades remotas; la telefonía móvil para permitir el acceso a la información y la comunicación entre profesionales y pacientes; la historia clínica informatizada, que posibilita el acceso colectivo de un equipo sanitario de trabajo a los datos de un paciente, entre otros.

La informatización de los servicios dirigidos a los pacientes requiere una gestión eficiente de los recursos tecnológicos. Es imprescindible una adaptación tecnológica de los sistemas de Información sanitarios con nuevos y mejores protocolos de comunicación, por lo que, en este sentido, el uso del software libre juega un papel fundamental, posibilitando el desarrollo colaborativo en red, su reutilización y disminuyendo los gastos económicos.

Actualmente la migración hacia Software Libre en el Ministerio de Salud (MINSAL) ha avanzado un 95% según nos comentaba el Ing. Carlos Juan Martín Pérez director de tecnologías de Información y comunicaciones, del Ministerio de Salud de El Salvador.

2.5. El software en Educación

Desde hace unos años las TIC y los centros educativos van unidos de la mano. Trabajamos con software en ámbitos diferentes: en secretaría, en los departamentos, en las aulas de informática, portátiles, etc. Los principales usos que damos al software en un centro educativo son:

- Gestiones relacionadas con la educación: actas, memorias, listados, etc.
- Uso con fines educativos, las TIC como herramienta que podemos usar para mejorar el aprendizaje de nuestros alumnos.

Todo software lleva una licencia que a modo de contrato nos dice lo que podemos hacer con dicho software. En líneas generales, sabemos que hay que comprar el software para poder usarlo de forma legal y que conlleva un gasto que no siempre podemos asumir desde el centro.

La alternativa más usual a esta problemática pasa por la instalación de las aplicaciones sin las licencias pertinentes (más conocido como piratería de software).

Una de las principales alternativas que tenemos para solventar este tipo de problemas pasa por el uso de software libre. Mediante su uso, entre otras muchas ventajas, nos liberaremos de esta problemática y, además, las familias de los alumnos no se verán en la necesidad de "piratear" el software que necesitan para el colegio.

Dentro del mundo del software libre, existen multitud de aplicaciones para realizar cualquier tarea que requiera un ordenador. Dentro de las principales alternativas libres tratare-

mos de analizar las mejores soluciones relacionadas con el mundo educativo para sustituir a los programas de pago.

Para finalizar, aprovechando la eclosión que ha tenido la llamada web 2.0 en internet, veremos las principales herramientas como alternativas a las aplicaciones instaladas en local.

2.5.1. Ventajas y desventajas del software libre en la educación

Según sus impulsores, el software libre fomenta el conocimiento y la participación de forma equitativa y colaborativa en el proceso educativo en particular, y en la cultura en general, pone al estudiante en situación de «crear», de ser usuario y productor al mismo tiempo. Los gobiernos y las instituciones educativas deberían utilizar y enseñar exclusivamente software libre porque este les permitiría adquirir independencia pedagógica y herramientas de conocimiento abiertas y accesibles a sus estudiantes y a toda la comunidad. Difundir el conocimiento y enseñar a los estudiantes a ser buenos miembros de esa comunidad. Por todo esto, el software libre parece tener beneficios en el campo educativo como los siguientes:

1. Libertad para elegir el programa que cubra de mejor manera las necesidades educativas.
2. Beneficio económico: Las licencias del software libre son completamente gratuitas, por lo que puede ser eliminado el pago en licencia de programas no libres e invertir el dinero en otra área.
3. Uso por parte de los estudiantes de los mismos programas en casa y escuela, de forma legal.
4. Existencia de comunidades de intercambio educativo.
5. Desarrollo independiente de aplicaciones: Se favorece el crecimiento de desarrolladores “pequeños” de software creando así una amplia oferta de programas educativos libres.

6. Conocimiento ampliado sobre la informática: El software libre permite que los estudiantes conozcan cómo trabajan los programas y fomenta el interés, la investigación y el aprendizaje de nuevos conceptos en el campo de la informática.
7. Diversidad de herramientas educativas en diversas áreas del conocimiento.
8. La posibilidad de copiar y difundir el software, es una forma de evitar que la propia institución educativa lleve a sus alumnos y docentes a la realización de copias ilegales de software privativo incurriendo en la violación de la ley .
9. Enseñando el software libre, las escuelas pueden formar ciudadanos preparados para vivir en una sociedad digital libre. Lo que implica la no dependencia del software privativo de las mega-corporaciones.

Existen algunas desventajas, como en cualquier tipo de software, como por ejemplo:

1. El software libre se compra o se adquiere sin garantías explícitas del fabricante o autor.
2. Las modificaciones o problemas encontrados requieren de la dedicación de recursos institucionales, así como a la adaptación a las necesidades del usuario que lo utilice.
3. Los usuarios deben tener unos conocimientos mínimos de programación para modificarlo y adaptarlo al contexto educativo.
4. Los usuarios deben estar al corriente de las modificaciones que se les haga al software para evitar confusiones.

Al utilizar un tipo de software en concreto, se debe pensar en las características de los usuarios para adaptarlo a sus necesidades y además la utilidad a la que se va a dedicar, el contexto educativo, infraestructuras, etc.

2.6. Historia De Gnu/Linux En El Salvador.

Las primeras referencias de GNU/Linux en El Salvador datan de 1998, donde en la comunidad de programadores se manejaba el concepto de un nuevo sistema operativo, gratuito y descargable desde Internet; para ese entonces los usuarios llegaban a los sitios de descarga, bajaban las imágenes iso del sistema operativo, creaban su CD de instalación y se embarcaban en la aventura de hacer funcionar sus computadoras por medio de la ayuda que ofrecía Internet. “El primer contacto que tuve con este sistema fue cuando fui a visitar a un amigo que había descargado de Internet un sistema operativo, muy entusiasmado me mostró las diferentes características que había descubierto (acceso remoto a través de ssh), sin embargo, fue dos años más tarde que empecé mi culturización en la utilización del sistema operativo GNU/Linux. A mediados del año 2000 con ayuda del Grupo Linux de El Salvador instalé por primera vez un servidor de Internet (Redhat 7.2) para la empresa en la que trabajaba, en ese año la divulgación del Sistema Operativa era poca, y la documentación en español escasa, pero gracias a la disponibilidad y paciencia del Grupo GNU/Linux tuve éxito en ese proyecto.” (Mejia, 2011).

El Grupo GNU/Linux de El Salvador se formó a principios del año 1999 con una reunión en la que participaron: Kirlian Zepeda, Max Dreyfus, Jaime López, Angel Posada, Carlos López, Francisco Díaz y Mario Arriaza. En el año 2000 se realizaron las primeras actividades masivas del grupo en la Universidad Centroamericana (UCA), las autoridades de la UCA prestaron un edificio completo para el desarrollo de charlas, talleres y demostraciones. En los años 2001 a 2004 el Grupo GNU/Linux realizó muchas actividades, en diferentes lugares, pero principalmente en universidades del país dando charlas informativas, talleres, demostraciones del sistema y festivales de instalación, la actividad del Grupo GNU/Linux dio un gran empuje para que otros usuarios decidieran probar y migrar sus aplicaciones a este nuevo sistema operativo.

En el año 2005 el Grupo GNU/Linux de El Salvador empezó el proceso de legalización

para convertirse en una Asociación y poder realizar proyectos de mayor envergadura, el impulso dado al sistema operativo sigue creciendo, han surgido nuevos grupos de usuarios (la comunidad de usuarios Debian de El Salvador, grupos de usuarios de GNU/Linux de las Universidades Don Bosco, Matías y UES) y cada vez hay más usuarios de GNU/Linux en nuestro país.

2.7. Situación actual del software libre en el MINSAL

Considerando que MINSAL cada 5 años gastara Un Millón de Dólares en materia de licenciamiento para los equipos actuales y alrededor de \$ 50,000 cada año en concepto de comprar antivirus, (Sin considerar los nuevos equipos que serían requerimiento incluir por el aumento de las necesidades) queda demostrado que en el período de los 7 años transcurridos desde la migración de Software Privativo a Software Libre, se ha obtenido un ahorro de más de 2.3 Millones, lo que se ha convertido en la compra e instalación de nuevos servidores, computadoras de escritorio, Laptops, contratación de nuevo personal para la creación, mantenimiento, desarrollo de software y capacitación, habilitación de plantas eléctricas, compra de teléfonos VoIP, Administrar y estructurar la Red Nacional, Mantenimiento y actualización de Sitio Web, etc (Avilés, Bermúdez, Climaco y Rodríguez, 2018).

2.8. Antecedentes del software libre en el MINED.

El Ministerio de Educación de El Salvador ha iniciado los procesos de migración hacia el uso de software libre en las escuelas y en las oficinas administrativas, así como, en las departamentales de educación. El proceso ha iniciado con capacitaciones del personal y esperan que para el 1 de julio de 2012, migrar completamente hacia el uso de LibreOffice, sustituyendo la versión privativa de Microsoft llamada Office. Por cierto ésta última es la suite de más alto costo económico en el mercado salvadoreño.¹

Según se lee en miportal.edu.sv: “Este nuevo software ofrece muchas ventajas, particular-

¹Tomado de: <http://informativo.mined.gob.sv:8090/DNP/GTIC/LibreOffice/>

mente en las administraciones públicas, entre las cuales se pueden mencionar el ahorro en costos económicos, libertad de elegir el proveedor, mejor accesibilidad, mantenimiento y desarrollo compartido, no hay virus, control total sobre el proyecto y razones de seguridad nacional, entre otras”, según el sitio web gestionado por el Ministerio de Educación.

La migración avanzó en principio con la utilización de navegadores privativos hacia el software libre, luego avanzaría hacia la migración de la ofimática libre, se espera que el salto más grande se dé cuando se trasladen a un sistema operativo libre que, sería la próxima etapa del plan. Sin embargo, en las escuelas ya hay laboratorios de informática que utilizan el sistema operativo con una distribución Ubuntu, ante la imposibilidad de que el MINED pueda pagar licencias privativas y, en otras, desde ya hace varios años utilizan ofimáticas libres.

Frente a la crisis sistémica que experimenta el mundo, la alternativa de más crecimiento es el uso del software libre, la misma Europa sumida en una grave crisis económica adopta masivas migraciones hacia el uso del software libre, el escenario para el final del año (2012) puede variar significativamente el mapa de uso de este tipo de herramientas alternativas.

2.8.1. Software libre en el Ministerio de Educación

■ Primeros Pasos.

1. El Ministerio escuchó de Linux, por lo que se implementó su uso en servidores con Red Hat en el año 2006.
2. Desconocían totalmente sus ventajas en torno a la filosofía de la Libertad.
3. Uso restringido a técnicos administradores de sistemas.

4. Carencia de especialización en el uso.

■ **Esfuerzos Iniciales.**

1. Los desarrolladores empezaron a formarse en el uso de tecnologías abiertas (java).
2. Aparecieron desarrolladores con fortalezas en web y PHP.
3. Se probaban distribuciones para conocer cuál era más amigable para los técnicos.

■ **Confusiones más comunes.**

1. Se promovió el uso del término Linux, más que de GNU/Linux.
2. Se usaron diferentes distribuciones esperando encontrar una que simulara al de la empresa de la ventana.
3. Se difundió el término de open source como sinónimo de software libre.

■ **Técnicos conocedores.**

1. Aparecieron más especialistas internos y externos, relacionados con sistemas operativos libres.
2. Las instituciones rducativas tenían encargados de aulas informáticas que poseían voluntad y empeño para aprender y usar software libre.
3. Internamente se dispuso una normativa donde dice que software es al que se le dará atención en la institución, y algunas herramientas nombradas son de software libre.
4. En los centros escolares de Educación Media se empezó a promover el uso de software libre a los estudiantes.

■ **Inversión Externa.**

1. Debido a proyectos educativos, se empezó a recibir apoyo de instituciones externas, consultores y organizaciones internacionales con expertos en la temática de software libre.
2. Se tuvieron proyectos educativos totalmente relacionados con software libre; pero aún sólo en uso.

■ **Empoderamiento.**

1. Los proyectos con apoyo internacional tuvieron que seguir, por lo que se brindaron espacios y oportunidades para que conocedores apoyaran, tanto de instituciones privadas como organizaciones civiles, (entusiastas y comunidades de software libre).
2. La institución afrontó su necesidad de apoyo en el uso de las nuevas herramientas tecnológicas.
3. Se permitió que la comunidad de software libre apoyara en el programa Cerrando la Brecha del Conocimiento (CBC).
4. Se dan certificaciones en línea acerca del uso de GNU/Linux, Ofimática Libre, y Aulas Virtuales en grado digital 4 desde el año 2012.
5. Nuevos equipos fueron implementados a nivel de usuarios administrativos utilizando completamente software libre.
6. Se inició un proceso de migración a uso de herramientas de ofimática para toda la institución
7. Se implementó un espejo de Debían interno para el soporte a los equipos instalados.
8. Se desarrollaron formaciones internas para difundir la filosofía y el uso de las nuevas herramientas.

9. En los centros escolares los equipos brindados ahora se les incluye software libre también.
 10. Al igual que en la institución, los Centros Educativos pasaran al usar Libre Office como herramienta de oficina.
 11. En la dotación de nuevos equipos a escuelas de escasos recursos, se les ha incluido exclusivamente software libre a los equipos multicliente.
 12. Las donaciones de mini laptops “Lempitas” realizada en el 2014 a centros escolares, también incluye solamente software libre.
- **Desarrollo Actual.**
1. Se necesita apoyo de especialistas para dar un mejor soporte a los proyectos y programas implementados a la fecha.
 2. Un cambio siempre requiere sacrificio.
 3. Es necesario pasar del uso de software libre a la creación de software libre.
 4. Afrontar nuevos retos es posible, sí estos se realizan a paso lento.

2.9. Software Libre como factor de desarrollo en El Salvador

■ **Económico**

El Salvador es un país en vías de desarrollo, con altas concentraciones de población en las ciudades, los niveles educativos de la población son muy bajos, con problemas serios en el medio ambiente por el mal uso de los recursos naturales y por falta de un sistema adecuado que proteja este recurso. Aunque los niveles educativos de la población son bajos, existen esfuerzos por llevar la tecnología informática a las escuelas, proyectos de acercamiento de Internet a la población, esto es bueno porque puede ser un elemento de desarrollo para las personas sin embargo necesita ser encausado hacia un plan de desarrollo económico de la nación.

A partir de los Acuerdos de Paz en 1992, El Salvador se fue consolidando como una economía basada en actividades del sector de servicios. El peso de la agricultura fue disminuyendo durante el conflicto armado y los precios internacionales de los productos primarios profundizaron esta tendencia. Desde 1990, según datos del Banco Central de Reserva, más del 50 % de la producción interna del país corresponde actividades al sector terciario. Después de la aceleración económica que trajo la firma de los Acuerdos y la reconstrucción, en 1995 se da el pique del crecimiento económico y a partir de 1996, el país ha entrado en recesión. También, en medio de la coyuntura internacional, se inició la aplicación de las políticas de los programas de ajuste estructural (PAE) y los programas de estabilización económica (PEE), estos programas facilitaron medidas que provocaron el surgimiento de sectores terciarios de amplia importancia y rápido crecimiento, tal como el sector financiero. En este sentido, la economía salvadoreña es una economía terciaria, donde los servicios se han posicionado como un eje clave.

Estas actividades se caracterizan por ser actividades que utilizan más tecnología y cuyo proceso de innovación es más rápido que en las actividades industriales y agrícolas. En este sentido, la innovación y creación de nuevos sistemas son elementos esenciales para la competitividad que se puede tener como país. La brecha tecnológica existente entre los países es sumamente amplia. Esta brecha, además, es difícilmente superable, dada la poca capacidad de países como El Salvador de poder invertir en tecnología. Por ejemplo, para 1998, los gastos en Investigación y Desarrollo (I+ D) salvadoreños representaban menos del 0.1 % del valor de la producción nacional, mientras, que los gastos realizados por EE.UU. rondaban alrededor del 3 % del valor de su producción. Estas brechas tecnológicas se reproducen al interior del país. El acceso a la información y sistemas de software propietarios es costoso y se convierte en una barrera más para las pequeñas empresas frente a las grandes. En este sentido, desde el factor económico los sistemas de software libre pueden aportar mucho a países como el nuestro, pues, aunque no es un software "gratis" el costo es relativamente pequeño en comparación

con los sistemas propietarios.

Si hacemos una pequeña reflexión en cuanto al costo del hardware y software, el costo de una computadora personal oscila por los \$400, las licencias de uso del sistema operativo y programas de aplicación oscilan por los \$400 dólares también, si revisamos los costos más bajos podemos comprar un sistema operativo propietario por \$120, el procesador de texto y la hoja de cálculo por \$220, también es indispensable comprar un antivirus con un costo de \$69 anual. Podemos ver que el costo del hardware es de \$400 y el del software de \$409. Qué pasaría si podemos bajar el costo del software hasta en un 95 %, esto significa capital que se puede invertir en otras áreas de la empresa (o evitamos el endeudamiento de las empresas para tener acceso a la tecnología).

Sin embargo, el beneficio no sólo puede verse a nivel micro. A nivel agregado esto puede significar mayor producción, mayor empleo y mejor desempeño económico. Incluso, el ahorro que podría darse, a un nivel estatal con relación al uso de sistemas propietarios podría ser reorientado a incrementar los gastos para la Investigación y Desarrollo y, por tanto redundar en beneficios aún mayores para el país.

■ **Tecnológico**

Desde el factor tecnológico el software libre también ofrece ventajas muy fuertes, pues permite acceder a la tecnología de punta a nivel mundial, al no ser restrictivo en cuanto a la forma de cómo se desarrollan los programas, el avance informático depende de los usuarios y programadores. Sin embargo, si vemos el software libre como factor de desarrollo de un país, podemos institucionalizarlo en la educación formal esto le dará un empuje a la formación técnica sobre herramientas de software libre potenciando el desarrollo económico a través de la tecnología.

El acceso a las nuevas tecnologías genera la oportunidad de desarrollo intelectual y tec-

nológico que a la vez potencia el desarrollo económico porque abre nuevos mercados (mercados tecnológicos), las posibilidades que se abren sobre este aspecto son ilimitadas pues al no ser restrictiva la tecnología permite que podamos crear en cualquier momento nuevos productos sobre la base del conocimiento compartido que existe a través de Internet.

El acceso al software libre es una fuente de desarrollo tecnológico que está al alcance de países como el nuestro, acceso a tecnología de punta a través de Internet, permite obtener información relativa al desarrollo de programas. Y este puede ser un eje de desarrollo tecnológico para un país como el nuestro, donde se pueden formar una nueva generación de programadores comprometidos con el desarrollo del país que puedan liberarnos de la dependencia tecnológica y económica actual.

■ **Social**

El principal componente del software libre son los principios que defiende, El software libre es consecuencia de un fenómeno social, donde las personas que desarrollan software lo hacen para satisfacer una necesidad particular de ellos en primera instancia, después comparten su solución con otros y éstos a la vez incorporan mejoras para satisfacer sus propias necesidades, esto es posible al tener un software que permita las libertades de: utilizar, estudiar, modificar y distribuir las modificaciones hechas al software.

La oportunidad de poder observar el trabajo desarrollado por otros permite que las personas puedan entender exactamente cómo funcionan los programas, pueden hacer observaciones sobre cómo mejorarlo o incluso pueden mejorarlo ellos mismos, por ello el software libre promueve la interacción de personas de una misma región o diferentes regiones, también propicia la colaboración entre las personas para satisfacer necesidades comunes. El software libre estimula la creatividad y el deseo de investigar de las personas. Esto es posible a través de la sólida base del conocimiento sobre desarrollo de software que existe para poder iniciar nuevos proyectos, prueba de ellos es la gran

cantidad de proyectos que actualmente se están desarrollando sobre software libre a través de Internet en el mundo.

La educación es un factor de desarrollo muy importante de los pueblos, un pueblo educado y libre es capaz de desarrollarse en todas las áreas de su vida, el software libre promueve el desarrollo intelectual de las personas a través del auto aprendizaje, la capacidad de poder ver y tocar las nuevas tecnologías permite un rápido desarrollo educativo de las comunidades de software de código abierto, el auto aprendizaje es el principal eje de desarrollo intelectual en el mundo actualmente. (Osorio, 2008).

2.9.1. Casos de éxito en migración al software libre en educación.

- **Instituto Especializado de Educación Superior de Profesionales de la Salud de El Salvador (IEPROES).**

El instituto ahorró 38 % en costos de hardware, rebajó la cuenta de consumo de electricidad, e incrementó el acceso a computadores. "Todo funciona muy bien, y el dinero que usábamos para comprar anti-virus ahora puede ser invertido en más estaciones Multi-Usuario", asegura Patricia Echavarría, Profesora de software libre en el IEPROES, "Nosotros recomendamos y nos sentimos muy orgullosos de trabajar con sistemas Multi-Usuario y software libre". Por su parte, Oscar Dávila quien es el Director del Departamento de Sistemas y los estudiantes están muy contentos con la nueva solución. "La parte de los ahorros es muy buena, pero todo el software libre disponible es definitivamente excelente", dice Oscar. "Los estudiantes ahora vienen con sus portátiles para que les instalemos Linux."

El IEPROES pasó de tener 150 computadores viejos con Windows, a 200 estaciones de cómputo nuevas con Linux, para dar servicio a sus estudiantes en 3 sucursales distintas. «Uno de los grandes éxitos de la solución de Useful es que los usuarios no sienten la diferencia entre Windows y Linux, y no se dan cuenta que están trabajando en una

máquina virtual», argumenta Roberto Arévalo, Gerente de Ventas de SISCOM.

Hoy, estudiantes y profesores del IEPROES están disfrutando los computadores nuevos y la riqueza de las aplicaciones disponibles de software libre sin ningún costo. La solución se ha vuelto un ejemplo comprobado de cómo la tecnología Multi-Usuario puede ayudar al gobierno salvadoreño a migrar sus sistemas a software libre, expandir acceso a computadores, y reducir costos de implementación. (IEPROES, s.f.)

■ **Software libre en la Universidad Luterana.**

La Universidad Luterana Salvadoreña inició el recorrido de adoptar el software libre en enero de 2009, cuando se migra el primer Centro de Cómputo de Windows XP a Ubuntu 8.10, gracias a la decisión contundente de no seguir pagando altas sumas de dinero en concepto de licencias privativas. En la actualidad cerca de 160 equipos informáticos son utilizados con ese software realmente alternativo, representando el 98.2%, tanto en la Sede Central ubicada en San Salvador, como en su Centro Regional de Cabañas. (Molina Medrano, 2013).

A la par del ahorro económico que se ha mantenido durante cuatro años, también se ha experimentado otro tipo de beneficios, como el aumento de la vida útil de los equipos, el rescate de varios que se pensaba eran obsoletos, adquisición de equipo con características básicas que funcionan eficientemente, menos necesidad de soporte y mantenimiento, mayor seguridad en la gestión y transferencia de datos, entre muchos más que podríamos seguir enumerando, y que no sólo abarcan lo técnico y lo económico, sino también lo social.

2.10. Marco Legal

2.10.1. Tipos de licencias

El software no se vende, se licencia. Una licencia es aquella autorización formal con carácter contractual que un autor de un software da a un interesado para ejercer "actos de

explotación legale”. Es decir, el software no se compra, sino que se adquieren una serie de derechos sobre el uso que se le puede dar. En las licencias de software libre esos derechos son muy abiertos y permisivos, apenas hay restricciones al uso de los programas. De ahí que ayude al desarrollo de la cultura. Pueden existir tantas licencias como acuerdos concretos se den entre el autor y el licenciatarario. Desde el punto de vista del software libre, existen distintas variantes del concepto o grupos de licencias:

- Licencias GPL. Una de las más utilizadas es la Licencia Pública General de GNU (GNU GPL). El autor conserva los derechos de autor (copyright), y permite la redistribución y modificación bajo términos diseñados para asegurarse de que todas las versiones modificadas del software permanecen bajo los términos más restrictivos de la propia GNU GPL. Esto hace que sea imposible crear un producto con partes no licenciadas GPL: el conjunto tiene que ser GPL. Es decir, la licencia GNU GPL posibilita la modificación y redistribución del software, pero únicamente bajo esa misma licencia. Y añade que si se reutiliza en un mismo programa código "A" licenciado bajo licencia GNU GPL y código "B" licenciado bajo otro tipo de licencia libre, el código final "C", independientemente de la cantidad y calidad de cada uno de los códigos "A" y "B", debe estar bajo la licencia GNU GPL. En la práctica esto hace que las licencias de software libre se dividan en dos grandes grupos, aquellas que pueden ser mezcladas con código licenciado bajo GNU GPL (y que inevitablemente desaparecerán en el proceso, al ser el código resultante licenciado bajo GNU GPL) y las que no lo permiten al incluir mayores u otros requisitos que no contemplan ni admiten la GNU GPL y que por lo tanto no pueden ser enlazadas ni mezcladas con código gobernado por la licencia GNU GPL. En el sitio web oficial de GNU hay una lista de licencias que cumplen las condiciones impuestas por la GNU GPL y otras que no²¹.

Aproximadamente el 60 por ciento del software licenciado como software libre emplea una licencia GPL.

²¹Free Software Foundation, Inc. «Licencias».

- Licencias LGPL

La Licencia Pública General Reducida de GNU, o más conocida por su nombre en inglés GNU Lesser General Public License, es una licencia creada por la (FSF) que garantiza la libertad de compartir y modificar el software cubierto por ella, asegurando que el software es libre para todos sus usuarios. Esta licencia se aplica a cualquier programa o trabajo que contenga una nota puesta por el propietario de los derechos del trabajo estableciendo que su trabajo puede ser distribuido bajo los términos de esta. El "Programa", utilizado en lo subsecuente, se refiere a cualquier programa o trabajo original, y el "trabajo basado en el Programa" significa que el programa o cualquier trabajo derivado del mismo bajo la ley de derechos de autor.

- Licencias AGPL

La Licencia Pública General de Affero (en inglés Affero General Public License, también Affero GPL o AGPL) es una licencia copyleft derivada de la Licencia Pública General de GNU diseñada específicamente para asegurar la cooperación con la comunidad en el caso de software que corra en servidores de red. La Affero GPL es íntegramente una GNU GPL con una cláusula nueva que añade la obligación de distribuir el software si este se ejecuta para ofrecer servicios a través de una red de ordenadores. La Free Software Foundation recomienda que el uso de la GNU AGPLv3 sea considerado para cualquier software que usualmente corra sobre una red.

- Licencias Estilo BSD

Llamadas así porque se utilizan en gran cantidad de software distribuido junto a los sistemas operativos BSD. El autor, bajo tales licencias, mantiene la protección de copyright únicamente para la renuncia de garantía y para requerir la adecuada atribución de la autoría en trabajos derivados, pero permite la libre redistribución y modificación, incluso si dichos trabajos tienen propietario. Son muy permisivas, tanto que son fácilmente absorbidas al ser mezcladas con la licencia GNU GPL con quienes son com-

patibles. Puede argumentarse que esta licencia asegura “verdadero” software libre, en el sentido que el usuario tiene libertad ilimitada con respecto al software, y que puede decidir incluso redistribuirlo como no libre. Otras opiniones están orientadas a destacar que este tipo de licencia no contribuye al desarrollo de más software libre (normalmente utilizando la siguiente analogía: “una licencia BSD es más libre que una GPL si y solo si se opina también que un país que permita la esclavitud es más libre que otro que no la permite”).

- Licencia PSFL

La Python Software Foundation License, anteriormente Python License, es una licencia de software libre permisiva, al estilo de la licencia BSD, es decir, que cumple con los requisitos OSI para ser declarada licencia de software libre; además, es compatible con la licencia GPL. A diferencia de la licencia GPL, y como la mayoría de licencias tipo BSD, la licencia PSFL no es una licencia copyleft, y permite modificaciones del código fuente, así como la creación de trabajos derivados, sin requerir que ni las modificaciones ni los trabajos derivados tengan que ser a su vez de código abierto. La licencia PSFL está dentro de las listas de licencias aprobadas tanto por la Free Software Foundation como por la Open Source Initiative.

- Licencia MIT

La licencia MIT es una de tantas licencias de software que ha empleado el Instituto Tecnológico de Massachusetts (MIT, Massachusetts Institute of Technology) a lo largo de su historia, y quizás debería llamarse más correctamente licencia X11, ya que es la licencia que llevaba este software de muestra de la información de manera gráfica X Window System originario del MIT en los años 1980. Pero ya sea como MIT o X11, su texto es idéntico. El texto de la licencia no tiene copyright, lo que permite su modificación. No obstante esto, puede no ser recomendable e incluso muchas veces dentro del movimiento del software de código abierto desaconsejan el uso de este texto

para una licencia, a no ser que se indique que es una modificación, y no la versión original. La licencia MIT es muy parecida a la licencia BSD en cuanto a efectos.

- Licencias MPL y derivadas

Esta licencia es de Software Libre y tiene un gran valor porque fue el instrumento que empleó Netscape Communications Corp. para liberar su Netscape Communicator 4.0 y empezar ese proyecto tan importante para el mundo del Software Libre: Mozilla. Se utilizan en gran cantidad de productos de software libre de uso cotidiano en todo tipo de sistemas operativos. La MPL es Software Libre y promueve eficazmente la colaboración evitando el efecto "viral" de la GPL (si usas código licenciado GPL, tu desarrollo final tiene que estar licenciado GPL). Desde un punto de vista del desarrollador la GPL presenta un inconveniente en este punto, y lamentablemente mucha gente se cierra en banda ante el uso de dicho código. No obstante la MPL no es tan excesivamente permisiva como las licencias tipo BSD. Estas licencias son denominadas de copyleft débil. La NPL (luego la MPL) fue la primera licencia nueva después de muchos años, que se encargaba de algunos puntos que no fueron tomados en cuenta por las licencias BSD y GNU. En el espectro de las licencias de software libre se la puede considerar adyacente a la licencia estilo BSD, pero perfeccionada.

- Licencia CDDL

Common Development and Distribution License (Licencia Común de Desarrollo y Distribución), también conocida como Sun Public License (SPL) versión 2, es una licencia de código abierto (OSI) y libre, producida por Sun Microsystems, basada en la Mozilla Public License o MPL, versión 1.1. La licencia CDDL fue enviada para su aprobación al Open Source Initiative el 1 de diciembre de 2004, y fue aprobada como una licencia de código abierto a mediados de enero de 2005. En el primer borrador hecho por el comité de divulgación de licencias OSI, la CDDL es una de las nueve licencias más populares, mundialmente usadas o con fuertes comunidades.

- Copyleft

Hay que hacer constar que el titular de los derechos de autor (copyright) de un software bajo licencia copyleft puede también realizar una versión modificada bajo su copyright original, y venderla bajo cualquier licencia que desee, además de distribuir la versión original como software libre. Esta técnica ha sido usada como un modelo de negocio por una serie de empresas que realizan software libre (por ejemplo MySQL); esta práctica no restringe ninguno de los derechos otorgados a los usuarios de la versión copyleft.

En España, toda obra derivada está tan protegida como una original, siempre que la obra derivada parta de una autorización contractual con el autor. En el caso genérico de que el autor retire las licencias copyleft, no afectaría de ningún modo a los productos derivados anteriores a esa retirada, ya que no tiene efecto retroactivo. En términos legales, el autor no tiene derecho a retirar el permiso de una licencia en vigencia. Si así sucediera, el conflicto entre las partes se resolvería en un pleito convencional.

- Licencias EPL

La Licencia Pública Eclipse (EPL) es una licencia utilizada por la Fundación Eclipse para su software. Sustituye a la Licencia Pública Común (CPL) y elimina ciertas condiciones relativas a los litigios sobre patentes. La Licencia Pública de Eclipse está diseñado para ser una licencia de software favorable a los negocios y cuenta con disposiciones más débiles que las licencias copyleft contemporáneas. El receptor de programas licenciados EPL pueden utilizar, modificar, copiar y distribuir el trabajo y las versiones modificadas, en algunos casos están obligados a liberar sus propios cambios.

- Licencia APSL

La Apple Public Source License es la licencia de código abierto y de software libre bajo la cual se publicó el sistema operativo de Apple, Darwin. Esta licencia fue adoptada de forma voluntaria para aumentar la colaboración de la comunidad de la que se origina

gran parte de Darwin. La primera versión de APSL fue aprobada por Open Source Initiative (OSI). La versión 2.0, lanzada el 29 de julio de 2003 es compatible con las directrices de la Free Software Foundation, y también está aprobada por la OSI. La Free Software Foundation aprobó la Apple Public Source License 2.0 como una licencia libre ya que reúne los requisitos y afirma que está bien usar y mejorar el software ya publicado bajo esta licencia pero no recomienda utilizarla para proyectos nuevos.

■ Licencia Apache

La licencia Apache (Apache Software License para versiones anteriores a 2.0) es una licencia de software libre creada por la Apache Software Foundation (ASF). La licencia (con versiones 1.0, 1.1 y 2.0) requiere la conservación del aviso de copyright y el disclaimer, pero no es una licencia copyleft, ya que no requiere la redistribución del código fuente cuando se distribuyen versiones modificadas ni siquiera que se tengan que distribuir como software libre/open source, solo exige que se mantenga una noticia que informe a los receptores que en la distribución se ha usado código con la Licencia Apache. Se deben añadir dos archivos en el directorio principal de los paquetes de software redistribuidos:

- LICENSE - Una copia de la licencia
- NOTICE - Un documento de texto, que incluye los «avisos» obligatorios del software presente en la distribución.

■ Licencia PHP

La licencia PHP es la licencia bajo la cual se publica el lenguaje de programación PHP. De acuerdo a la Free Software Foundation es una licencia de software libre no copyleft y una licencia de código abierto según la Open Source Initiative. Debido a la restricción en el uso del término "PHP", no es compatible con la licencia GPL.

2.10.2. Licencias para software

Licencias de software libre compatibles con la GPL

- Licencia Pública General de GNU (GPL) versión 3 ²²

Esta es la última versión de la GPL de GNU: una licencia de software libre con copyleft. Es la que recomendamos para la mayoría de los paquetes de software.

Nótese que la GPLv3 no es compatible con la GPLv2 por sí misma. Sin embargo, la mayoría del software publicado bajo la GPLv2 permite que se utilice bajo los términos de las versiones posteriores de la GPL. Cuando este sea el caso, puede utilizar el código bajo la GPLv3 para lograr la combinación deseada.

- Licencia Pública General de GNU (PL) versión 2

Esta es la versión previa de la GPL de GNU: una licencia de software libre con copyleft. Recomendamos utilizar la última versión para la mayoría del software.

Nótese que la GPLv2, por sí misma, no es compatible con la GPLv3. Sin embargo, la mayoría del software publicado bajo la GPLv2 permite que se utilice también bajo los términos de las versiones posteriores de la GPL. Cuando este sea el caso, puede utilizar el código bajo la GPLv3 para lograr la combinación deseada.

- Licencia Pública General Reducida de GNU (LGPL) versión 3

Esta es la última versión de la LGPL: una licencia de software libre, pero sin un sólido copyleft, ya que permite el enlazado con módulos privativos. Es compatible con la GPLv3. La recomendamos solo para circunstancias especiales.

²²<http://www.gnu.org/licenses/license-list.html#NetscapeJavaScript>

Nótese que la GPLv3 no es compatible con la GPLv2 por sí misma. Sin embargo, la mayoría del software publicado bajo la GPLv2 permite que se utilice también bajo los términos de las versiones posteriores de la GPL. Cuando este sea el caso, puede utilizar el código bajo la GPLv3 para lograr la combinación deseada.

- Licencia Pública General Reducida de GNU (LGPL) versión 2.1

Esta es la versión previa de la LGPL: una licencia de software libre, pero sin un sólido copyleft, ya que permite el enlazado con módulos privativos. Es compatible con la GPLv2 y la GPLv3. Generalmente recomendamos la última versión de la LGPL, solo para circunstancias especiales.

- Licencia Pública General Affero de GNU (AGPL) versión 3

Esta es una licencia de software libre con copyleft. Sus términos son en la práctica los mismos de la GPLv3, con un párrafo adicional en la sección 13 que permite a los usuarios que interactúan con el software bajo esta licencia en una red, recibir la fuente de tal software. Recomendamos a los desarrolladores que consideren el uso de la AGPL de GNU para todo el software que se ejecute de forma habitual en una red.

Nótese que la AGPL de GNU no es compatible con GPLv2. Tampoco es técnicamente compatible con la GPLv3 en sentido estricto: no se puede tomar código publicado bajo la AGPL de GNU, copiarlo o modificarlo a gusto bajo los términos de la GPLv3, o viceversa. Sin embargo, sí le está permitido combinar módulos separados o archivos fuente publicados bajo ambas licencias y obtener así un único proyecto, lo cual le otorgará a muchos programadores los permisos necesarios para hacer los programas que deseen.

- Licencia Completamente Permisiva de GNU

Esta es una licencia de software libre laxa, permisiva, compatible con la GPL de

GNU, que recomendamos usar en los pequeños archivos de soporte de los paquetes de GNU, como README, etc. Todos los desarrolladores, si lo desean, pueden usarla en situaciones similares.

Las versiones más antiguas de esta licencia no tenían la segunda oración en la que se expresa la ausencia de garantía. El mismo análisis se aplica a ambas versiones.

- Licencia Apache, versión 2.0

Esta es una licencia de software libre compatible con la versión 3 de la GPL de GNU.

Nótese que esta licencia no es compatible con la versión 2 de la GPL de GNU porque tiene algunos requisitos que no están en esa versión de la GPL, por ejemplo ciertas disposiciones sobre las indemnizaciones y la terminación por patentes. La disposición sobre las patentes es buena, por eso para programas de una cierta envergadura recomendamos usar la licencia Apache 2.0 en lugar de otras licencias laxas, permisivas.

- Licencia Artística 2.0

Esta es una licencia de software libre, compatible con la GPL gracias a la opción de reposición de la licencia.

- Licencia Artística Clarificada

Esta es una licencia de software libre, compatible con la GPL. Contiene los cambios mínimos necesarios para corregir las imprecisiones de la Licencia Artística 1.0.

- Berkeley Database License (también conocida como la «Sleepycat Software Product License»).

Esta es una licencia de software libre compatible con la GPL de GNU.

- Boost Software License.

Esta es una licencia de software libre laxa, permisiva, sin copyleft, compatible con la GPL de GNU.

- Licencia BSD Modificada.

Esta es la licencia BSD original, modificada por la eliminación de la cláusula de publicidad. Es una licencia de software libre laxa, permisiva, sin copyleft, compatible con la GPL de GNU.

A esta licencia se la llama algunas veces «licencia BSD de 3 cláusulas». Como licencia laxa, permisiva, la BSD modificada no es tan mala, aunque es preferible la Apache 2.0. Sin embargo, es peligroso recomendar el uso de la «licencia BSD», incluso en casos especiales como por ejemplo para programas pequeños, porque fácilmente se podría producir una confusión que llevaría al uso de la defectuosa licencia BSD original. Para evitar ese riesgo, se puede sugerir el uso de la licencia X11. Las licencias X11 y BSD modificada son más o menos equivalentes. De todas maneras, para programas de una cierta magnitud es mejor la licencia Apache 2.0 porque previene la trampa de las patentes.

- CC0

La licencia CC0 es una contribución al dominio público por parte de Creative Commons. Una obra publicada bajo la CC0 se entrega al dominio público en la extensión máxima permitida por la ley. Si por cualquier razón esto no fuera posible, alternativamente la CC0 también sirve como licencia laxa, permisiva. Tanto las obras en dominio público como la licencia laxa proporcionada por CC0 son compatibles con la GPL de GNU. Si desea publicar su obra en el dominio público, le recomendamos que use CC0.

- CeCILL, versión 2

La CeCILL es una licencia de software libre, explícitamente compatible con la GPL de GNU.

El texto de la CeCILL usa un par de términos tendenciosos que deberían evitarse: «propiedad intelectual» y «protección»); esta decisión fue inapropiada porque al leerse, la licencia tiende a difundir las presuposiciones que dichos términos sugieren. Sin embargo, esto no causa ningún problema en particular a los programas publicados bajo la CeCILL.

- The Clear BSD License

Esta es una licencia de software libre, compatible tanto con la GPLv2 como con la GPLv3. Está basada en la licencia BSD modificada, a la que se le ha agregado una cláusula que dice expresamente que no se concede al usuario ninguna licencia sobre patentes. Es por esto que recomendamos ser cuidadosos cuando se utiliza software que está bajo esta licencia; primero se debe considerar si el licenciataria podría demandar al usuario por infringir una patente. Si el desarrollador se niega a conceder licencias sobre patentes para ponerle una trampa al usuario, sería mejor evitar utilizar el programa.

- Cryptix General License

Esta es una licencia de software libre laxa, permisiva, sin copyleft, compatible con la GPL de GNU. Es muy similar a la licencia X11.

- Licencia eCos, versión 2.0

La licencia eCos versión 2.0 es compatible con la GPL. Consiste en la misma GPL más una excepción que permite el enlazado con software que no se encuentra bajo la GPL. Tiene las mismas desventajas que la LGPL.

- Educational Community License 2.0

Esta es una licencia de software libre compatible con la GPLv3. Se basa en la

licencia Apache 2.0. El alcance de la licencia de patentes se modificó en modo tal que, en el caso de que un empleado de una organización trabaje en un proyecto, la organización no tenga que poner bajo esta licencia todas sus patentes. Este licenciamiento de patentes y la cláusula de indemnización en la sección 9 hacen a la licencia en cuestión incompatible con la GPLv2.

- Eiffel Forum License, versión 2

Esta es una licencia de software libre, compatible con la GPL de GNU. Las versiones anteriores de la licencia Eiffel no son compatibles con la GPL. EU DataGrid Software License

- EU DataGrid Software License

Esta es una licencia de software libre laxa, permisiva, sin copyleft, compatible con la GPL de GNU.

- Licencia de Expat

Esta es una licencia de software libre laxa, permisiva, sin copyleft, compatible con la GPL de GNU. Algunas veces se la llama ambiguamente «MIT License».

Para programas de una cierta magnitud es mejor usar la licencia Apache 2.0 porque evita la trampa de las patentes.

- Licencia FreeBSD

Esta es la licencia BSD original en la que se ha eliminado la cláusula de la publicidad, y otra. A veces se la denomina «2-clause BSD license» (licencia BSD de dos cláusulas). Es una licencia de software libre laxa, permisiva, sin copyleft, compatible con la GPL de GNU.

Nuestros comentarios acerca de la licencia BSD Modificada se aplican también a esta licencia.

- Freetype Project License

Esta es una licencia de software libre compatible con la GPLv3. Posee algunas características que la hacen incompatible con GPLv2.

- Historical Permission Notice and Disclaimer

Esta es una licencia de software libre laxa, permisiva y débil, que es compatible con la GPL. Es similar a la licencia de Python 1.6a2 y versiones anteriores.

- Licencia de la iMatix Standard Function Library

Esta es una licencia de software libre compatible con GPL.

- Licencia de imlib2

Una licencia de software libre compatible con GPL. El autor nos ha explicado que las opciones de la GPL para proveer el código fuente significa que dicho código «ha sido puesto en manos del público», según sus palabras.

- Independent JPEG Group License

Esta es una licencia de software libre compatible con la GPL de GNU. Los autores nos aseguraron que los desarrolladores que documenten los cambios de la forma requerida por la GPL también estarán cumpliendo con los requisitos de esta licencia.

- Licencia informal

Una «licencia informal» es una declaración de tipo «Haga lo que quiera con esto» o «Usted puede redistribuir y modificar este código».

En los Estados Unidos, se supone que la interpretación de estas licencias se hace en base a la aparente intención del autor. Por tanto, es probable que el significado de estas licencias sea el que aparentan, lo cual haría que fueran licencias libres sin copyleft y compatibles con la GPL de GNU. Sin embargo, una redacción desacer- tada podría darles un significado diferente.

Sin embargo, muchos otros países adoptan un enfoque más rígido respecto a las licencias de copyright. No podemos predecir cómo podrían interpretar una declaración informal los tribunales de tales países, podrían incluso decidir que no se trata de una licencia.

Si desea que su código sea libre, no cree innecesariamente una fuente de problemas para los usuarios. Elija y aplique una licencia de software libre consolidada.

- Intel Open Source License

Esta es una licencia de software libre compatible con la GPL de GNU.

- Licencia ISC

Esta licencia es a veces también conocida como licencia OpenBSD. Es una licencia de software libre y compatible con la GPL de GNU.

Esta licencia ha elegido una formulación desafortunada: otorga a los destinatarios «Permiso para utilizar, copiar, modificar y distribuir este software». Son las mismas palabras empleadas en la licencia de Pine que, según luego alegó la Universidad de Washington, prohibían la distribución de versiones modificadas del software.

ISC nos ha dicho que no comparte la interpretación de la Universidad de Washington, y hemos de creerles. Además, ISC actualizó la licencia para que dijera «Permiso para utilizar, copiar, modificar y/o distribuir este software». Si bien la inclusión de «y/o» no resuelve completamente el problema, no vemos razón alguna para evitar el software distribuido bajo esta licencia. Sin embargo, para evitar que esta terminología cause problemas en el futuro, animamos a los desa-

rrolladores a que elijan otra licencia para sus trabajos. La licencias FreeBSD es igualmente permisiva y breve. No obstante, si busca una licencia laxa y débil, le recomendamos la licencia Apache 2.0.

- Licencia Pública de Mozilla (MPL), versión 2.0

Esta es una licencia de software libre. La sección 3.3 proporciona una compatibilidad indirecta entre esta licencia y la versión 2.0 de la GPL de GNU, la versión 2.1 de la LGPL de GNU, la versión 3 de la AGPL de GNU y todas las versiones posteriores de estas licencias. Cuando reciba una obra bajo la MPL 2.0, puede hacer una «obra más grande» que combine la obra bajo la MPL con otras que están bajo licencias de GNU. Al hacerlo, la sección 3.3 le autoriza a distribuir la obra cubierta por la MPL bajo los términos de dichas licencias de GNU, con una condición: debe asegurarse que los archivos, que originalmente estaban disponibles bajo la MPL, sigan estando disponibles bajo los términos de la MPL. En otras palabras, cuando haga una combinación de este tipo, los archivos que estaban originalmente bajo la MPL pueden licenciarse de forma dual bajo la MPL y una/s licencia/s de GNU. El resultado final es una obra mayor, que como total, estará cubierta por la/s licencia/s de GNU. Los que reciban la combinación tendrán la opción de usar los archivos originalmente cubiertos por la MPL, bajo los términos de la MPL o distribuir la obra, en el todo, bajo las condiciones de una de las licencias de GNU sin más restricciones.

Es importante entender que la condición de distribuir archivos bajo los términos de la MPL solo se aplica a quien, en primer lugar, crea y distribuye la obra ampliada. Si se aplicase a quienes la reciben, sería una restricción adicional, incompatible con la GPL y AGPL. Dicho esto, cuando colabore con un proyecto que ya existe, normalmente recomendamos que mantenga todos sus cambios bajo la misma licencia, incluso cuando no se le solicite. Si recibe una obra bajo

una licencia de GNU, donde algunos archivos estén bajo la MPL, solo debería eliminar la MPL de esos archivos cuando exista una poderosa razón que lo justifique.

Verifique las notas sobre la licencia en el software cubierto bajo la MPL antes de realizar una obra más amplia sobre él. Quien publica bajo la MPL 2.0 puede elegir eliminar esta compatibilidad, añadiendo una indicación en las notas sobre la licencia que diga que esta obra es «Incompatible With Secondary Licenses.» [incompatible con licencias secundarias]. Cualquier obra que incluya esta indicación no es compatible con la GPL ni la AGPL.

El software bajo versiones previas de la MPL puede actualizarse a la versión 2.0, pero en el software que no esté disponible bajo una de las licencias de GNU, debe indicarse que es incompatible con licencias secundarias. Esto implica que el software que solo está disponible bajo la versión previa de la MPL sigue siendo incompatible con la GPL y la AGPL.

- NCSA/University of Illinois Open Source License

Esta licencia está basada en los términos de las licencias Expat y BSD modificada. Es una licencia de software libre laxa, permisiva, sin copyleft, compatible con la GPL de GNU.

- Licencia de Netscape JavaScript

Es la separación de la Licencia Pública de Netscape y la GPL de GNU. Por esta razón es una licencia de software libre, compatible con la GPL de GNU, pero con un copyleft débil.

- OpenLDAP License, versión 2.7

Es una licencia de software libre sin copyleft, permisiva, compatible con la GPL de GNU.

- Licencia de Perl 5 y anteriores

Esta licencia consiste en la separación de la Licencia Artística 1.0 y la GPL de GNU; en otras palabras, puede elegir entre cualquiera de las anteriores. Se califica como una licencia de software libre pero puede no ser un verdadero copyleft. Es compatible con la GPL de GNU porque esta última constituye una de las alternativas.

Recomendamos utilizar esta licencia para cualquier paquete hecho en Perl 4 o Perl 5, para promover la coherencia y la uniformidad en cuanto a lo que se programa en este lenguaje. Fuera de él, urgimos a que esta licencia no sea utilizada, ya que es preferible la GPL de GNU.

- Dominio Público

El dominio público no se refiere a tener una licencia; más bien, significa que el material no se encuentra bajo copyright y no es necesaria ninguna licencia. Prácticamente hablando, si una obra es de dominio público, puede a su vez tener una licencia de software libre totalmente permisiva sin copyleft. El material en dominio público es compatible con la GPL de GNU.

Si desea publicar su obra en el dominio público, le animamos a usar herramientas formales para hacerlo. A quienes realizan pequeñas contribuciones a GNU les pedimos firmar un formulario de renuncia, esa es una solución. Si está trabajando en un proyecto que no tiene políticas formales de colaboración, CC0 es una buena herramienta que cualquiera puede usar. Dedicar formalmente su obra al dominio público, y proporciona una licencia alternativa cuando no sea legalmente posible.

- Licencia de Python 2.0.1, 2.1.1, y versiones posteriores

Esta es una licencia de software libre, compatible con la GPL de GNU. Nótese,

sin embargo, que las versiones intermedias de Python (1.6b1 hasta 2.0 y 2.1) se encuentran bajo una licencia diferente.

- Licencia de Python 1.6a2 y versiones anteriores

Esta es una licencia de software libre compatible con la GPL de GNU. Nótese, sin embargo, que las versiones nuevas de Python se encuentran bajo otras licencias (ver puntos previo y siguientes).

- Licencia de Ruby

Esta es una licencia de software libre, compatible con la GPL mediante una cláusula explícita de licencia dual.

- SGI Free Software License B, versión 2.0

La SGI Free Software License B versión 2.0 es una licencia de software libre. Es esencialmente idéntica a la licencia X11, con una alternativa opcional para proveer las notas sobre la licencia.

Las versiones anteriores de la SGI Free Software License B no eran licencias de software libre, a pesar de su nombre. Sin embargo, todas incluían cláusulas que permiten cambiar a versiones más recientes de la misma, si así se desea. Por lo tanto, si un programa se publicó bajo cualquier versión de la SGI Free Software License B, se puede utilizar bajo los términos de esta versión libre.

- Standard ML of New Jersey Copyright License

Esta es una licencia de software libre laxa, permisiva, sin copyleft, compatible con la GPL de GNU.

- Unicode, Inc. License Agreement for Data Files and Software

Esta es una licencia que Unicode Inc. ha aplicado a la Unicode Character Database: diversos archivos de datos que los desarrolladores pueden utilizar para implementar el estándar Unicode en sus propios programas. Es una licencia laxa,

permisiva, compatible con todas las versiones de la GPL.

Si desea utilizar archivos cubiertos por este acuerdo de licencias en su propio software, no debería ser un problema, pero le aconsejamos que también incluya una copia completa del texto. Algunos archivos contienen cláusulas que no son libres, o ninguna información sobre la licencia, por lo que incluir una copia de este acuerdo ayuda a evitar confusión en caso que otras personas deseen distribuir su software. Por supuesto, también tendrá que seguir las condiciones de este acuerdo para distribuir los archivos, las cuales son bastante simples.

Por favor asegúrese que los archivos que esté usando estén cubiertos por este acuerdo. Otros archivos publicados por Unicode Inc. están bajo los Unicode Terms Of Use (términos de uso de Unicode), una licencia diferente y privativa que aparece en la misma página pero cubre distintos archivos. Una breve explicación al principio del acuerdo detalla cuáles son los archivos que cubre.

Por favor no utilice este acuerdo de licencias para su propio software. Si desea usar una licencia laxa, permisiva para su proyecto, por favor utilice la licencia Expat para programas pequeños y la Apache 2.0 para programas de una cierta envergadura. Estas dos son mucho más comunes y ampliamente reconocidas en la comunidad del software libre.

- Licencia Universal Permisiva (UPL)

Esta es una licencia de software libre laxa, permisiva, sin copyleft, compatible con la GPL de GNU. Esta licencia ofrece la posibilidad de conceder licencias sobre patentes junto con el software. No obstante, cuando se elija poner una obra bajo una licencia laxa, para evitar la trampa de las patentes recomendamos la licencia Apache 2.0.

- The Unlicense

La «Unlicense» es una contribución al dominio público. Un trabajo publicado bajo la «Unlicense» se entrega al dominio público con la extensión máxima permitida por la ley. También viene con una licencia laxa adicional que ayuda a cubrir los casos donde la entrega es inadecuada. Tanto las obras en el dominio público como la licencia laxa proporcionada por «Unlicense» son compatibles con la GPL de GNU.

Si desea publicar su obra en el dominio público, le recomendamos que use la CC0. CC0 también proporciona la entrega al dominio público, con una licencia alternativa, y es más concisa y madura que la «Unlicense».

- Licencia de Vim, versión 6.1 o posterior

Esta es una licencia de software libre, con copyleft parcial, pero no realmente. Es compatible con la GPL por una cláusula explícita de conversión.

- W3C Software Notice and License

Esta es una licencia de software libre compatible con GPL.

- License of WebM

La implementación del WebM de Google está cubierta por la Licencia BSD Modificada. Google también provee una licencia separada (llamada erróneamente «Additional IP Rights Grant») para patentes que controla o posee, y que inevitablemente son infringidas por su implementación del WebM. El software cubierto por la GPL se puede distribuir en cumplimiento de los términos de esta licencia: permite a los distribuidores ejercer todos los derechos que garantiza la GPL, cumpliendo a la vez con todas sus condiciones. Por lo tanto, todas las licencias del WebM son libres y compatibles con GPL.

- WTFPL, Versión 2

Esta es una licencia de software libre laxa, permisiva, sin copyleft, compatible con la GPL de GNU.

No recomendamos esta licencia. Si usted quiere usar una licencia laxa, permisiva, para un programa pequeño, le recomendamos la licencia X11. Los programas más significativos generalmente tienen que tener copyleft, pero si usted está decidido a usar una licencia laxa, permisiva, para un programa de este tipo, le aconsejamos que use la Apache 2.0 porque esta licencia protege a los usuarios contra la trampa de las patentes.

- Licencia de bibliotecas WxWidgets

La licencia WxWidgets es una licencia de software libre compatible con la GPL. Está constituida por la LGPL 2.0 (GPL Reducida, versión 2.0) o cualquier otra versión posterior, más una autorización adicional que permite poner las distribuciones binarias que usan la biblioteca bajo una licencia a elección del distribuidor (incluso privativa). Es un copyleft débil, incluso más débil que el de la LGPL, por lo que recomendamos su uso únicamente en circunstancias especiales.

- Licencia de bibliotecas WxWindows

Un antiguo nombre de la Licencia de bibliotecas WxWidgets.

- Licencia X11

Esta es una licencia de software libre laxa, permisiva, sin copyleft, compatible con la GPL de GNU. Versiones más antiguas de XFree86 utilizaban esta misma licencia, como así también algunas de las versiones actuales de XFree86. Las últimas versiones de XFree89 son distribuidas bajo la licencia XFree86 1.1.

Esta licencia es a veces llamada MIT license, pero este término es confuso, ya que MIT ha utilizado varias licencias distintas para software.

Esta licencia es aceptable para un programa pequeño. Los programas grandes generalmente tienen que tener copyleft, pero si usted está decidido a usar una licencia laxa, permisiva para uno de estos programas, le aconsejamos que use la Apache 2.0 pues tal licencia protege a los usuarios contra la trampa de las patentes.

- XFree86 1.1 License

Esta es una licencia de software libre laxa, permisiva, sin copyleft, compatible con la versión 3 de la GPL.

Nótese que esta licencia es incompatible con la versión 2 de la GPL debido a sus requisitos, que se aplican a toda la documentación en una distribución que contenga reconocimientos al trabajo de otros autores. Existen diversas variantes de XFree86, y solo algunas de ellas utilizan esta licencia. Otras continúan utilizando la licencia X11.

- Licencia de ZLib

Esta es una licencia de software libre compatible con la GPL.

- Zope Public License, versiones 2.0 y 2.1

Esta es una licencia de software libre laxa, permisiva, sin copyleft, compatible con la GPL de GNU.

Licencias de software libre incompatibles con la GPL

- Licencia Pública General Affero, versión 1

La Licencia Pública General Affero es una licencia de software libre con copyleft pero incompatible con la GPL de GNU. Consiste en la versión 2 de la GPL de GNU, con una sección adicional agregada por Affero con el consentimiento de la

FSF. La nueva sección, 2(d), cubre la distribución de programas de aplicación a través de servicios web o redes de computadoras.

Esta licencia ha sido reemplazada por la versión 3 de la Licencia Pública General Affero de GNU; por favor utilice esta última en su lugar.

- Academic Free License, todas las versiones hasta la 3.0

La Academic Free License es una licencia de software libre, sin copyleft e incompatible con la GPL de GNU. Las versiones recientes contienen ciertas cláusulas similares a las de la Open Software License, y deben evitarse por las mismas razones.

- Licencia Apache, versión 1.1

Esta es una licencia de software libre permisiva, sin copyleft. Tiene algunos requisitos que la vuelven incompatible con la GPL de GNU, como fuertes prohibiciones en el uso de nombres relacionados con Apache.

- Licencia Apache, versión 1.0

Esta es una licencia de software libre laxa, permisiva, sin copyleft, que contiene una cláusula publicitaria. Esto crea problemas en la práctica, como aquellos de la licencia BSD original, incluyendo la incompatibilidad con la GPL de GNU.

- Apple Public Source License (APSL), versión 2

Esta es una licencia de software libre incompatible con la GPL de GNU. Recomendamos no utilizarla para el software nuevo que se escriba, pero está bien usar y mejorar el software publicado bajo esta licencia.

- BitTorrent Open Source License

Esta es una licencia de software libre, pero incompatible con la GPL por las mismas razones que la Jabber Open Source License.

- Licencia BSD Original

A esta licencia también se la suele llamar «licencia BSD de 4 cláusulas».

Esta es una licencia de software libre laxa, permisiva, sin copyleft, con un serio fallo; la «nociva cláusula de publicidad de BSD». Dicho fallo no es fatal, o sea, no convierte en privativo el software que está bajo esta licencia, pero sí causa problemas en la práctica, incluyendo la incompatibilidad con la GPL de GNU.

Le instamos a no utilizar la licencia BSD original para el software que escriba. Si desea utilizar una licencia de software libre laxa, permisiva, sin copyleft, es mucho mejor utilizar la licencia BSD modificada, la licencia X11, o la Expat. Aún mejor, para programas de una cierta envergadura, es preferible usar la licencia Apache 2.0 pues implementa medidas contra las trampas de las patentes. Sin embargo, no existe razón para no utilizar programas publicados bajo la licencia BSD original.

- CeCILL-B versión 1

La CeCILL-B es una licencia de software libre. Es incompatible con la GPL porque incluye requisitos que no están presentes en la GPL. El requisito relativo a los reconocimientos de la sección 5.3.4 excede los de la GPL. Incluye también el extraño requisito de que haga el «máximo esfuerzo» para lograr que las terceras partes se avengan a «cumplir con las obligaciones establecidas en ese Artículo». No publique software bajo esta licencia.

- CeCILL-C versión 1

La CeCILL-C es una licencia de software libre con un copyleft débil al modo de la Licencia Pública General Reducida de GNU. Es incompatible con la GPL de GNU porque no contiene la cláusula explícita de compatibilidad con la GPL que se encuentra en la CeCILL básica. No publique software bajo esta licencia.

- Common Development and Distribution License (CDDL), versión 1.0

Esta es una licencia de software libre. Tiene un copyleft débil que se aplica a cada archivo (como la versión 1 de la Licencia Pública de Mozilla), lo cual la hace incompatible con la GPL de GNU. Esto significa que un módulo cubierto por la GPL y otro cubierto por la CDDL no pueden ser legalmente enlazados entre sí. Es por ello que urgimos a no utilizar la CDDL.

En el comunicado de la FSF *Interpreting, enforcing and changing the GNU GPL, as applied to combining Linux and ZFS* se ofrece un ejemplo ilustrativo de por qué no se deben combinar obras bajo licencia CDDL con obras bajo la GPL. También es lamentable que en la CDDL se use el término «propiedad intelectual».

- Common Public Attribution License 1.0 (CPAL)

Esta es una licencia de software libre. Se basa en la versión 1 de la Licencia Pública de Mozilla y es incompatible con la GPL por las mismas razones: posee varios requisitos para las versiones modificadas del software que no existen en la GPL. También exige que se publique la fuente del programa si se va a permitir su uso.

- Common Public License versión 1.0

Esta es una licencia de software libre. Lamentablemente, su copyleft débil y la elección de cláusulas legales la hacen incompatible con la GPL de GNU.

- Condor Public License

Las versiones recientes de Condor (de 6.9.5 en adelante) se publican bajo la Licencia Apache 2.0. Solo las versiones antiguas de Condor usan esta licencia.

La Condor Public License es una licencia de software libre. Contiene un par de requisitos que la hacen incompatible con la GPL de GNU, incluyendo fuertes restricciones en el uso de nombres relacionados con Condor, y exige que quienes

redistribuyen «representen y garanticen» que cumplirán con las leyes de exportación de los Estados Unidos de América (si este cumplimiento fuera una condición efectiva de la licencia, no sería una licencia de software libre).

- Eclipse Public License, versión 1.0

La Eclipse Public License es similar a la Common Public License, y nuestros comentarios sobre la CPL se aplican igualmente a la EPL. El único cambio es que la EPL elimina el léxico general sobre represalias en los casos de juicios por infracción de patentes, específicamente contra los colaboradores del programa cubierto por la EPL.

- Licencia Pública Eclipse, versión 2.0

En lo referente a la compatibilidad con la GPL, la versión 2.0 de la Licencia Pública Eclipse es prácticamente equivalente a la versión version 1.0. El único cambio es que ofrece de forma explícita la posibilidad de designar a la versión 2 o posterior de la GPL de GNU como «licencia secundaria» para una determinada parte del código.

Si un colaborador inicial publica una parte específica de código y designa la versión 2 de la GPL de GNU como licencia secundaria, eso garantiza explícitamente la compatibilidad de ese código con tales versiones de la GPL. (Para los usuarios, esto es aproximadamente equivalente a publicar ese fragmento de código bajo una licencia dual EPL | GPL). No obstante, sin esta designación la EPL2 sigue siendo incompatible con la GPL.

- Licencia Pública de la Unión Europea (EUPL) versión 1.1

Esta es una licencia de software libre. Por sí misma posee un copyleft comparable con el de la GPL e incompatible con el mismo. Permite la distribución la obra según los términos de otras licencias seleccionadas, algunas de las cuales tienen un

copyleft débil, como por ejemplo la Eclipse Public License y la Common Public License. Por lo tanto, los desarrolladores no pueden contar con esta licencia si desean un copyleft fuerte.

La EURL permite cambiar por la GPLv2, porque la GPLv2 está en la lista de licencias alternativas que los usuarios pueden elegir. También permite, indirectamente, cambiar la licencia por la GPL versión 3, porque existe una manera de cambiar por la CeCILL v2, y a su vez la CeCILL v2 permite cambiar por cualquier versión de la GPL de GNU.

Para hacer este cambio de licencias de dos pasos, en primer lugar es necesario escribir un poco de código y licenciarlo bajo la CeCILL v2, o encontrar algún módulo que ya esté disponible en esa licencia y añadirlo al programa. Añadiendo ese código al programa cubierto por la EURL se obtienen las bases para cambiar la licencia del programa a la CeCILL v2. Luego es necesario escribir una porción de código y licenciarlo bajo la GPLv3+, o encontrar un módulo adecuado que ya esté disponible bajo la GPLv3+ y añadirlo al programa. Añadiendo ese código al programa cubierto por la CeCILL se obtienen las bases para cambiar la licencia del programa a la GPLv3+.

- Licencia Pública de la Unión Europea (EURL) versión 1.2

Esta es una licencia de software libre. Por sí misma posee un copyleft comparable al de la GPL e incompatible con él. No obstante, permite la distribución la obra según los términos de otras licencias seleccionadas, y algunas de ellas (la Licencia Pública Eclipse, en particular) tienen un copyleft más débil. Por lo tanto, los desarrolladores no pueden contar con esta licencia si desean un copyleft fuerte.

La EURL permite cambiar por la GPLv2 solo y por la GPLv3 solo, ya que estas

licencias están en la lista de licencias alternativas que los usuarios pueden elegir. También permite, indirectamente, cambiar la licencia por la GPL versión 3, pues existe una manera de cambiar por la CeCILL v2, y a su vez la CeCILL v2 permite cambiar por cualquier versión de la GPL de GNU.

Para hacer este cambio de licencias en dos pasos, en primer lugar es necesario escribir un poco de código y licenciarlo bajo la CeCILL v2, o encontrar algún módulo que ya esté disponible en esa licencia y añadirlo al programa. Al añadir ese código al programa cubierto por la EUPL se sientan las bases para cambiar la licencia del programa a la CeCILL v2. Luego es necesario escribir una porción de código y licenciarlo bajo la GPLv3+, o encontrar un módulo adecuado que ya esté disponible bajo la GPLv3+ y añadirlo al programa. Al añadir ese código al programa cubierto por la CeCILL se sientan las bases para cambiar la licencia del programa a la GPLv3+.

- Licencia Fraunhofer FDK AAC

Esta es una licencias de software libre, pero hasta ahí llega. Es incompatible con cualquier versión de la GPL de GNU.

Esta licencia contiene una cláusula especialmente peligrosa que dice que no se concede al usuario ninguna licencia sobre patentes, y le invita a comprar alguna. Debido a esto, y a que el autor de la licencia es conocido por sus agresiones con patentes, le aconsejamos que tenga mucho cuidado si utiliza o redistribuye software que esté bajo esta licencia: debe primero considerar si el licenciador no podría estar lanzándole el anzuelo para que cometa una infracción de patentes. Si llega a la conclusión de que el programa es un cebo para que caiga en la trampa de las patentes, lo más sensato sería evitar el programa.

Es posible que las patentes pertinentes hayan expirado. Dependiendo de si Fraunhofer tiene aún patentes en vigor que cubran la obra, el software podría ser una trampa, o no. (Por supuesto, cualquier programa está potencialmente amenazado por las patentes, y la única manera de poner fin a esta amenaza es cambiar la ley de patentes para poner el software a salvo de las patentes.)

- Licencia Gnuplot

Esta es una licencia de software libre incompatible con la GPL de GNU.

- Licencia Pública IBM, versión 1.0

Esta es una licencia de software libre. Lamentablemente contiene cláusulas legales que la hacen incompatible con la GPL de GNU.

- Jabber Open Source License, versión 1.0

Esta es una licencia de software libre incompatible con la GPL. Permite cambiar la licencia por otras de cierto tipo, aquellas que incluyen todos los requisitos de la licencia Jabber. La GPL no está entre ese tipo de licencias, por lo que la licencia Jabber no permite el cambio a la GPL. Por lo tanto no es compatible.

- LaTeX Project Public License 1.3a

No hemos escrito un análisis completo de esta licencia, pero es una licencia de software libre, con requisitos de distribución menos restrictivos que la LPPL 1.2 (ver la siguiente). Es incompatible con la GPL porque algunas versiones modificadas deben incluir una copia (o hacer referencia a) una versión no modificada.

- LaTeX Project Public License 1.2

Esta licencia es una declaración incompleta de los términos de distribución de LaTeX. Como licencia, es de software libre, pero incompatible con la GPL por poseer varios requisitos que esta última no posee.

Esta licencia posee complejas y molestas restricciones en cuanto a la manera de

publicar una versión modificada, incluyendo un requisito que cae solo por poco en el lado bueno de la línea que establece lo que es aceptable: que cualquier archivo modificado debe tener un nuevo nombre.

La razón por la cual este requisito es aceptable para LaTeX, es que TeX posee una función que permite mapear nombres de archivos para especificar «utiliza el archivo ABC cuando se solicita el archivo XYZ. Con esta herramienta, el requisito es simplemente molesto; sin ella, el requisito sería un serio obstáculo, y tendríamos que concluir que dicho requisito hace que el programa no sea libre.

Esta condición puede causar problemas con algunas modificaciones relevantes. Por ejemplo, si se quisiera migrar una obra cubierta por la LPPL a otro sistema que carezca de una herramienta similar para el mapeado, pero que aún así obligase a los usuarios a solicitar archivos por su nombre, podría ser necesario implementar tal herramienta para que el programa sea libre. Esto sería una molestia, pero el hecho de que una licencia pueda hacer que el código no sea libre si se migra a un contexto muy diferente, no hace que no sea libre en el contexto original.

La LPPL dice que algunos archivos, en ciertas versiones de LaTeX, pueden tener restricciones adicionales que podrían convertirlos en privativos. Por esta razón, es muy probable que para obtener una versión de LaTeX que sea software libre sea necesario efectuar una cuidadosa revisión.

La LPPL hace la controvertida afirmación de que el simple hecho de tener archivos en una máquina a la que unas pocas personas pueden conectarse para acceder a los mismos, constituye distribución. Creemos que los tribunales no sostendrán esta afirmación, pero no es bueno que el público comience a hacerla. Por favor no

utilice esta licencia para ningún otro proyecto.

- Lucent Public License, versión 1.02 (Plan 9 license)

Esta es una licencia de software libre, pero incompatible con la GPL de GNU por sus cláusulas legales. Recomendamos que no utilice esta licencia para el software nuevo que se escribe, aunque sí está bien utilizar y hacer mejoras al Plan 9 bajo esta licencia.

- Microsoft Public License (Ms-PL)

Esta es una licencia de software libre; posee un copyleft débil, es incompatible con la GPL de GNU. Es por esta razón que le instamos a no utilizar esta licencia.

- Microsoft Reciprocal License (Ms-RL)

Esta es una licencia de software libre basada en la Microsoft Public License, posee una cláusula que hace su copyleft apenas un poco más fuerte. También es incompatible con la GPL de GNU. Le instamos a no utilizarla por dicha razón.

- Licencia Pública de Mozilla (MPL), versión 1.1

Esta es una licencia de software libre que no posee un copyleft fuerte. A diferencia de la licencia X11, posee algunas restricciones complejas que la hacen incompatible con la GPL de GNU. En concreto, un módulo cubierto por la GPL y otro por la MPL no pueden ser legalmente enlazados. Por esta razón le instamos a no utilizar la MPL 1.1.

Sin embargo, la MPL 1.1 posee una previsión (sección 13) que permite a un programa (o a partes del mismo) ofrecer la elección de otra licencia simultánea. Si parte de un programa permite el uso de la GPL de GNU como alternativa, o cualquier otra licencia compatible con la GPL, dicha parte del programa está bajo una licencia compatible con la GPL.

La versión 2.0 de la MPL contiene mejoras, incluyendo la compatibilidad predefinida con la GPL. Para información más detallada, véase el artículo correspondiente.

- Netizen Open Source License (NOSL), versión 1.0

Esta es una licencia de software libre que, en esencia, es la misma que la Licencia Pública de Mozilla versión 1.1. Tal como la MPL, la NOSL posee restricciones complejas que la hacen incompatible con la GPL de GNU. Esto significa que un módulo cubierto por la GPL y otro por la NOSL no pueden ser legalmente enlazados. Por esta razón urgimos a no utilizar esta licencia.

- Licencia Pública de Netscape (NPL), versiones 1.0 y 1.1

Esta es una licencia de software libre, sin copyleft fuerte, incompatible con la GPL de GNU. Consiste en la Licencia Pública de Mozilla versión 1.1 con una cláusula agregada que permite a Netscape utilizar el código que se añade, incluso en las versiones privativas de dicho programa. Por supuesto que no le otorgan a usted permisos para utilizar el código de ellos en forma análoga. Le urgimos a no utilizar esta licencia.

- Nokia Open Source License

Es similar a la Licencia Pública de Mozilla versión 1: una licencia de software libre incompatible con la GPL de GNU.

- Old OpenLDAP License, versión 2.3

Esta es una licencia de software libre permisiva y sin copyleft, con un par de requisitos (en secciones 4 y 5) que la vuelven incompatible con la GPL de GNU. Nótese que la última versión de la OpenLDAP posee una licencia diferente que sí es compatible con la GPL de GNU.

Le urgimos a no utilizar la antigua versión de la licencia OpenLDAP para el

software que usted escriba. Sin embargo, no hay razón para no ejecutar programas que han sido publicados bajo esta licencia.

- Open Software License, todas las versiones hasta la 3.0

La Open Software License es una licencia de software libre. Es incompatible con la GPL de GNU por varios motivos.

Las versiones recientes de la Open Software License poseen una cláusula que requiere que los distribuidores intenten obtener la aceptación explícita de la licencia. Esto implica que distribuir software OSL en sitios FTP comunes, enviar parches a las listas de correo electrónico comunes, o guardar el software en un sistema de control de versiones común, constituye posiblemente un incumplimiento de la licencia e implicaría una posible terminación de la misma. Por eso, la Open Software License dificulta el desarrollo de software libre con las herramientas habituales. Por esta razón, y por ser incompatible con la GPL, recomendamos no utilizar ninguna versión de la OSL para ningún tipo de software.

Le urgimos a no utilizar la Open Software License para el software que escriba. Sin embargo, no hay razón para no utilizar programas que se encuentren bajo esta licencia.

- OpenSSL license

La licencia de OpenSSL es una mezcla de dos licencias, una es la llamada «Licencia OpenSSL» y la otra es la licencia de SSLeay. Se debe cumplir con ambas. La combinación da como resultado una licencia de software libre con copyleft, incompatible con la GPL de GNU. También posee una cláusula de publicidad, como en la licencia BSD original y como en la licencia Apache 1.

Recomendamos utilizar GNUTLS en lugar de OpenSSL para el software que escriba. Sin embargo no hay razón para no utilizar OpenSSL y aplicaciones que funcionen con OpenSSL.

- Phorum License, versión 2.0

Esta es una licencia de software libre incompatible con la GPL. La sección 5 es la que la hace incompatible.

- Licencia PHP, versión 3.01

Esta licencia se utiliza generalmente para PHP4. Es una licencia de software libre sin copyleft. Es incompatible con la GPL de GNU porque contiene fuertes restricciones sobre el uso de «PHP» en el nombre de los productos derivados. Recomendamos no utilizar esta licencia salvo para añadidos a PHP.

- Licencia de Python 1.6b1 a 2.0 y 2.1

Esta es una licencia de software libre incompatible con la GPL de GNU. La incompatibilidad principal es que la licencia de Python está gobernada por las leyes del Estado de Virginia, en los Estados Unidos de América, y la GPL no lo permite.

- Q Public License (QPL), versión 1.0

Esta es una licencia de software libre sin copyleft e incompatible con la GPL de GNU. También causa grandes inconvenientes prácticos, porque las fuentes modificadas solo pueden distribuirse como parches.

Recomendamos evitar el uso de la QPL para todo lo que se escriba y utilizar software cubierto por ella solo en caso de ser absolutamente necesario. Sin embargo, esta salvedad ya no se aplica a Qt, porque Qt se ha publicado bajo la GPL de GNU.

Como la QPL es incompatible con GPL de GNU, no se puede coger un programa cubierto con la GPL y enlazarlo con uno cubierto con la QPL, de ninguna manera.

Sin embargo, si usted ha escrito un programa que utiliza una biblioteca cubierta por QPL (llamémosla XYZ) y desea publicarlo bajo la GPL de GNU, puede hacerlo fácilmente. El conflicto para su programa lo puede resolver agregando una nota como la siguiente:

Puede hacerlo, legalmente, si es el titular del copyright del programa. Agréguelo en los archivos fuente después de una nota que diga que el programa está cubierto por la GPL de GNU.

- RealNetworks Public Source License (RPSL), versión 1.0

La RPSL es una licencia de software libre, incompatible con la GPL por una serie de razones: requiere que los trabajos derivados sean puestos bajo esta misma licencia, y ordena que cualquier litigio se lleve a cabo en Seattle, Washington.

- Sun Industry Standards Source License 1.1

Esta es una licencia de software libre, sin un copyleft fuerte, incompatible con la GPL de GNU por detalles más que por cuestiones de política relevantes.

- Sun Public License

Es esencialmente igual a la Licencia Pública de Mozilla versión 1: una licencia de software libre incompatible con la GPL de GNU. Por favor no la confunda con la Sun Community Source License, que no es una licencia de software libre.

- Licencia de xinetd

Esta es una licencia de software libre con copyleft, incompatible con la GPL. La incompatibilidad se debe a que impone restricciones adicionales a la redistribución de versiones modificadas que contradicen los requisitos de redistribución

especificados en la GPL.

- Yahoo! Public License 1.1

Esta es una licencia de software libre. Posee un copyleft similar al de la Licencia Pública de Mozilla. También hace una elección legal particular en la sección 7. Ambas características la hacen incompatible con la GPL. A su vez, lamentablemente esta licencia utiliza el término «propiedad intelectual».

- Licencia Zend, versión 2.0

Esta licencia se utiliza en una parte de PHP4. Es una licencia de software libre sin copyleft, incompatible con la GPL de GNU, y tiene problemas prácticos como aquellos de la BSD original.

Recomendamos que no utilice esta licencia para nada de lo que escriba.

- Zimbra Public License 1.3

Esta licencia es idéntica a la Licencia Pública de Yahoo! 1.1, excepto que la licencia la proporciona VMWare en lugar de Yahoo!. Nuestras observaciones también se aplican en este caso. Es una licencia incompatible con la GPL, parcialmente con copyleft y de software libre.

- Zope Public License version 1

Esta es una licencia de software libre laxa, bastante permisiva, sin copyleft, que presenta problemas prácticos al igual que la licencia BSD original, incluyendo incompatibilidad con la GPL de GNU.

Le urgimos a que evite utilizar la ZPL versión 1 para el software que escriba. Sin embargo no existe razón para evitar ejecutar programas publicados bajo esta licencia, como versiones previas de Zope. La versión 2.0 de la Licencia Pública de Zope es compatible con la GPL.

Licencias para software que no son libres

Las siguientes licencias no se califican como de software libre. Una licencia que no es libre es automáticamente incompatible con la GPL de GNU.

Por supuesto, le urgimos a evitar el uso de licencias de software que no son libres y el uso de software privativo en general.

No hay manera posible de incluir aquí todas las licencias privativas conocidas. Después de todo, cada compañía productora de software privativo posee la suya. Nos concentraremos en licencias que son frecuentemente confundidas como de software libre, pero que de hecho no lo son.

- Sin licencia

Si el código fuente no está acompañado de una licencia que otorgue a los usuarios las cuatro libertades esenciales, no es software libre, a menos que de manera explícita y válida se haya puesto en el dominio público.

Algunos programadores piensan que el código que no tiene licencia está automáticamente en el dominio público. Según la actual ley de copyright, no es así. Más bien, todas las obras que se pueden poner bajo copyright, incluyendo los programas informáticos, tienen copyright si no se especifica lo contrario. Los usuarios no tienen ninguna libertad si no existe ninguna licencia que la otorgue. En algunos países, los usuarios que descargan código desprovisto de licencia pueden violar el copyright simplemente si compilan o ejecutan el código.

Para que un programa sea libre, quienes poseen el copyright sobre el mismo deben otorgar a los usuarios las cuatro libertades esenciales. El documento que se

utiliza para otorgar las libertades se denomina licencia de software libre. Es esta la función de las licencias de software libre.

En algunos países los autores están autorizados a poner el código en el dominio público, pero para eso es necesario seguir un procedimiento específico. Si desea poner su código en el dominio público, el método que recomendamos es usar la licencia CC0, que también sirve en otros países como licencia más o menos equivalente al dominio público. Sin embargo, en la mayoría de los casos es mejor que el código tenga copyleft para asegurarse de que las libertades lleguen a todos los usuarios del mismo.

El código escrito por los empleados del Gobierno de los Estados Unidos de América constituye una excepción especial debido a que, según la ley de copyright de ese país, dicho código está en el dominio público. Sin embargo, esto no se aplica a las obras que una empresa escribe por encargo del Gobierno de los EE. UU. Tampoco se aplica a otros países, en muchos de los cuales está permitido que el Estado posea el copyright de las obras escritas por Gobierno.

- Aladdin Free Public License

A pesar de su nombre, esta no es una licencia de software libre porque no permite cobrar las distribuciones y prohíbe empaquetar software bajo ella con cualquier otro tipo de software por el cual se cobre un precio.

- Apple Public Source License (APSL), versión 1.x

Las versiones 1.0, 1.1 y 1.2 no son licencias de software libre. Por favor no utilice estas licencias ni software que esté bajo las mismas. La versión 2.0 de la APSL sí es de software libre.

- Licencia Artística 1.0

No podemos decir que esta sea una licencia de software libre porque es demasiado ambigua; algunos pasajes son demasiado ingeniosos para su propio beneficio, y su significado no queda claro. Le urgimos a evitar su uso, excepto como parte de la disyuntiva licencia de Perl.

- Licencia Pública de AT&T

La Licencia Pública de AT&T no es una licencia libre. Presenta varios problemas serios:

- La licencia sobre patentes queda anulada por cualquier modificación del código respectivo, sin importar cuan pequeña sea.
- Se debe exigir un acuerdo por escrito cuando se distribuyen las fuentes o parches.
- Requiere que se notifique a AT&T si se distribuye un parche.
- La licencia puede quedar terminada sin que exista culpa del usuario, bajo la sección 8/3.
- Pone como condición de la licencia el cumplimiento de las leyes de control de exportaciones.
- Algunas versiones de la licencia requieren que se presten servicios de soporte.
- Algunas versiones estipulan que no se puede vender una copia del software a un precio mayor que el coste de la distribución.

La licencia posee otras características odiosas:

- Otorga privilegios amplios a AT&T, que van mucho más allá de la utilización del código que usted ha escrito, incluso su código modificado.
- Afirma que es necesario tener una licencia de AT&T para poner un enlace al sitio web de esta compañía. Esto no es un problema práctico, ya que la licencia dice que otorga el permiso para realizar dicho enlace (de todas maneras, nadie debería poner enlaces a sitios sobre software que no es libre).

Pero no se debería poner tal requisito, ni propagarlo.

- Code Project Open License, versión 1.02

La licencia Code Project Open License no es de software libre. La sección 5.6 restringe el modo en que se puede usar la obra. La sección 5.4 impide la distribución comercial del software por sí mismo, y según cómo se interprete la sección 3.4, puede que tampoco tenga permiso para distribuir el software de ningún modo.

- Commons Clause

La «Cláusula de los bienes comunes» es una licencia no libre, ya que prohíbe vender copias del programa, e incluso ejecutar el programa si se emplea para poner en marcha algún servicio comercial. Para colmo, tergiversa los términos «bienes comunes» y «vender».

Exhortamos a la gente a rechazar los programas bajo esta licencia y a desarrollar alternativas libres. Si se dispone de una versión previa en forma de software libre, una posibilidad es continuar el desarrollo de dicha versión.

- CNRI Digital Object Repository License Agreement

Esta licencia no es libre debido al Artículo 3, que incluiría el requisito de no violar la licencia de ningún programa que el usuario ejecute, incluso programas privativos.

- eCos Public License, versión 1.1

Esta era la vieja licencia de eCos. No es una licencia de software libre porque requiere enviar todas las versiones modificadas que se publiquen a un desarrollador inicial especificado. También existe léxico en la licencia de cuyo significado no estamos seguros y que podría resultar problemático.

Al día de hoy eCos está disponible bajo la GPL de GNU con permisos adicionales

para el enlazado con programas privativos.

- GPL para Programas de Computación de la Administración Pública

La GPL-PA (cuyo nombre original en Portugués es «Licença Pública Geral para Administração Pública») no es libre por varias razones:

1. Permite su uso únicamente en «circunstancias normales».
2. No permite la distribución del código fuente sin los binarios.
3. Sus permisos expiran después de 50 años.

- Hacktivismo Enhanced-Source Software License Agreement (HESSLA)

No es una licencia de software libre porque limita lo que la persona puede hacer con el software y también limita sustancialmente lo que se puede hacer con las versiones modificadas del programa.

- Jahia Community Source License

La Jahia Community Source License no es de software libre. El uso del código fuente está limitado a fines de investigación.

- The JSON License

Esta es la licencia de la implementación original del formato JSON de intercambio de datos. Esta licencia utiliza como base la Expat, pero agrega una cláusula que exige lo siguiente: «El software deberá ser utilizado para el Bien y no para el Mal». Esta es una restricción al uso y por lo tanto entra en conflicto con la libertad 0. Puede que resulte imposible hacer cumplir esta restricción, pero no podemos hacer conjeturas. Como consecuencia, esta licencia no es libre.

- Antigua licencia de ksh93

ksh93 solía venir con una licencia original que no era de software libre. Una de las razones es que requería que todas las modificaciones se enviaran al desarrollador.

- Licencia de Lha

La licencia lha debe ser considerada como no libre porque es tan ambigua que no se puede estar seguro qué permisos otorga.

- Microsoft's Shared Source CLI, C#, y Jscript License

Esta licencia no permite la distribución comercial, solo concede permiso de uso para fines comerciales bajo ciertas circunstancias. Microsoft posee otras licencias que describe como «Shared Source», algunas de las cuales poseen diferentes restricciones.

- NASA Open Source Agreement

La NASA Open Source Agreement, versión 1.3, no es una licencia de software libre porque incluye una previsión que requiere que los cambios hechos sean un «diseño original». El desarrollo de software libre se basa en combinar código de terceras partes y la licencia de NASA no lo permite.

Urgimos a no se utilizar esta licencia. Además, si usted es ciudadano de los Estados Unidos de América, por favor escriba a la NASA solicitándole que utilice una verdadera licencia de software libre.

- Licencia Oculus Rift SDK

Esta no es una licencia de software libre; contiene varios defectos graves.

1. Solo se puede redistribuir el programa libOVR completo.
2. Los derechos de redistribución pueden determinarse en base a condiciones imprecisas.
3. Quienes hacen versiones modificadas están obligados a enviarlas a Oculus cuando así se les solicite.
4. Se permite el uso únicamente con su producto.
5. Las versiones anteriores de las licencias se reemplazan completamente con versiones nuevas, lo cual significa que se pueden quitar los permisos que se

habían otorgado.

Puede que tenga otros errores graves; después de haber visto estos ya no buscamos más.

- Open Public License

Esta no es una licencia de software libre porque requiere enviar cada versión modificada que se publique a un desarrollador inicial específico. Existen, a su vez, términos en dicha licencia de cuyo significado no estamos seguros y que podrían ser problemáticos.

- Peer-Production License

La licencia Peer-Production no es de software libre porque pone límites con respecto a quiénes pueden redistribuir el programa y para qué propósito. Tampoco otorga a nadie el permiso para ejecutar el programa.

La licencia Peer-Production contiene varias cláusulas específicamente diseñadas para representaciones artísticas, y no tenemos nada en contra de su uso para obras de arte, aunque sabemos que hay quienes abogan por su uso también para el software. Esta licencia no debe utilizarse para software, manuales u otras obras que deben ser libres.

- Licencia de PINE

La licencia de PINE no es de software libre porque en la mayoría de los casos prohíbe la distribución de versiones modificadas. También restringe los medios que se pueden utilizar para vender copias.

Nótese que Alpine, sucesor de Pine, se encuentra bajo la Licencia Apache, versión 2.0.

- Old Plan 9 license

Esta licencia no es de software libre; carece de libertades esenciales como por ejemplo la libertad de realizar y utilizar versiones modificadas en forma privada. Por supuesto, se debe evitar el uso de esta licencia y de cualquier programa que se encuentre bajo la misma. Véase [Un análisis detallado de esta licencia](#).

En septiembre de 2002 se observó que la licencia publicada para Plan 9 había sido modificada, agregando aún más restricciones, aunque estaba todavía fechada 20/09/00. Sin embargo, un cambio posterior en 2003 convirtió a Plan 9 en software libre.

- Reciprocal Public License

La Licencia Pública Recíproca es una licencia que no es libre debido a tres problemas:

1. Impone límites al precio que se puede cobrar por la copia inicial.
2. Requiere que se notifique al autor original cada vez que se publique una versión modificada.
3. Requiere la publicación de toda versión modificada utilizada por cualquier organización, incluso si se usa en forma privada.

- Licencia Scilab

Esta no es una licencia de software libre porque no permite la distribución comercial de versiones modificadas. Por suerte, a partir de la versión 5.0.0, Scilab es software libre, publicado bajo la CeCILL versión 2.

- Licencia Scratch 1.4

Esta no es una licencia de software libre porque no permite la redistribución comercial. Además, la cláusula 4 restringe substancialmente la funcionalidad de las obras modificadas.

Hay nuevas versiones de software Scratch distribuidas bajo la GPL de GNU, pero no recomendamos algunas de esas nuevas versiones porque dependen del software privativo Adobe Air.

- Simple Machines License

A pesar de su nombre, es una licencia para software, y no es libre por varias razones:

1. Debe obtenerse el permiso del licenciatarario antes de distribuir el software.
2. No se permite vender copias del software.
3. Es posible que los permisos otorgados por la licencia caduquen en caso de haberse obtenido la copia de alguien que no cumplió con lo establecido por la licencia.

- Antigua Licencia Squeak

La licencia Squeak original, aplicada al software, no es una licencia de software libre porque requiere que todos los usuarios, cualquiera sea el país en el que se encuentren, obedezcan a las leyes de control de exportaciones de los Estados Unidos de América. Cuando se aplica a los tipos de letras, tampoco permite su modificación.

Además, requiere que los usuarios indemnicen al desarrollador, lo cual es suficiente para que los usuarios lo piensen dos veces antes de utilizar esta licencia. Las versiones recientes de Squeak están publicadas bajo una licencia estilo Expat con algunas partes del código puestas bajo la Licencia Apache 2.0.

- Sun Community Source License

Esta no es una licencia de software libre. Carece de libertades esenciales, como la de publicar versiones modificadas. Por favor no utilice esta licencia y evite cualquier software publicado bajo la misma.

- Sun Solaris Source Code (Foundation Release) License, Versión 1.1

Esta no es una licencia de software libre. Prohíbe la redistribución como así también el uso comercial del software, y puede ser revocada.

- Sybase Open Watcom Public License, versión 1.0

No es una licencia de software libre. Obliga a que se publique el código fuente siempre que se «despliegue» el software en cuestión, y en la definición de «desplegar» se incluyen muchas formas de uso privado.

- Licencia SystemC «Open Source», versión 3.0

Esta licencia requiere que todos aquellos que reciban el software ayuden activamente al licenciatarario en la protección de sus marcas comerciales. No es razonable imponer tal condición a los derechos de los usuarios, por lo que la licencia no es libre. También posee otros problemas prácticos: algunos de sus requisitos son ambiguos y utiliza el término «propiedad intelectual».

A pesar del nombre, no está claro si esta licencia entra dentro de la categoría «open source». De todos modos, nuestra evaluación no se basa en eso.

- Licencia Truecrypt 3.0

Esta licencia no es libre por varias razones. Dice que si no se entiende la licencia no se puede usar el programa. Impone condiciones para consentir la ejecución de una copia del programa. Impone condiciones para programas distintos que «dependen de» Truecrypt. La condición acerca de la marca registrada se aplica a «materiales relacionados».

Hay otros puntos de la licencia que podrían ser inaceptables, y ante la incertidumbre habíamos postergado la publicación de nuestra evaluación. La publicamos ahora para explicar que no lamentamos la desaparición de Truecrypt. Existen pro-

gramas libres que cumplen la misma función.

- Licencia Pública de la Research Foundation de la Universidad de Utah

La Licencia Pública de la Research Foundation de la Universidad de Utah no es una licencia libre porque no permite la redistribución con fines comerciales. También pretende impedir la utilización del software con fines comerciales, e incluso brindar asesoramiento sobre el programa bajo pago. Probablemente resulte imposible hacer cumplir tales restricciones bajo las leyes de copyright de los Estados Unidos de América, pero puede que sea posible en otros países; el mero hecho de imponerlas es vergonzoso.

El uso de esta licencia por parte de la Universidad de Utah es un ejemplo de la peligrosa tendencia en las universidades de restringir el conocimiento en lugar de contribuir a su esparcimiento público.

Si una universidad intenta imponer una licencia como esta al software que usted está escribiendo, no pierda las esperanzas. Con firmeza, perseverancia y adecuada planificación, es posible derrotar a los administradores avaros de las universidades.

- Licencia YaST

Esta no es una licencia de software libre. Prohíbe la distribución con fines de lucro e imposibilita la inclusión de programas en muchas colecciones de software libre hechas en CD-ROM, como las que venden algunas compañías y organizaciones.

2.11. Herramientas para la creacion de la plataforma

2.11.1. PHP (Hypertext Preprocessor)

PHP es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en el documento HTML en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página Web resultante. PHP ha evolucionado por lo que ahora incluye también una interfaz de línea de comandos que puede ser usada en aplicaciones gráficas independientes. Puede ser usado en la mayoría de los servidores web al igual que en casi todos los sistemas operativos y plataformas sin ningún costo.

PHP se considera uno de los lenguajes más flexibles, potentes y de alto rendimiento conocidos hasta el día de hoy, lo que ha atraído el interés de múltiples sitios con gran demanda de tráfico, como Facebook, para optar por el mismo como tecnología de servidor. Fue creado originalmente por Rasmus Lerdorf en 1995. Actualmente el lenguaje sigue siendo desarrollado con nuevas funciones por el grupo PHP. Este lenguaje forma parte del software libre publicado bajo la licencia PHP, que es incompatible con la Licencia Pública General de GNU debido a las restricciones del uso del término PHP.

Características.

- Orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos.
- Es considerado un lenguaje fácil de aprender, ya que en su desarrollo se simplificaron distintas especificaciones, como es el caso de la definición de las variables primitivas, ejemplo que se hace evidente en el uso de php arrays.
- El código fuente escrito en PHP es invisible al navegador web y al cliente, ya que

es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador.

- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.
- Capacidad de expandir su potencial utilizando módulos (llamados ext's o extensiones).
- Posee una amplia documentación en su sitio web oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite aplicar técnicas de programación orientada a objetos.
- No requiere definición de tipos de variables aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución.
- Tiene manejo de excepciones (desde PHP5).
- Si bien PHP no obliga a quien lo usa a seguir una determinada metodología a la hora de programar, aún haciéndolo, el programador puede aplicar en su trabajo cualquier técnica de programación o de desarrollo que le permita escribir código ordenado, estructurado y manejable. Un ejemplo de esto son los desarrollos que en PHP se han hecho del patrón de diseño Modelo Vista Controlador (MVC), que permiten separar el tratamiento y acceso a los datos, la lógica de control y la interfaz de usuario en tres componentes independientes.
- Debido a su flexibilidad ha tenido una gran acogida como lenguaje base para las aplicaciones WEB de manejo de contenido, y es su uso principal.

2.11.2. CodeIgniter

Probablemente ya sepamos que un framework es un programa para desarrollar otros programas, CodeIgniter, por tanto, es un programa o aplicación web desarrollada en PHP para la creación de cualquier tipo de aplicación web bajo PHP. Es un producto de código libre, libre de uso para cualquier aplicación.

Como cualquier otro framework, Codeigniter contiene una serie de librerías que sirven para el desarrollo de aplicaciones web y además propone una manera de desarrollarlas que debemos seguir para obtener provecho de la aplicación. Esto es, marca una manera específica de codificar las páginas web y clasificar sus diferentes scripts, que sirve para que el código esté organizado y sea más fácil de crear y mantener. CodeIgniter implementa el proceso de desarrollo llamado Model View Controller (MVC), que es un estándar de programación de aplicaciones, utilizado tanto para hacer sitios web como programas tradicionales.

CodeIgniter no es magia, pero contiene muchas ayudas para la creación de aplicaciones PHP avanzadas, que hacen que el proceso de desarrollo más rápido. A la vez, define una arquitectura de desarrollo que hará que programemos de una manera más ordenada y contiene diversas herramientas que ayudan a hacer aplicaciones más versátiles y seguras.

CodeIgniter y otros frameworks PHP pueden ayudarte a dar el salto definitivo como desarrollador PHP, creando aplicaciones web más profesionales y con código más reutilizable, con la diferencia que Code Igniter está creado para que sea fácil de instalar en cualquier servidor y de empezar a usar que cualquier otro framework. Además muchas de sus utilidades y modos de funcionamiento son opcionales, lo que hace que goces de mayor libertad a la hora de desarrollar sitios web.

Ventajas.

- Las páginas se procesan más rápido, el núcleo de CodeIgniter es bastante ligero.
- Es sencillo de instalar, basta con subir los archivos al ftp y tocar un archivo de confi-

guración para definir el acceso a la bd.

- Reutilización de código, desarrollo ágil.
- Existe abundante documentación en la red.
- Facilidad de edición del código ya creado.
- Facilidad para crear nuevos módulos, páginas o funcionalidades.
- Acceso a librerías públicas y clases. Entre otras, hay librerías para el login, paginador, calendarios, fechas.
- Estandarización del código. Fundamental cuando hay que tocar código hecho por otra persona o cuando trabaja más de una persona en un mismo proyecto.
- URLs amigables con SEO. Hoy en día creo que nadie duda de la importancia del posicionamiento web.
- Separación de la lógica y arquitectura de la web, el MVC.
- CodeIgniter es bastante menos rígido que otros frameworks. Define una manera de trabajar, pero podemos seguirla o no (esto puede convertirse en un inconveniente también)
- Cualquier servidor que soporte PHP+MySQL sirve para CodeIgniter.
- CodeIgniter se encuentra bajo una licencia open source, es código libre.
- CodeIgniter usa una versión modificada del Patrón de Base de Datos Active Record. Este patrón permite obtener, insertar y actualizar información en tu base de datos con mínima codificación. Permite queries más seguras, ya que los valores son escapadas automáticamente por el sistema.

2.11.3. Atom

Atom es un editor de código fuente y de texto de código abierto para macOS , Linux y Microsoft Windows con soporte para complementos escritos en Node.js y Git Control integrado, desarrollado por GitHub. Atom es una aplicación de escritorio construida utilizando tecnologías web. La mayoría de los paquetes que se extienden tienen licencias de software libre y son construidas y mantenidas por la comunidad. Atom se basa en Electron (anteriormente conocido como Atom Shell), un marco que permite las aplicaciones de escritorio multiplataforma utilizando Chromium y Node.js. Está escrito en CoffeeScript y Menos . Se pudo usar como un entorno de desarrollo integrado (IDE), hasta que esa característica se retiró en diciembre de 2018. Atom se lanzó de la versión beta, como versión 1.0, el 25 de junio de 2015. Sus desarrolladores lo llaman un «editor de texto hackeable para el siglo XXI».

2.11.4. MariaDB

MariaDB es un sistema de gestión de bases de datos derivado de MySQL con licencia GPL (General Public License). Es desarrollado por Michael (Monty) Widenius fundador de MySQL, la fundación MariaDB y la comunidad de desarrolladores de software libre. Introduce dos motores de almacenamiento nuevos, uno llamado Aria que reemplaza a MyISAM y otro llamado XtraDB en sustitución de InnoDB. Tiene una alta compatibilidad con MySQL ya que posee las mismas órdenes, interfaces, API y bibliotecas, siendo su objetivo poder cambiar un servidor por otro directamente.

Extensiones y características nuevas:

- MariaDB maneja hasta 32 segmentos clave por clave
- Se agregó `--abort-source-on-error` al cliente mysql
- Precisión de microsegundos en la lista de procesos
- Pool de hilos de ejecución o procesos

- Eliminación de tablas
- Extensiones de prueba mysqltest
- Columnas virtuales
- Estadísticas extendidas para el usuario
- Caché de claves segmentadas
- Autenticación a través de plugins
- Especificación de motor de almacenamiento en CREATE TABLE
- Mejoras a la tabla INFORMATION SCHEMA.PLUGINS
- Se agregó `--rewrite-db` como opción en mysqlbinlog al cambiar de base de datos usada
- Reporte de Procesos para ALTER TABLE y LOAD DATA INFILE

CAPITULO III
FORMULACIÓN DE SISTEMAS DE HIPÓTESIS.

3. Formulación del sistema de hipótesis

3.1. Hipótesis de la investigación

3.1.1. Hipótesis general.

Hi: El uso de Software libre causa un impacto social, económico y educativo en las instituciones de educación media del municipio de San Miguel.

Ho: El uso de Software libre no causa un impacto social, económico y educativo en las instituciones de educación media del municipio de San Miguel.

3.1.2. Hipótesis Específicas.

Hipótesis Específica I

El uso del software libre brinda beneficios económicos, educativos y de carácter social a la población estudiantil, planta docente y administrativa de las instituciones de educación media del municipio de San Miguel.

Hipótesis Específica II

La inadecuada formación y capacitación en herramientas tecnológicas libres dificulta la implementación de la normativa propuesta por el MINED acerca de la migración al uso de software libre en las aulas de Centro de Recursos de Aprendizajes (Aulas CRA).

Hipótesis Específica III

El Software libre proporciona cambios en los modelos pedagógicos, brindando materiales educativos, acceso a recursos bibliográficos, etc. Así como cambios en la gestión administrativa de las instituciones de educación media del municipio de San Miguel.

3.2. Operacionalización de Hipótesis.

| Objetivo | Hipótesis | Variables | Indicadores |
|---|--|---|---|
| Identificar el Impacto del uso del Software Libre en las instituciones de educación media, en el Municipio de San Miguel. | <p>Hipótesis Específica I</p> <p>El uso del software libre brinda beneficios económicos, educativos y de carácter social a la población estudiantil, planta docente y administrativa de las instituciones de educación media del municipio de San Miguel.</p> | <p>VI:</p> <p>El uso del software libre brinda beneficios.</p> <p>VD:</p> <p>Beneficios económicos y de carácter Social</p> | <p>VI:</p> <ul style="list-style-type: none"> Poco presupuesto al MINED Adaptación de nuevas tecnologías análisis de impacto de la nueva tecnología <p>VD:</p> <ul style="list-style-type: none"> Alto costos de licencias privadas Resistencia al cambio |
| | <p>Hipótesis Específica II</p> <p>La inadecuada formación y capacitación en herramientas tecnológicas libres dificulta la implementación de la normativa propuesta por el MINED acerca de la migración al uso de software libre en las aulas de Centro de Recursos de Aprendizajes (Aulas CRA).</p> | <p>VI:</p> <p>Formación y capacitación en Herramientas tecnológicas libres</p> <p>VD:</p> <p>Implementación de Normativa propuesta por el MINED</p> | <p>VI:</p> <ul style="list-style-type: none"> Escasa cantidad de personal profesional. Capacitaciones. Uso adecuado de las herramientas de software libre Manuales. <p>VD:</p> <ul style="list-style-type: none"> Inexperiencia en el uso de software libre. Acceso a recursos software. Incompatibilidad de Hardware/Software. |
| | <p>Hipótesis Específica III</p> <p>El Software libre proporciona cambios en los modelos pedagógicos, brindando materiales educativos, acceso a recursos bibliográficos, etc. Así como cambios en la gestión administrativa de las instituciones de educación media del municipio de San Miguel.</p> | <p>VI:</p> <p>Cambios en modelos pedagógicos en Instituciones Educativas.</p> <p>VD:</p> <p>Cambios en la gestión Administrativas de las Instituciones educativas</p> | <p>VI:</p> <ul style="list-style-type: none"> Control Ministerio de Educación. Nivel de innovación Uso adecuado de las herramientas de software libre. Personal capacitado. <p>VD:</p> <ul style="list-style-type: none"> conocimiento en el uso de estándares establecidos en la normativa Capacitación de herramientas de tecnologías libres. |

Figura 1: Tabla de Operacionalización de Hipótesis.

CAPITULO IV
METODOLOGIA DE LA INVESTIGACION.

4. Metodología de la investigación

4.1. Tipo de investigación

Uno de los principales objetivos que tiene la investigación es intentar descubrir algo, con el fin de obtener datos que ayuden a generar nuevos conocimientos. Para ello se emplean diferentes estrategias y metodologías. La realidad que se percibe es subjetiva, y a través de la investigación se procura explicar una realidad determinada para que resulte algo más objetiva.

4.1.1. Método de Investigación: cuantitativo

Utiliza la recolección de datos para probar hipótesis con base en la medición numérica y el análisis estadístico, con el fin establecer pautas de comportamiento y probar teorías. El enfoque cuantitativo (que representa, como dijimos, un conjunto de procesos) es secuencial y probatorio. Cada etapa precede a la siguiente y no podemos “brincar” o eludir pasos. El orden es riguroso, aunque desde luego, podemos redefinir alguna fase. Parte de una idea que va acotándose y, una vez delimitada, se derivan objetivos y preguntas de investigación, se revisa la literatura y se construye un marco o una perspectiva teórica. De las preguntas se establecen hipótesis y determinan variables; se traza un plan para probarlas (diseño); se miden las variables en un determinado contexto; se analizan las mediciones obtenidas utilizando métodos estadísticos, y se extrae una serie de conclusiones respecto de la o las hipótesis.

4.1.2. Investigación descriptiva

Según los tipos de investigación que existen, la que mejor se adaptó a este proyecto es de tipo descriptiva, porque esta permite puntualizar las características de la población que está estudiando. Esta metodología se centra más en el “qué”, en lugar del “por qué” del sujeto de investigación, la cual se aplicara en las Instituciones de educación media del municipio

de San Miguel. Existen 3 métodos distintos para llevar a cabo la investigación descriptiva:

1. **Método de observación:** El método observacional es el más eficaz para llevar a cabo la investigación descriptiva. Se utilizan tanto la observación cuantitativa como la observación cualitativa.
 - **La observación cuantitativa:** Es la recopilación objetiva de datos que se centran principalmente en números y valores. Los resultados de la observación cuantitativa se obtienen utilizando métodos de análisis estadísticos y numéricos como la edad, la forma, el peso, el volumen, la escala, etc.
 - **La observación cualitativa:** Este método no involucra mediciones o números, sólo características de monitoreo. En este caso, el investigador observa a los encuestados a distancia. Dado que se encuentran en un ambiente cómodo, las características observadas son naturales y efectivas. En la investigación descriptiva, el investigador puede elegir entre ser un observador completo, observar como participante, un participante observador o un participante completo.
2. **Método de estudio de caso:** Implican un análisis profundo y el estudio de individuos o grupos. Además, conduce a una hipótesis y amplía el alcance de la investigación de un fenómeno. Sin embargo, no deben utilizarse para determinar la causa y el efecto, ya que no tiene la capacidad de hacer predicciones precisas porque podría haber un sesgo por parte del investigador. La otra razón por la que los estudios de caso no son una forma precisa de llevar a cabo la investigación descriptiva es porque podría haber un encuestado atípico y su descripción conduciría a generalizaciones deficientes y se alejaría de la validez externa.
3. **Encuesta de Investigación:** Las encuestas son una herramienta de retroalimentación muy popular para la investigación de mercado. Para que una encuesta reúna datos de buena calidad, debe tener buenas preguntas, y haber un equilibrio entre preguntas abiertas y cerradas. El método de encuesta puede realizarse en línea o hacer encuestas

offline, lo que la convierte en la opción preferida para la investigación descriptiva cuando el tamaño de la muestra es muy grande. La investigación se centró en los conocimientos que poseen los diferentes estudiantes de Educación Media referente al tema de software libre, La cual dio los resultados necesarios para obtener el nivel de conocimiento e impacto tanto psicológico, social y educativo que ocasiono el tema en estudio.

4.1.3. Investigación Tecnológica

Proceso de conocimiento planeado, consciente, racional y crítico a través del cual los seres humanos diseñan herramientas, máquinas y procesos para incrementar su control y su comprensión del entorno material. La Investigación Tecnológica logra determinar un saber hacer técnico. Este saber es un recurso rentable que brinda beneficios económicos, por ello se protege con el secreto o se patenta investigar, idear e innovar.

La investigación tecnológica es la actividad que permite descubrir nuevos conocimientos, para luego encontrarles aplicaciones prácticas para la mejora de diseños, productos, equipos y de procesos industriales. El objetivo más general de la investigación tecnológica es obtener conocimiento para resolver problemas o necesidades de la sociedad.

La tecnología es una estructura de instrumentos, técnicas y procedimientos que emplea el conocimiento científico para controlar, transformar o crear determinados objetos o procesos con la finalidad de descripción y producción, tanto de problemas referidos al ámbito y/o a las soluciones orientadas en el mismo orden. Quiere decir que la tecnología se refiere a un tipo de conocimiento científico aplicado con la intención de mejorar rutinas concretas, sobre todo en áreas pragmáticas como el sector industrial, por ejemplo, dado el caso en las ciencias de la ingeniería. Sin embargo, los avances en la tecnología aplicada a procesos particulares dependen en cierta medida de los avances tecnológicos que la humanidad logre en otras áreas del conocimiento científico. Esta evolución es la que ha permitido que el proceso comunicativo del hombre tenga hoy características casi ilimitadas en cuanto a tiempo y espacio (Canga

Larequi, 1988).

Puede concebirse entonces la tecnología o la investigación tecnológica en dos extremos, investigar (ámbito científico) y transformar (ámbito tecnológico), el resultado implica un nuevo estado en el objeto de estudio. La investigación tecnológica comprende con mayor énfasis la transformación, cuyo fin es obtener conocimiento para lograr modificar la realidad en estudio, persiguiendo un conocimiento práctico. “Como resultado de una investigación tecnológica se obtienen conocimientos que establecen con detalle: acciones, requisitos, características, diseño, materiales, costos, responsables, métodos, instrumentos, y demás circunstancias, que describen el qué y el cómo, con lo que se promueve el logro de los objetivos, generalmente predeterminados en el área de producción” (García Córdoba, 2007, p.81)²³.

4.2. Técnicas e Instrumentos de Investigación

Son el conjunto de reglas y procedimientos que permiten al investigador establecer la relación con el objeto o sujeto de la investigación. Toda técnica o instrumento de recolección de los datos debe reunir dos requisitos esenciales: confiabilidad y validez. La confiabilidad de un instrumento se refiere al grado en que su aplicación repetida al mismo objeto produce iguales resultados. Y la validez se refiere al grado en que un instrumento realmente mide la variable que pretende medir. De tal manera, la información recolectada tiene que ser verdadera y justificada para ser uso de ella.

4.2.1. Instrumentos Primarios

Nos permitió obtener información directamente de la realidad, recolectándolos mediante instrumentos primarios.

Entre estos se mencionan los siguientes:

²³<http://www.eumed.net/rev/cccss/13/ibrs.html>

- **Observación Indirecta o no interactiva:** Durante el desarrollo del presente trabajo se utilizaron la técnica de la observación indirecta, es decir, cuando el investigador entra en conocimiento del hecho o fenómeno, a través de las observaciones realizadas anteriormente por otra persona, y para efecto de recopilar la información de la forma más válida y confiable, se recolectarán los datos y/o la información. Se utilizó durante todo el proyecto para obtener información como documentos, noticias, artículos y otro tipo de información necesaria para abordar la temática en el uso del software libre.

- **Fuentes primarias de información:** Se utilizarían como fuente primaria de información a la población de estudio, la cual sería representada por los estudiantes, docentes y directores de las instituciones públicas de las cuales se tomará la muestra. Por lo que, para el desarrollo de la investigación, se encuestara al director, subdirector, docentes y estudiantes de las instituciones, los cuales están involucrados directamente; de tal forma que se pudiera conocer la situación actual de las instituciones.

- **Entrevista:** Se utilizó para conocer la opinión de los directores y encargados de las aulas CRA de las instituciones educativas en el uso de software libre; También para conocer los avances que ha tenido en las mismas el cumplimiento de la normativa del MINED.

- **Encuesta:** Con este instrumento se realizará la recolección de datos en las Instituciones Educativas. Con los resultados obtenidos se podrán formular las conclusiones del proyecto.

4.2.2. Instrumentos Secundarios

Son registros escritos que proceden también de un contacto con la práctica, pero que ya han sido elegidos y procesados por otros investigadores.

Entre estos se mencionan a continuación:

- **Internet:** Se encontró información relevante para el desarrollo de la investigación, referencias de autores, libros e investigaciones, datos específicos e información general de gran utilidad.
- **La fuente bibliográfica:** Se encontró libros, revistas, reportes y estadísticas que nos ayudaron a realizar nuestra investigación.
- **La fuente de información institucional:** Fue proporcionada por las instituciones de las cuales se recolectó información para desarrollar nuestra investigación. Las fuentes fueron datos impresos, de forma digital o videos que contenían datos e información importante.

4.3. Universo.

El municipio de san miguel tiene 174 instituciones educativas de los cuales 41 son privados y 133 públicos.

| Institución Educativas | Numero de Instituciones |
|--|-------------------------|
| Centro Escolares | 112 |
| Colegios | 23 |
| Escuelas | 10 |
| Complejos Educativos | 9 |
| Liceos | 7 |
| Institutos | 7 |
| Jardines Infantiles | 2 |
| Ciudad Martell Republica de Francia | 1 |
| Centro Internacional de Programación de Computadoras | 1 |
| Centro de Formación en Ciencias Comerciales | 1 |
| Centro Cultural Salvadoreño Americano de San Miguel | 1 |
| Total | 174 |

Cuadro 1: Instituciones Educativas del Municipio de San Miguel

4.4. Poblacion.

Se tomó en cuenta la población de los estudiantes, docentes y encargados de Aulas CRA de Educación Media de la ciudad de San Miguel, debido a que en su plan de estudio ya está incorporada la materia de informática y nuestro tema de investigación tiene un gran auge debido a que el Ministerio de Educación está impulsando la utilización de software libre en los Centros Educativos. Para la población estudiantil es de gran importancia el conocer las distintas herramientas informáticas libres, son los que presentan mayores inquietudes referentes a la temática que se está abordando. El diseño de la investigación se orientó a medir sólo a la población en estudio, ya que ésta es única y al momento en que se pasaron los instrumentos, se consideró la relación que tienen las variables de las hipótesis.

Para desarrollar esta investigación se tomó como población solo las Instituciones de educación media del municipio de san miguel.

| Instituciones de Educación Media | Numero de Instituciones |
|---|--------------------------------|
| Complejos Educativos | 9 |
| Liceos | 7 |
| Institutos | 7 |
| Total | 23 |

Cuadro 2: Instituciones de Educación Media

4.5. Muestra.

Para determinar la muestra de la población en estudio, se consideró la población estudiantil de Educación Media del municipio de San Miguel.

Se realizara la muestra probabilística estratificada proporcional de los estudiantes.

4.5.1. Tamaño de la Muestra.

La muestra se obtuvo de tal manera que permitió establecer un mínimo de error posible respecto de la población. Para que una muestra sea fiable, es necesario que su tamaño sea obtenido mediante procesos matemáticos que eliminen la incidencia del error. (Marchal, 2005)

Fórmula para calcular el tamaño de la muestra:

$$N = \frac{(Z^2 \cdot P \cdot Q \cdot N)}{(N - 1)E^2 + Z^2 P \cdot Q}$$

Dónde:

Z: valor crítico 1.96

P: proporción que ocurra el evento = 50 %

Q: proporción que no ocurra el evento = 50 %

N: tamaño de la población

E: error muestral = 5 %

4.5.2. Calculo de la Muestra.

Muestra de Instituciones. Para el total de Muestra de Instituciones:

$$N = \frac{(1,96^2(0,5)(0,5)(23))}{(23 - 1)0,05^2 + 1,96^2(0,5)(0,5)}$$

$$N = 9$$

El total de muestra de instituciones fue de: 9

4.5.3. Tipo de muestreo.

Después de calcular el tamaño de la muestra se utilizó el tipo de muestreo probabilístico estratificado proporcional y muestreo por conveniencia.

El muestro por conveniencia se aplicó al momento de seleccionar las 9 instituciones de muestra debido a que selecciono las instituciones más céntricas del municipio e instituciones donde se imparte materias sobre informática, el método estratificado proporcional se utilizó para calcular el número de encuestas que se pasaran por cada institución.

Instituciones en Estudio.

Las Instituciones seleccionadas a estudiar se detallan a continuación:

Población Estudiantil en Estudio.

| Instituciones Educativas | Total de Estudiantes |
|---|----------------------|
| Instituto Nacional Joaquín Ernesto Cárdenas | 1553 |
| Instituto Nacional Isidro Menéndez | 1858 |
| Instituto Nacional Metropolitano | 189 |
| Instituto Nacional De San Antonio Silva | 284 |
| Instituto Nacional Francisco Gavidia | 838 |
| Complejo Educativo Sor Cecilia Santillana Ahuactzin | 224 |
| Complejo Educativo Ofelia Herrera | 456 |
| Complejo Educativo Confederación Suiza | 175 |
| Complejo Educativo Católico Santa Sofía | 133 |
| Total | 5,710 |

Cuadro 3: Instituciones Educativas en estudio del Municipio de San Miguel

Para el total de población estudiantil:

$$N = \frac{(1,96^2(0,5)(0,5)(5,710))}{(5,710) - 1)0,05^2 + 1,96^2(0,5)(0,5)}$$

$$N = 360$$

El total de muestra de la población fue de: 360

El total de encuestas por institución se detalla a continuación: Debido a la alta población

| Instituciones Educativas | Porcentaje % | Encuestas |
|---|---------------------|------------------|
| Instituto Nacional Joaquín Ernesto Cárdenas | 27.19 % | 98 |
| Instituto Nacional Isidro Menéndez | 32.55 % | 117 |
| Instituto Nacional Metropolitano | 4.97 % | 18 |
| Instituto Nacional De San Antonio Silva | 3.31 % | 12 |
| Instituto Nacional Francisco Gavidia | 14.68 % | 53 |
| Complejo Educativo Sor Cecilia Santillana Ahuactzin | 3.92 % | 14 |
| Complejo Educativo Ofelia Herrera | 7.99 % | 29 |
| Complejo Educativo Confederación Suiza | 3.06 % | 11 |
| Complejo Educativo Católico Santa Sofía | 2.33 % | 8 |
| TOTAL | 100 % | 360 |

Cuadro 4: Total de Encuestas por Institución

concentrada en algunas instituciones, se tomó a bien aplicar los instrumentos a los alumnos de los terceros años de bachillerato y segundos años generales de cada institución.

Se pasara la encuesta a algunas instituciones por secciones, ya que cuentan con bachillerato técnico vocacional en desarrollo de software; quedando de la siguiente manera:

- En el Instituto Nacional Joaquín Ernesto Cárdenas de 98 encuestas se pasaran 42 que corresponden a una sección de la especialidad

- En el Instituto Nacional Isidro Menéndez de 117 encuestas se pasaran 32 que corresponden a una sección de la especialidad.
- Instituto Nacional Francisco Gavidia 53 encuestas se pasaran 28 que corresponden a una sección de la especialidad.

Tomando en cuenta este criterio se tomaran el total de 214 encuestas como muestra.

4.6. Técnicas e instrumentos para el análisis de los datos.

4.6.1. Procedimiento de recolección de datos.

El proceso de recolección de datos es la planificación de un instrumento de medición que cumpla con los requisitos técnicos para poder aplicarlo en la muestra de la investigación. Para la aplicación de la encuesta se hará de la siguiente manera:

- Se seleccionará la población que será estudiada para la obtención de información, en este caso serán los directores o subdirectores y docentes de las instituciones públicas que se tomaron como muestra.
- Se les darán las indicaciones respectivas para la realización de la encuesta.
- Obtenidos los datos, se analizarán y se espera que los resultados sean de ayuda para el problema planteado y para la realización de la investigación

4.6.2. Procedimiento para procesar e interpretar los datos.

Para el proceso de los datos se recopilará toda la información obtenida de los instrumentos implementados y con ello se seleccionará todo lo que se requiere para el desarrollo del sistema,

con el fin de priorizar las necesidades y contar con información válida.

Para representar la información se realizará lo siguiente:

- Se tabularán los datos obtenidos de las encuestas.
- Se presentará cada de una de las preguntas que contiene la encuesta.
- Se formulará el objetivo de cada pregunta de la encuesta.
- Se hará una representación de los datos en forma porcentual en un gráfico circular, para una mejor comprensión y análisis de los datos.
- Análisis e interpretación de los resultados.

CAPITULO V
TABULACIÓN DE RESULTADOS Y
COMPROBACIÓN DE HIPÓTESIS.

5. Tabulación de resultados y comprobación de hipótesis

5.1. Tabulación de Encuestas

A continuación se presenta la información obtenida de las encuestas realizadas a los estudiantes, docentes y encargados de aulas informáticas de los centros escolares públicos del municipio de San Miguel, departamento de San Miguel.

5.1.1. Encuesta dirigida a Estudiantes

Pregunta 1: ¿Qué grado cursas actualmente?

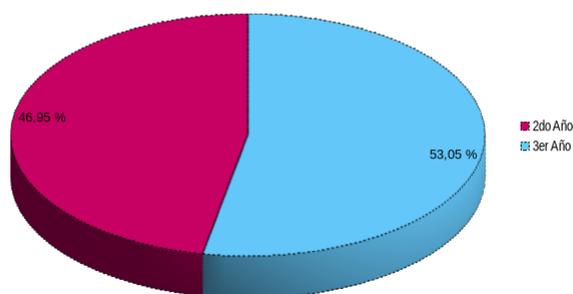


Figura 2: Pregunta 1

| Bachillerato | Cantidad | Porcentaje |
|-----------------|----------|------------|
| 1° Bachillerato | 0 | 0.00 % |
| 2° Bachillerato | 100 | 46.95 % |
| 3° Bachillerato | 113 | 53.05 % |
| Total | 213 | 100.00 % |

Cuadro 5: Pregunta 1

Pregunta 2: ¿Cuántos años has recibido clases de informática?

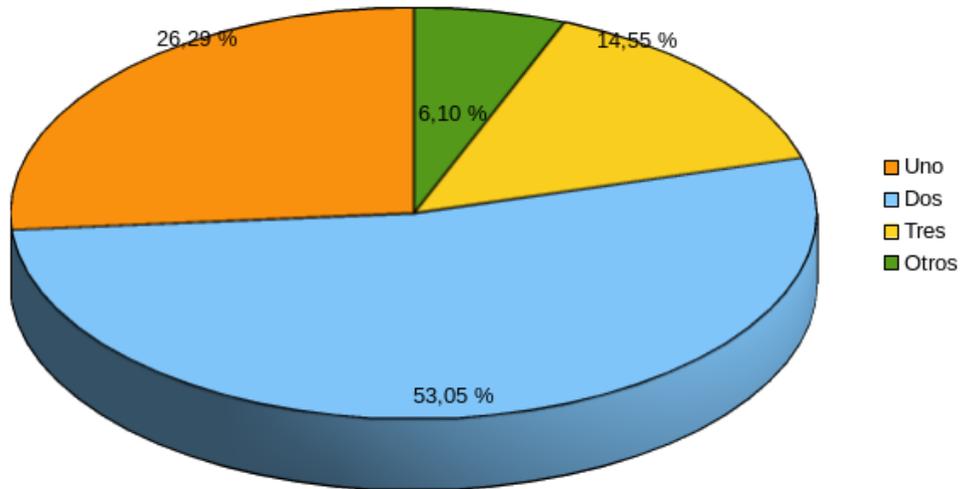


Figura 3: Pregunta 2

| Años | Cantidad | Porcentaje |
|-------|----------|------------|
| Uno | 56 | 26.29 % |
| Dos | 113 | 53.05 % |
| Tres | 31 | 14.55 % |
| Otros | 13 | 6.10 % |
| Total | 213 | 100 % |

Cuadro 6: Pregunta 2

Analisis:

En la encuesta se tomó en cuenta la cantidad de años que los estudiantes han recibido clases de informática, con los resultados se puede observar que el 26.29% solo han recibido un año, el 53.05% dos años, el 14.55% tres años y el 6.10% han recibido más de tres años.

Pregunta 3: ¿Haces uso de herramientas informáticas para las clases en la institución?

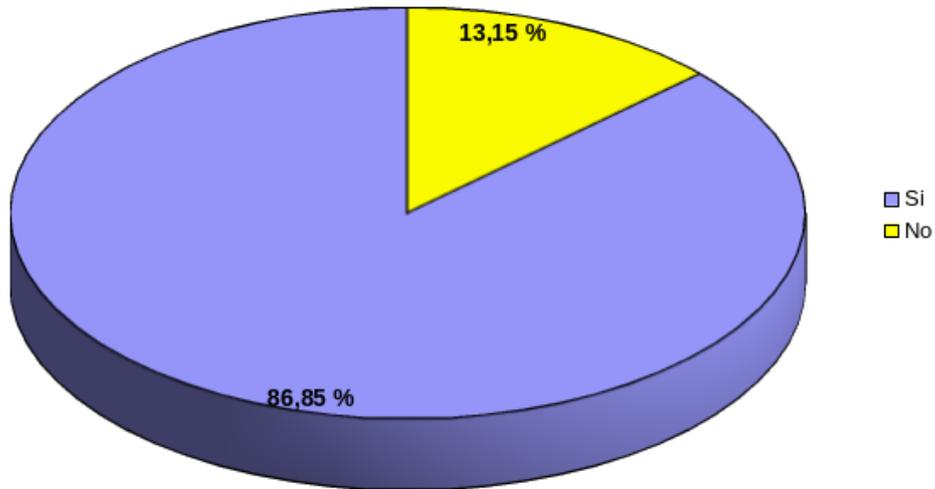


Figura 4: Pregunta 3

| Usas Herramientas | Cantidad | Porcentaje |
|-------------------|----------|------------|
| Si | 185 | 86.85 % |
| No | 28 | 13.15 % |
| Total | 213 | 100 % |

Cuadro 7: Pregunta 3

Analisis:

En la encuesta se pregunta si el estudiante hace uso de herramientas informáticas en su institución y como resultado tenemos que 86.85 % que corresponde a un total de 185 estudiantes si hacen uso y 28 estudiantes que conforman el 13.15 % respondieron que no.

Pregunta 4: ¿Tienes conocimientos o has escuchado hablar sobre el software libre?

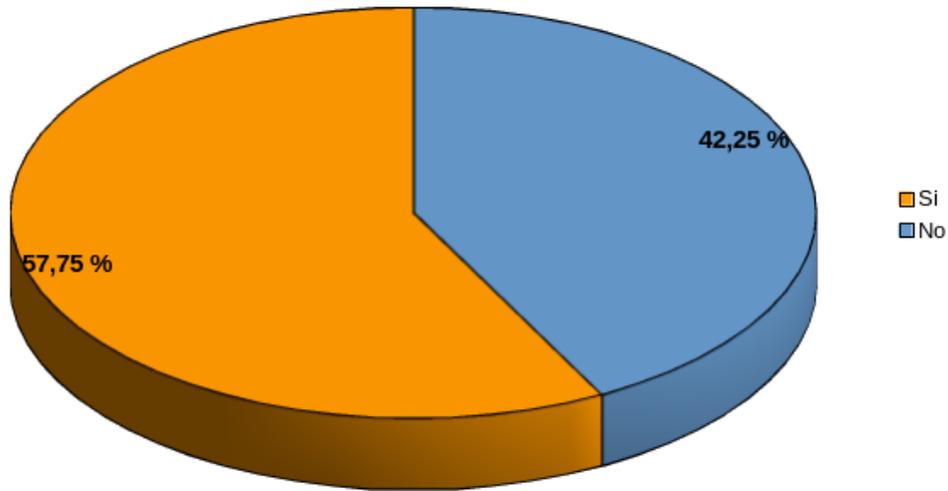


Figura 5: Pregunta 4

| Tiene conocimiento | Cantidad | Porcentaje |
|--------------------|----------|------------|
| Si | 123 | 57.75 % |
| No | 90 | 42.25 % |
| Total | 213 | 100 % |

Cuadro 8: Pregunta 4

Analisis:

Se preguntó a los estudiantes si tienen conocimientos o han escuchado hablar sobre el software libre el 57.75 % contestó que si tienen conocimiento y el 42.25 % contestó que no tenía conocimientos, esto nos indica que no todos los estudiantes tienen conocimientos sobre el software libre aunque utilicen el software libre en las instituciones.

Pregunta 5: Si has escuchado hablar del software libre, ¿Cuáles de los siguientes valores crees que lo sustentan?

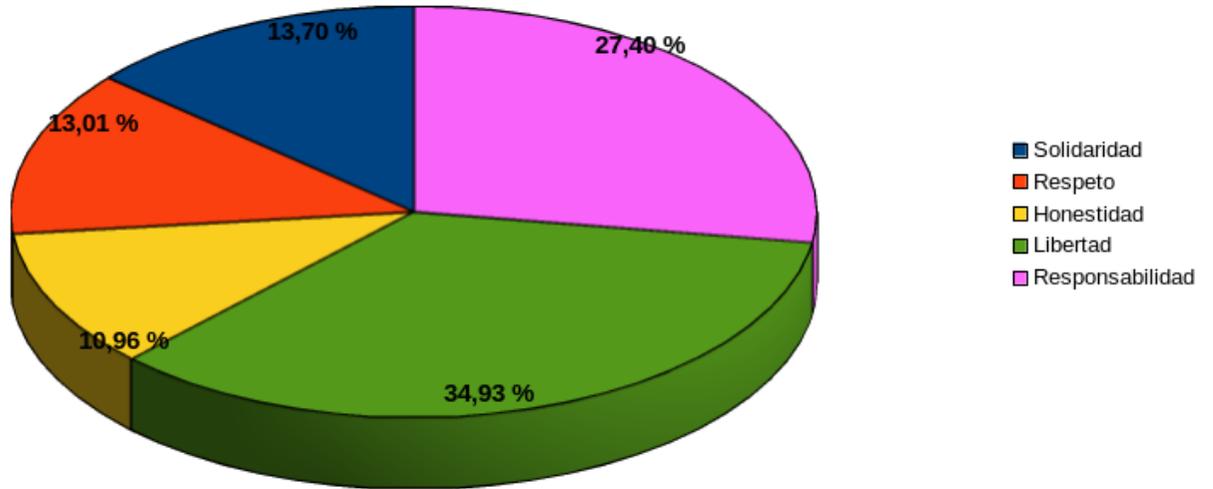


Figura 6: Pregunta 5

| Valores | Cantidad | Porcentaje |
|-----------------|----------|------------|
| Solidaridad | 40 | 13.70 % |
| Respeto | 38 | 13.01 % |
| Honestidad | 32 | 10.96 % |
| Libertad | 102 | 34.93 % |
| Responsabilidad | 80 | 27.40 % |
| Total | 292 | 100 % |

Cuadro 9: Pregunta 5

Análisis:

Preguntamos en la encuesta sobre cuáles son los valores que sustentan al software libre, los valores de Libertad con 102 estudiantes que corresponde al 34.93 % y el de Responsabilidad con 80 estudiantes que corresponde al 27.40 % fueron los más sobresalientes

Pregunta 6: ¿Conoces el sistema operativo GNU-Linux?

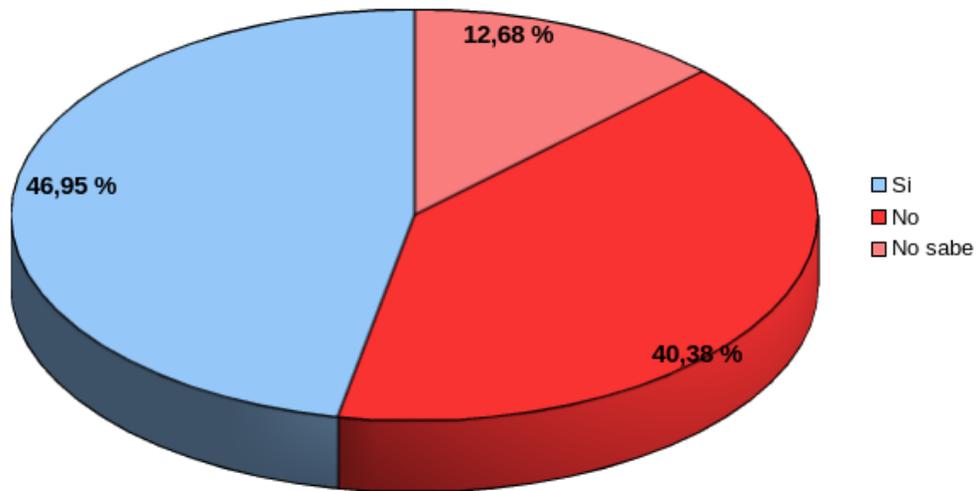


Figura 7: Pregunta 6

| Conoce GNU-Linux | Cantidad | Porcentaje |
|------------------|----------|------------|
| Si | 100 | 46.95 % |
| No | 86 | 40.38 % |
| No sabe | 27 | 12.68 % |
| Total | 213 | 100 % |

Cuadro 10: Pregunta 6

Analisis:

Preguntamos en la encuesta sobre cuáles son los valores que sustentan al software libre, los valores de Libertad con 102 estudiantes que corresponde al 34.93 % y el de Responsabilidad con 80 estudiantes que corresponde al 27.40 % fueron los más sobresalientes.

Si conoces el sistema operativo GNU-Linux. Marca algunas de las distribuciones que conoces:

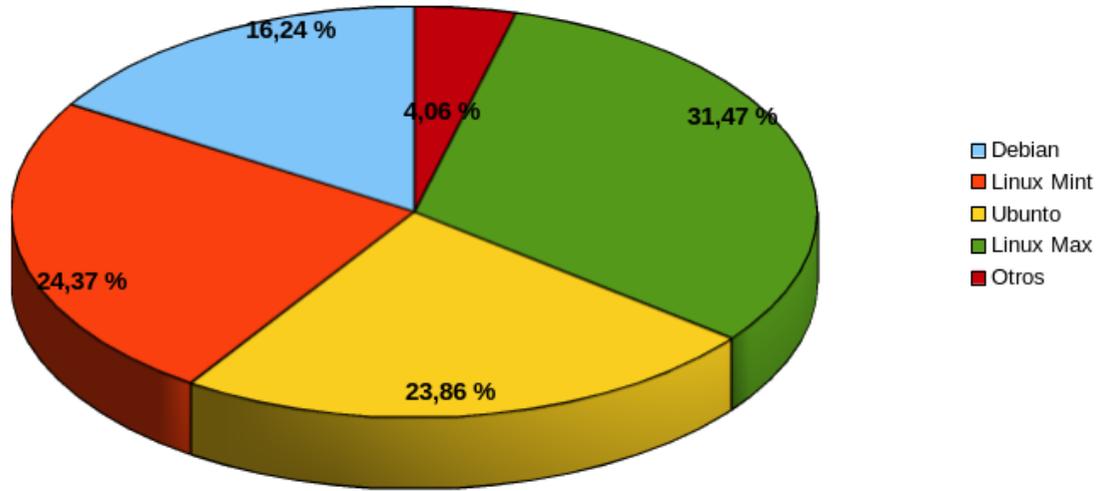


Figura 8: Pregunta 6a

| Distribución | Cantidad | Porcentaje |
|--------------|----------|------------|
| Debian | 32 | 16.24 % |
| Linux Mint | 48 | 24.37 % |
| Ubuntu | 47 | 23.86 % |
| Linux Max | 62 | 31.47 % |
| Otros | 8 | 4.06 % |
| Total | 213 | 100 % |

Cuadro 11: Pregunta 6a

Analisis:

Se pregunta a los estudiantes si conocen las distribuciones de GNU-Linux las respuestas nos muestran que Linux Max con un 31.47 % y Linux Mint con un 24.37 % son las distribuciones más conocidas y las menos conocidas son Ubuntu con el 23.86 % y Debían 16.24.

Pregunta 7: ¿Conoces otras herramientas informáticas basadas en software libre?

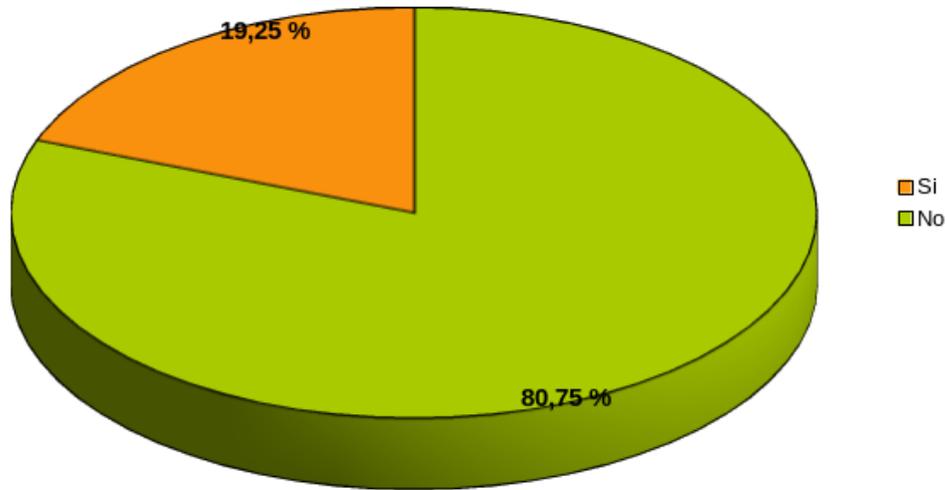


Figura 9: Pregunta 7

| Conoce | Cantidad | Porcentaje |
|--------|----------|------------|
| Si | 41 | 19.25 % |
| No | 172 | 80.75 % |
| Total | 213 | 100 % |

Cuadro 12: Pregunta 7

Analisis:

Se preguntó a los estudiantes si conocen otras herramientas informáticas basadas en software libre, el 80.75 % respondió que no conocían y el 19.25 % dijo que si conocían otras herramientas informáticas.

Menciona alguna de ellas:

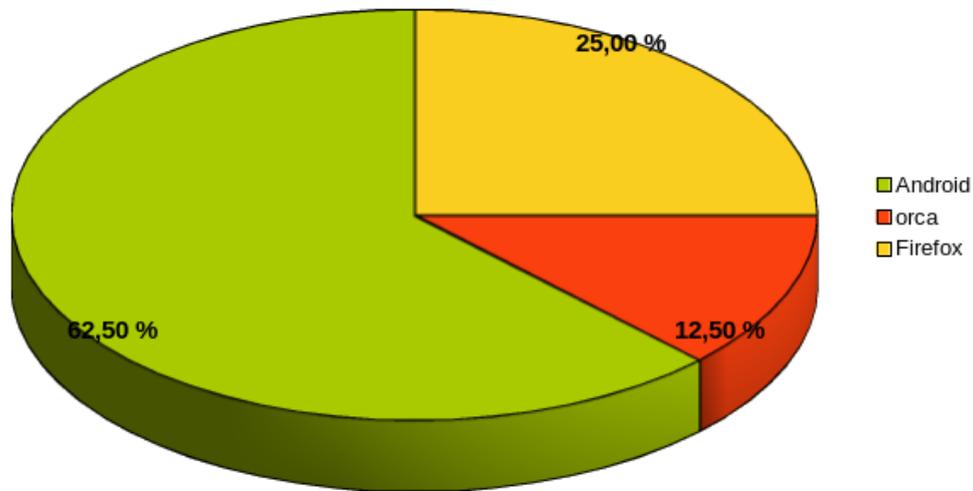


Figura 10: Pregunta 7

| Herramientas | Cantidad | Porcentaje |
|--------------|----------|------------|
| Android | 5 | 62.50 % |
| Orca | 1 | 12.50 % |
| Firefox | 2 | 25.0 % |
| Total | 8 | 100 % |

Cuadro 13: Pregunta 7a

Analisis:

Luego en la misma pregunta se dice que si conoce otras herramientas que las mencione, de las cuales las más mencionadas fueron Android, Firefox y Orca.

5.1.2. Encuesta dirigida a docentes

Pregunta 1: ¿Que programas de ofimática enseña a sus alumnos?



Figura 11: Pregunta 1

| Programs | Total |
|------------------------|--------------|
| Paquete de Office | 9 |
| Paquete de LibreOffice | 7 |
| Corel Draw | 2 |
| Total | 18 |

Cuadro 14: Pregunta 1

Analisis:

Por los resultados se puede observar que la mayoría de docentes usan un software diferente al Software libre, los 9 docentes tomados como muestra usan programas de ofimática privativas, y solo 7 usan software libre. Podemos determinar que los docentes hacen un mayor uso de tecnologías propietarias, es decir de Microsoft y menos de software libre.

Pregunta 2: ¿Enseña programación a los estudiantes?

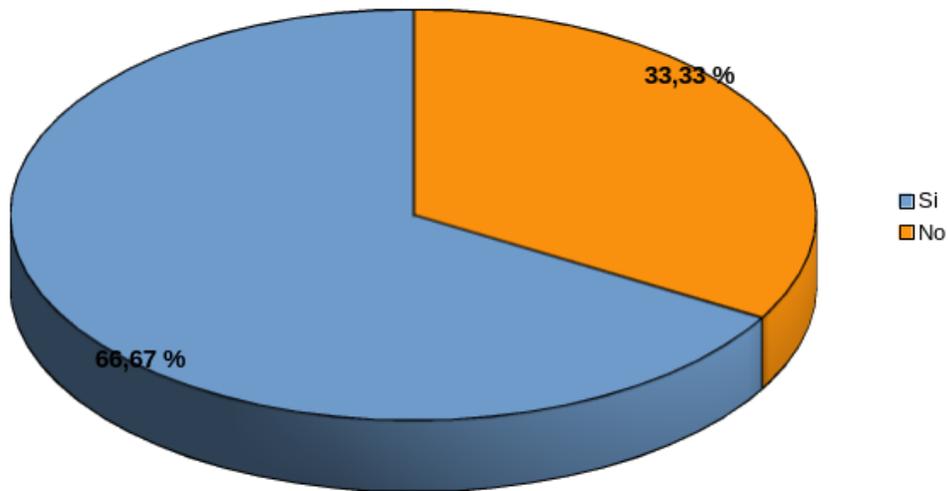


Figura 12: Pregunta 2

| Enseña programación | Cantidad | Porcentaje |
|---------------------|----------|------------|
| Si | 6 | 66.67 % |
| No | 3 | 33.33 % |
| Total | 9 | 100 % |

Cuadro 15: Pregunta 2

Analisis:

Se pregunta a los docentes si enseña clases de programación en la instituciones 66.67 % respondió que si enseña programación y el 33.33 % que no.

Pregunta 2.1: Si la respuesta anterior fue si, ¿qué lenguajes de programación imparte sus alumnos?

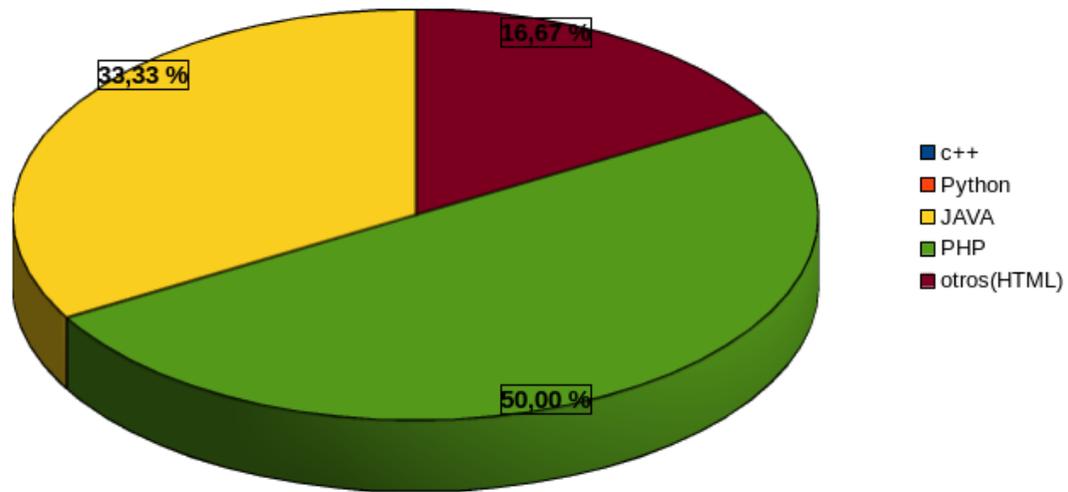


Figura 13: Pregunta 2.1

| Lenguajes | Cantidad | Porcentaje |
|-------------|----------|------------|
| c++ | 0 | 0.0 % |
| Python | 0 | 0.0 % |
| JAVA | 2 | 33.33 % |
| PHP | 3 | 50.00 % |
| Otros(HTML) | 1 | 16.67 % |
| Total | 6 | 100 % |

Cuadro 16: Pregunta 2.1

Analisis:

Si en la pregunta anterior se respondió que sí, se le pregunta sobre cuál es el lenguaje de programación que enseña, en donde el 50.00 % enseña PHP, el 33.33 % JAVA y un 16.67 % respondió que otros, en los cuales se mencionó HTML.

Pregunta 3: Si respondió que si a la pregunta 2; ¿Que entorno de programación integrado (IDE) o que editores utiliza como herramienta para programar en clases?

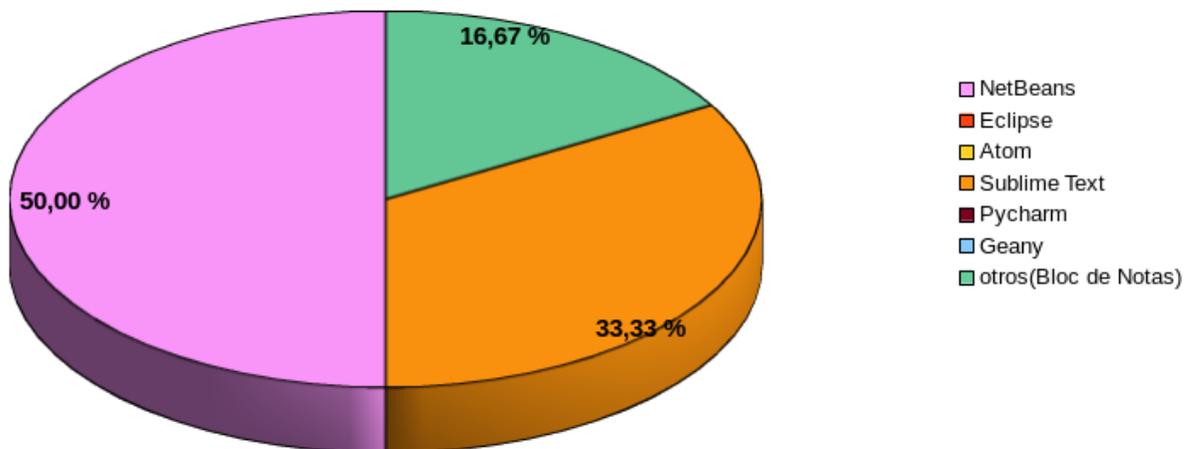


Figura 14: Pregunta 3

| Editores | Cantidad | Porcentaje |
|----------------------|----------|------------|
| NetBeans | 3 | 50.00 % |
| Eclipse | 0 | 0.00 % |
| Atom | 0 | 0.00 % |
| Sublime Text | 2 | 33.33 % |
| Pycharm (HTML) | 0 | 0.00 % |
| Geany | 0 | 0.00 % |
| Otros(Bloc de Notas) | 1 | 16.67 % |
| Total | 6 | 100 % |

Cuadro 17: Pregunta 3

Analisis:

Siguiendo la secuencia anterior, sobre si enseña programación, a los que respondieron que sí ,se pregunta cuál es el entorno o que editores de texto utiliza, NetBeans 50.00 % es el más utilizado seguido por Sublime Text 33.33 % y en otros se menciona Bloc de Notas con un porcentaje de 16.67 % .

Pregunta 4: ¿Enseña sobre base de datos a los estudiantes?

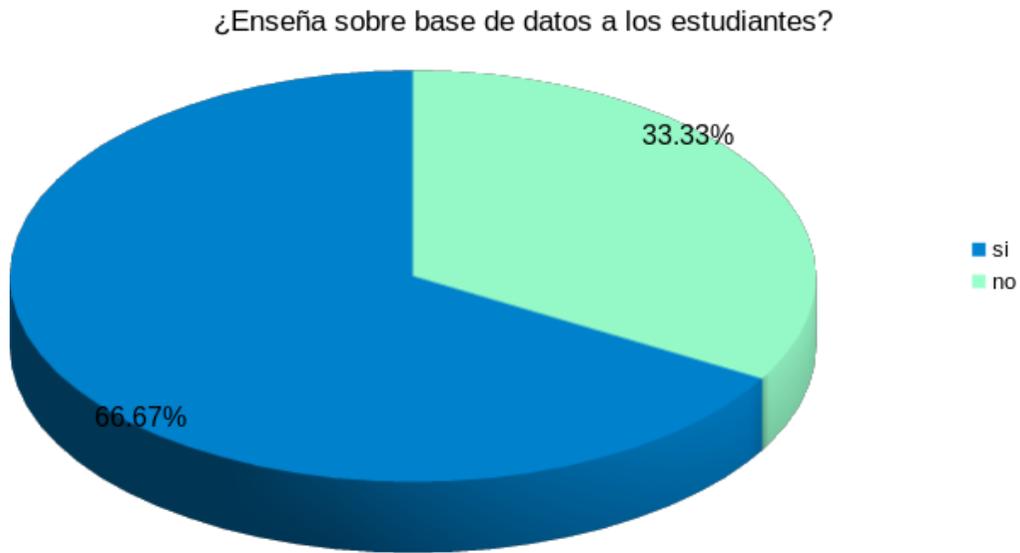


Figura 15: Pregunta 4

| Enseña Base de datos | Cantidad | Porcentaje |
|----------------------|----------|------------|
| Si | 6 | 77.78 % |
| No | 3 | 22.22 % |
| Total | 9 | 100 % |

Cuadro 18: Pregunta 4

Analisis:

Se pregunta a los docentes si utiliza base de datos, el 77.78% responde que si utiliza y un 22.22% que no utiliza base de datos, como podemos observar esta pregunta con la anterior tienen similitud ya que si se enseña programación, también utiliza base de datos.

Pregunta 4A: Si la respuesta anterior fue si, ¿Que sistemas de Gestores de Bases de Datos imparte a sus alumnos?

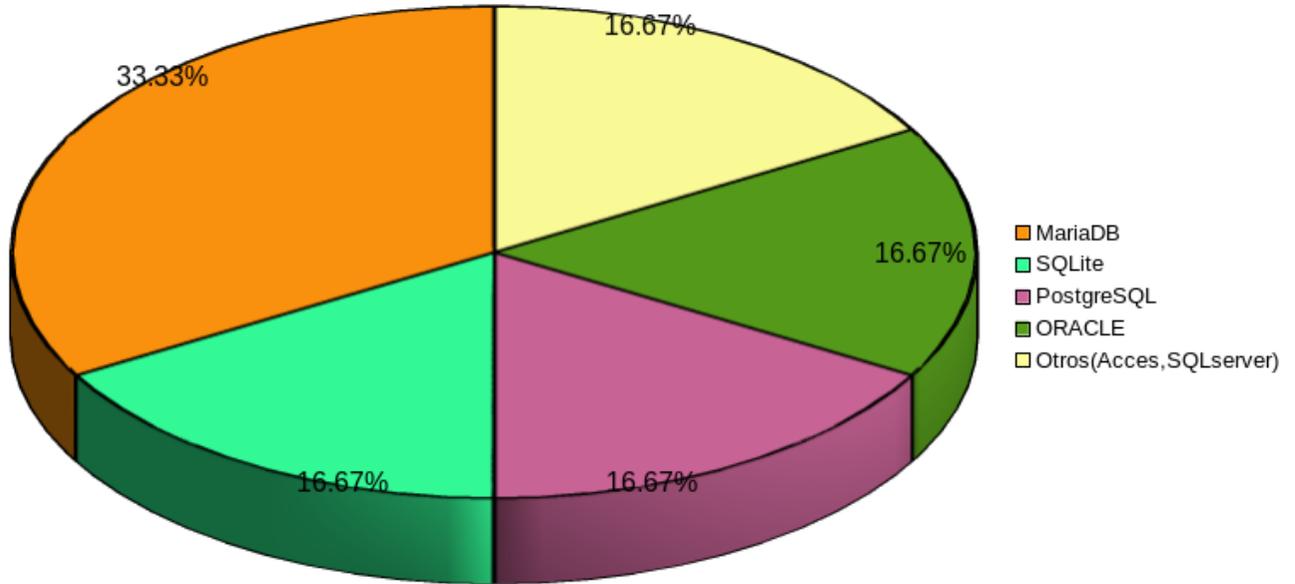


Figura 16: Pregunta 4A

| Gestores | Cantidad | Porcentaje |
|------------------------|----------|------------|
| MariaDB | 2 | 33.33 % |
| SQLite | 1 | 16.67 % |
| PostgreSQL | 1 | 16.67 % |
| ORACLE | 1 | 16.67 % |
| Otros(Acces,SQLserver) | 1 | 16.67 % |
| Total | 6 | 100 % |

Cuadro 19: Pregunta 4A

Analisis:

Para los docentes que utilizan base de datos, se les pregunta que cuales son los gestores de datos que utiliza para enseñar, MariaDB 33.33 % fue la que más utilizan las demás gestores mencionados todos tienen el 16.67 % y en Otros(Acces, SQLserver) también representa el 16.67 %

Pregunta 5: ¿Tiene conocimiento sobre la Normativa del MINED a cerca de la implementación de software Libre?

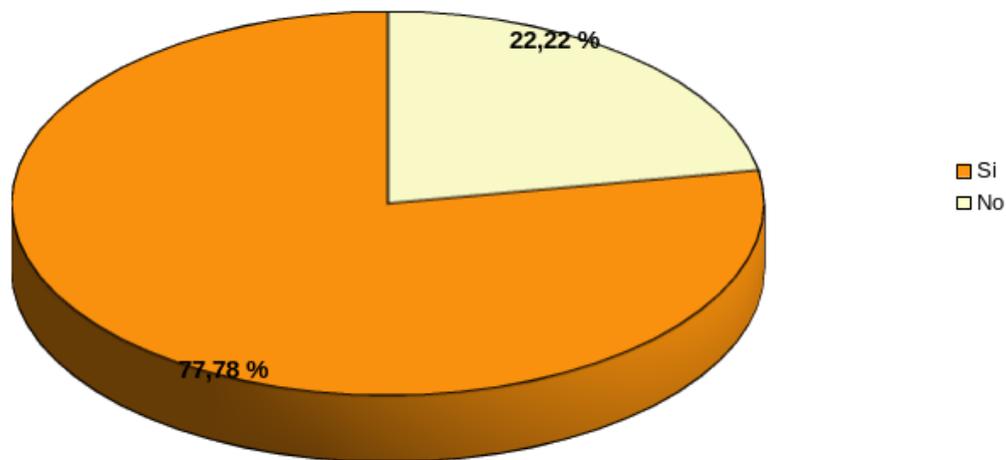


Figura 17: Pregunta 5

| Conocimiento sobre la normativa | Cantidad | Porcentaje |
|---------------------------------|----------|------------|
| Si | 7 | 77.78 % |
| No | 2 | 22.22 % |
| Total | 9 | 100 % |

Cuadro 20: Pregunta 5

Analisis:

Se pregunta a los docentes si tienen conocimientos sobre la Normativa del MINED a cerca de la implementación de software Libre de los cuales un 77.78 % contesto que sí, y el 22.22 % que no, esto nos indica que la mayoría de docentes conocen sobre la normativa de implantación del software libre.

Pregunta 6: ¿Has recibido capacitaciones de parte del MINED sobre el uso de software libre?

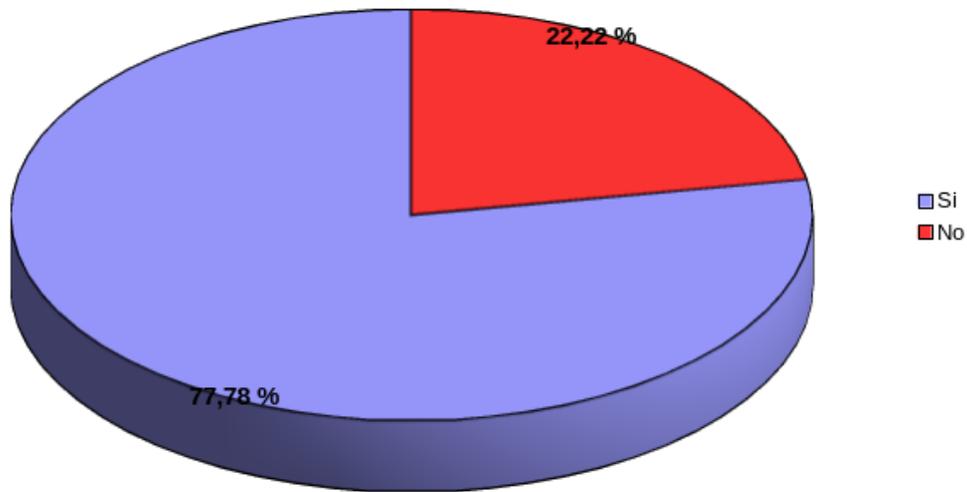


Figura 18: Pregunta 6

| Capacitación del MINED | Cantidad | Porcentaje |
|------------------------|----------|------------|
| Si | 7 | 77.78 % |
| No | 2 | 22.22 % |
| Total | 9 | 100 % |

Cuadro 21: Pregunta 6

Análisis:

También se preguntó a los docentes si han recibido capacitaciones de parte del MINED sobre el uso de software libre, el resultado fue un 77.78 % contestó que sí y el 22.22 % que no, esta pregunta tiene repuestas similares a la anterior ya que son preguntas relacionadas.

Especifique tipo de capacitación:

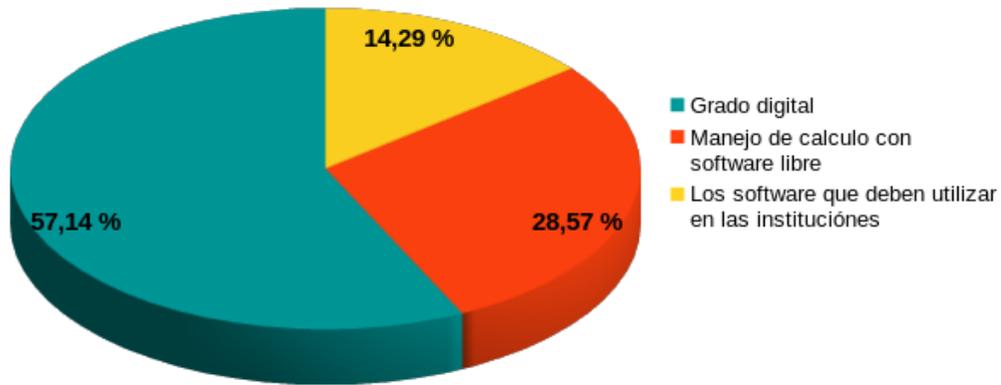


Figura 19: Pregunta 6A

| Capacitación del MINED | Cantidad | Porcentaje |
|--|----------|------------|
| Grado digital | 4 | 57.14 % |
| Manejo de cálculo con software libre | 2 | 28.57 % |
| Los software que deben utilizar en las instituciones | 1 | 14.29 % |
| Total | 7 | 100 % |

Cuadro 22: Pregunta 6A

Analisis:

Siguiendo la secuencia se preguntó a los docentes cuales son las capacitaciones que impartió el Mined, el 57.14 % respondió que grado digital el 28.57 % Manejo de cálculo con software libre y el 14.29 % Los software que deben utilizar en las instituciones estas fueron las capacitaciones que los docentes recibieron.

Pregunta 7: ¿Considera usted que tendría algún inconveniente para desarrollar sus clases haciendo uso del Software Libre?

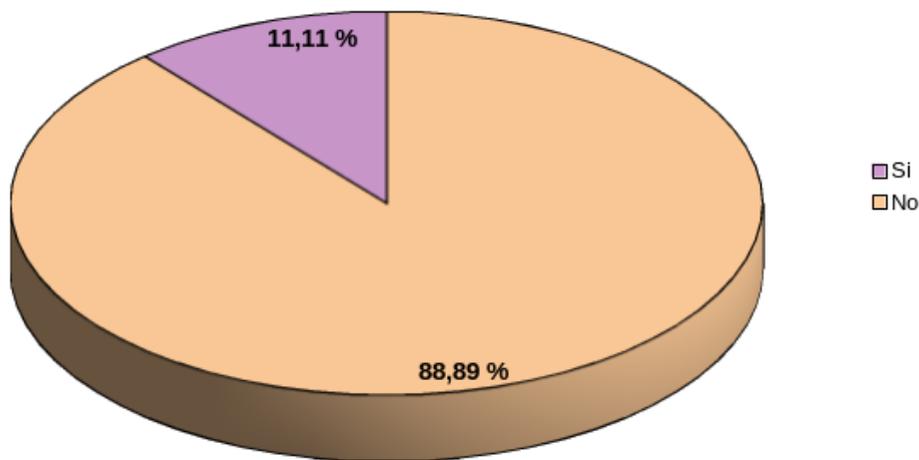


Figura 20: Pregunta 7

| Inconvenientes | Cantidad | Porcentaje |
|----------------|----------|------------|
| Si | 3 | 11.11 % |
| No | 6 | 89.89 % |
| Total | 9 | 100 % |

Cuadro 23: Pregunta 7

Análisis:

Se pregunta a los docentes si tendría algún inconveniente para desarrollar sus clases haciendo uso del Software Libre en esta respuesta el 89.89 % contestó que no tendría inconveniente y un 11.11 % respondió que si tendría inconvenientes.

Pregunta 7A: Si tuvo algunos inconvenientes, detalle algunos que considere importantes:

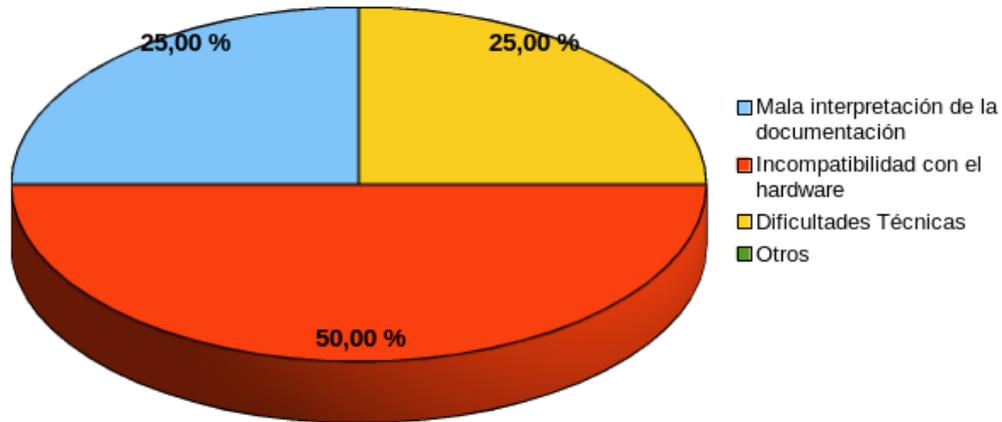


Figura 21: Pregunta 7A

| Inconvenientes | Cantidad | Porcentaje |
|---|----------|------------|
| Mala interpretación de la documentación | 1 | 25.00 % |
| Incompatibilidad con el hardware | 2 | 50.00 % |
| Dificultades Técnicas | 1 | 25 % |
| Total | 4 | 100 % |

Cuadro 24: Pregunta 7A

Análisis :

Con referencia a la pregunta anterior se pregunta cuáles son los inconvenientes más importantes que tuvo, el inconveniente con mayor porcentaje fue la Incompatibilidad con el hardware con un 50.00 % y un 25.00 % Mala interpretación de la documentación y Dificultades Técnicas con otro 25.00 %.

5.1.3. Encuesta dirigida a encargados de CRA

Pregunta 1: ¿Cuáles son las funciones que desempeña en el aula CRA?

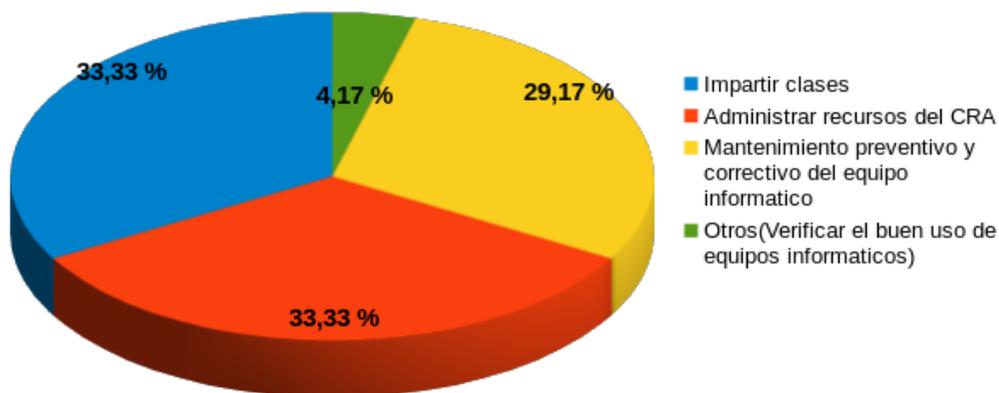


Figura 22: Pregunta 1

| Funciones | Respuestas | Porcentaje |
|--|------------|------------|
| Impartir clases | 8 | 66.66 % |
| Administrar recursos del CRA | 8 | 66.66 % |
| Mantenimiento preventivo y correctivo del equipo informático | 7 | 29.17 % |
| Otros (Verificar el buen uso de equipos informáticos) | 1 | 4.17 % |

Cuadro 25: Pregunta 1

Análisis:

También preguntamos sobre cuál es la función que tienen los encargados de aulas CRA, el 66.66 % nos contestó que era impartir clases y administrar los recursos del CRA, El 29 % brindar mantenimiento preventivo y correctivo del equipo informático y un 4.17 % se encarga de verificar el buen uso de los equipos.

Pregunta 2: ¿Equipos con los que cuenta el aula CRA?

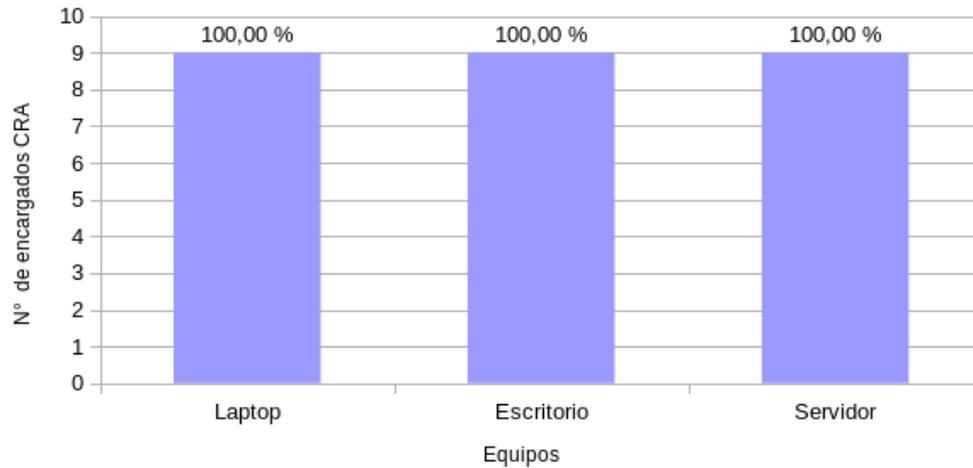


Figura 23: Pregunta 2

| Equipo | Respuestas | Porcentaje |
|------------|------------|------------|
| Laptop | 9 | 100 % |
| Escritorio | 9 | 100 % |
| Servidor | 9 | 100 % |

Cuadro 26: Pregunta 2

Análisis:

Se consultó a los encargados de aulas CRA acerca de los diferentes equipos con los que cuentan, El 100 % respondió que utiliza laptop, computadoras de escritorios y servidor. Los resultados nos muestran que las aulas CRA están lo suficiente preparadas para poder recibir clases de informática

Pregunta 2.1: ¿Mencione algunas características de hardware y software? Hardware

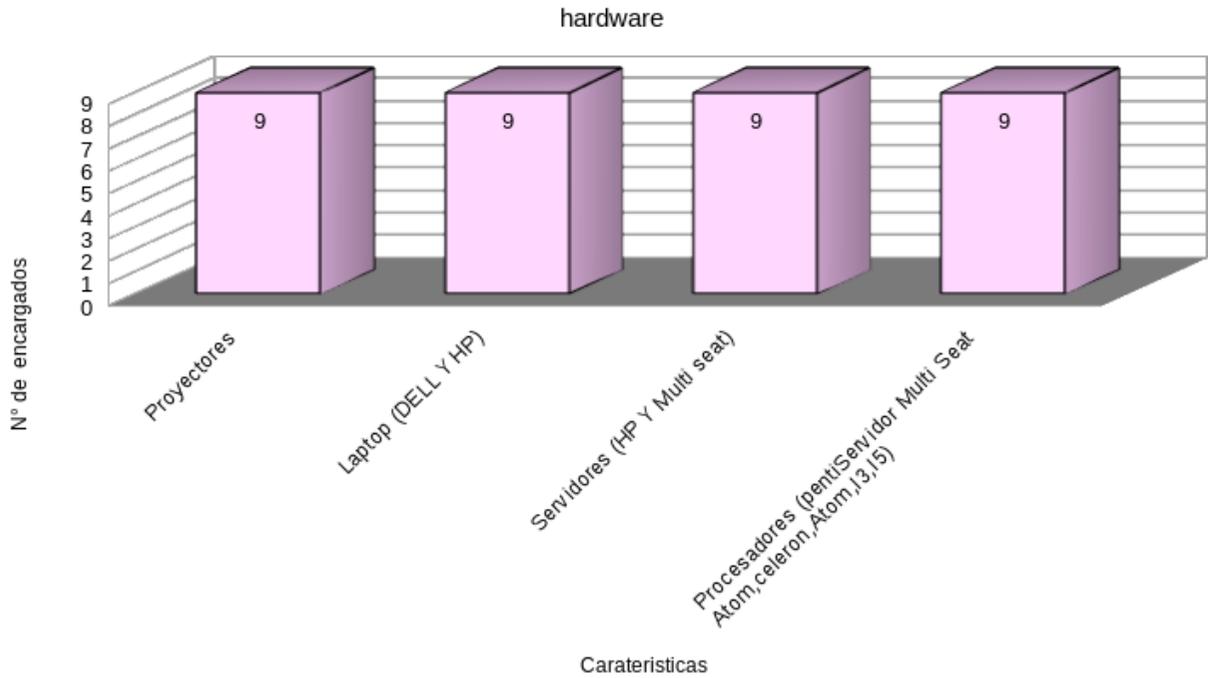


Figura 24: Pregunta 2.1

| Hardware | Respuestas | Porcentaje |
|--|------------|------------|
| Proyectorres | 9 | 100 % |
| Laptop (DELL Y HP) | 9 | 100 % |
| Servidores (HP Y Multi seat) | 9 | 100 % |
| Procesadores (Atom, Celeron, Atom, I3, I5) | 9 | 100 % |

Cuadro 27: Pregunta 2.1

Análisis:

Se pregunta sobre las características del software y el software que tienen en las instituciones, en el hardware, lo que más se tiene son proyectores, Laptop (DELL Y HP), Procesadores (Atom, Celeron, Atom, I3, I5) y Servidores (HP Y Multi seat).

Software

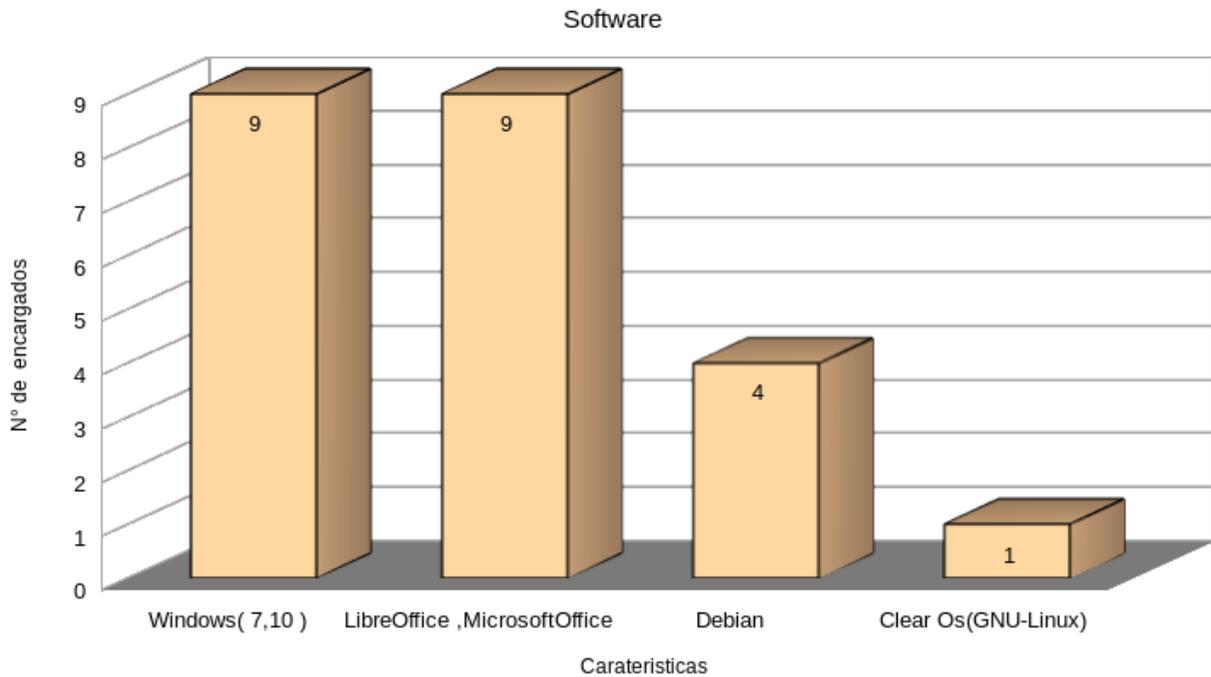


Figura 25: Pregunta 2.1

| Software | Respuestas | Porcentaje |
|------------------------------|------------|------------|
| Windows (7,10) | 9 | 100 % |
| LibreOffice, MicrosoftOffice | 9 | 100 % |
| Debian | 4 | 44.4 % |
| ClearOs(GNU-Linux) | 11.11 | 100 % |

Cuadro 28: Pregunta 2.1

Analisis:

En el software tenemos que el 100 % poseen Windows (7,10) y Libre Office, Microsoft Office un 44.4 % utilizan también Debían y el 11.11 % ClearOs esto nos indica que en la mayoría Aulas CRA poseen software privativo y software libre, con la diferencia que utilizan más software privativo (Windows)

Pregunta 3: ¿Posee conexión a internet en el aula CRA?

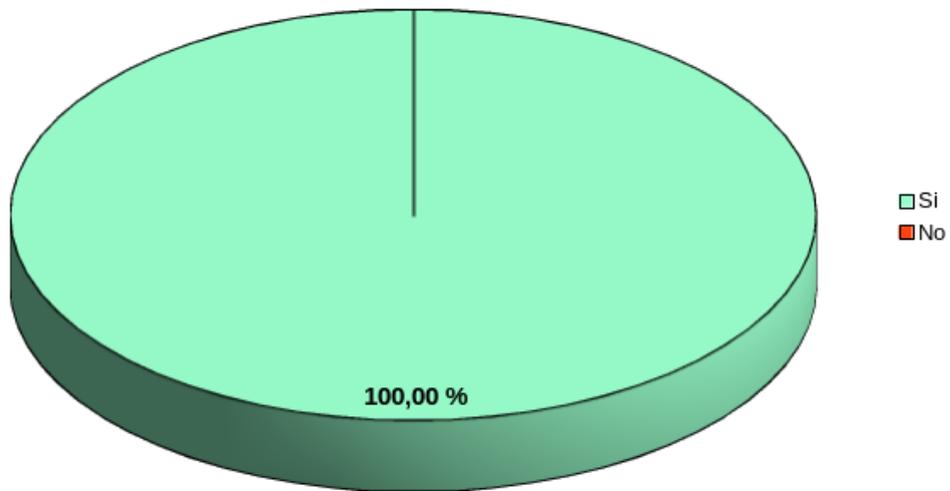


Figura 26: Pregunta 3

| Conexión a internet | Respuestas | Porcentaje |
|---------------------|------------|------------|
| Si | 9 | 100 % |
| No | 9 | 0.0 % |

Cuadro 29: Pregunta 3

Análisis:

Se pregunta a los encargados de aulas CRA si posee conexión a internet en el aula, el 100 % contestó que si tienen conexión a internet

Pregunta 4: ¿Tiene conocimiento de la normativa del MINED en la implementación del software Libre en las Aulas Informáticas?

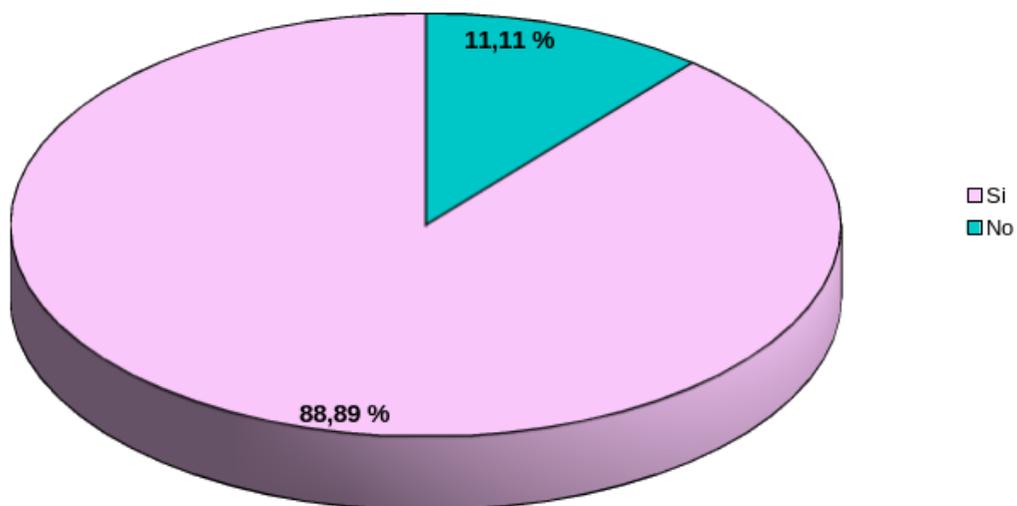


Figura 27: Pregunta 4

| Conocimiento de Normativa | Respuestas | Porcentaje |
|---------------------------|------------|------------|
| Si | 8 | 88.89 % |
| No | 1 | 11.11 % |

Cuadro 30: Pregunta 4

Analisis: También se le pregunta al encargado de aula CRA si tiene conocimiento de la normativa del MINED en la implementación del software Libre en las Aulas informáticas, el 88.89% respondió que sí y el 11.11% que contesto que no.

Pregunta 5: ¿Ha realizado los cambios de software privativo a software libre según la normativa?

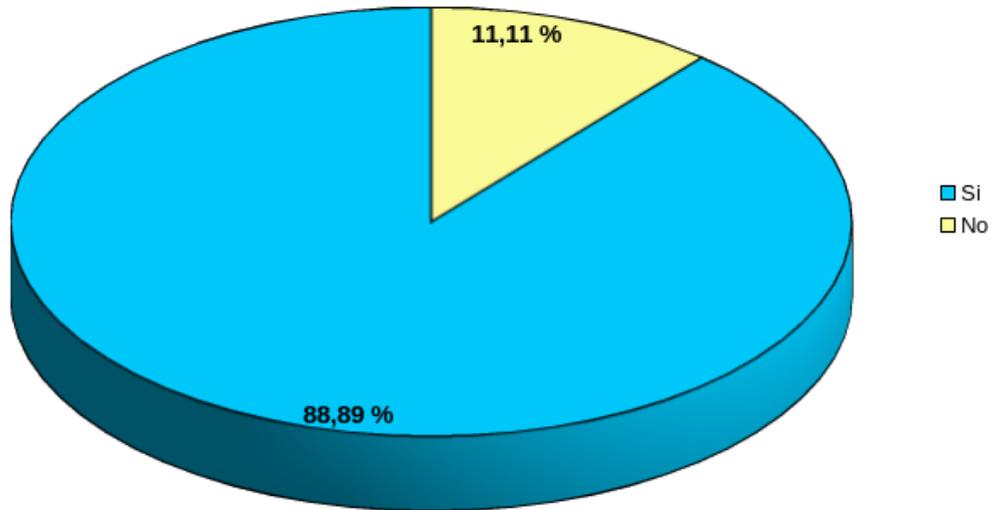


Figura 28: Pregunta 5

| Cambios según la Normativa | Respuestas | Porcentaje |
|----------------------------|------------|------------|
| Si | 8 | 88.89 % |
| No | 1 | 11.11 % |

Cuadro 31: Pregunta 5

Analisis:

Con referencia a la pregunta anterior se pregunta al encargado de aula CRA si ha realizado los cambios de software privativo a software libre según la normativa, el 88.89 % respondió que si realizo el cambio y el 11.11 % que contesto que no.

Pregunta 6: Si la respuesta anterior fue si, ¿qué inconvenientes ha tenido en la migración de software privativo a software libre?

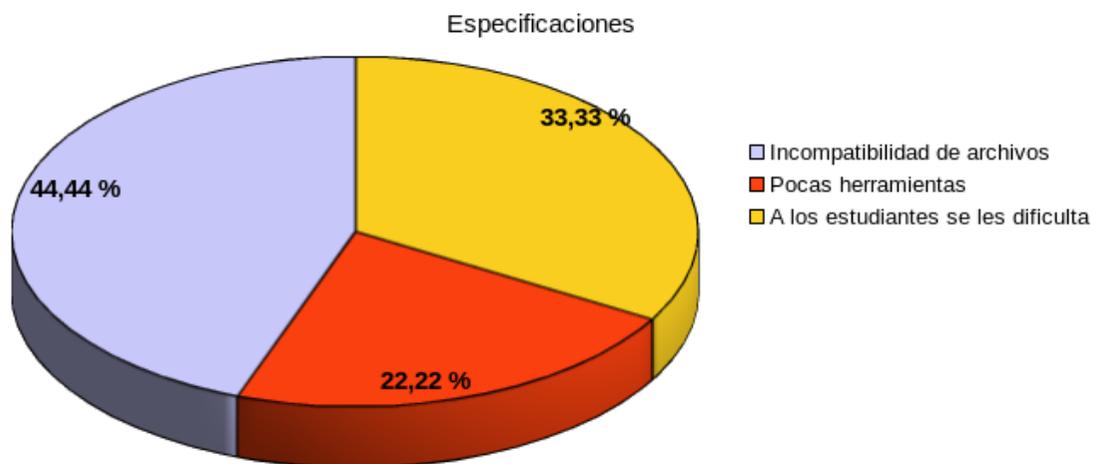


Figura 29: Pregunta 6

| Inconvenientes | Respuestas | Porcentaje |
|------------------------------------|------------|------------|
| Incompatibilidad de archivos | 4 | 44.44 % |
| Pocas herramientas | 2 | 22.22 % |
| A los estudiantes se les dificulta | 3 | 33.33 % |

Cuadro 32: Pregunta 6

Analisis:

Con referente a la respuesta anterior a los que respondieron que sí, se pregunta ¿qué inconvenientes ha tenido en la migración de software privativo a software libre?, el mayor porcentaje lo tiene la Incompatibilidad de archivos con el 44.44 %, el otro inconveniente es que a los estudiantes se les dificulta con el 33.33 % y el ultimo inconveniente es que hay Pocas herramientas con 22.22 %

Pregunta 7: ¿Ha recibido alguna capacitación para el uso de distribuciones de GNU-Linux?

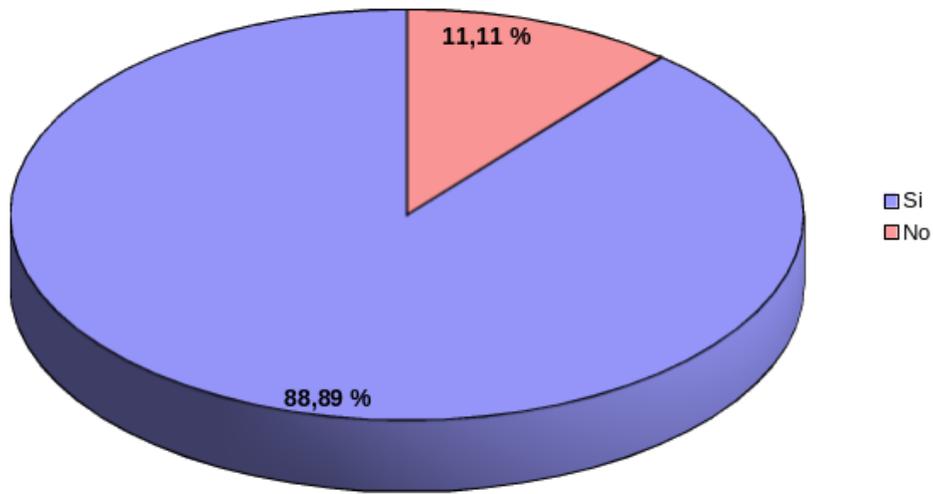


Figura 30: Pregunta 7

| Ha recibido capacitación | Respuestas | Porcentaje |
|--------------------------|------------|------------|
| Si | 8 | 88.89 % |
| No | 1 | 11.11 % |

Cuadro 33: Pregunta 7

Analisis:

También se les pregunto si ha recibido alguna capacitación para el uso de distribuciones de GNU-Linux el 88.89 % contestó que sí y el 11.11 % contestó que no.

Pregunta 8: ¿Cuál es la distribución de GNU-Linux que utiliza?

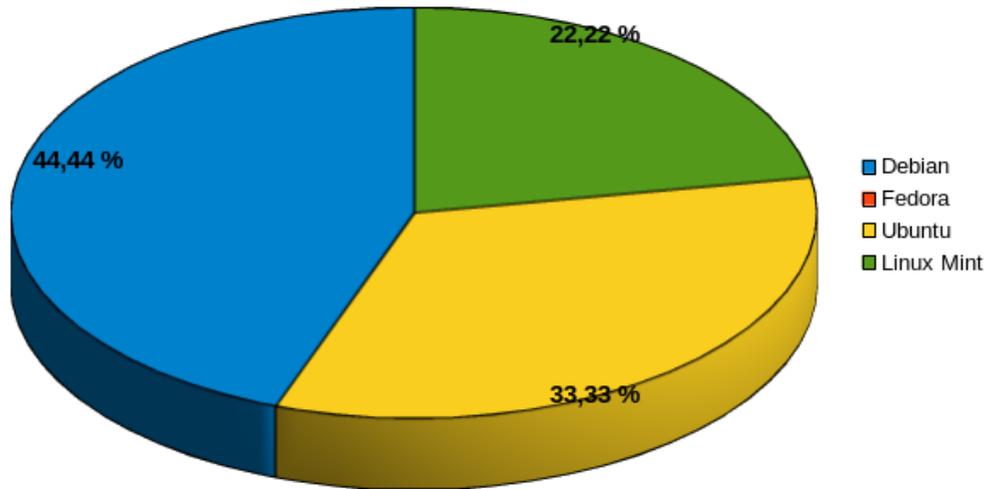


Figura 31: Pregunta 8

| Distribución | Respuestas | Porcentaje |
|--------------|------------|------------|
| Debian | 4 | 44.44 % |
| Fedora | 0 | 0.00 % |
| Ubuntu | 3 | 33.33 % |
| Linux Mint | 2 | 22.22 % |

Cuadro 34: Pregunta 8

Analisis:

Con referente a lo anterior a los que contestaron que sí, se le pregunta cuales son las distribuciones de GNU-Linux que utiliza, la más utilizada fue Debían con el 44.44 % seguida por Ubuntu con el 33.33 % y Linux Mint con el 22.22 %

Pregunta 9: ¿Considera que la migración de software privativo a software libre tiene beneficios en la enseñanza de los estudiantes?

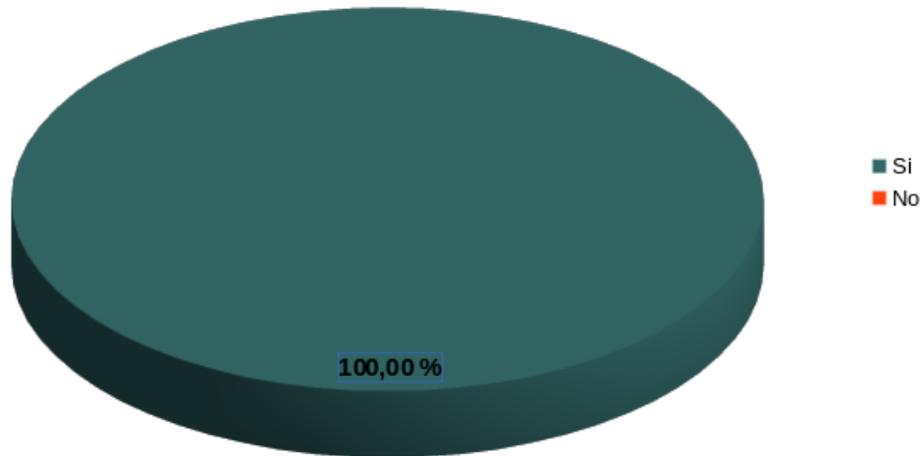


Figura 32: Pregunta 9

| Tiene Beneficios | Respuestas | Porcentaje |
|------------------|------------|------------|
| Si | 9 | 100 % |
| No | 0 | 0.00 % |

Cuadro 35: Pregunta 9

Analisis:

Se pregunta si Considera que la migración de software privativo a software libre tiene beneficios en la enseñanza de los estudiantes, el 100 % respondió que si tiene beneficios.

Porque

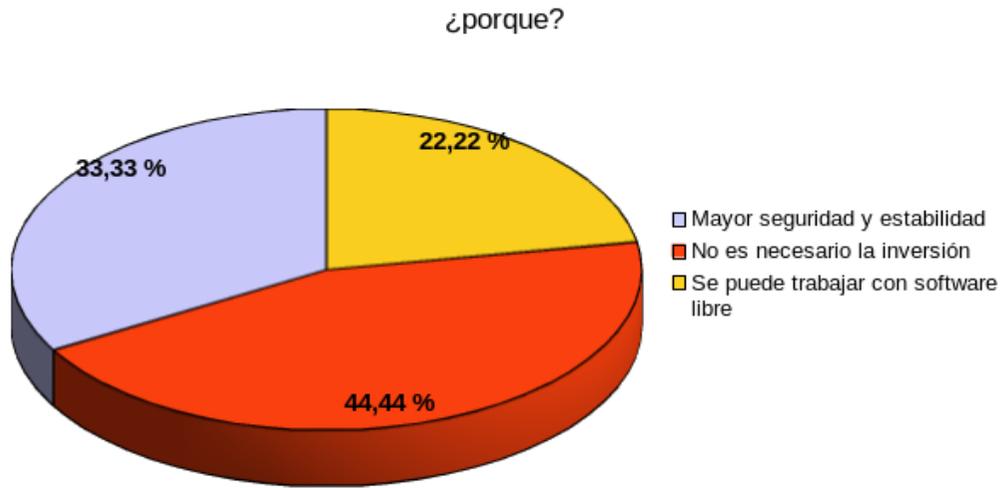


Figura 33: Pregunta 9

| Por qué | Respuestas | Porcentaje |
|--------------------------------------|------------|------------|
| Mayor seguridad y estabilidad | 3 | 33.33 % |
| No es necesario la inversión | 4 | 44.44 % |
| Se puede trabajar con software libre | 2 | 22.22 % |

Cuadro 36: Pregunta 9a

Analisis:

Con referencia a la pregunta anterior, se pregunta cuáles son los beneficios, el 44.44 % contesto que No es necesario la inversión, el 33.33 % contesto que hay una Mayor seguridad y estabilidad y el 22.22 % contesto que Se puede trabajar con software libre

Analisis General

Analisis encuestas a estudiantes.

Por las observaciones realizadas al visitar las instituciones educativas, se constató que cuentan al menos con un aula informática, donde los estudiantes reciben formación académica en el área de informática y que el 86.85 % hace uso de herramientas informáticas para las clases en la institución

El tema de software libre en las Instituciones Educativas no ha tenido gran impacto, a pesar de existir una normativa emitida por el MINED, no se han realizado los esfuerzos necesarios para implementarlo en las AI. Los resultados demuestran en la pregunta número 4 que solamente el 57.75 % tiene conocimientos o ha escuchado hablar sobre el software libre, mientras que un 42.25 % dijo no conocerlo, como podemos ver se obtuvo un porcentaje bastante cerrado en cuanto a las respuestas.

Analisis encuestas a docentes.

En las encuestas dirigidas a docentes se les pregunta sobre de ofimática enseña a sus alumnos, 9 de 9 de los docentes ofimática de Microsoft todos enseñan software privativo y 7 de 9 docentes usan software libre ofimática de libre Office y 2 de los 9 usan Corel Draw que también es un software privativo, el 66.67 % enseñan programación a sus estudiantes entre los lenguajes que se enseñan están el 50.00 % enseña PHP, el 33.33 % JAVA y un 16.67 % respondió que otros, en los cuales se mencionó HTML. También el 77.78 % hacen uso de base de datos entre los gestores de bases de datos están MariaDB con 33.33 % y SQLite, PostgreSQL, ORACLE, Otros (Acces, SQLserver) cada uno con el 16.67 %.

Los docentes en su mayoría se encontraban preparados técnicamente, debido a que han recibido capacitación por parte del Ministerio de Educación que brindara ayuda sobre la utilización de software libre en las Instituciones Educativas. El 77.78 % han recibido las capacitaciones que representa a 7 de 9 docentes y el 22.22 % que representa 2 de 9 docentes no

han recibido capacitaciones, entre las capacitaciones que han recibido se mencionan grado digital con el 57.14 % , manejo de cálculo con software libre con el 28.57 % y Los software que deben utilizar en las instituciones con el 14.29 %. Y por último se pregunta a los docentes si tendría algún inconveniente para desarrollar sus clases haciendo uso del Software Libre el 89.89 % contesto que no tendría inconveniente y un 11.11 % respondió que sí tendría inconvenientes. Entre los inconvenientes mencionados tenemos, El 50.00 % Incompatibilidad con el hardware, el 25.00 % Mala interpretación de la documentación y el 25 % Dificultades Técnicas

Analisis encuestas a dirigentes CRA.

Se pregunta a los encargados de aulas CRA cuáles son las funciones que ellos desempeñan el 66.66 % nos contestó que era impartir clases y administrar los recursos del CRA, El 29 % brindar mantenimiento preventivo y correctivo del equipo informático y un 4.17 % se encarga de verificar el buen uso de los equipos. Todas las instituciones educativas contaban con equipo informático ya sea computadoras de escritorio, laptop, proyectores, servidores los cuales son utilizados para impartir las clases informáticas y por supuesto todos tienen conexión a internet.

En cuanto a preparación técnica por parte de los coordinadores de aula CRA respecto a la utilización de software libre, se les cuestionó si tenían conocimiento sobre la utilización de software libre el 88.89 % dijo tener capacitación para el uso de distribuciones de GNU-Linux y el 11.11 % contesto que no. Siendo Debian con el 44.44 % seguida por Ubuntu con el 33.33 % y Linux Mint con el 22.22 % las distribuciones de GNU-Linux, también se pregunta si ha realizado los cambios de software privativo a software libre según la normativa las respuestas fueron el 88.89 % dijo haber hecho los cambios y el 11.11 % contesto que no. Es decir que la mayoría de coordinadores poseían conocimientos para implementar el software libre en las instituciones educativas también se les pregunta si considera que la migración de software privativo a software libre tiene beneficios en la enseñanza de los estudiantes el

100 % dijo que si entre los beneficios mencionados tenemos, Mayor seguridad y estabilidad 33.33 % el 44.44 % respondió que no es necesario la inversión y el 22.22 % respondió que se puede trabajar con software libre.

5.2. Comprobación de Hipótesis

5.2.1. Hipótesis general.

Hi: El uso de Software libre causa un impacto social, económico y educativo en las instituciones de educación media del municipio de San Miguel.

Impacto Social: Al preguntar a los estudiantes si Tiene conocimientos o había escuchado hablar sobre el software libre más del 57 % respondió que sí, al igual que los docentes y los encargados de aulas CRA la mayoría respondió si conocer sobre software libre, lo que indica que si ha habido un impacto social tanto en estudiantes, docentes y encargados de Aulas CRA.

Impacto Económico: Se les pregunta a los encargados del CRA sobre los beneficios que ellos consideran que tiene el software libre el 44.44 % contestó que No es necesario la inversión, el 33.33 % contestó que hay una Mayor seguridad y estabilidad y el 22.22 % contestó que Se puede trabajar con software libre En este sentido la población estudiantil, es beneficiada ya sea en una forma de ahorro o uso de tecnologías libre; era considerable ya que no tendrían que invertir en licencias de Microsoft y gozar de los beneficios de software libre. Con las plataformas libres los costos de inversión en software propietario, podrían invertirse en la mejora educativa.

Impacto Educativo: como podemos ver en el análisis el software libre si ha impactado en el sector educativo como podemos ver un porcentaje poco mayor a la media tiene conocimientos sobre el software libre, También los docentes y encargados de aulas CRA tienen conocimientos sobre la normativa del MINED en la implementación del software Libre en las Aulas informáticas, y también la mayoría de docentes ya han sido capacitados por parte del MINED para poder implementarlo.

5.2.2. Hipótesis Específicas.

Hipótesis Específica I

El uso del software libre brinda beneficios económicos, educativos y de carácter social a la población estudiantil, planta docente y administrativa de las instituciones de educación media del municipio de San Miguel.

Según las encuestas realizadas a estudiantes, docentes y encargados de aulas CRA el software libre brinda beneficios a todas estas áreas de las instituciones, beneficios económicos por que el software libre es gratuito, permite reducir costos, las licencias son libres, educativo porque los estudiantes pueden interactuar con programas gratuitos y social porque brinda un cambio de percepción en los estudiantes y docentes a vivir en una sociedad independiente, solidaria y libre.

Hipótesis Específica II

La inadecuada formación y capacitación en herramientas tecnológicas libres dificulta la implementación de la normativa propuesta por el MINED acerca de la migración al uso de software libre en las aulas de Centro de Recursos de Aprendizajes (Aulas CRA).

Si bien es cierto según las encuestas realizadas a los docentes y encargados de aulas CRA afirmaron haber recibido capacitación y haber implementado la normativa propuesta por el MINED acerca de la migración al uso de software libre en las aulas de Centro de Recursos de Aprendizajes pero sin embargo siguen trabajando con el software privativo, cuando se preguntó sobre la ofimática que utilizan 9 de los 9 docentes usan Microsoft office y solamente 6 de los 9 docentes utilizan en simultáneo libreOffice. Esto nos indica que a pesar de haber implementado la normativa siguen usando software privativo, lo cual se deduce que la mayoría de docentes siguen enseñando software privativo en los centros educativos.

Hipótesis Específica III

El Software libre proporciona cambios en los modelos pedagógicos, brindando materiales educativos, acceso a recursos bibliográficos, etc. Así como cambios en la gestión administrativa de las instituciones de educación media del municipio de San Miguel.

Se puede deducir que las herramientas de software libre si proporciona cambios en los modelos pedagógicos pues facilita que los alumnos dispongan en su casa de las mismas herramientas educativas que utilizan en su centro, de forma 100 % legal. Poder utilizar estas aplicaciones fuera del aula sólo es posible si están disponibles de forma abierta y gratuita en cualquier ámbito y con cualquier finalidad.

CAPITULO VI
CONCLUSIONES Y RECOMENDACIONES

6. Conclusiones y Recomendaciones

6.1. Conclusiones

- El Software libre si ha causado un impacto económico, educativo y social en la población estudiantil, planta docente y administrativa de las instituciones de educación media del municipio de San Miguel.
- El Software libre ha proporcionado cambios en los modelos pedagógicos, brindando materiales educativos, acceso a recursos bibliográficos, etc. Así como cambios en la gestión administrativa de las instituciones de educación media del municipio de San Miguel.
- La elaboración de una plataforma virtual para registrar información útil sobre el uso del software libre en el país proporciona una herramienta que brinde orientación a organizaciones tanto gubernamentales cómo no gubernamentales sobre los beneficios que ofrece el software libre en la administración y operación de dichas instituciones.
- La plataforma virtual se ha desarrollado puramente con herramientas basadas en software libre, es multiplataforma y responsive, lo cual permite acceder a ella desde cualquier navegador web, dispositivos móviles, portátiles y computadoras de escritorio.
- Al ser llevado a cabo el proyecto con herramientas de software libre, los costos de desarrollo se limitan únicamente a hora-humano y los gastos que ello implica, alimentación, transporte, etc.

6.2. Recomendaciones

6.2.1. Poblacion estudiantil.

- Referente a considerar aprender alternativa a las tecnologías de Microsoft para incrementar las oportunidades en el área profesional.
- Incorporar tecnología libre para poder realizar proyectos de software y tecnología para motivar el proceso de enseñanza-aprendizaje.
- Realizar el proceso de migración hacia las tecnologías libres que permita explorar la plataforma de Linux con LibreOffice Writer en un ambiente educativo.
- Implementar las tecnologías de software libre para poder evaluar el código fuente de las aplicaciones y conocer el funcionamiento de las mismas.
- Explorar las ventajas y desventajas de la plataforma de software libre para poder comparar con las tecnologías de software de Microsoft.
- Facilitar las herramientas de formación a la población educativa en general para realizar la implantación del software libre de forma eficiente.

6.2.2. Poblacion docente.

- Promover las tecnologías libres y sus beneficios intangibles dentro de un ambiente educativo.
- Aprender las ventajas y desventajas del software libre para realizar una comparación con las tecnologías propietarias.
- Promover el uso de la plataforma de Linux para motivar al estudiante a realizar proyectos de software.

- Hacer uso de las herramientas que el MINED posee para certificaciones del software libre, así se podrá contar con docentes formados en tecnológicas de uso libre. MINED

- Facilitar al coordinador y docente planes de estudio, guías y manuales de usuario para realizar la implementación del software libre de una forma efectiva y que permita brindar el un mayor acceso a la población estudiantil.

- Seguir brindando oportunidades con recurso tecnológico a las Instituciones Educativas.

- El MINED debe proporcionar capacitaciones y documentación técnica sobre el uso de GNU-Linux, para que los coordinadores cuenten con herramientas que les faciliten el cambio de sistema operativo.

- Además del paquete de ofimática libre que recomienda el MINED, las Instituciones Educativas pueden optar por utilizar WPS Office, que es un paquete similar a Microsoft y es totalmente libre.

- El Ministerio de Educación debió de utilizar distintos medios para dar conocer la normativa de tal manera que llegara a conocerse por todos los que conforman las diferentes Instituciones Educativas.

- Continuar los controles y auditorías de software para medir el nivel de cumplimiento y evitar caer en problemas legales.

- Proporcionar manuales de usuario que permitan realizar exitosamente la migración a software libre, así como también hacer uso de algún medio como por ejemplo, línea telefónica, chat, asistencia técnica personalizada que permitan solventar cualquier duda por parte de los encargados de aulas CRA.

- Proporcionar de equipo tecnológico actualizado las Aulas Informáticas de las Instituciones Educativas, ya que la mayoría de esta cuentan con equipo obsoleto y no permiten

instalar el software libre.

- Motivar a docentes para que hagan uso de las herramientas con las que se cuenta, por ejemplo, certificaciones de grado digital o formación en el área informática por parte de los encargados de Aulas Informáticas.

6.2.3. Coordinadores de aulas informáticas.

- Promover el uso de las herramientas del software libre mediante la implementación de GNU-Linux Max con Libre Office.
- Comunicar los inconvenientes acerca del proceso de implementación a las entidades correspondientes para facilitar la implementación de GNU-Linux
- Realizar la formación mediante capacitaciones y certificación de grado digital para facilitar el proceso de implementación de GNU-Linux
- Fomentar el uso de código bajo licencias de software libre para el desarrollo de proyectos bajo estas licencias.
- Promover de guías de estudio como apoyo a la población estudiantil en el uso de herramientas de GNU-Linux.
- Realizar una calendarización y planificación de clases donde necesite hacer uso del equipo informático con el que cuenta la institución para desarrollar sus clases. Es importante priorizar los contenidos para mostrar en cuales será necesario.

7. Documentación de la plataforma

7.1. Documentación plataforma

7.1.1. Introducción

En el presente capítulo se desarrollará la plataforma donde se recopilará la información obtenida en la investigación. Esta plataforma web dará apoyo a la gestión educativa; permite un fácil acceso a información sobre la temática del software libre. En este proyecto se pretende elaborar una plataforma web que unifique la gestión documental en donde los usuarios tengan acceso a compartir información en base a sus experiencias personales con el uso de herramientas informáticas basadas en software libre.

7.1.2. Objetivo

Organizar y recopilar toda la información por estudiante, docentes, administradores de centros de cómputo y encargados de IT de las instituciones.

7.2. Diseño y desarrollo de la plataforma

El diseño de la plataforma está orientada especialmente a la recopilación de información que sirva de apoyo a la toma de decisiones sobre el uso, implementación y aplicación de herramientas de software libre en las instituciones tanto gubernamentales como no gubernamentales. Actualmente, las plataformas educativas se utilizan también para crear espacios de discusión y construcción de conocimiento por parte de grupos de investigación, o para la implementación de comunidades virtuales y redes de aprendizaje, por parte de grupos de personas unidos en torno a una temática de interés.

Desarrollo del software

En el presente capítulo se desarrollará la plataforma donde se recopilará la información obtenida en la investigación esta plataforma web dará apoyo a la gestión educativa. Esta plataforma típicamente permiten el rápido acceso a información sobre la temática del software libre.

Entorno de Ejecución.

Entorno de ejecución

| Sistema operativo | Gestor de BD | Lenguaje de Desarrollo | Framework |
|---------------------|--------------|------------------------|------------------------|
| Linux | MariaDB | PHP | Framework |
| Version 9.0 Stretch | MariaDB | PHP 7.2 | CodeIgniter, Bootstrap |

Cuadro 37: Entorno de Ejecución

Tecnologías utilizadas.

HTML



(HyperText Markup Language, versión 5) es la quinta revisión importante del lenguaje básico de la World Wide Web, HTML. HTML5 especifica dos variantes de sintaxis para HTML: una «clásica», HTML (text/html), conocida como HTML5, y una variante XHTML conocida como sintaxis XHTML 5 que deberá servirse con sintaxis XML (application/xhtml+xml).

PHP (Hypertext Preprocessor)



Lenguaje de scripting ejecutado en el servidor web, es el sucesor de un producto llamado PHP/FI. Creado en 1994 por Rasmus Lerdorf, la primera versión fue un conjunto simple de archivos binarios CGI escritos en C.

El PHP, es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor. La meta del lenguaje es permitir rápidamente a los desarrolladores la generación dinámica de páginas. Puede ser usado en la mayoría de los servidores web al igual que en casi todos los sistemas operativos y plataformas sin ningún costo.

CodeIgniter



CodeIgniter es un framework para desarrollo de aplicaciones, desarrollada en PHP que utiliza el MVC(Modelo Vista Controlador). Esto permite a los programadores o desarrolla-

dores Web mejorar su forma de trabajar, además de dar una mayor velocidad a la hora de crear páginas Webs es un marco web de desarrollo rápido de software de código abierto, para su uso en la creación de sitios web dinámicos con PHP, para la creación de cualquier tipo de aplicación web bajo PHP.

Su objetivo es permitirle desarrollar proyectos mucho más rápido que lo que podría hacer si escribiera el código desde cero, proveyéndole un rico conjunto de bibliotecas para tareas comunes, así como y una interfaz sencilla y una estructura lógica para acceder a esas bibliotecas. CodeIgniter le permite enfocarse creativamente en su proyecto al minimizar la cantidad de código necesaria para una tarea dada

Atom



Es un editor de código fuente de código abierto para macOS, Linux, y Windows¹ con soporte para múltiples plug-in escritos en Node.js y control de versiones Git integrado, desarrollado por GitHub. Atom es una aplicación de escritorio construida utilizando tecnologías web.² La mayor parte de los paquetes tienen licencias de software libre y está desarrollados y mantenidos por la comunidad de usuarios.³ Atom está basado en Electron (Anteriormente conocido como Atom Shell), Un framework que permite crear aplicaciones de escritorio multiplataforma usando Chromium y Node.js.⁴ Está escrito en CoffeeScript y Less.⁷ También puede ser utilizado como un entorno de desarrollo integrado (IDE).⁸ Atom liberó su beta en la versión 1.0, el 25 de junio de 2015.

7.2.1. Requerimientos de usuario.

Al ser una plataforma web se necesita un equipo con el desempeño suficiente para los navegadores actuales. Los requerimientos mínimos de usuario son:

| Hardware | Especificación |
|-------------|---------------------------------|
| Procesador | Core i3, 3.7 GHz CPU o Superior |
| Memoria RAM | 4GB o Más |
| Disco Duro | 500GB o Más |

Cuadro 38: Requerimientos de hardware

| Software | Especificación |
|-------------------------|--|
| Sistema Operativo | Debian 8 o superior, ya sea de 32bits o 64bits |
| Gestor de Base de Datos | Maria DB |
| Editor de Texto | Atom |
| Framework | CodeIgniter |

Cuadro 39: Requerimientos de software

7.2.2. Requerimientos del sistema.

El sistema va a requerir un servidor en el cual estará alojada la base de datos y la aplicación desde la cual se tendrá acceso por medio de una red desde otro equipo de cómputo el cual deberá tener los requerimientos mínimos ya mencionados en el punto anterior a este.

Requisitos funcionales del sistema CRUD de Usuarios

- La plataforma web debe poder registrar usuarios en la base de datos.
- En la plataforma web el usuario registrado podrá agregar una investigación.
- En la plataforma web el usuario registrado podrá modificar una investigación.
- En la plataforma web el usuario registrado podrá eliminar una investigación.

CRUD deAdministrador.

- La plataforma web el administrador debe poder registrar usuarios en la base de datos.
- En la plataforma web el administrador podrá agregar una investigación a la plataforma.
- En la plataforma web el administrador podrá modificar una investigación a la plataforma.
- En la plataforma web el administrador podrá eliminar una investigación a la plataforma.
- En la plataforma web el administrador podrá agregar a los usuarios de la plataforma.
- En la plataforma web el administrador podrá modificar los usuarios de la plataforma.
- En la plataforma web el administrador podrá eliminar usuarios de la plataforma.
- Autenticación de los usuarios, los usuarios requieren iniciar sesión para poder tener acceso a todos los procesos del sistema.
- El sistema controlará el acceso y lo permitirá solamente a usuarios autorizados.
- Los usuarios deben ingresar al sistema con un nombre de usuario y contraseña.

Requerimientos no funcionales.

- La plataforma debe ser amigable y de fácil uso por parte de los usuarios.
- La plataforma debe tener Correcta redacción y ortografía en las pantallas.
- La plataforma no debe permitir el cierre de una operación hasta que todos sus procesos hayan sido terminados y cerrados satisfactoriamente.

7.2.3. Composición y función de la plataforma

La plataforma está compuesta por módulos dónde un usuario pueda registrarse y pueda subir información. La plataforma se encargará de unificar en un solo lugar, clasificar por tipo de documento y por categoría. Esta plataforma la utilizará todo el personal encargado de aulas CRA, encargados de IT que estén registrados y los visitantes podrán ver las investigaciones que los usuarios hayan registrado. Además deberá haber un usuario administrador que se encargará de autorizar usuarios, publicar investigaciones y llevar un control de las altas y bajas de documentos.

7.2.4. Diagrama de Base de Datos.

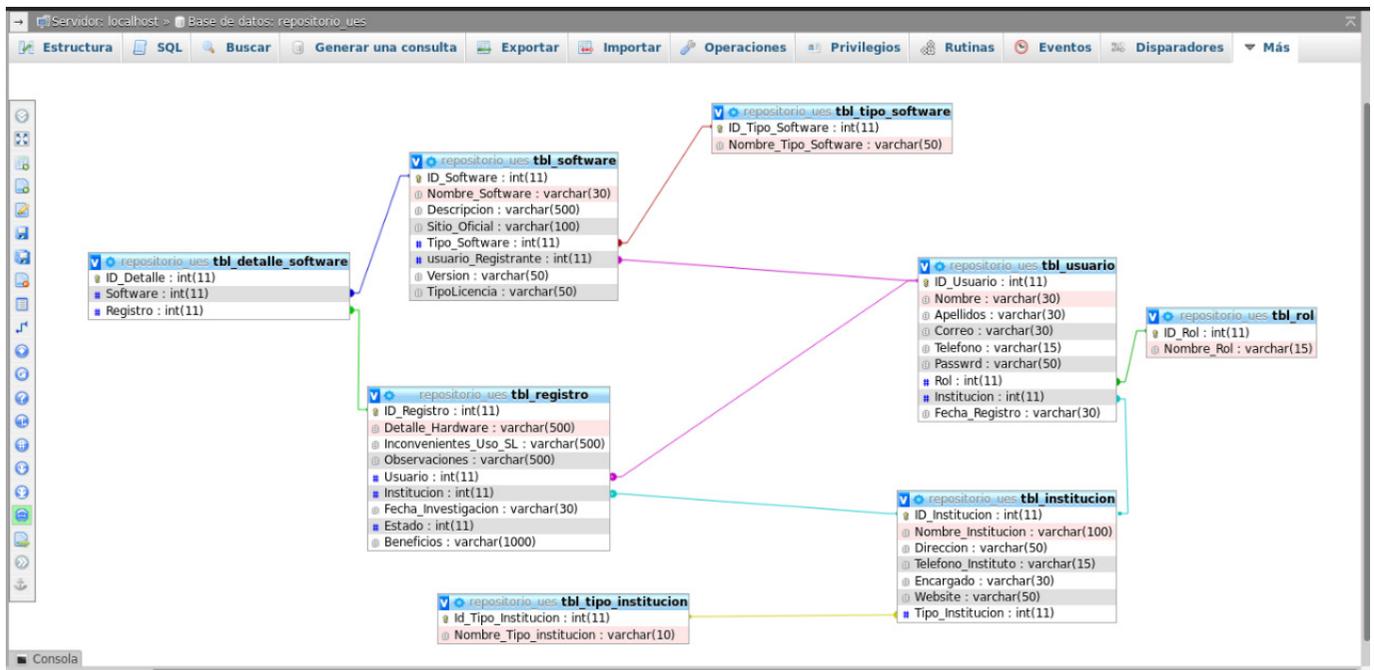


Figura 34: Diagrama Base de Datos

7.2.5. Diagrama de Clases.

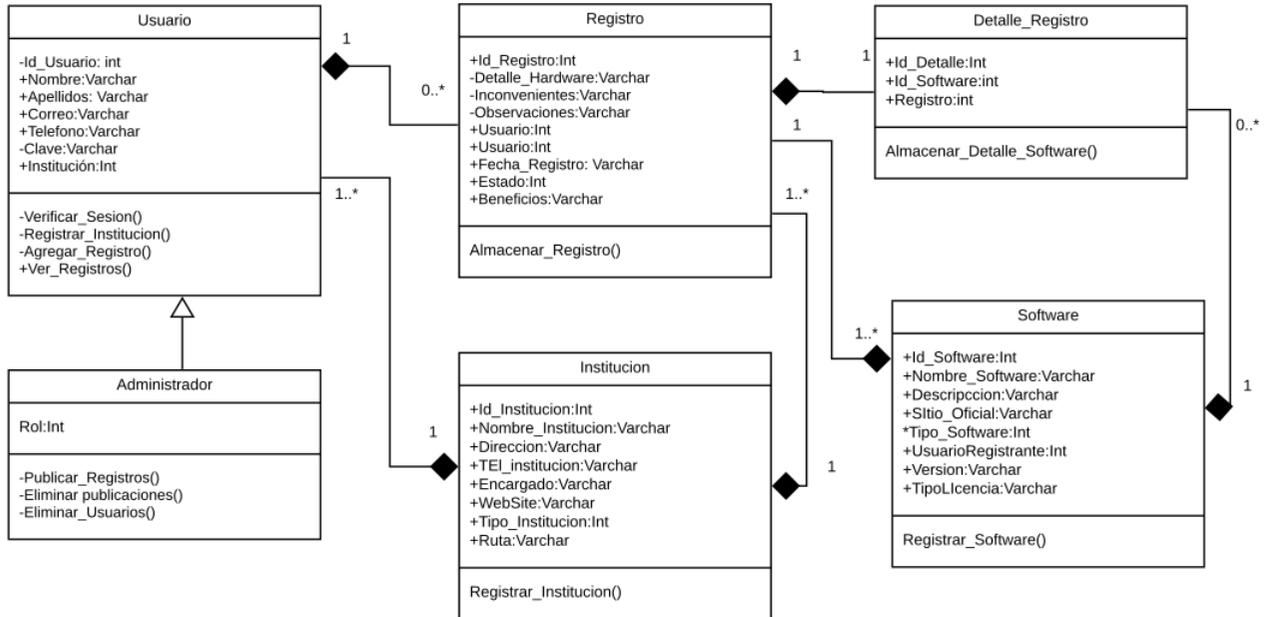


Figura 35: Diagrama de Clases

7.3. Diccionario de Base de Datos.

7.3.1. Tabla detalle de Software

tbl_detalle_software

| Columna | Tipo | Nulo | Predeterminado | Enlaces a | Comentarios | MIME |
|--------------------------|---------|------|----------------|-----------------------------|-------------|------|
| ID_Detalle (Primaria) | int(11) | No | | | | |
| Software | int(11) | No | | tbl_software -> ID_Software | | |
| Registro | int(11) | No | | tbl_registro -> ID_Registro | | |

Índices

| Nombre de la clave | Tipo | Único | Empaquetado | Columna | Cardinalidad | Cotejamiento | Nulo | Comentario |
|--------------------|-------|-------|-------------|------------|--------------|--------------|------|------------|
| PRIMARY | BTREE | Sí | No | ID_Detalle | 1 | A | No | |
| Software | BTREE | No | No | Software | 1 | A | No | |
| | | | | Registro | 1 | A | No | |
| Registro | BTREE | No | No | Registro | 1 | A | No | |

Figura 36: Tabla detalle de Software

7.3.2. Tabla Institución

tbl_institucion

| Columna | Tipo | Nulo | Predeterminado | Enlaces a | Comentarios | MIME |
|------------------------------|--------------|------|----------------|---|-------------|------|
| ID_Institucion (Primaria) | int(11) | No | | | | |
| Nombre_Institucion | varchar(100) | No | | | | |
| Direccion | varchar(50) | No | | | | |
| Telefono_Instituto | varchar(15) | No | | | | |
| Encargado | varchar(30) | No | | | | |
| Website | varchar(50) | No | | | | |
| Tipo_Institucion | int(11) | No | | tbl_tipo_institucion -> Id_Tipo_Institucion | | |

Índices

| Nombre de la clave | Tipo | Único | Empaquetado | Columna | Cardinalidad | Cotejamiento | Nulo | Comentario |
|--------------------|-------|-------|-------------|------------------|--------------|--------------|------|------------|
| PRIMARY | BTREE | Sí | No | ID_Institucion | 3 | A | No | |
| Tipo_Institucion | BTREE | No | No | Tipo_Institucion | 3 | A | No | |

Figura 37: Tabla Institución

7.3.3. Tabla Registros

tbl_registro

| Columna | Tipo | Nulo | Predeterminado | Enlaces a | Comentarios | MIME |
|------------------------|---------------|------|----------------|-----------------------------------|-------------|------|
| ID_Registro (Primaria) | int(11) | No | | | | |
| Detalle_Hardware | varchar(500) | No | | | | |
| Inconvenientes_Uso_SL | varchar(500) | No | | | | |
| Observaciones | varchar(500) | No | | | | |
| Usuario | int(11) | No | | tbl_usuario -> ID_Usuario | | |
| Institución | int(11) | No | | tbl_institucion -> ID_Institucion | | |
| Fecha_Investigacion | varchar(30) | No | | | | |
| Estado | int(11) | No | | | | |
| Beneficios | varchar(1000) | No | | | | |

Índices

| Nombre de la clave | Tipo | Único | Empaquetado | Columna | Cardinalidad | Cotejamiento | Nulo | Comentario |
|--------------------|-------|-------|-------------|-------------|--------------|--------------|------|------------|
| PRIMARY | BTREE | Sí | No | ID_Registro | 1 | A | No | |
| Institucion | BTREE | No | No | Institucion | 1 | A | No | |

Figura 38: Tabla Registros

7.3.4. Tabla Rol

tbl_rol

| Columna | Tipo | Nulo | Predeterminado | Enlaces a | Comentarios | MIME |
|----------------------------|-------------|------|----------------|-----------|-------------|------|
| ID_Rol (<i>Primaria</i>) | int(11) | No | | | | |
| Nombre_Rol | varchar(15) | No | | | | |

Índices

| Nombre de la clave | Tipo | Único | Empaquetado | Columna | Cardinalidad | Cotejamiento | Nulo | Comentario |
|--------------------|-------|-------|-------------|---------|--------------|--------------|------|------------|
| PRIMARY | BTREE | Sí | No | ID_Rol | 2 | A | No | |

Figura 39: Tabla Rol

7.3.5. Tabla de Software

tbl_software

| Columna | Tipo | Nulo | Predeterminado | Enlaces a | Comentarios | MIME |
|---------------------------------|--------------|------|----------------|---------------------------------------|-------------|------|
| ID_Software (<i>Primaria</i>) | int(11) | No | | | | |
| Nombre_Software | varchar(30) | No | | | | |
| Descripción | varchar(500) | No | | | | |
| Sitio Oficial | varchar(100) | No | | | | |
| Tipo_Software | int(11) | No | | tbl_tipo_software -> ID_Tipo_Software | | |
| usuario_Registrante | int(11) | Sí | NULL | tbl_usuario -> ID_Usuario | | |
| Versión | varchar(50) | No | | | | |
| TipoLicencia | varchar(50) | No | | | | |

Índices

| Nombre de la clave | Tipo | Único | Empaquetado | Columna | Cardinalidad | Cotejamiento | Nulo | Comentario |
|---------------------|-------|-------|-------------|---------------------|--------------|--------------|------|------------|
| PRIMARY | BTREE | Sí | No | ID_Software | 1 | A | No | |
| Tipo_Software | BTREE | No | No | Tipo_Software | 1 | A | No | |
| tbl_software_ibfk_2 | BTREE | No | No | usuario_Registrante | 1 | A | Sí | |

Figura 40: Tabla Software

7.3.6. Tabla Tipo de Institución

tbl_tipo_institucion

| Columna | Tipo | Nulo | Predeterminado | Enlaces a | Comentarios | MIME |
|---|-------------|------|----------------|-----------|-------------|------|
| Id_Tipo_Institucion (<i>Primaria</i>) | int(11) | No | | | | |
| Nombre_Tipo_institucion | varchar(10) | No | | | | |

Índices

| Nombre de la clave | Tipo | Único | Empaquetado | Columna | Cardinalidad | Cotejamiento | Nulo | Comentario |
|--------------------|-------|-------|-------------|---------------------|--------------|--------------|------|------------|
| PRIMARY | BTREE | Sí | No | Id_Tipo_Institucion | 2 | A | No | |

Figura 41: Tabla Tipo de Institución

7.3.7. Tabla Tipo de Software

tbl_tipo_software

| Columna | Tipo | Nulo | Predeterminado | Enlaces a | Comentarios | MIME |
|--------------------------------------|-------------|------|----------------|-----------|-------------|------|
| ID_Tipo_Software (<i>Primaria</i>) | int(11) | No | | | | |
| Nombre_Tipo_Software | varchar(50) | No | | | | |

Índices

| Nombre de la clave | Tipo | Único | Empaquetado | Columna | Cardinalidad | Cotejamiento | Nulo | Comentario |
|--------------------|-------|-------|-------------|------------------|--------------|--------------|------|------------|
| PRIMARY | BTREE | Sí | No | ID_Tipo_Software | 2 | A | No | |

Figura 42: Tabla Tipo de Software

7.3.8. Tabla Usuario

tbl_usuario

| Columna | Tipo | Nulo | Predeterminado | Enlaces a | Comentarios | MIME |
|--------------------------------|-------------|------|----------------|-----------------------------------|-------------|------|
| ID_Usuario (<i>Primaria</i>) | int(11) | No | | | | |
| Nombre | varchar(30) | No | | | | |
| Apellidos | varchar(30) | No | | | | |
| Correo | varchar(30) | No | | | | |
| Teléfono | varchar(15) | No | | | | |
| Passwrđ | varchar(50) | No | | | | |
| Rol | int(11) | No | | tbl_rol -> ID_Rol | | |
| Institucion | int(11) | No | | tbl_institucion -> ID_Institucion | | |
| Fecha_Registro | varchar(30) | No | | | | |

Índices

| Nombre de la clave | Tipo | Único | Empaquetado | Columna | Cardinalidad | Cotejamiento | Nulo | Comentario |
|--------------------|-------|-------|-------------|-------------|--------------|--------------|------|------------|
| PRIMARY | BTREE | Sí | No | ID_Usuario | 1 | A | No | |
| Rol | BTREE | No | No | Rol | 1 | A | No | |
| | | | | Institucion | 1 | A | No | |
| Institucion | BTREE | No | No | Institucion | 1 | A | No | |

Figura 43: Tabla Usuario

7.3.9. Caso de Uso

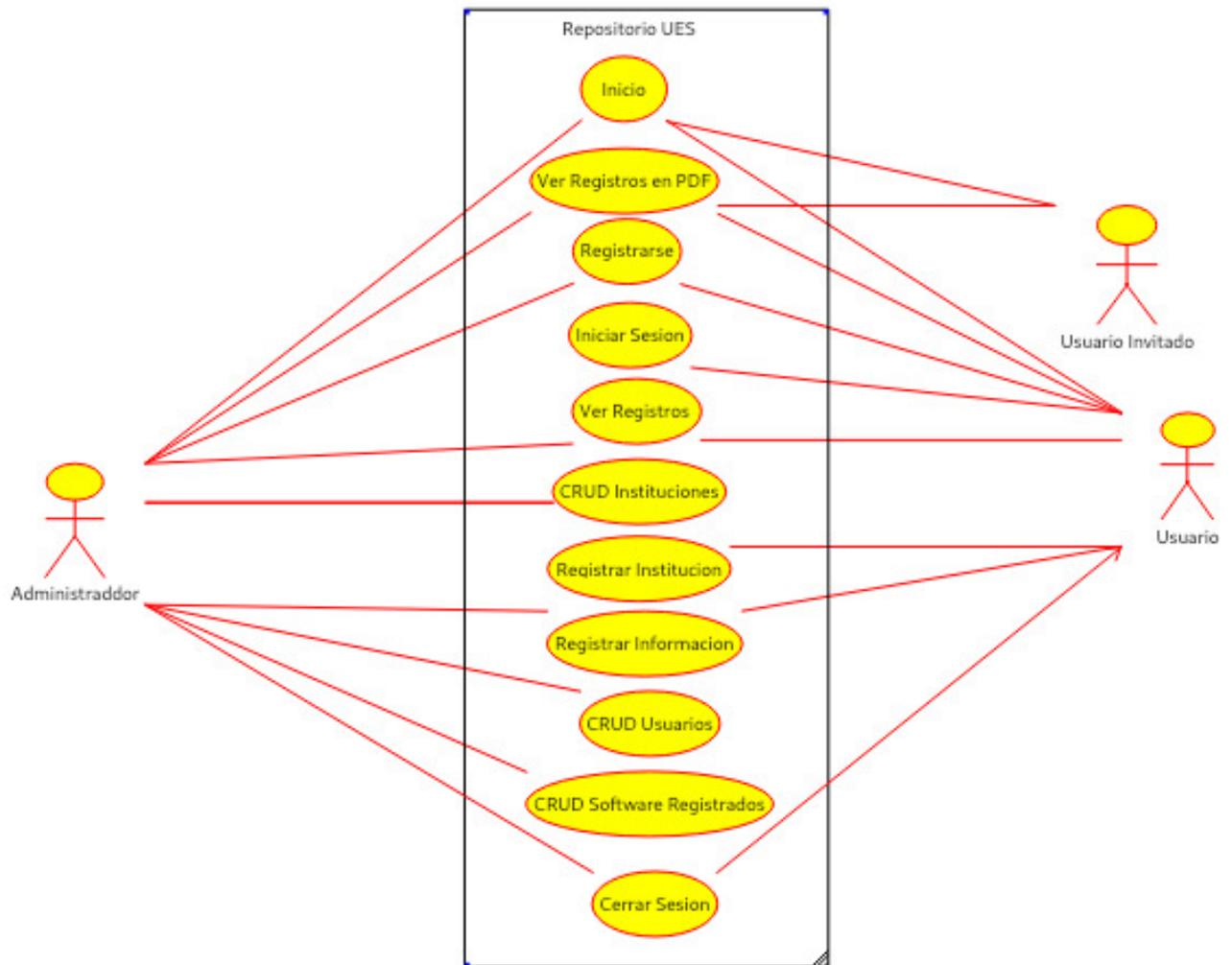


Figura 44: Caso de uso

8. Bibliografía

- [1] Williams, Sam (2 de octubre de 2010). Free as in Freedom (2.0): Richard Stallman and the Free Software Revolutoin (2 edición). GNU Press. p.71.
- [2] Rodríguez, G. (2008). «El software libre y sus implicaciones jurídicas». Revista de Derecho. Scielo. Consultado el 8 de abril de 2018
- [3] Sampieri, R. H. (2014) metodología de la investigación
- [4] Metodología de la Investigación (Sexta Edición ed.). Mexico: McGRAW-HILL.
- [5] García Córdoba, 2007, p.81 (La Investigación Tecnológica)
- [6] Recuperado de <http://softwarelibre20.blogspot.com/2013/04/motivaciones-del-software-libre.html>
- [7] Recuperado de <https://opensource.org/>
- [8] Recuperado de <http://www.carlospes.com/minidiccionario/mainframe.php>
- [9] Recuperado de www.gnome.org (en inglés). Consultado el 1 de abril de 2018.
- [10] Recuperado de www.gnu.org. Consultado el 21 de diciembre de 2016.
- [11] Recuperado de <http://www.oekonux.org/>
- [12] Recuperado de <https://docs.hipatia.net/dsl/>
- [13] Recuperado de <https://www.openbsd.org/>
- [14] Recuperado de <http://informativo.mined.gob.sv:8090/DNP/GTIC/LibreOffice/>
- [15] Recuperado de <http://www.gnu.org/licenses/license-list.html> NetscapeJavaScript
- [16] Recuperado de <http://www.eumed.net/rev/cccss/13/ibrs.html>

ANEXOS

9. Anexos

9.1. Encuestas

9.1.1. Encuestas dirigidas a encargados aulas CRA



UNIVERSIDAD DE EL SALVADOR
FACULTAD MULTIDISCIPLINARIA ORIENTAL
DEPARTAMENTO DE INGENIERIA Y ARQUITECTURA

Encuesta dirigida a los docentes a nivel de educación media del municipio de San Miguel, del departamento de San Miguel.

Objetivo: Conocer el nivel de utilización de las tecnologías de información y comunicación, especialmente el uso de herramientas relacionadas al software libre en el proceso de enseñanza a nivel de educación media.

Indicación: Marque la respuesta que considere conveniente.

1-¿Qué programas de ofimática enseña a sus alumnos? (Selección múltiple)

Procesador de Texto (Nombre) _____

Hojas de cálculo (Nombre) _____

Presentaciones (Nombre) _____

Otros (Tipo y Nombre) _____

2- ¿Enseña Programación a los estudiantes?

Sí

No

2.1- Si la respuesta anterior fue sí, ¿Que lenguajes de programación imparte a sus alumnos?

C++ JAVA

Python PHP

Otro (Especifique): _____

3- Si respondió que sí a la pregunta 2; ¿Que entorno de programación integrado (IDE) o que editores utiliza como herramienta para programar en clases?

NetBeans

Eclipse

Atom

Sublime Text

Figura 45: Encuesta a encargados aulas CRA

4- ¿Tiene conocimiento de la normativa del MINED en la implementación del software libre en las Aulas Informáticas?

Sí

No

5- ¿Ha realizado los cambios de software privativo a software libre según la normativa?

Sí

No

6- Si la respuesta anterior fue sí, ¿Qué inconvenientes ha tenido en la migración de software privativo a software libre?

Especifique:

7- ¿Ha recibido alguna capacitación para el uso de distribuciones de GNU-Linux?

Sí

No

8- ¿Cuál es la distribución de GNU-Linux que utiliza?

Debían

Fedora

Ubuntu

Linux Mint

9- ¿Considera que la migración de software privativo a software libre tiene beneficios en la enseñanza de los estudiantes?

Sí

No

¿Porque?

Figura 46: Encuesta a encargados aulas CRA

9.1.2. Encuestas dirigidas a docentes



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

Encuesta dirigida a los docentes a nivel de educación media del municipio de San Miguel, del departamento de San Miguel.

Objetivo: Conocer el nivel de utilización de las tecnologías de información y comunicación, especialmente el uso de herramientas relacionadas al software libre en el proceso de enseñanza a nivel de educación media.

Indicación: Marque la respuesta que considere conveniente.

1-¿Qué programas de ofimática enseña a sus alumnos? (Selección múltiple)

Procesador de Texto (Nombre) _____

Hojas de cálculo (Nombre) _____

Presentaciones (Nombre) _____

Otros (Tipo y Nombre) _____

2- ¿Enseña Programación a los estudiantes?

Sí

No

2.1- Si la respuesta anterior fue sí, ¿Que lenguajes de programación imparte a sus alumnos?

C++ JAVA

Python PHP

Otro (Especifique): _____

3- Si respondió que sí a la pregunta 2; ¿Que entorno de programación integrado (IDE) o que editores utiliza como herramienta para programar en clases?

NetBeans Sublime Text

Eclipse Pycharm

Atom Geany _____

Otro especifique _____

Figura 47: Encuesta a docentes

4- ¿Enseña sobre bases de datos a los Estudiantes?
 Sí No

4A- Si la respuesta anterior fue sí, ¿Qué sistemas de Gestores de Bases de Datos imparte a sus alumnos?

MariaDB

SQLite

PostgreSQL

ORACLE

Otro (Especifique): _____

5- ¿Tiene conocimiento sobre la normativa del MINED a cerca de la implementación de software libre?
 Sí No

6- ¿Ha recibido capacitaciones de parte del MINED sobre el uso de software libre?
 Sí No

Especifique el tipo de capacitación:

7- ¿Considera usted que tendría algún inconveniente para desarrollar sus clases haciendo uso del Software Libre?
 Sí No

Si tuvo algunos inconvenientes, detalle algunos que considere importantes:

Mala interpretación de la documentación oficial

Incompatibilidad con el hardware

Dificultades técnicas

Otros (Especifique) _____

Figura 48: Encuesta a docentes

9.1.3. Encuestas dirigidas a estudiantes



UNIVERSIDAD DE EL SALVADOR
FACULTAD MULTIDISCIPLINARIA ORIENTAL
DEPARTAMENTO DE INGENIERIA Y ARQUITECTURA

Encuesta dirigida a estudiantes de educación media del municipio de San Miguel, del departamento de San Miguel.

Objetivo: Conocer el impacto social y académico que ha generado el uso de software libre en los alumnos de educación media del municipio de San Miguel, departamento de San Miguel.

Indicación: Marca la respuesta que, a conciencia, consideras conveniente.

1- ¿Qué grado cursas actualmente?

1° Bachillerato 2° Bachillerato 3° Bachillerato

2- ¿Cuántos años has recibido clases de informática?

Un año Dos Años Tres años Otro

(Especifique): _____

3- ¿Haces uso de herramientas informáticas para las clases en la institución?

Sí No

4- ¿Tienes conocimiento o has escuchado hablar sobre el software libre?

Sí No

Especifica: _____

5- Si has escuchado hablar del software libre, ¿Cuáles de los siguientes valores crees que lo sustentan?

Solidaridad

Respeto

Honestidad

Libertad

Responsabilidad

Figura 49: Encuesta a estudiantes

6- ¿Conoces el sistema operativo GNU-Linux?

Sí

No

No Sabe

Si conoces el sistema operativo GNU/Linux. Marca algunas de las distribuciones que conoces:

Debian

Linux Mint

Ubuntu

Linux Max

Otro:

7- ¿Conoces otras herramientas informáticas basadas en Software libre?

Sí

No

Menciona algunas de ellas:

Figura 50: Encuesta a estudiantes

9.2. Manuales

9.2.1. Manual de instalación

En este manual se explicará cómo instalar cada una de las herramientas necesarias para el correcto funcionamiento de la plataforma web. El usuario accederá a la aplicación gracias a la conexión a internet o de red local, necesitarán únicamente tener instalado o ser compatible con algún navegador de los siguientes: Chrome 44.0 o superior Mozilla Firefox 38.0 o superior.

Instalacion de servidor web. Se procede a instalar los servicios apache, php, Mysql administrar la base de datos, para ello se instala programas donde vienen los distintos servicios en un solo paquete preconfigurado como Xampp.

1. Ingresa a la Web de descarga de xampp:

<https://www.apachefriends.org/es/index.html>



Figura 51: Pagina de descarga de la xampp

2. Haz clic en XAMPP para Linux. Está en el medio de la página. Esto hará que el archivo de instalación de XAMPP se empiece a descargar a tu computadora.



Figura 52: Descarga para linux

3. Luego hacer clic en Guardar archivo o seleccionar la carpeta Descargas como la ubicación para guardar antes de continuar.

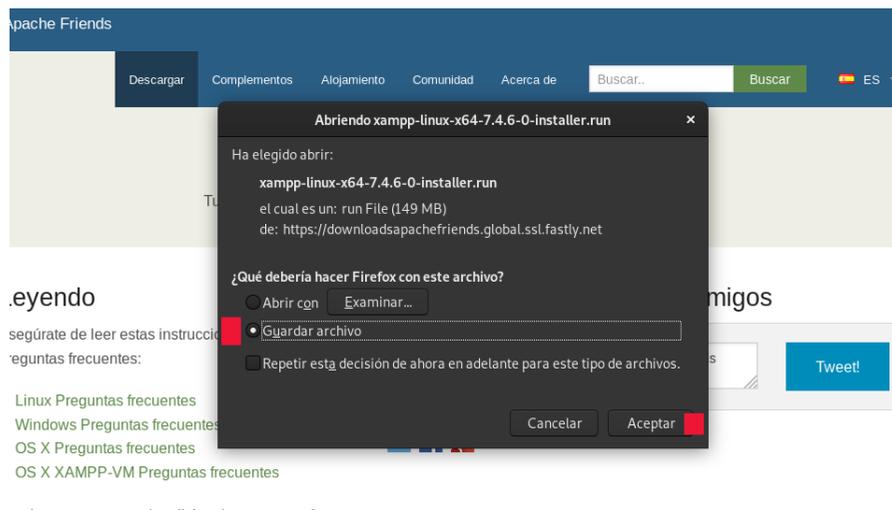


Figura 53: Guardando la descarga

- Deja que la descarga se complete. Una vez que el archivo de instalación de XAMPP se termine de descargar a tu computadora, puedes continuar.

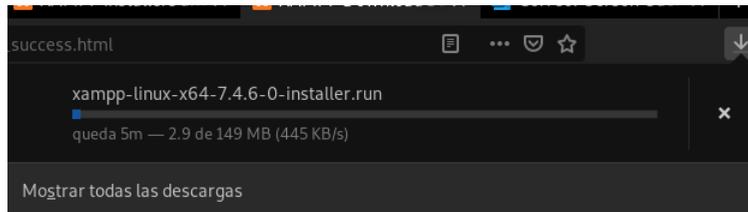


Figura 54: Descarga de la aplicación

- Abre la Terminal. Haz clic en el icono de la aplicación Terminal, Cámbiate a súper usuario escribiendo **su** y luego la contraseña , una vez en súper usuario entre al directorio Descargas. Escribe **cd Descargas** y presiona Entrar.



Figura 55: Ventana de terminal

- Posteriormente ejecute los siguientes comandos, uno a la vez: **chmod +x xampp-linux-64-7.0.6-0-installer.run** . Este para que el archivo descargado sea ejecutable. **./ xampp-linux-64-7.0.6-0-installer.run** Comando de instalación.

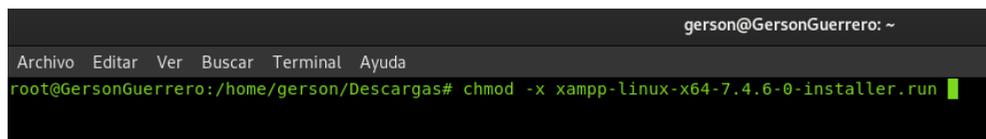


Figura 56: Ventana de terminal

7. Sigue las instrucciones de instalación. Una vez que aparezca la ventana de instalación, haz lo siguiente: Haz clic en **Siguiente (Next)** tres veces. Desmarca la casilla **Aprender más acerca de Bitnami para XAMPP (Learn more about Bitnami for XAMPP)**. Haz clic en **Siguiente** y luego otra vez en **Siguiente** para empezar a instalar XAMPP.



Figura 57: Ventana de instalación

8. Haz clic en **Finalizar (Finish)**. Esta opción está en la parte inferior de la ventana. Al hacerlo, se cerrará la ventana de instalación. En este punto, estás listo para ejecutar XAMPP.



Figura 58: Finalizar instalación

9. Haz clic en la pestaña Administrar servidores (**Manage Servers**). Esta opción está en la parte superior de la ventana. Haz clic en Iniciar todos (**Start All**). Está en la parte inferior de la ventana. Al hacerlo, los componentes activos de XAMPP se empezarán a ejecutar.

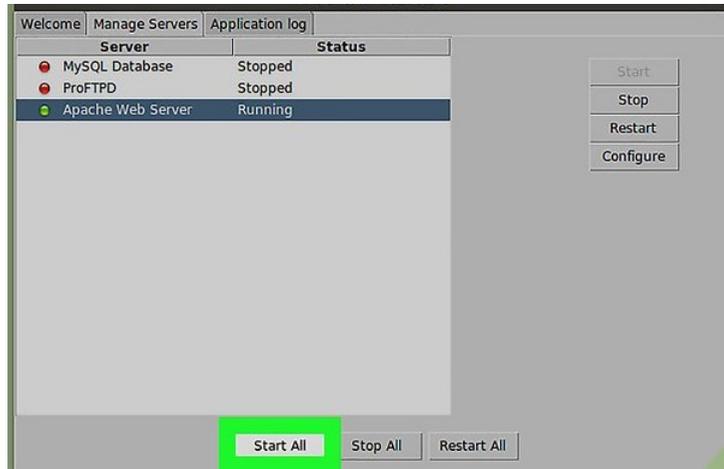


Figura 59: Ejecutando xampp

10. Comprobamos en el sitio web **localhost**



Welcome to XAMPP for Linux 7.2.26

You have successfully installed XAMPP on this system! Now you can start using Apache, MariaDB, PHP and other components. You can find more info in the [FAQs](#) section or check the [HOW-TO Guides](#) for getting started with PHP applications.

XAMPP is meant only for development purposes. It has certain configuration settings that make it easy to develop locally but that are insecure if you want to have your installation accessible to others. If you want have your XAMPP accessible from the internet, make sure you understand the implications and you checked the [FAQs](#) to learn how to protect your site. Alternatively you can use [WAMP](#), [MAMP](#) or [LAMP](#) which are similar packages which are more suitable for production.

Start the XAMPP Control Panel to check the server status.

Community:

Figura 60: localhost

Instalación y configuración de CodeIgniter

1. Descarga y descomprime el paquete de CodeIgniter Lo primero que debemos hacer es ingresar al sitio oficial de CodeIgniter <https://codeigniter.com/> donde encontrara diferentes versiones de CodeIgniter.

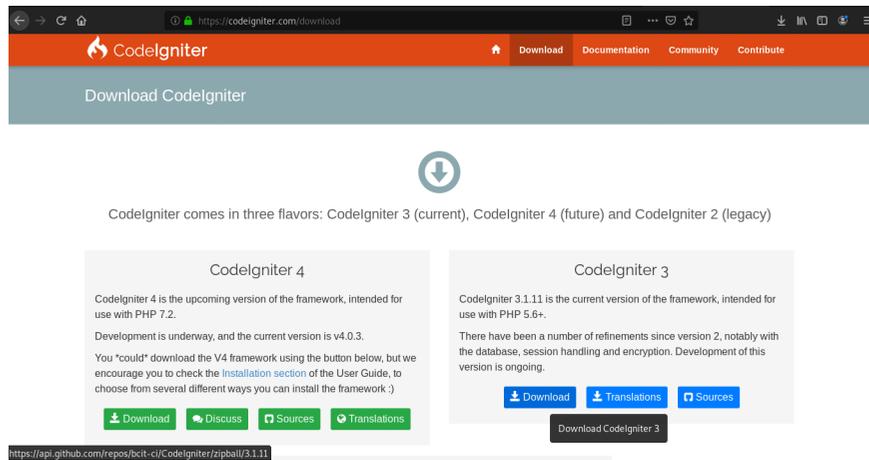


Figura 61: Sitio oficial

2. Una vez en el sitio oficial damos clic en el enlace que dice **Download** para descargar el archivo una vez descargado lo descomprimimos.

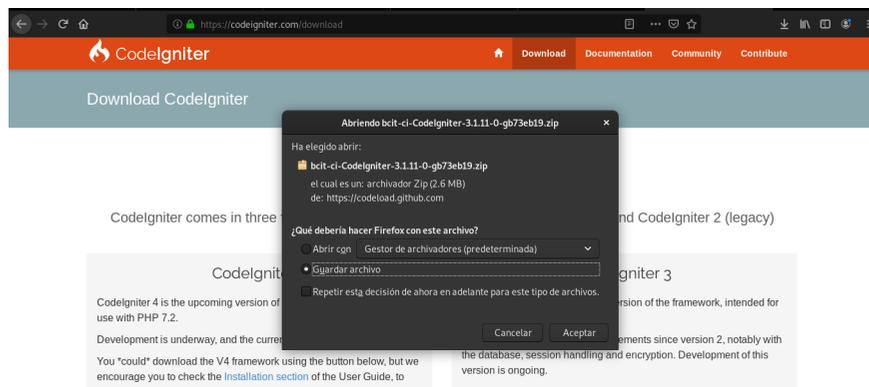


Figura 62: carpeta de descarga

3. Luego siga los siguientes pasos:

- **Descomprima el paquete**, el archivo que ha descargado, en este caso el archivo que le ha sido proporcionado por el desarrollador del Sistema.
- **Suba CodeIgniter al servidor.** Ahora debe subir todos los archivos descomprimidos al servidor web.

- **Opción A:**

Si estás programando en su ordenador local, para pruebas y desarrollo, deberás tener un servidor instalado, Xampp. En este caso tendrá que copiar simplemente los archivos de codeIgniter en el directorio del servidor, ubicado dentro de la carpeta del Equipo **/opt/lampp/htdocs**. Es dentro de la carpeta o directorio htdocs donde tendrá que pegar y guardar el archivo que ha sido proporcionado por el desarrollador del Sistema.

- **Opción B:**

Subir CodeIgniter a un servidor web de Internet, en un espacio que tengas contratado de hosting, deberás subir por FTP (utilizando la herramienta FileZilla) todos los archivos, Copiando CodeIgniter (Framework que ha sido proporcionado por el desarrollador del Sistema) en la raíz del dominio o subdominio, para que todas las páginas del dominio se sirvan a través del framework PHP.

- **Configura la URL base de tu aplicación web** Abra el archivo `application/config/config.php` con un editor de texto (ATOM) y establezca su URL base, la url para acceder a la raíz de CodeIgniter donde ha subido el código del framework.

Instalación de ATOM

La instalación de Atom en cualquier versión de Debian es relativamente fácil. Ingrese a la página web oficial <https://atom.io/> en donde encontrara el programa en formato .deb o en formato .rpm.

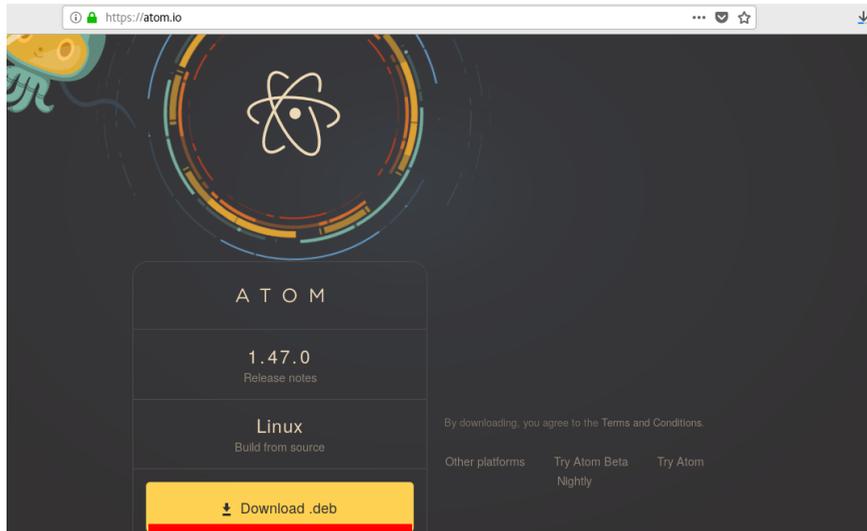


Figura 63: sitio oficial atom

4. Guardamos la descarga del paquete de Atom

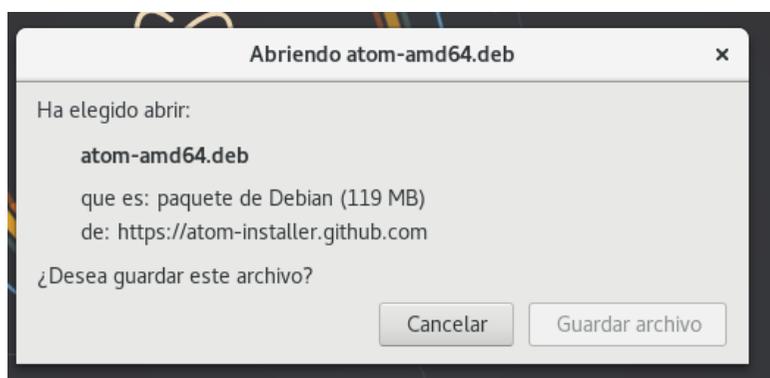
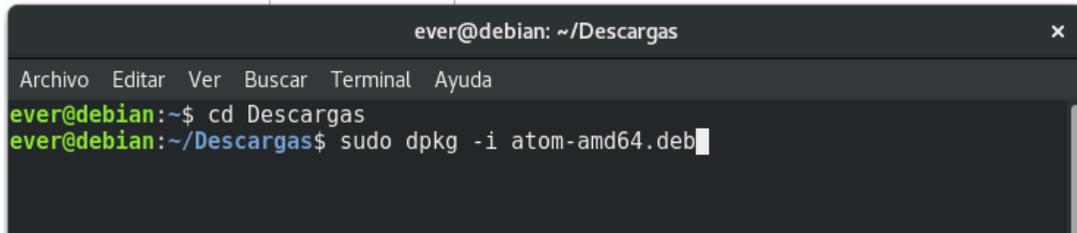


Figura 64: Descarga atom

5. Una vez que hemos descargado el paquete, abrimos una terminal, y posteriormente nos ubicamos en la carpeta donde está el paquete (ubicación del paquete descargado).

Posteriormente ejecute el siguiente comando: **sudo dpkg -i atom-amd64.deb**

Nota: atom-amd64.deb es el nombre del paquete.

A terminal window titled 'ever@debian: ~/Descargas' with a close button in the top right corner. The window contains a menu bar with 'Archivo', 'Editar', 'Ver', 'Buscar', 'Terminal', and 'Ayuda'. The terminal shows the following commands and output:

```
ever@debian:~$ cd Descargas
ever@debian:~/Descargas$ sudo dpkg -i atom-amd64.deb
```

Figura 65: Terminal

Esto instalará el editor de código Atom en nuestro sistema operativo. Pero no lo encontraremos en castellano sino en inglés. Para ello solo hemos de ir a **Packages** → Settings View y seleccionar la opción **Install Packages/themes**.

9.2.2. Manual De Administrador

En este documento se explican los contenidos y acciones más relevantes a realizar siendo un usuario con rol de Administrador en la plataforma. Este manual ha sido creado para facilitar al Administrador la gestión/administración de la plataforma web, se presentan las diferentes funciones que el administrador puede ejecutar dentro de la plataforma.

- Primero abrimos el navegador y pondremos la siguiente dirección:

<http://localhost/Repositorio2.0> se nos mostrara el siguiente menú:



Figura 66: Bienvenido

- En este menú aparece la página de inicio, donde se pueden ver los registros, también hay información sobre la plataforma, hay una opción de registrarse y la opción de iniciar sesión en el caso del administrador iniciara sesión como administrador.

- Primero iniciamos como usuario de administrador llenando los campos de **Correo** >**Contraseña** luego en **Iniciar sesión**

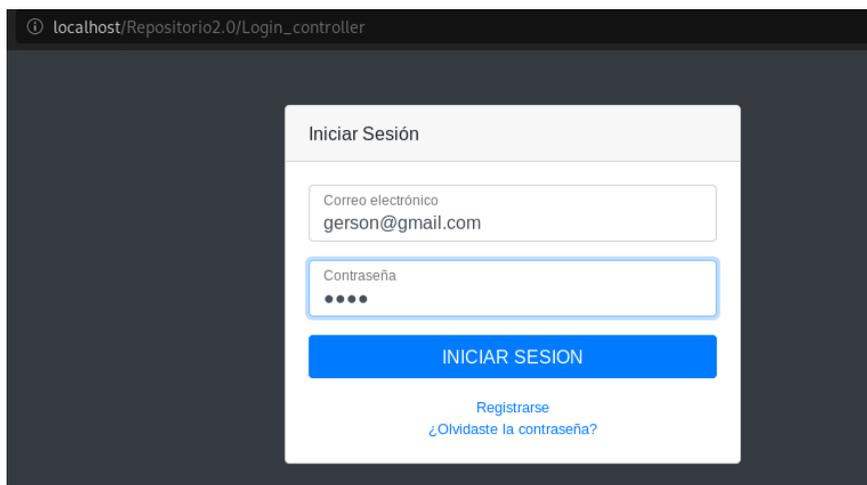


Figura 67: Login

- Una vez iniciemos sesión se nos aparecerá el siguiente menú donde aparecen las opciones de **Registros**, **Usuarios**, **Instituciones Registradas** y **software registrado** podemos ver los detalles de cada una de estas opciones.

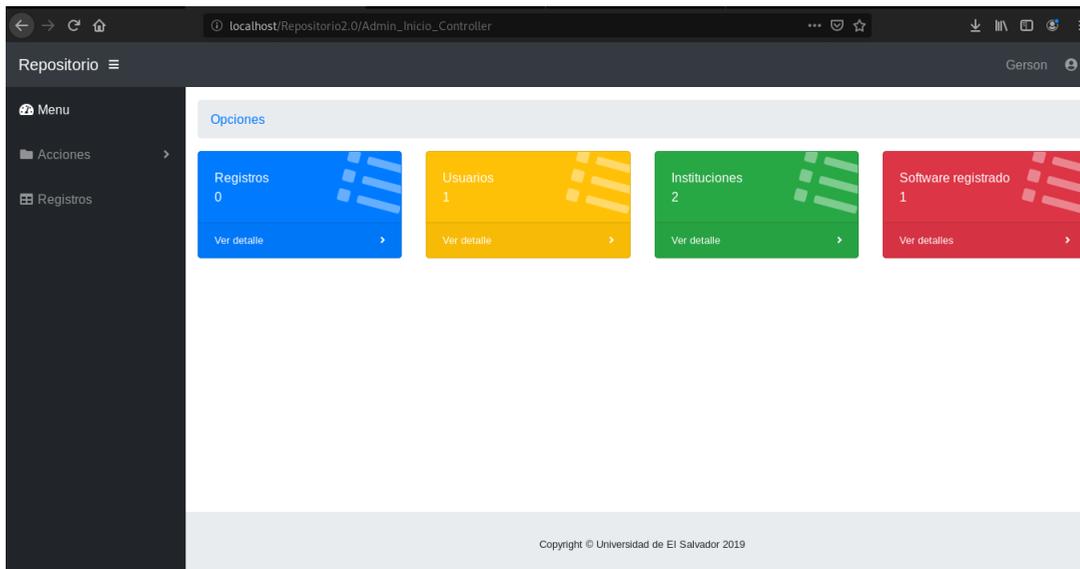


Figura 68: Menu

- Podemos ver la opción de **Registros** donde aparecerá el Nombre del encargado IT , Nombre de la Institución, Fecha de Registro, Estado y las acciones donde los da la opciones de editar y eliminar los Registros.



Figura 69: Registro

- Podemos ver la opción de **Software registrado** donde aparecerá el nombre del software, Descripción, Sitio Oficial, Tipo de Software, y las acciones.

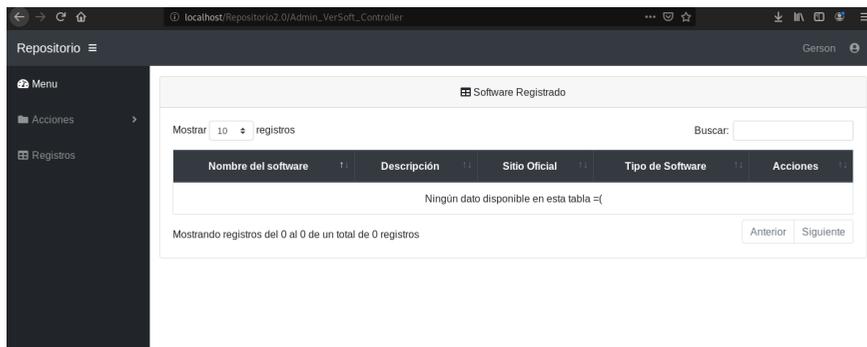


Figura 70: Software registrado

- Podemos ver la opción de **usuarios** donde aparecerá el nombre de los usuarios registrados donde aparece, el ID, Nombre, Apellido, Correo teléfono y las acciones donde los da la opciones de editar y eliminar los usuarios.

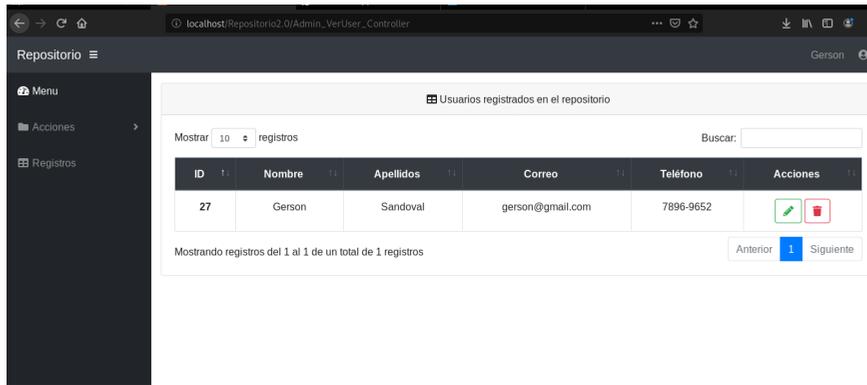


Figura 71: Usuarios registrados

- Podemos ver la opción de **Instituciones Registradas** donde aparecerá el nombre de las instituciones registradas, Dirección, Teléfono, Encargado o Representante Sitio Web y las acciones donde los da la opción de editar y eliminar instituciones.

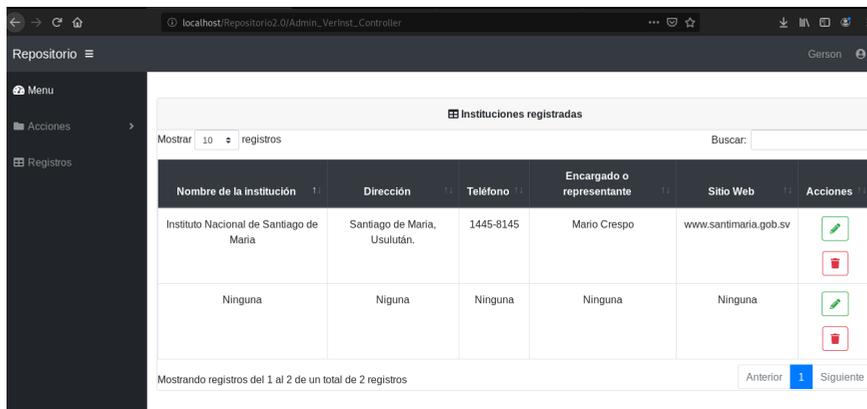


Figura 72: Instituciones Registradas

- Luego Desplegar la opciones de **Acciones** donde aparecerá los menús de **Agregar Información, Registrar Información, Registrar Institución.**

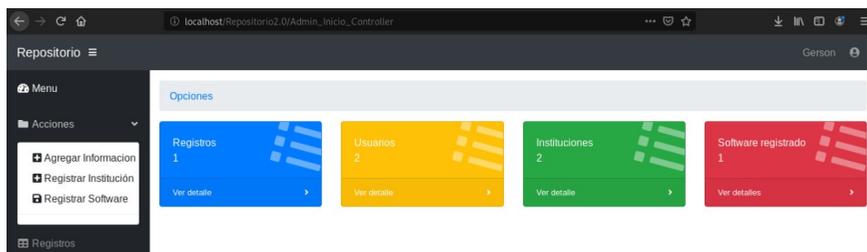


Figura 73: Acciones

- En la acción de **Agregar información** nos manda al formulario donde aparece los campos de Usuario, Fecha de Registro, Detalle del Hardware que utiliza en la institución, las observaciones, institución, inconvenientes que se presentaron en el uso de software libre, beneficios percibidos, agregar software y llenar los campos del detalle del software luego selecciona guardar registro.

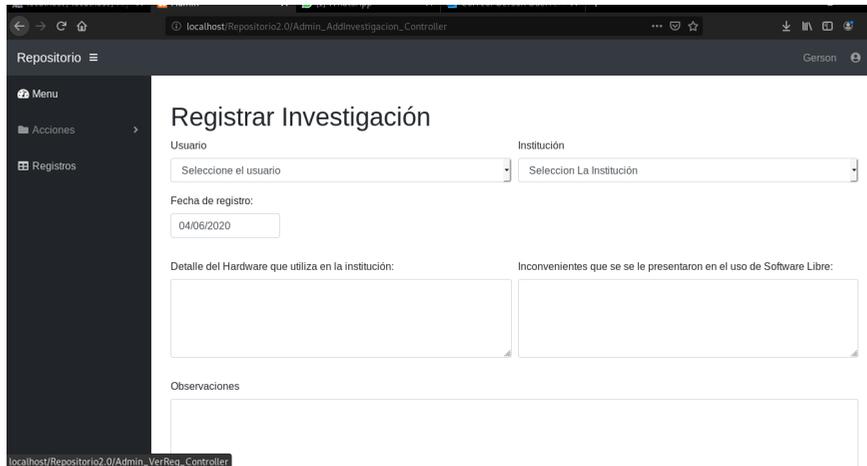


Figura 74: Agregar Información

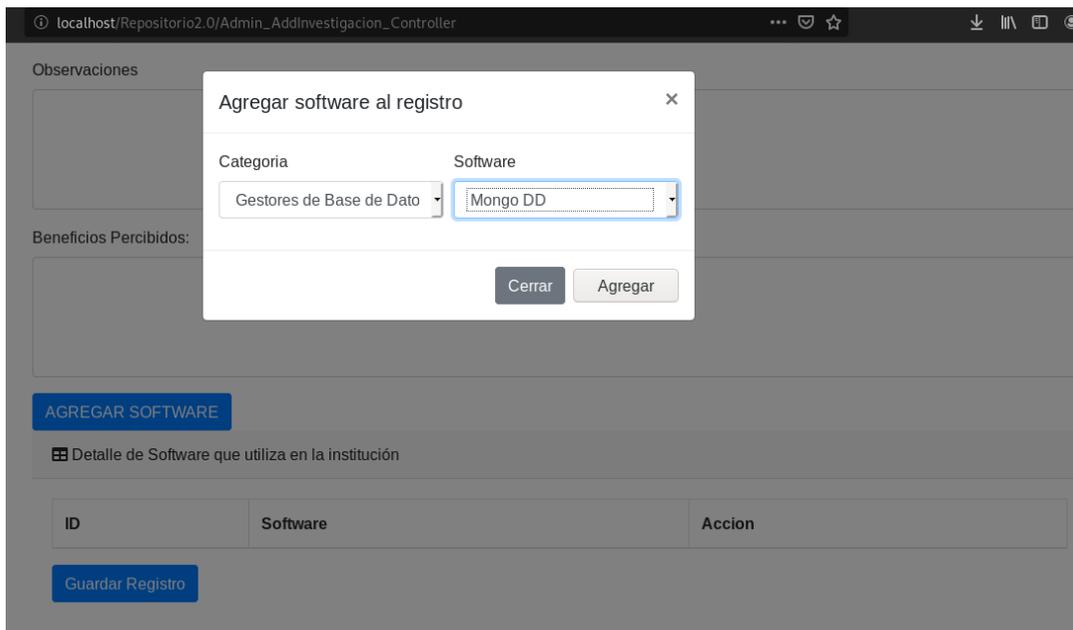
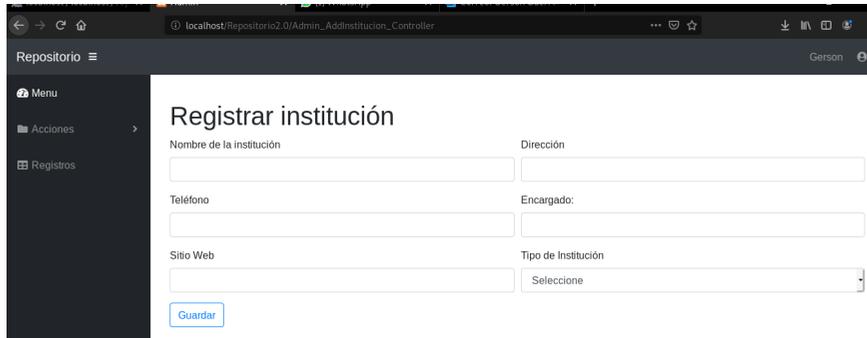


Figura 75: Agregar Información

- En la acción de **Agregar Institución** nos manda al formulario de registrar institución donde aparecen los campos de Nombre de la Institución, Teléfono, Sitio Web, Dirección, Encargado, Tipo de Institución luego de llenar los campos seleccione en **Guardar**.



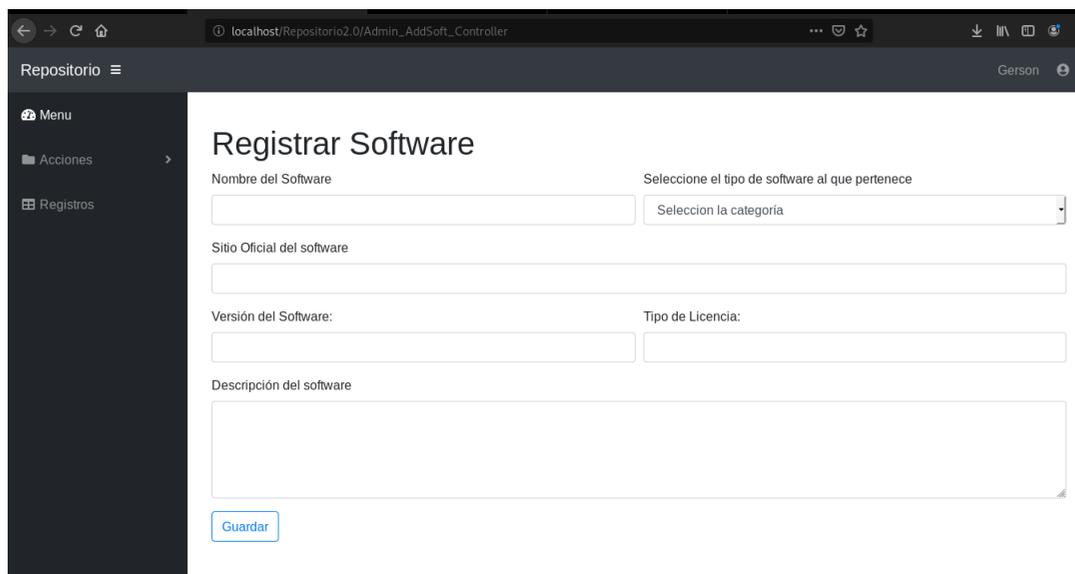
The screenshot shows a web browser window with the URL `localhost/Repositorio2.0/Admin_AddInstitucion_Controller`. The page title is 'Registrar institución'. On the left, there is a dark sidebar menu with 'Menu', 'Acciones', and 'Registros'. The main content area contains the following form fields:

- Nombre de la institución:
- Teléfono:
- Sitio Web:
- Dirección:
- Encargado:
- Tipo de Institución:

A blue 'Guardar' button is positioned at the bottom left of the form.

Figura 76: Agregar Institución

- En la acción de **Agregar Software** nos manda al formulario Registrar Software donde están los campos de Nombre del Software, Sitio Oficial del Software, Versión del Software, Descripción del Software, Seleccione el tipo de software a que pertenece y el Tipo de Licencia luego de llenar los campos seleccione en **Guardar**.



The screenshot shows a web browser window with the URL `localhost/Repositorio2.0/Admin_AddSoft_Controller`. The page title is 'Registrar Software'. On the left, there is a dark sidebar menu with 'Menu', 'Acciones', and 'Registros'. The main content area contains the following form fields:

- Nombre del Software:
- Sitio Oficial del software:
- Versión del Software:
- Descripción del software:
- Tipo de Licencia:
- Seleccione el tipo de software al que pertenece:

A blue 'Guardar' button is positioned at the bottom left of the form.

Figura 77: Agregar Software

- Ingresamos al ícono de perfil donde aparece la opción de actualizar mi perfil y cerrar sesión, en la opción de mi perfil aparece el formulario de mi perfil donde está el Nombre, Apellido, Correo, Contraseña, Teléfono y la opción de actualizar información.

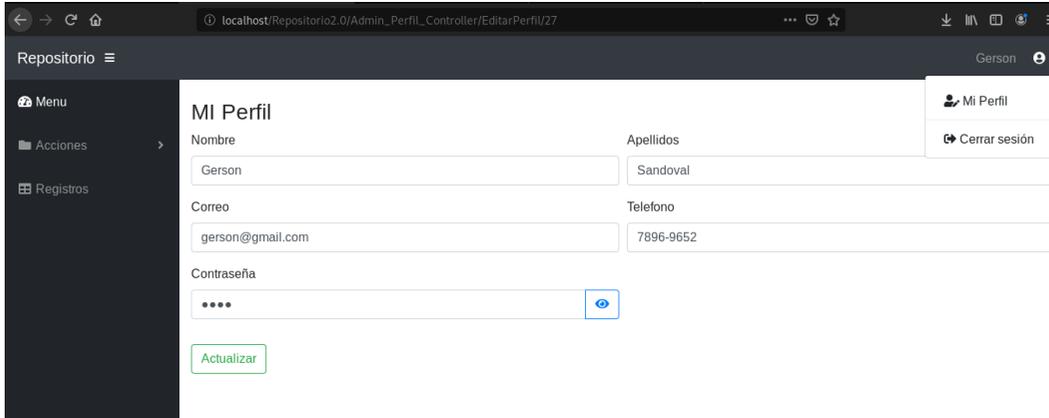


Figura 78: Perfil

- También está el icono de cerrar sesión donde se abrirá la ventana de cerrar sesión y pregunta si realmente desea cerrar sesión y aparece la opción de cancelar y continuar.

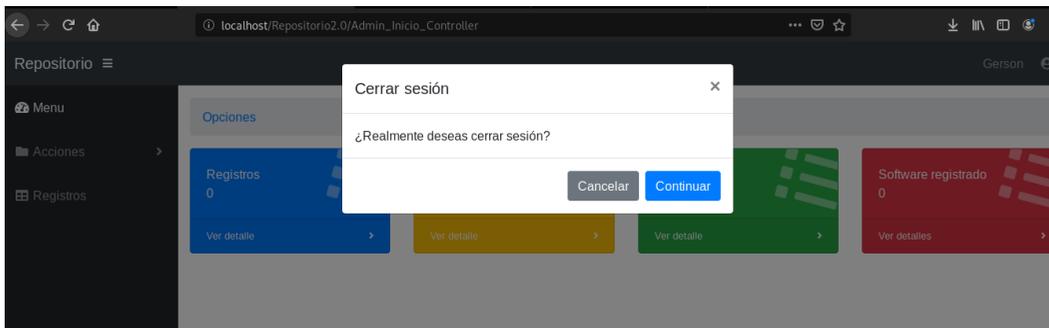


Figura 79: Cerrar Sesión

9.2.3. Manual De Usuario

Se presenta el manual técnico de plataforma web que está diseñada para realizar las operaciones de Registro de usuarios donde cada usuario podrá agregar nuevas investigaciones a la plataforma.

- Para acceder a la plataforma web puede ingresar desde la página web en la siguiente dirección `http://localhost/Repositorio2.0` se nos mostrara el siguiente menú:



Figura 80: Bienvenido

- En este menú aparece la página de inicio, donde se pueden ver los registros, también hay información sobre la plataforma, hay una opción de registrarse y la opción de iniciar sesión en el caso del usuario.

- Si es usuario puede llenando los campos de **Correo** > **Contraseña** luego en **Iniciar sesión** para poder ingresar al menú de usuario, donde podrá agregar información y también podrá ver la información que él ha ingresado.

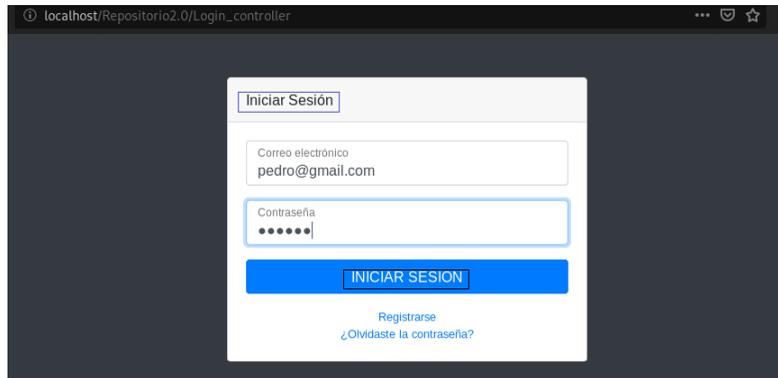


Figura 81: Iniciar Sesión

- De no estar registrado como usuario podrá ingresar a la opción de **Registrarse** y llenar los campos que requiere el formulario.

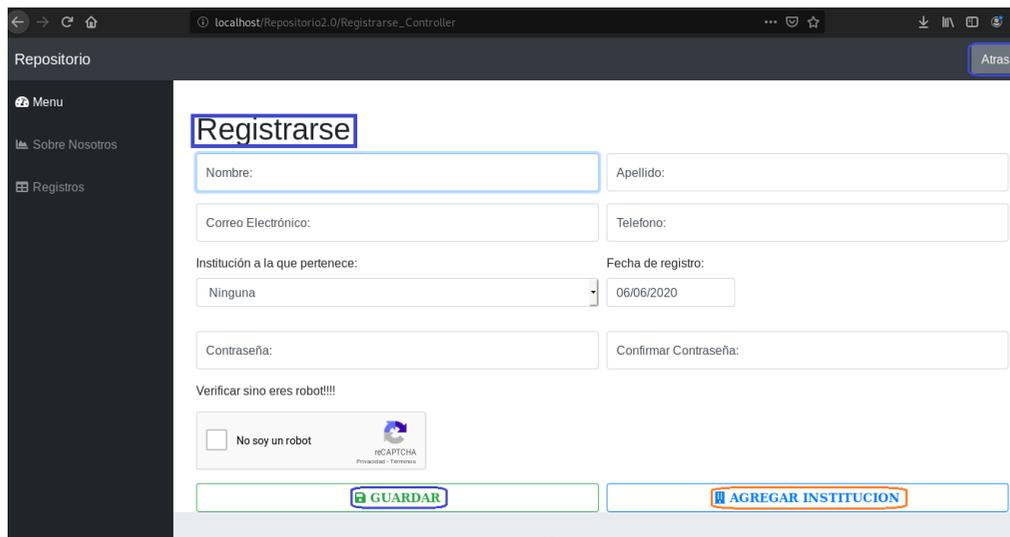


Figura 82: Registrarse

- Una vez el usuario ingresa al sitio web, lo primero que observa es la página principal del sitio. Esta es descrita a continuación. En la página principal podemos ver las opciones de **Mis Registros** , **Mi Institución** , **Software Registrado** también en la parte superior izquierda vemos un menú de **Acciones y Registros**.El usuario puede agregar , Modificar y Eliminar su información.

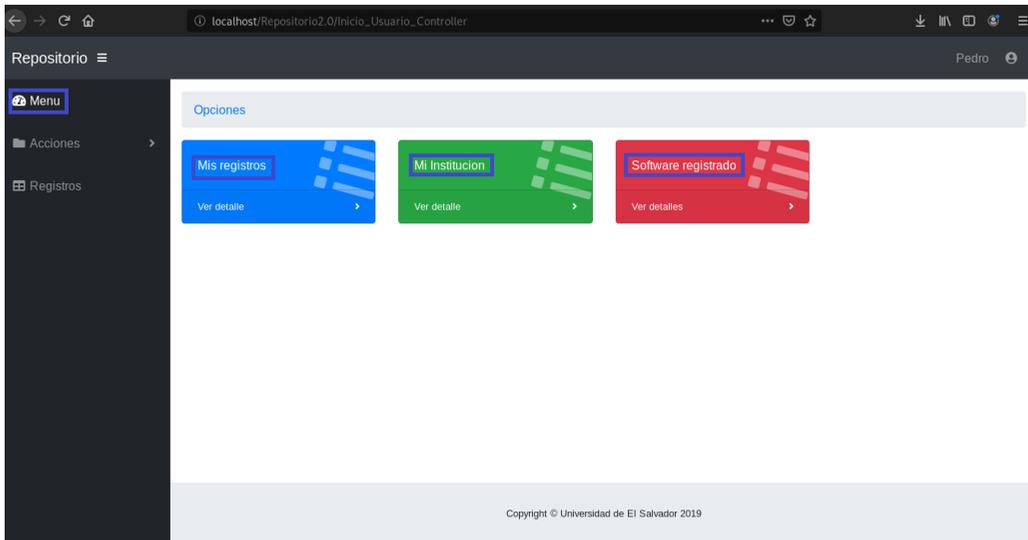


Figura 83: Inicio Usuario

- Si ingresamos en **Mis Registros** podemos ver los registros que el usuario ha creado. También aparece el icono de la opción de ver en detalle del registro agregado.

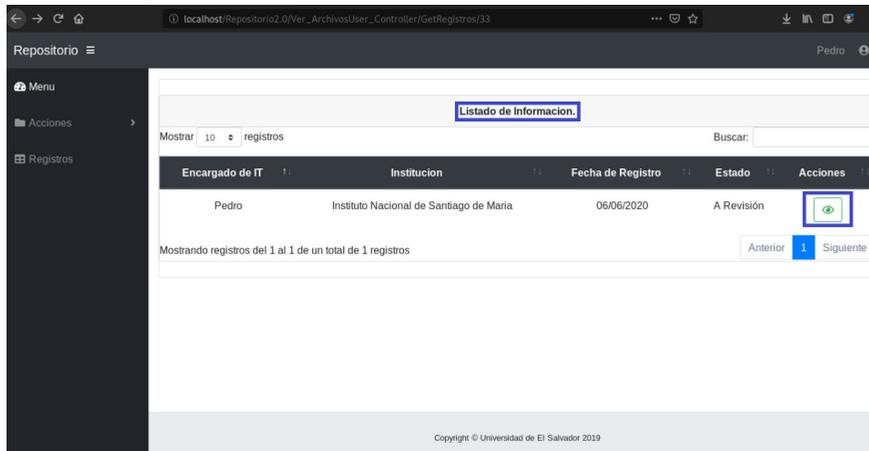


Figura 84: Mis Registros

- Si ingresamos en **Mi institución** podemos ver las instituciones que el usuario a ingresado. También aparece el icono de la opción de editar la institución agregada.

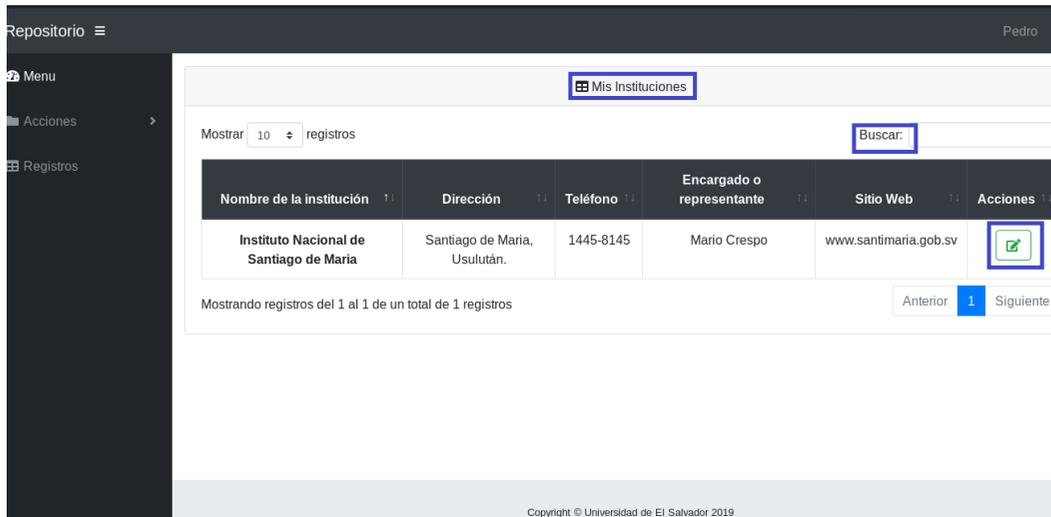


Figura 85: Instituciones

- Si ingresamos en **Software Registrado** podemos ver los software que el usuario ha ingresado.

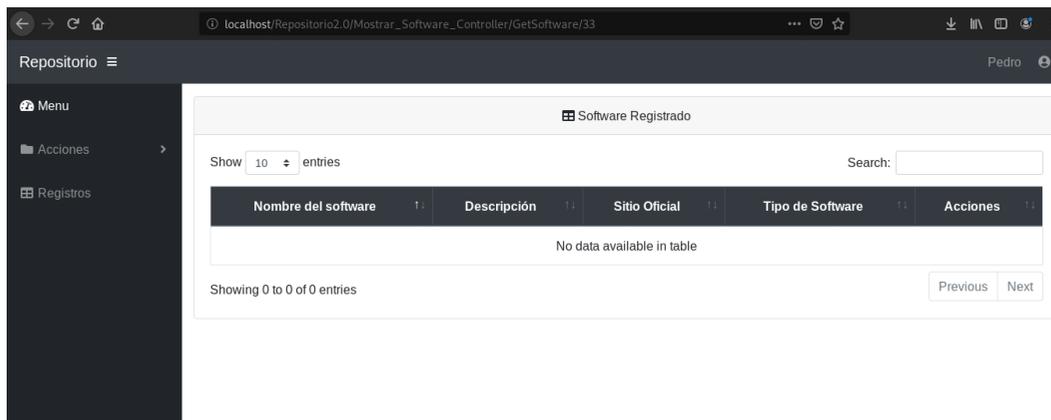


Figura 86: Software registrados

- En la esquina superior izquierda tenemos el apartado de **Acciones**, donde el usuario puede Agregar Información y Registrar Software

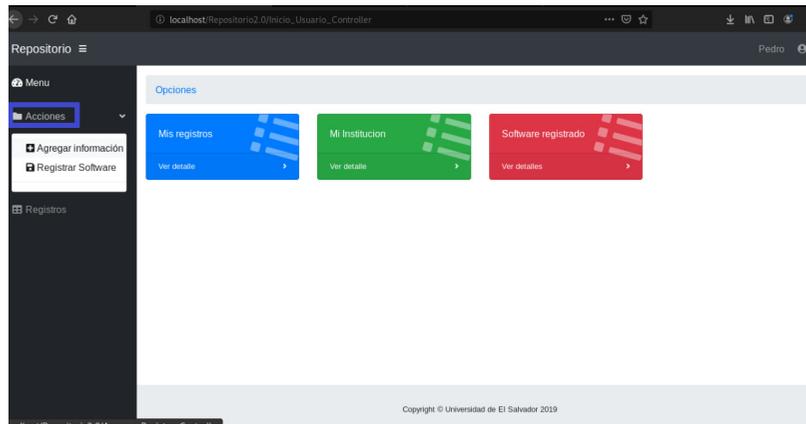


Figura 87: Acciones

- En la opción de **Agregar Información** se abre el formulario de Registrar Información donde automáticamente aparecerá el nombre del usuario registrado y la institución que ha registrado, aparece la fecha de registro, detalle del software que utiliza la institución, los inconvenientes que se presentaron en el uso del software libre, las observaciones, los beneficios también aparece el botón de agregar software. Después de completar todos los campos del formulario el usuario podrá guardar el registro.



Figura 88: Registra Información

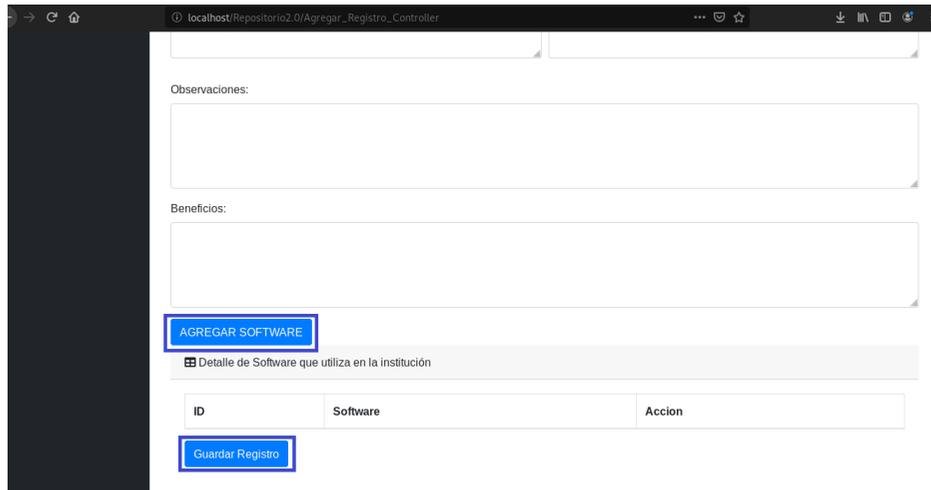


Figura 89: Registrar Información

- En la opción de **Registrar Software** el Usuario puede Registrar el software llenando los siguientes campos, Nombre del software, Sitio Oficial del Software, Versión y Descripción del software, Seleccione al tipo de software que pertenece según Categoría, Tipo de Licencia, En el campo de registrante aparece automáticamente el nombre del usuario.

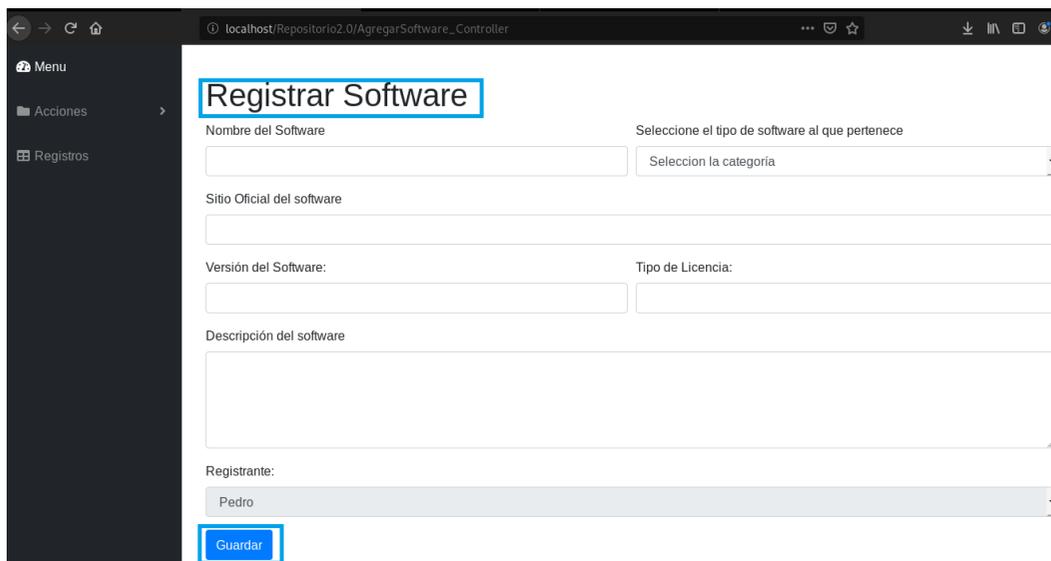


Figura 90: Registrar Software

- También está el formulario de cerrar sesión donde nos pregunta si realmente deseamos cerrar sesión

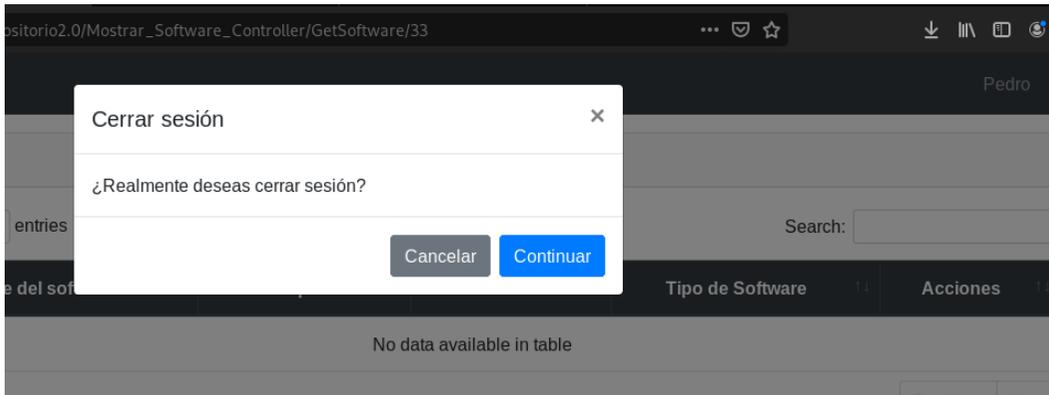


Figura 91: Cerrar Sesión

9.3. Glosario

- a) **Sistema Operativo:** Es el software principal o conjunto de programas de un sistema informático que gestiona los recursos de hardware y provee servicios a los programas de aplicación de software, ejecutándose en modo privilegiado respecto de los restantes.
- b) **Software:** Se conoce como software al soporte lógico de un sistema informático, que comprende el conjunto de los componentes lógicos necesarios que hacen posible la realización de tareas específicas.
- c) **Hardware:** Se refiere a las partes físicas, tangible de un sistema informático; sus componentes eléctricos, electrónicos, electromecánicos y mecánicos.
- d) **Imagen ISO:** Es un archivo informático dónde se almacena una copia o imagen exacta de un sistema de archivos.
- e) **Ofimática:** Designa al conjunto de técnicas, aplicaciones y herramientas informáticas que se utilizan en funciones de oficina para optimizar, automatizar, mejorar tareas y procedimientos relacionados.
- f) **Navegador Web:** Es un software, aplicación o programa que permite el acceso a la Web interpretando la información de distintos tipos de archivos y sitios web para que éstos puedan ser visualizados.
- g) **Servidor (hardware) :** Un servidor basado en hardware es una máquina física integrada en una red informática en la que, además del sistema operativo, funcionan uno o varios servidores basados en software.
- h) **Servidor (Software) :** Un servidor basado en software es un programa que ofrece un servicio especial que otros programas denominados clientes (clients) pueden usar a nivel local o a través de una red.
- i) **Código fuente:** En el contexto de la informática, el código fuente se define como el conjunto de líneas de textos, que son las directrices que debe seguir la

computadora para realizar dicho programa; por lo que es en el código fuente, donde se encuentra escrito el funcionamiento de la computadora.

- j)* **Kernel:** Es un software que constituye una parte fundamental del sistema operativo, y se define como la parte que se ejecuta en modo privilegiado. Es el principal responsable de facilitar a los distintos programas acceso seguro al hardware de la computadora, o en forma básica, es el encargado de gestionar recursos, a través de servicios de llamada al sistema.
- k)* **Blob binario:** Es un término peyorativo para un objeto cargado en el núcleo de un sistema operativo de código abierto, sin tener su código fuente disponible.