



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA

**TRABAJO DE TITULACIÓN PREVIO LA OBTENCIÓN DEL TÍTULO
DE INGENIERO EN SISTEMAS E INFORMÁTICA**

**TEMA: DESARROLLO E IMPLANTACIÓN DE UN ECOSISTEMA
INFORMÁTICO PARA LA ENSEÑANZA Y NORMALIZACIÓN DEL
PENSAMIENTO COMPUTACIONAL PARA LA UNIVERSIDAD DE
LAS FUERZAS ARMADAS ESPE UTILIZANDO LA HERRAMIENTA
APPGINI**

AUTOR: CORAL CASTILLO, GIOVANNY FRANCISCO

DIRECTOR: PHD. DELGADO RODRÍGUEZ, RAMIRO NANAC

SANGOLQUÍ, ENERO 2018

CERTIFICADO



**DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA**

CERTIFICACIÓN

Certifico que el trabajo de titulación, “*DESARROLLO E IMPLANTACIÓN DE UN ECOSISTEMA INFORMÁTICO PARA LA ENSEÑANZA Y NORMALIZACIÓN DEL PENSAMIENTO COMPUTACIONAL PARA LA UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE UTILIZANDO LA HERRAMIENTA APPGINI*” realizado por el señor **GIOVANNY FRANCISCO CORAL CASTILLO**, ha sido revisado en su totalidad y analizado por el software anti-plagio, el mismo cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, por lo tanto me permito acreditarlo y autorizar al señor **GIOVANNY FRANCISCO CORAL CASTILLO** para que lo sustente públicamente.

Sangolquí, 15 de enero del 2018

Firma

Dr. Ramiro Delgado PhD

c.c. 170701917-S

AUTORÍA DE RESPONSABILIDAD



DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA

AUTORÍA DE RESPONSABILIDAD

Yo, **GIOVANNY FRANCISCO CORAL CASTILLO**, con cédula de identidad N°172010920-4, declaro que este trabajo de titulación “**DESARROLLO E IMPLANTACIÓN DE UN ECOSISTEMA INFORMÁTICO PARA LA ENSEÑANZA Y NORMALIZACIÓN DEL PENSAMIENTO COMPUTACIONAL PARA LA UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE UTILIZANDO LA HERRAMIENTA APPGINI**” ha sido desarrollado considerando los métodos de investigación existentes, así como también se ha respetado los derechos intelectuales de terceros considerándose en las citas bibliográficas.

Consecuentemente declaro que este trabajo es de mi autoría, en virtud de ello me declaro responsable del contenido, veracidad y alcance de la investigación mencionada.

Sangolquí, 15 de enero del 2018

GIOVANNY FRANCISCO CORAL CASTILLO

C.C.1720109204

AUTORIZACIÓN



**DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA**

AUTORIZACIÓN

Yo, **GIOVANNY FRANCISCO CORAL CASTILLO**, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar en la biblioteca Virtual de la institución el presente trabajo de titulación **“DESARROLLO E IMPLANTACIÓN DE UN ECOSISTEMA INFORMÁTICO PARA LA ENSEÑANZA Y NORMALIZACIÓN DEL PENSAMIENTO COMPUTACIONAL PARA LA UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE UTILIZANDO LA HERRAMIENTA APPGINI”** cuyo contenido, ideas y criterios son de mi autoría y responsabilidad.

Sangolquí, 15 de enero del 2018

GIOVANNY FRANCISCO CORAL CASTILLO

C.C.1720109204

DEDICATORIA

Dedico este trabajo especialmente a mi familia: Silvia, Giovanni y Daniela quienes siempre me han apoyado en todo mi desarrollo profesional, estando siempre en las buenas y las malas. Gracias por ese apoyo incondicional y amor a pesar de todo.

A mis abuelos: José, Elsa, Campo, Mercedes y a mi tía Ruth quienes con el transcurrir de la vida me enseñaron a vivirla y apreciarla cada día más.

Giovanny Francisco Coral

AGRADECIMIENTO

Agradezco primeramente a mi Dios que me ha dado fuerzas a pesar de las altas y bajas, gracias por permitirme cumplir una de mis grandes metas en la vida.

Agradezco a mis padres y a mi hermana que, sin su apoyo y amor incondicional, no podría haber cumplido esta meta en mi vida.

A mi novia Estefanía que siempre me ha apoyado en las buenas y en las malas sin importar condición ni circunstancia.

A los ingenieros que participaron en todo el proceso de aprendizaje en la carrera, los mismos que aportaron con sus experiencias y conocimientos para forjarme como profesional.

Al ingeniero Ramiro Delgado, más que un profesor un gran amigo que supo brindarme siempre su apoyo y conocimiento para lograr este objetivo.

A mis compañeros y amigos con quienes viví experiencias invaluable e inolvidables durante la carrera.

Gracias a todos.

Giovanny Francisco Coral

ÍNDICE DE CONTENIDOS

CERTIFICADO	ii
AUTORÍA DE RESPONSABILIDAD.....	iii
AUTORIZACIÓN.....	iv
DEDICATORIA.....	v
AGRADECIMIENTO	vi
ÍNDICE DE TABLAS.....	xi
ÍNDICE DE FIGURAS.....	xiii
GLOSARIO	xv
RESUMEN.....	xvi
ABSTRACT.....	xvii
CAPÍTULO I.....	1
INTRODUCCIÓN	1
1.1 ANTECEDENTES.....	1
1.2 PLANTEAMIENTO DEL PROBLEMA	2
1.3 JUSTIFICACIÓN.....	4
1.4 OBJETIVOS.....	5
1.4.1 Objetivo General.....	5
1.4.2 Objetivos Específicos.....	6
1.5 ALCANCE.....	6
1.6 FACTIBILIDAD.....	9
1.6.1 Factibilidad Técnica.....	9
1.6.2 Factibilidad Económica	10
1.6.3 Factibilidad Legal	13
1.6.4 Factibilidad Operativa.....	13
CAPÍTULO II	14
MARCO TEÓRICO.....	14
2.1 PENSAMIENTO COMPUTACIONAL.....	14
2.1.1 Orígenes del pensamiento computacional.	14
2.1.2 Definición del pensamiento computacional.....	17
2.1.3 Características principales del pensamiento computacional.....	18

2.1.4	Investigaciones y proyectos existentes	22
2.2	HERRAMIENTAS DEL MODELO.....	23
2.2.1	AppGini.....	23
2.2.2	MySQL.	30
2.2.1	Origen de MySQL.....	30
2.3	DEFINICIÓN BÁSICA DE MYSQL.....	31
2.4	COMPARATIVA DE MYSQL FRENTE A OTROS DBMS.....	31
2.5	VENTAJAS Y DESVENTAJAS DE MYSQL.....	32
2.5.1	PHP	33
2.5.2	Origen de PHP.	34
2.5.3	Definición de PHP.	35
2.5.4	Ventajas y puntos que considerar sobre PHP	36
2.5.5	Git.	38
2.5.6	Características Generales de Git.	39
2.5.7	BitBucket.	40
2.5.8	PHPStorm	40
2.6	METODOLOGÍA DE DESARROLLO SCRUM.....	41
2.6.1	Introducción.	41
2.6.2	Principios de SCRUM.....	42
2.6.3	Roles (Scrum Team).	42
2.6.4	Herramientas de Trabajo.....	43
2.6.5	Ciclo de Vida.	44
2.7	TRABAJOS RELACIONADOS	45
CAPÍTULO III.....		46
ANÁLISIS, ARQUITECTURA Y DISEÑO DEL PROTOTIPO		46
2.8	ESPECIFICACIÓN DE REQUERIMIENTOS IEEE-830	46
2.8.1	Introducción.	46
2.8.2	Propósito.	46
2.8.3	Alcance.	46
2.8.4	Definiciones, siglas y nomenclatura.	47
2.8.5	Personal involucrado.....	48
2.8.6	Referencias.....	48

2.8.7	Resumen.....	48
2.8.8	Descripción General.....	49
2.8.9	Perspectiva del producto.	50
2.8.10	Interfaces del sistema.	50
2.8.11	Interfaces del Usuario.	51
2.8.12	Interfaces de hardware.	52
2.8.13	Interfaces de software.	53
2.8.14	Interfaces de comunicación.....	53
2.8.15	Restricciones de memoria.	54
2.8.16	Operaciones.....	54
2.8.17	Funcionalidad del producto.....	55
2.8.18	Características del usuario.	55
2.8.19	Restricciones del sistema.	56
2.8.20	Requisitos Específicos.	56
2.8.21	Clasificación de requerimientos funcionales.	57
2.8.22	Requerimientos no funcionales.....	58
2.8.23	Propósito.	58
2.8.24	Alcance.	58
2.8.25	Referencias.....	58
2.8.26	Suposiciones y dependencias.....	59
2.8.27	Usabilidad.	59
2.8.28	Fiabilidad.	59
2.8.29	Acceso.....	59
2.8.30	Rendimiento.....	59
2.8.31	Capacidad.....	60
2.8.32	Compatibilidad.....	60
2.8.33	Mantenimiento.	60
2.8.34	Diseño de restricciones.	60
2.9	ESPECIFICACIÓN DE CASOS DE USO	60
2.9.1	Introducción.	60
2.9.2	Descripción general de los actores.....	61
2.9.3	Caso de uso: Ingreso al sistema.	61

2.9.4	Caso de uso II: Administración de usuarios.	63
2.9.5	Caso de uso III: Administrador de grupos.	65
2.9.6	Caso de uso IV: Cambio de contraseña.	66
2.9.7	Caso de Uso V: Administrar Estudiantes.....	67
2.9.8	Caso de Uso VI: Administrar Materias.....	68
2.9.9	Caso de Uso VII: Administrar Matrículas.	70
2.9.10	Caso de Uso VIII: Administrar Comentarios.....	72
2.9.11	Caso de Uso IIX: Administrar Pruebas.....	74
2.9.12	Caso de Uso IX: Administrar Talleres.....	77
CAPÍTULO IV		80
DESARROLLO DEL PROTOTIPO		80
3.1	DESARROLLO DE LA PLATAFORMA.....	80
3.2	PRUEBAS TÉCNICAS	92
3.2.1	Prueba de Performance / Desempeño.	92
3.2.2	Prueba de Stress.	93
3.3	DISCUSIÓN DE LA EVALUACIÓN.....	94
3.3.1	Tipo de investigación utilizada.	94
3.3.2	Enfoque metodológico de la investigación.	96
3.3.3	Técnicas específicas Aplicadas.....	97
3.3.4	Técnicas e Instrumentos de Recolección de Información.....	99
3.3.5	Población.....	101
3.3.6	Resultados del Test de Usabilidad Aplicado a Docentes y Estudiantes.	102
Interpretación de los Resultados		120
CAPÍTULO V.....		122
CONCLUSIONES Y RECOMENDACIONES.....		122
4.1	Conclusiones	122
4.2	Recomendaciones.....	123
REFERENCIAS BIBLIOGRÁFICAS.....		124

ÍNDICE DE TABLAS

Tabla 1	<i>Requerimientos de hardware</i>	11
Tabla 2	<i>Costos de requisitos de Software</i>	12
Tabla 3	<i>Costos de Talento Humano</i>	12
Tabla 4	<i>Gastos Totales</i>	12
Tabla 5	<i>Características Principales del Pensamiento Computacional</i>	19
Tabla 6	<i>Comparativa de MySQL frente a otros DBMS</i>	32
Tabla 7	<i>Ventajas y Desventajas de MySQL</i>	33
Tabla 8	<i>Definición y Nomenclatura</i>	47
Tabla 9	<i>Detalles de Personal Involucrado</i>	48
Tabla 10	<i>Referencias</i>	48
Tabla 11	<i>Características de los usuarios</i>	55
Tabla 12	<i>Caso de Uso I: Ingreso al Sistema</i>	61
Tabla 13	<i>Caso de Uso II: Administrador de Usuario</i>	63
Tabla 14	<i>Caso de Uso III: Administrador de Grupos</i>	65
Tabla 15	<i>Caso de Uso IV: Cambio de Contraseña</i>	66
Tabla 16	<i>Caso de Uso V: Administrar Estudiantes</i>	67
Tabla 17	<i>Caso de Uso VI: Administrar Materias</i>	69
Tabla 18	<i>Caso de Uso VII: Administrar Matrículas</i>	70
Tabla 19	<i>Caso de Uso VIII: Administrar Comentarios</i>	72
Tabla 20	<i>Caso de Uso IIX: Administrar Pruebas</i>	74
Tabla 21	<i>Caso de Uso IX: Administrar Talleres</i>	77
Tabla 22	<i>Prueba de Performance</i>	92
Tabla 23	<i>Prueba de Stress 1</i>	93
Tabla 24	<i>Prueba de Stress 2</i>	93
Tabla 25	<i>Población de estudio</i>	101
Tabla 26	<i>Pregunta 1 de Profesores</i>	102
Tabla 27	<i>Pregunta 2 de Profesores</i>	103
Tabla 28	<i>Pregunta 3 de Profesores</i>	104
Tabla 29	<i>Pregunta 4 de Profesores</i>	105
Tabla 30	<i>Pregunta 5 de Profesores</i>	106
Tabla 31	<i>Pregunta 6 de Profesores</i>	106
Tabla 32	<i>Pregunta 7 de Profesores</i>	107
Tabla 33	<i>Pregunta 8 de Profesores</i>	108
Tabla 34	<i>Pregunta 9 de Profesores</i>	108
Tabla 35	<i>Pregunta 1 de Estudiantes</i>	109
Tabla 36	<i>Pregunta 2 de Estudiantes</i>	110
Tabla 37	<i>Pregunta 3 de Estudiantes</i>	111
Tabla 38	<i>Pregunta 4 de Estudiantes</i>	112
Tabla 39	<i>Pregunta 5 de Estudiantes</i>	112
Tabla 40	<i>Pregunta 6 de Estudiantes</i>	113

Tabla 41 <i>Pregunta 7 de Estudiantes</i>	114
Tabla 42 <i>Pregunta 1 al administrador</i>	115
Tabla 43 <i>Pregunta 1 al administrador</i>	115
Tabla 44 <i>Pregunta 3 al administrador</i>	116
Tabla 45 <i>Pregunta 4 al administrador</i>	117
Tabla 46 <i>Pregunta 5 al administrador</i>	118
Tabla 47 <i>Pregunta 6 al administrador</i>	118

ÍNDICE DE FIGURAS

Figura 1 Inicio de Sesión de Bushido	5
Figura 2 Inicio de Sesión Nipro-OCD Pedidos	5
Figura 3 Arquitectura del Ecosistema I	6
Figura 4 Arquitectura del Ecosistema 2.....	7
Figura 5 Elementos del Pensamiento Computacional	21
Figura 6 Logo de AppGini.....	24
Figura 7 Funcionamiento de AppGini 1	28
Figura 8 Funcionamiento de AppGini 2	29
Figura 9 Logo de PHP	33
Figura 10 Funcionamiento de PHP.....	37
Figura 11 Interpretación mediante PHP.....	38
Figura 12 Logo de Git.....	38
Figura 13 Logo de BitBucket	40
Figura 14 Logo de PHPStorm.....	40
Figura 15 Interfaz Inicial de PHPStorm	41
Figura 16 Ciclo de Vida Scrum	44
Figura 17 Interfaz del aplicativo (administrador)	50
Figura 18 Interfaz del aplicativo (administrador) 2	51
Figura 19 Caso de Uso I: Ingreso al Sistema.....	63
Figura 20 Caso de uso II: Administrador de Usuarios.....	64
Figura 21 Caso de uso III: Administrador de Grupos.....	66
Figura 22 Caso de Uso V: Administrar Estudiantes	68
Figura 23 Caso de Uso VI: Administrar Materias	70
Figura 24 Caso de Uso VII: Administrar Matrículas.....	72
Figura 25 Caso de Uso VIII: Administrar Comentarios	73
Figura 26 Caso de Uso IIX: Administrar Pruebas	75
Figura 27 Caso de Uso IIX: Prueba	76
Figura 28 Caso de Uso IIX: Respuestas	76
Figura 29 Caso de Uso IX: Administrar Talleres	78
Figura 30 Caso de Uso IX: Administración de Talleres Detallado	79
Figura 31 MER de la Base de Datos.....	89
Figura 32 Entorno de Desarrollo de AppGini.....	90
Figura 33 Entorno de Desarrollo de PHPStorm.....	91
Figura 34 Archivos de Sistema siendo Editados	91
Figura 35 Pregunta 1 Profesores.....	103
Figura 36 Pregunta 2 Profesores	104
Figura 37 Pregunta 3 Profesores.....	105
Figura 38 Pregunta 4 Profesores.....	105
Figura 39 Pregunta 5 Profesores.....	106
Figura 40 Pregunta 6 Profesores.....	107

Figura 41 Pregunta 7 Profesores	107
Figura 42 Pregunta 8 Profesores	108
Figura 43 Pregunta 9 Profesores	109
Figura 44 Pregunta 1 Estudiantes	110
Figura 45 Pregunta 2 Estudiantes	111
Figura 46 Pregunta 3 Estudiantes	111
Figura 47 Pregunta 4 Estudiantes	112
Figura 48 Pregunta 5 Estudiantes	113
Figura 49 Pregunta 6 Estudiantes	114
Figura 50 Pregunta 7 Estudiantes	114
Figura 51 Pregunta 1 Administrador	115
Figura 52 Pregunta 2 Administrador	116
Figura 53 Pregunta 3 Administrador	117
Figura 54 Pregunta 4 Administrador	117
Figura 55 Pregunta 5 Administrador	118
Figura 56 Pregunta 6 Administrador	119

GLOSARIO

Apache: Se trata de un servidor web que trabaja mediante el protocolo HTTP de código abierto para diversas plataformas web.

Array: Son mecanismos de organización que facilitan la gestión de los datos por parte de los dispositivos lógicos que se sirven de él.

Back-End: dicho concepto se refiere al conjunto de procesos y protocolos que se ejecutan del lado del servidor web o el cliente destinado gestionar los ingresos y accesos a diferentes módulos dependiendo de los niveles de usuario y compartiendo los datos concernientes al manipular la base de datos.

Cliente FTP: es aquel que emplea el protocolo FTP para conectarse con un servidor web y transmitir archivos hacia otro alojamiento.

Front-end: El termino se refiere a todas aquellas tecnologías y protocolos que se ejecutan del lado del dispositivo o computador, esto aplicado a la naturaleza de las aplicaciones web se refiere entonces a los procesos y protocolos que se corren desde un navegador web.

FTP: son las siglas del término conocido como protocolo de transferencias de archivos y se refiere a una serie de estándares web de transferencias de archivos entre dispositivos conectados mediante una red TCP/IP (Familia de protocolos del internet).

Librerías: se le llama así a la agrupación de utilidades funcionales codificadas en un determinado de lenguaje de programación.

Sript: Este término se refiere a la unidad más sencilla de un archivo de programación informática casi siempre de carácter de texto plano.

Servidor web: es aquel sistema de información que se encarga de interpretar los procesos y protocolos que están destinados a ejecutarse del lado del proveedor de las aplicaciones o sistemas.

RESUMEN

El Pensamiento Computacional es un movimiento educativo a nivel internacional, el mismo involucra la resolución de problemas, diseño de sistemas y el entendimiento del pensamiento humano utilizando los conceptos fundamentales de las Ciencias de la Computación por lo tanto este proyecto se enfoca en el desarrollo e implantación de un Ecosistema Informático que consiste en la combinación de un LMS (Learning Management System) y un Portafolio Electrónico que sirva de apoyo para la enseñanza y normalización del Pensamiento Computacional en la Universidad de las Fuerzas Armadas – ESPE así como una base para macro-proyectos educativos. Para el desarrollo del ecosistema se ha utilizado la metodología SCRUM junto con el IDE de desarrollo PHPStorm, el framework de desarrollo AppGini que trabaja con el lenguaje de programación PHP y con el DBMS MySQL para la base de datos; también se usó la herramienta de versionamiento Git para el alojamiento del código en la nube con su cliente llamado BitBucket y se obtuvo los requerimientos del sistema a partir de las experiencias y conocimientos de los docentes expertos en el Pensamiento Computacional de la Universidad de las Fuerzas Armadas – ESPE. La Aplicación finalmente estará alojada en un servidor web Apache en Amazon Web Services para su correcto funcionamiento y despliegue.

Palabras clave:

- **PENSAMIENTO COMPUTACIONAL.**
- **CIENCIAS DE LA COMPUTACIÓN.**
- **ECOSISTEMA INFORMÁTICO.**
- **APPGINI.**
- **GIT.**

ABSTRACT

Computational Thinking is an educational movement at an international level, it involves the resolution of problems, the design of systems and the knowledge of human thought that uses the fundamental concepts of Computer Science so this project focuses on the development and implementation of a Computer Ecosystem consisting of the combination of an LMS (Learning Management System) and an Electronic Portfolio that will support the teaching and normalization of Computational Thinking at the University of the Armed Forces - especially as a basis for macro educational projects. For the development of the ecosystem the SCRUM methodology has been used together with the development IDE PHPStorm, the AppGini development framework that works with the PHP programming language and with the MySQL DBMS for the database; the version tool for hosting the code in the cloud with its BitBucket client was also used and the system requirements of the experiences and knowledge of the experts in Computational Thinking of the University of the Armed Forces - ESPE were obtained. The Application will be available on an Apache web server in Amazon Web Services for its correct operation and deployment.

Keywords:

- **COMPUTATIONAL THINKING.**
- **COMPUTER'S SCIENCE.**
- **COMPUTER ECOSYSTEM.**
- **APPGINI.**
- **GIT.**

CAPÍTULO I

INTRODUCCIÓN

1.1 ANTECEDENTES

En los últimos años se ha impulsado un movimiento educativo a nivel internacional relacionado con la introducción del “Pensamiento Computacional”, la programación informática y la robótica en los diferentes niveles educativos. De hecho, algunos sistemas educativos están incluyendo nuevas competencias en sus currículos vinculados con este tema (Borrega J. A., 2016).

El término Pensamiento Computacional fue propuesto por Jeannette M. Wing (ACM, 2006), ella lo define como: “Pensamiento Computacional involucra la resolución de problemas, diseño de sistemas, y el entendimiento del pensamiento humano utilizando los conceptos fundamentales de las Ciencias de la Computación”. Debido al impacto que generó este concepto se han realizado numerosas investigaciones y proyectos pilotos para poder implantarlo en las mallas curriculares y syllabus desde los niveles más bajos hasta la educación superior principalmente a nivel de la Unión Europea que ha sido pionera en este ámbito, así como sus similares en El Reino Unido y Estados Unidos.

La revisión sistemática de literatura realizada por Jesús Acevedo Borrega en el año 2016 cuyo título es: “El Pensamiento Computacional en la educación obligatoria, una revisión sistemática de literatura”, en la que expone que las repercusiones del Pensamiento Computacional en todos los niveles de la Educación son innumerables. Es por ello que en los últimos diez años se aprecia un aumento de la producción científica que estudia el concepto en todos sus ámbitos.

Al momento, no se ha podido encontrar un ecosistema que ayude tanto al seguimiento como al soporte para el desarrollo del pensamiento computacional (Charoula, et al., 2016) , a razón de todos los niveles educativos, existen aplicativos como por ejemplo SCRATCH (Programación Visual) (Scratch, 2017) o CodeAcademy (Cursos de Programación Online) (CodeAcademy, 2017), mediante los cuales de alguna manera se contribuye al desarrollo del Pensamiento Computacional (Silapachote & Srisuphab, 2016), sin embargo, este no puede tener un seguimiento deseado.

Con estos antecedentes, la Universidad de las Fuerzas Armadas ESPE a través del Grupo de Investigación de Innovación Educativa (InnovaEdu) (InnovaEDU, 2017), propone desarrollar un ecosistema informático para el desarrollo del Pensamiento Computacional compuesto por un LMS (Learning Management System) y un Portafolio Electrónico a fin de establecer de forma experimental la línea base de la institución respecto al pensamiento computacional y definir las acciones que contribuyan a mejorar las habilidades y destrezas computacionales en los estudiantes a través de recursos y actividades en varios módulos implementados en el ecosistema.

1.2 PLANTEAMIENTO DEL PROBLEMA

En 2006, el Parlamento Europeo y el Consejo de la Unión Europea publicaron una serie de recomendaciones sobre ocho competencias claves para el aprendizaje a lo largo de la vida, entre las cuales se encontraba la Competencia Digital (Europa Ec, 2016). En el mismo año, el término “pensamiento computacional” fue propuesto por Jeanette Wing, centrando la atención no en el uso seguro de la tecnología, sino en la comprensión de sus conceptos básicos subyacentes. La tendencia actual de integrar el pensamiento computacional en la enseñanza obligatoria hace que la investigación de su relación con el término Competencia Digital sea aún más relevante (Bocconi, Chiocciariello, Dettori, Ferrari, & Engelhardt, 2016).

La introducción del Pensamiento Computacional en la enseñanza obligatoria requiere formación del profesorado. Según Eurostat (Europa Ec, 2016), hay un total de 2 millones de maestros de primaria y 2,5 millones de profesores de secundaria en los 28 países de la Unión Europea. Y es que, por ejemplo, en Italia, el Plan de Escuela Digital del Ministerio de Educación (Piano Nazionale Scuola Digitale. (PNSD), 2017) prevé la formación de 157.000 docentes, entre marzo de 2016 y diciembre de 2017, a través de talleres presenciales, cursos en línea y formación en cascada. En Francia, el proyecto de formación del profesorado Class'Code dirigido por la SIF (Société Informatique de France) y gestionado por el INRIA (el instituto nacional francés de Computación y Matemáticas Aplicadas) estima que 300.000 profesores recibirán formación en Pensamiento Computacional.

Asimismo, El proyecto Computing At School (CAS) (Codejobs, 2013), propone un enfoque innovador para apoyar el nuevo currículo de computación en Inglaterra: los docentes

experimentados se convierten en formadores del CAS, que trabajan con alrededor de 40 docentes en su comunidad local. Estos formadores del CAS reciben un curso de formación de 5-10 días durante seis meses. Actualmente hay más de 350 docentes, con el objetivo de reclutar 600 para 2018. Las oportunidades de formación docente debatidas en la literatura se centran principalmente en aspectos pedagógicos en lugar de en habilidades tecnológicas. La mayoría de la formación parece estar diseñada para los docentes de todas las asignaturas, a veces centrándose en los docentes de STEM (Science, Technology, Engineering, Mathematics).

De esta manera, Los enfoques pedagógicos abordados incluyen narración, resolución de problemas, pedagogías deductivas e inductivas centradas en modelos computacionales, y simulación. A menudo, las acciones formativas están diseñadas específicamente para ser prácticas, de manera que los docentes puedan aplicar más fácilmente sus nuevas habilidades en sus clases. Aunque se han desarrollado varios MOOCs (Cursos online masivos y abiertos), parece que el componente presencial de la formación tiene gran relevancia. (Bocconi, Chiocariello, Dettori, Ferrari, & Engelhardt, 2016).

Hasta la fecha, luego de revisadas varias fuentes bibliográficas se ha determinado que en el país no existe ninguna propuesta ni planificación para el desarrollo del Pensamiento Computacional en la educación básica y secundaria. A nivel universitario en la Universidad de las Fuerzas Armadas ESPE tampoco se evidencia ninguna planificación, investigación o cursos enfocados en el tema.

A partir de varias reuniones con el grupo de investigación se ha determinado la necesidad del desarrollo de un ecosistema como alternativa de solución. Para agilizar este proceso se ha escogido la herramienta AppGini debido a que aplica un desarrollo ágil y sistemático, así como el lenguaje de programación PHP y el gestor de base de datos MySQL, y luego por la experiencia que tiene el Tesista en el uso de la herramienta descrita anteriormente. Es importante también acotar que se garantiza una administración centralizada de usuarios, actividades, recursos y seguridades en el ecosistema propuesto.

1.3 JUSTIFICACIÓN

Cuando conviene hablar del Pensamiento Computacional lo primordial es aclarar que el concepto no es solamente relevante en el mundo digital, sino que cabe tener en cuenta que es una metodología de análisis que posee una amplia variedad de beneficios educativos e intelectuales que aseguran que la aplicación de las denominadas ‘abstracciones’ facilitan y sustentan todas las aptitudes intelectuales y de razonamiento, y que esto se ve resumido en que posee un sinnúmero de aplicaciones en la vida cotidiana.

En Ecuador no existe un registro de investigaciones o proyectos relacionados con el Pensamiento Computacional en la Educación Primaria, Secundaria o Universitaria.

Se realizará el desarrollo de un ecosistema informático con la herramienta AppGini debido a que se cuenta con una amplia experiencia en la herramienta teniendo como casos de éxito sistemas publicados en internet como por ejemplo: “Nipro-OCD Pedidos” el cual es un sistema de pedidos realizado para la empresa Nipro Medical Corp y que se lo puede ver en: nipro-ocd.com/pedidos/; así como un sistema de reconocimientos de gestión humana realizado para la parte de recursos humanos llamado “Bushido” y que se lo puede ver en: bushidoecuador.com/ así mismo para la misma empresa durante la permanencia del tesista en dicha empresa; se muestra en la Figura 1 y Figura 2 respectivamente las pantallas de inicio de sesión de las aplicaciones mencionadas anteriormente.

Debido al análisis del problema, es de vital importancia, desarrollar un ecosistema informático que es una plataforma desarrollada por componentes centrales hechas por el propietario y que se complementan con aplicaciones externas que permita el desarrollo y normalización del Pensamiento Computacional a través de las características combinadas de un Learning Management System (LMS) y un Portafolio Electrónico; con esto se obtendrá: estadísticas y referencias sobre el desarrollo del Pensamiento Computacional, por ello, en la Figura 1 se detalla a continuación el ecosistema de desarrollo para el proyecto.



Figura 1 Inicio de Sesión de Bushido
Fuente: (Nipro, 2017)



Figura 2 Inicio de Sesión Nipro-OCD Pedidos
Fuente: (Nipro, 2017)

1.4 OBJETIVOS

1.4.1 Objetivo General

Desarrollar e Implantar un Ecosistema informático compuesto por un LMS (Learning Management System) y un Portafolio Electrónico que facilite la enseñanza y normalización del Pensamiento Computacional para la Universidad de las Fuerzas Armadas ESPE utilizando la herramienta AppGini y la metodología SCRUM.

1.4.2 Objetivos Específicos

- 1.4.2.1 Establecer los Requerimientos y Reglas del Negocio aplicando encuestas y entrevistas a docentes e investigadores expertos en el área del Pensamiento Computacional.
- 1.4.2.2 Diseñar la arquitectura de información del ecosistema en base a los requerimientos establecidos en el numeral anterior utilizando las librerías de modelamiento de la herramienta AppGini.
- 1.4.2.3 Desarrollar el ecosistema utilizando la herramienta AppGini y la metodología SCRUM.
- 1.4.2.4 Implementar la solución propuesta validando los procesos desarrollados y realizar las pruebas recomendadas por la Ingeniería de Software.

1.5 ALCANCE

Se propone realizar el desarrollo de un ecosistema que permita validar el proyecto propuesto (Ver Figura 3) utilizando como grupo experimental a los estudiantes de 1er nivel de la carrera de Ingeniería en Tecnologías de la Información de la Universidad de las Fuerzas Armadas ESPE.

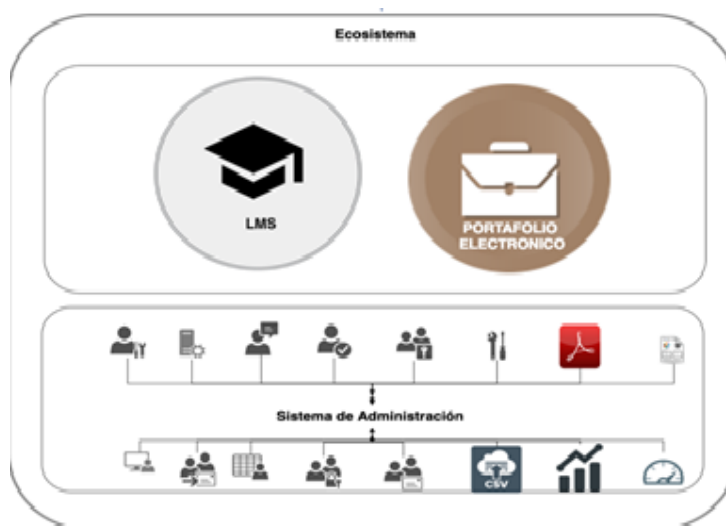


Figura 3 Arquitectura del Ecosistema I



Figura 4 Arquitectura del Ecosistema 2

El Ecosistema Informático permanecerá en una arquitectura (Ver Figura 4) alojada en un servidor de Amazon Web Service (Amazon, 2017) de distribución Centos 6 o superior, que proveerá los servicios de Apache HTTP para el alojamiento del proyecto en lenguaje PHP y el servicio de MySQL para la base de datos.

El Ecosistema Informático contará con las siguientes funcionalidades:

- Módulo General de Usuarios
 - Módulo de autenticación de usuarios.
 - Módulo de recuperación de usuarios.
 - Módulo de registro de usuarios.
 - Módulo de administración completa de Usuarios, Roles y Grupos.
- Módulo de Auditoria
 - Módulo de Auditoria y Monitoreo a nivel de usuarios y sus respectivas actividades.
- Módulo de Seguridad
 - Módulo de Seguridad del sistema contra ataques de tipo: SQL, XSS, CSRF.
 - Módulo de Seguridad de las transacciones de datos encriptados con MD5.
 - Módulo de configuración general (Hora, Fechas, SMTP, Mensajes de Bienvenida, Mensajería Individual y Grupal) del Ecosistema Informático.
- Módulo de Comunicaciones
 - Módulo de mensajería (e-mail) del sistema con notificaciones automáticas.

- Características del Sistema para dispositivos Móviles y Ambientes Web
 - Funcionalidad de redimensionamiento de la vista tanto para plataforma web y móvil.
 - Validaciones a nivel de Vista de Usuario y Base de datos para cada requerimiento del Ecosistema Informático.
 - Características de Escalabilidad, Estandarización, Usabilidad para los requerimientos del Ecosistema Informático.
- Módulo de Contenidos
 - Módulo de subida y difusión de Contenidos (Archivos - LMS).
 - Funcionalidades de generación de Pdf y archivos de Excel para cada requerimiento del Ecosistema Informático.
 - Módulo de seguimiento y evaluación de Contenidos (Archivos -Portafolio Electrónico).
 - Módulo de almacenamiento y replicación de recursos (Archivos – Portafolio Electrónico).
- Módulo de gestión de reportes para los contenidos en base a los criterios levantados en los requerimientos.
- Módulo gestión de reportes para Seguimiento y Evaluación de los Contenidos en base a los criterios levantados en los requerimientos

El Alcance del Ecosistema Informático será hasta completar las principales características de un LMS las cuales son:

- **Interactividad:** la persona como protagonista de su propio aprendizaje.
- **Flexibilidad:** hace referencia al hecho de que una plataforma pueda ser adaptada no solo a los planes de estudio sino también a los contenidos y estilos pedagógicos de la organización.
- **Escalabilidad:** se refiere a la capacidad de la plataforma para funcionar con diversa cantidad de usuarios según las necesidades de la organización.
- **Estandarización:** es la que permite utilizar cursos realizados por terceros.
- **Usabilidad:** es la facilidad con que las personas pueden utilizar la plataforma con el fin de alcanzar un objetivo concreto.

- **Funcionalidad:** son las prestaciones-características que hacen que esa plataforma sea adecuada (funcional) según los requerimientos y necesidades de los usuarios.
- **La ubicuidad:** la capacidad de una plataforma o sitio de hacer sentir al usuario omnipresente. Que la plataforma nos transmite la "tranquilidad" de que todo lo que necesitamos lo encontramos ahí. Es decir, esa plataforma pasa a formar parte de nuestro entorno personal de aprendizaje (PLE) de forma transversal y no ocupa simplemente el lugar de ser una url para bajar un contenido o entregar una tarea.

Un Portafolio Electrónico tiene algunas características que se listan a continuación:

- Administración de archivos
- Disponibilidad de archivos
- Segmentación y filtrado de archivos para los diferentes tipos de aprendizaje
- Disponibilidad inmediata del recurso; siempre alojado en el ecosistema.

En lo que se refiere al Seguimiento y Evaluación de los Contenidos del estudiante, los cuales serán determinados mediante las entrevistas y reuniones con investigadores y expertos en el área del Pensamiento Computacional.

1.6 FACTIBILIDAD

1.6.1 Factibilidad Técnica

El análisis de factibilidad técnica evalúa si el hardware y software están disponibles si se puede desarrollar la aplicación. También se evalúan las capacidades técnicas requeridas por cada alternativa del diseño que se esté considerando. La diferencia entre la realidad y el ideal es el resultado de una variedad de factores. En primer lugar, la mayoría de los sistemas de software operan con gran cantidad de resultados y fórmulas, actividades y algoritmos complejos. Por lo tanto, la presencia de defectos no solo es una función del software sino también de las expectativas de los usuarios. (Pfleeger, 2002)

El desarrollo del presente proyecto de investigación es factible debido a que el tesista ha puesto a disposición sus recursos y herramientas, para la realización del proyecto además de contar con todo el apoyo en los temas de información y logística para el desarrollo del

proyecto. Se cuenta también con el software AppGini que es de licencia comercial y que ya fue adquirido por el estudiante y que tiene amplia gama de documentación y herramientas disponibles para la realización del proyecto. Así mismo se cuenta con el servidor alojado en Amazon Web Services el cual proveerá toda la arquitectura necesaria para el desarrollo del presente proyecto. Actualmente el Grupo de Investigación de Innovación Educativa (InnovaEdu) proveerá todo su apoyo en la realización del proyecto en cuestión.

HARDWARE

- El sistema funcionara en un servidor alojado en la nube o cloud en Amazon Web Services.

SERVIDORES

Los servidores son las siguientes:

- Servidor de aplicaciones
- Servidor de Base de Datos
- Servidor Web

SOFTWARE

- IDE de desarrollo PHPStorm
- Base de Datos MySql
- Servidor Web Apache
- AppGini Development Tool

1.6.2 Factibilidad Económica

Debido a que se utilizará versiones Open Source a nivel de código y un software de licencia comercial como se muestra en la Tabla 2, pero de bajo coste para el desarrollo del

presente proyecto, presenta una factibilidad positiva, además del apoyo del “Grupo de Investigación de Innovación Educativa (InnovaEdu)” como facilitador para el desarrollo del proyecto de investigación (Ver Tabla 3).

En base a lo anterior se estima un costo de hardware bastante módico teniendo en cuenta las sumas arrojadas por las investigaciones previas realizadas por el investigador (Ver Tabla 1), esto debido también a la elección de herramientas de trabajos adaptables, ágiles y que se caracterizan por poseer un bajo uso de los recursos físicos y lógicos de los dispositivos que requieren para funcionar.

En ese orden de ideas, se anexa a continuación una serie de tablas que especifican los costos que supondrían el desarrollo e implementación del proyecto planteado como sumatoria total se puede visualizar en la Tabla 4 de gastos totales:

Tabla 1

Requerimientos de hardware

HADWARE			
Equipo	Cantidad	Costo Unitario	Costo Total
Portátil MAC BookPro i5	1	1500	1500
Impresora EPSON L220	1	200	200
Flash Memory Kingston 16 GB	1	12	12
TOTAL			1712

Tabla 2*Costos de requisitos de Software*

SOFTWARE			
Equipo	Cantidad	Costo Unitario	Costo Total
Sistema Operativo			
Windows 7	1	0	0
Microsoft Office			
2016	1	0	0
PHP	1	0	0
MySql	1	0	0
Servidor Apache	1	0	0
Hosting de Pruebas	6	70	420
AppGini	1	79.90	79.90
TOTAL			499.90

Tabla 3*Costos de Talento Humano*

TALENTO HUMANO CANTIDAD ESTIMADA POR 6 MESES DE DURACIÓN DEL PROYECTO	
DESARROLLADOR	2142
TOTAL	2142

Tabla 4*Gastos Totales*

TOTALES	
Equipo	Costo Unitario
Cantidad	Costo Total
HARDWARE	1712
SOFTWARE	499.90

CONTINÚA 

TALENTO HUMANO	2142
TOTAL	4353.90

1.6.3 Factibilidad Legal

En el desarrollo del presente sistema se estarán utilizando software de naturaleza de carácter privativo para los cuales se procedió a cancelar el monto requerido para obtener la licencia y el hosting del servidor para cumplir y respetar las condiciones de propiedad intelectual por ende no existe ningún impedimento jurídico dentro del país que impida el desarrollo o implementación de la aplicación planteado.

1.6.4 Factibilidad Operativa

El desarrollo del presente proyecto es factible debido a que el Tesista cuenta con el conocimiento necesario del software y lenguajes, el mismo que ha sido adquirido a lo largo de su formación académica.

Al finalizar el presente proyecto se tendrá disponible el código en el repositorio Bitbucket (Bitbucket, 2017), a través de la herramienta Git (Git, 2017), aquí se alojara todas las ramas o etapas que se hayan realizado durante todo el proceso de desarrollo del Ecosistema Informático, además el código será comentando y sustentado con su respectiva documentación en cada etapa de desarrollo para que pueda ser usado bien en su etapa final o en alguna etapa del desarrollo para futuros proyectos; se dejara en el repositorio todo el código, el modelo de la herramienta AppGini (AppGini, 2017) y el esquema de la base de datos del Ecosistema Informático; todo esto al acceso público y estableciendo al Tesista como revisor principal de los futuros aportes de la comunidad al proyecto mencionado.

Para este efecto, se adjunta la carta de auspicio del Grupo de Investigación de Innovación Educativa (InnovaEdu) (InnovaEDU, 2017) con la cual se compromete a brindar todo el apoyo tecnológico y humano requerido para la realización del mismo; la carta se anexa en la sección Anexos.

CAPÍTULO II

MARCO TEÓRICO

2.1 PENSAMIENTO COMPUTACIONAL

Ante todo, es necesario destacar que el presente trabajo de investigación encuentra sus fundamentos en el concepto del pensamiento computacional, por ende se consideró propicio que dicho concepto y las consiguientes reflexiones del tesista acerca de él se ubicasen como primer punto a tratar dentro del marco teórico del trabajo de grado.

2.1.1 Orígenes del pensamiento computacional.

El concepto de pensamiento computacional ha estado durante mucho tiempo ligado al mundo de las ingenierías de la informática y la computación, sin embargo, desde alrededor de una década, surgen corrientes que incorporan el pensamiento computacional como parte de la educación a través de experiencias de desarrollo de proyectos mediante lenguajes de programación visual o desarrollando el concepto mediante lápiz y papel.

Sin embargo, aunque el termino se encuentra en temprana madurez, debido al auge que ha tenido en los últimos años, existe ya documentación previa que data de más de 50 años, desde las bases del pensamiento computacional, cuando el matemático Seymour Papert, discípulo de Jean-Piaget junto a Marvin Minsky, co-fundó el MIT (Massachusetts Institute Of Technology) Artificial Intelligence Laboratory en 1967, donde desarrolló la primera versión de un lenguaje de programación visual, llamado “LOGO”. De este modo, en los últimos años el pensamiento computacional ha estado en énfasis sobre las manos de Jeannette Wing quien inicio el punto de partida para su aplicación en la educación en 2006, pero siendo Papert precursor de la idea de que los estudiantes, pudieran desarrollarlo a través de la programación mediante “LOGO”.

En continuación a la idea de Papert, se fueron sumando varios entornos de programación diseñados para que los usuarios aprendan a programar y a razonar mediante la lógica y la estructura del pensamiento computacional. Entre ellos, podemos destacar en orden

cronológico el LOGO de (Papert, 1980), posteriormente rediseñado por (Kafai y Resnick, 1996), seguidos de Etoys (Kay, 2010), Greenfoot (Greenfoot, 2015), Alice (Alice, 2015) y el popular, Scratch (Scratch, 2015) mundialmente utilizado en escuelas de más de 180 países en el mundo.

En este sentido, el termino de pensamiento computacional hace su aparición en la última década, en una columna de publicación para la Association for Computing Machinery (ACM) en Marzo de 2006, cuando Jeanette Wing explica que el pensamiento computacional “representa una actitud y unas habilidades universales que deberían ser de todos los individuos, no solo científicos o ingenieros computacionales”. Refiriéndose al pensamiento computacional como una habilidad y competencia que debe ser de todos los individuos para el desenvolvimiento y desarrollo basado en procesos y el entendimiento del comportamiento humano, a través del desarrollo e implementación de sistemas sobre procesos que ejecuten los humanos y computadores, para solventar problemas comunes.

En 2010, El Consejo Nacional de Investigación (NRC) de los Estados Unidos, realizó el “Taller sobre el alcance y la naturaleza del pensamiento computacional”, donde asistieron prestigiosos investigadores internacionales, incluyendo Jeannette Wing, de este taller, cabe destacar la acotación de la ventaja que aporta el pensamiento computacional según numerosos estudios, entre ellos (Kolodner, 2010) expresa que: “el pensamiento computacional permite a los jóvenes y niños, pensar diferente a la hora de resolver problemas, analizar asuntos cotidianos desde una forma distinta.” Entendiendo de esta manera, los elementos y herramientas que la tecnología tiene para ofrecer, aun mas con el pensamiento computacional.

Desde los anteriores análisis, la acotación de Wing, se convierte en el punto de partida del debate del pensamiento computacional, dando origen para otros conceptos como el propuesto por la Royal Society en 2012, el cual hace gran énfasis en que la computación no es solamente una obra humana, sino que también está presente en la naturaleza, por ejemplo, en el ADN. En ese orden de ideas, (RoyalSociety, 2012) afirma que: “se define al Pensamiento Computacional como el conjuntos de procedimientos cuyo propósito es la aplicación métodos y estrategias usado en el mundo de la computación, pero esta vez aplicado en el medio que nos rodea, de manera que se aborden las problemáticas desde otro enfoque más objetivo y sistemático que permita un entendimiento más eficiente del mismo, y poder encontrar de manera más ágil una solución rápida y práctica.”

Con lo anteriormente expuesto por la Royal Society, se podría decir que entonces, el pensamiento computacional es una habilidad íntegra, presente en el ser humano, no desarrollo del mismo, sino obra de la naturaleza, pero poco implementado o descubierto por parte de las personas, para el análisis y resolución de problemas y asuntos cotidianos.

De este modo, se agrega el MIT y el equipo de Scratch (Massachusetts Institute of Technology) (ScratchEd Team, 2015), quienes definen el pensamiento computacional como “un conjunto de conceptos, prácticas, y teorías fundamentadas en la informática, mediante las cuales, los estudiantes se desarrollan como pensadores computacionales a través del desarrollo de estrategias de diseño, resolución de problemas y maneras de colaborar.” Tal es el caso, que el pensamiento computacional, se ha ido introduciendo como un estudio formal de las competencias computacionales, siendo reconocido por numerosas instituciones y administraciones.

Aunado a lo anterior, se encuentra el ejemplo de Inglaterra, donde a partir del periodo académico 2014 – 2015, que deja en evidencia el auge de este concepto al observarse un variado número de pensum en los que su estudio ha sido incluido, por ello, encontramos casos como (Code.org, 2013), siendo este portal, gran promotor, que busca fomentar el aprendizaje de las artes de la programación en sus usuarios, asimismo, este portal, contiene en los instrumentos de aprendizaje, y guías de programación la aplicación de un análisis más sistemático sustentado en el pensamiento computacional, además de fomentar la implementación de mayor programación en los pensum escolares. Dicha iniciativa cuenta también con el apoyo de figuras relevantes de la industria como Microsoft, Facebook, entre otros gigantes de la industria en general.

Teniendo en cuenta, todo lo anteriormente expuesto, se puede sintetizar, que el pensamiento computacional, hace su primera aparición en 1967 con el matemático Seymour Papert, pero se enfatiza en 2006, con Jeannette Wing quien lo introduce como estrategia de aprendizaje orientada a resolver problemas mediante el pensamiento lógico y el razonamiento computacional. De este modo, surge en la última década y cada año más, herramientas que fomentan el aprendizaje mediante este nuevo paradigma de pensamiento, con el fin de formar personas netamente capacitadas para el desenvolvimiento óptimo en la era digital y el entendimiento de la tecnología como herramienta de solución.

2.1.2 Definición del pensamiento computacional.

El pensamiento computacional se entiende como una habilidad de pensamiento, por tanto, independiente de la tecnología, y es una forma específica de resolución de problemas y situaciones que implican la capacidad de desarrollar soluciones para ser implementadas por un computador, humano, o ambos.

En este sentido, (Wing, 2006) especifica que: el Pensamiento Computacional encuentra como su propósito principal el “solventar problemáticas mediante la estructuración de estrategias y sistemas de acuerdo con el comportamiento de la mente humana, sirviéndose de la documentación básica de la ciencia computacional”. Lo que refiere a un pensamiento para solventar problemas basándose en procesos y métodos que realiza un humano, diseñando sistemas que por sí solos no podrían. Además, también señala que: “el pensamiento computacional son todos los procesos que se ven implicados en la formulación de problemas y sus soluciones para que estas últimas estén representadas de manera que puedan llevarse a cabo de manera efectiva por un procesador de información”.

Siguiendo el orden de ideas anteriormente expuesto, (Wing, 2006) también señala que el ‘pensamiento computacional’ se trata de “una competencia que todo ser humano debería conocer para desarrollarse en la sociedad digital, pero sin embargo, no es una habilidad ‘mecánica’, ya que el pensamiento computacional se refiere a una manera de resolver problemas de forma inteligente, en la cual los conceptos computacionales se aplican para resolver problemas reales, comunicarnos con personas y solventar múltiples situaciones de la vida cotidiana.

Asimismo, (Olabe, 2015) afirma que el pensamiento computacional “es presentado como una ‘metodología’ que pretende implementar conceptos básicos de la informática y computación, para la resolución de todo tipo de problemas. Una forma para abordar los problemas de manera eficaz y con altas probabilidades del éxito, que sin el pensamiento computacional, no sería posible.”. Definiéndolo no como una habilidad, sino, como una metodología estructuradamente capaz de implementarse como herramienta para la resolución de problemas exitosamente.

Es por ello, que es importante destacar que el pensamiento computacional incluye innumerables repercusiones en distintos campos de la educación, así como en el desarrollo

cognitivo de los estudiantes que se ven involucrados en dicha actividad. Asimismo, las actividades derivadas del uso de la programación informática requieren de un continuo uso de memoria y altas capacidades de concentración para controlar las diversas variables que constituyen un programa, desarrollando ambas capacidades. Además, la relación que se produce entre la forma de programar un sistema, y la habilidad de solucionar problemas en el entorno del estudiante, se ven fomentados y alterados, causando en consecuencia un incremento considerable en la habilidad de pensamiento lógico, razonamiento y análisis para todos los involucrados en dicho proceso.

De esta forma, la asociación de profesores del pensamiento computacional y la sociedad internacional de la educación tecnológica (CSTA & ISTE, 2009) han concebido otro punto referencial significativo como concepto, donde además, se enumeran todas las operaciones que conforman el Pensamiento Computacional de forma práctica siendo que: el pensamiento computacional es un proceso de resolución de problemas que incluye, pero no se limita a las siguientes características:

- Formular problemas de una manera que nos permita usar un ordenador y otras herramientas para ayudar a resolverlos.
- Organizar y analizar datos de una manera lógica.
- Representar datos a través de abstracciones tales como modelos y simulaciones.
- Automatizar soluciones mediante el pensamiento algorítmico (una serie de pasos ordenados).
- Identificar, analizar e implementar posibles soluciones con el objetivo de conseguir la combinación más eficaz de pasos y recursos.
- Generalizar y transferir este proceso de resolución de problemas a una amplia variedad de problemas.

2.1.3 Características principales del pensamiento computacional.

El pensamiento computacional, viene acompañado de una serie de características que fomentan su desarrollo y comprensión a través de la aplicación de las mismas, por ello, evoluciona constantemente su integración en la sociedad actual, la cual se ve sumergida en la

era digital, y requiere cada día, pensadores acordes a dicha era. En este sentido, (Wing, 2006) resalta un conjunto de consideraciones a tener en cuenta en lo relacionado al pensamiento computacional, basado en ello. En la Tabla 5 presentada en la página continuación se pueden apreciar las acotaciones que la autora pretende que se tengan en cuenta:

Tabla 5

Características Principales del Pensamiento Computacional

#	CARACTERISTICA
1	Reformular un problema a uno parecido que sepamos resolver por reducción, encuadrarlo, transformar o simular.
2	Pensar recursivamente.
3	Procesar en paralelo.
4	Interpretar código como datos y datos como código.
5	Generalizar análisis dimensional.
6	Reconocer ventajas y desventajas del solapamiento.
7	Reconocer coste y potencia de tratamiento indirecto y llamada a proceso.
8	Juzgar un programa por simplicidad de diseño.
9	Utilizar abstracción y descomposición en un problema complejo o diseño de sistemas complejos.
10	Elegir una correcta representación o modelo para hacer tratable el problema.
11	Seguridad en utilizarlo, modificarlo en un problema complejo sin conocer cada detalle.
12	Modularidad ante múltiples usuarios.
13	Obtención previa de información y parámetros, y la organización de los procesos y metas anticipadamente para el futuro.
14	Prevención, protección, recuperarse de escenario peor caso.
15	Utilizar razonamiento para encontrar la solución.

CONTINÚA 

-
- 16 Planificar y aprender en presencia de incertidumbre.
 - 17 Buscar, buscar y buscar más.
 - 18 Utilizar muchos datos para acelerar la computación.
 - 19 Límite tiempo/espacio y memoria/potencia de procesado.
-

Fuente: (Wing, 2006)

De tal forma, se puede decir que el pensamiento computacional es un modelo de aprendizaje diferenciado notoriamente de las metodologías tradicionales y que puede implementarse en la mayoría de las actividades de educación que se realizan en aulas. Asimismo, (Olabe, 2015) agrega que “Si se fomentara el uso del pensamiento computacional en la educación, mediante herramientas como la anterior mencionada Scratch del MIT, no solo se alcanzaría la solución al problema determinado, sino más inmediato, y el incremento de las ya mencionadas competencias y habilidades.

Además, (Wing, 2006) sintetiza en una serie de artículos, los siguientes puntos respecto al pensamiento computacional:

- El proceso de pensamiento más importante y de mayor nivel en el Pensamiento Computacional es el proceso de abstracción.
- Un algoritmo es una abstracción de un proceso que admite entradas, ejecuta una secuencia de pasos y produce resultados para alcanzar una meta determinada.
- La computación es la automatización de nuestras abstracciones. Operamos mecanizando nuestras abstracciones, capas de abstracción y sus relaciones. La mecanización es posible gracias a nuestras anotaciones y modelos precisos y exactos.
- El Pensamiento Computacional utiliza la abstracción y la descomposición al abordar una tarea compleja o al diseñar un sistema complejo.
- La abstracción se utiliza en la definición de patrones, en la generalización desde ejemplos, y en la parametrización (generalización).

En ese orden de ideas, la autora busca enfatizar la importancia del uso de un pensamiento metodológico aplicado para la resolución de los problemas en la vida diaria mediante identificación de las diferentes etapas de la vida desde un enfoque “Entrada-Proceso-Salida” que permita a través de eso la generación de patrones y el estudio de la circunstancias en la que presentan las problemáticas.

De lo anteriormente expuesto por Wing, se puede tener en cuenta ciertos elementos resaltantes que integran fundamentalmente el pensamiento computacional, presentes en la Figura 5 y que se exponen a continuación:



Figura 5 Elementos del Pensamiento Computacional

- La abstracción: de acuerdo con la metodología esta palabra se refiere a convertir un problema que a simple vista se ve complicado a un nivel más sencillo y entendible mediante la simplificación de detalles innecesarios.
- El pensamiento algorítmico: Es la forma de encontrar la solución a través de una conceptualización clara de los pasos a seguir.
- La automatización: Es el proceso mediante el cual se ahorra trabajo, al programar un ordenador para que ejecute tareas de forma automática y mucho más rápido en comparación a un humano. A este respecto, (Lee, 2011) explica que “los programas que ejecutan los computadores no son más que automatizaciones de abstracciones”.
- La descomposición: Es la manera de pensar sobre los artefactos, teniendo en cuenta sus partes y componentes. De esta forma, agrega (Csizmadia, 2015) que “cada parte debe

analizarse, solventarse y desarrollarse por separado, ya que esto facilita la comprensión de grandes problemas y facilita el desarrollo de grandes sistemas.”

- La depuración: Se entiende como la aplicación sistemática de habilidades para análisis y evaluación, usando como test la localización y el pensamiento lógico y algorítmico para predecir y verificar los resultados.
- La generalización: Es comúnmente asociada con la identificación y localización de patrones, similitudes y relaciones, así como la explotación de características, siendo una nueva forma de resolver los nuevos problemas en base a las soluciones de problemas anteriores, fomentando la construcción de la experiencia previa, debido a esto, agrega (Csizmadia, 2015), que “es importante entonces, identificar patrones, ya que algoritmos que resuelven problemas específicos, pueden ser adaptados para resolver toda una clase de problemas similares, siguiendo un patrón”.

2.1.4 Investigaciones y proyectos existentes

Título: “Herramientas Informáticas para el Desarrollo del Pensamiento Computacional” (2012).

Autor(es): Hamilton Omar Pérez Narváez

Resumen: Dentro de las destrezas que requiere un habitante del siglo XXI, para desenvolverse en la sociedad, es el pensamiento computacional considerado uno, si no es que el más significativo, debido a sus aplicaciones en la vida real, ya que su objetivo es solventar problemas mediante la unificación de procesos mentales superiores, a través de herramientas informáticas, logrando así, que los resultados sean creativos, rápidos y aplicables. De esta forma, universidades de todo el mundo, han tomado tal idea, desarrollando sus propios entornos de trabajo o adaptando los ya existentes. El trabajo anteriormente citado, supone un recorrido desde sus inicios, desarrollo y dos experiencias particulares de lenguajes de programación para enseñanza, como son Scratch y Da Vinci.

Título: “Pensamiento Computacional: Que y Por Qué” (2010).

Autor(es): Jeannette M. Wing.

Resumen: En el Pensamiento Computacional la clave se encuentra entre el diseño y análisis de los problemas y sus soluciones interpretadas de una manera asertiva donde lo más

importante es el proceso de pensamiento en un alto nivel y enfoque y concentración lo que permite definir patrones, generalizar circunstancias y crear parámetros que permitan la solución del problema planteado.

Título: “Pensamiento Computacional a Través de la Programación: Paradigma de Aprendizaje” (2015).

Autor(es): Xabier Basogain Olabe, Miguel Ángel Olabe Basogain, Juan Carlos Olabe Basogain.

Resumen: El trabajo anteriormente citado, presenta la definición del Pensamiento Computacional buscando su aplicación en las aulas educativas a través de la esquematización de proyectos de programación, aunado a eso, se centra en urgencia, la finalidad y los puntos destacados del Pensamiento Computacional. Ejemplificando como se puede crear sistemas que tengan una sólida relación metodológica con el pensamiento computacional y en la orientación de la selección del lenguaje de programación adecuado, mostrando el contenido y las conclusiones arrojadas todo lo anterior enseñado mediante el uso de MOOC (Massive Open Online Courses) en la plataforma Miríada X.

2.2 HERRAMIENTAS DEL MODELO

Es aquí donde se procede a desglosar el conjunto de herramientas mediante las cuales fue posible el desarrollo del sistema y por consiguiente lograr la solución a la problemática que el presente trabajo de grado plantea, en ese orden de ideas, se definirá tanto el programa en el que se desarrolló el proyecto, el lenguaje en el que fue programado, la naturaleza del gestor de base de datos utilizado y otras herramientas varias que fueron de ayuda al momento del desarrollo. Por consiguiente:

2.2.1 AppGini.

La herramienta de diseño y desarrollo mediante el cual se llevó a cabo el proyecto recibe el nombre de AppGini (Ver Figura 6), que puede ser definido en síntesis como un programa

que permite el desarrollo de sistemas y aplicaciones y en la gestión rápida de sus respectivas bases de datos logrando un entorno y diseño amigable para sus consumidores. A continuación, se procede a definir más a fondo dicha herramienta de desarrollo:



Figura 6 Logo de AppGini

Fuente: (BigProf, 2017)

2.2.1.1 Características Generales.

Sobre AppGini se puede aportar que según (BigProf, 2017): “es un software de desarrollo que permite crear aplicaciones web mediante la generación del front-end sustentado principalmente en la base de datos. Sin la exigencia de haber desarrollado el back-end, para ello el programa exige una estructura optima de la base de datos con un adecuado uso de las relaciones; se procede a configurar otra serie de aspectos relacionados al proyecto y por último el programa procede a generar el código php que supondrá la funcionalidad del sistema y que luego será almacenada en el servidor que el desarrollador desee conveniente.”

Asimismo, AppGini tiene características que fueron consideradas interesantes y de gran utilidad por el investigador por la manera en que facilitan el desarrollo de software web, como:

- **Navegación de Datos:** Permite la manipulación completa y a conveniencia del desarrollador de todos los datos que maneja la aplicación o sistema, permitiendo ubicarlos, modificarlos y mostrarlos según los requerimientos que se posean.
- **Ordenamiento:** garantiza la clasificación personalizada y mediante tipos de todos los datos manejados.
- **Filtros:** proporciona herramientas de visualización y ordenamiento de los datos mediante parámetros específicos, de manera que se agilice la producción de diferentes reportes de acuerdo con los requerimientos fijados.
- **Edición:** permite la modificación de la información y contenido previamente plasmado en la aplicación o sistema.
- **Inserción:** garantiza la creación de nuevo contenido.

- **Eliminación:** permite suprimir contenido previamente plasmado en la aplicación.
- **Importación:** permite la interacción e intercambio de los datos de un proyecto a otro.
- **Exportación (.xls, .csv) :** garantiza la seguridad de los datos alojados en el sistema o aplicación, y del proyecto mismo al permitir la creación de archivos que cumplen la función de copias de seguridad que pueden utilizarse para ingresar datos a otro proyecto de AppGini o almacenarse en caso de pérdida de información.
- **Administración de Usuarios y Grupos:** Permite la creación de distintos niveles de usuarios y grupos así como la configuración de los accesos y visualización individualizada de información de acuerdo a dicho niveles.
- **Gran variedad de Claves foráneas:** para el desarrollo de estructura de base de datos
- **Diversidad de Estilos para desarrollo Front-end:** posee una gran cantidad de herramientas de personalización para que se pueda adaptar al proyecto de acuerdo los requerimientos y metas fijadas.
- **Soporte de Imágenes:** permite la inserción de imágenes en distintos formatos.
- **Soporte de subida de Archivos (.doc., .pdf, etc...):** Permite adjuntar archivos y documentos en variados formatos.
- **Soporte Geolocalización con la API Google Maps:** Soporta la inserción de ubicaciones y direcciones completamente compatibles con la API de mapas del Motor de Búsquedas Google.
- **Enriquecimiento de código fuente basado en la inserción de Hooks** (“insertado por parte del desarrollador mismo.”)

Se considera importante destacar también que AppGini trabaja exclusivamente mediante el lenguaje de programación conocido como PHP y gestiona las bases de datos por medio de la herramienta MYSQL, aunque, una de las características más resaltantes de esta herramienta de diseño es que no necesariamente le requiere al desarrollador un amplio conocimiento en el lenguaje de programación con el que trabaja ni una gran capacidad de manipulación de la herramienta gestora de base de datos con la que funciona, pero, tampoco niega la opción de una personalización avanzada del código que genera, esto, para aquellos desarrolladores que posean requerimientos más complejos o simplemente quieran obtener la ayuda de AppGini hasta cierto punto.

2.2.1.2 *Funcionamiento de AppGini.*

Según (BigProf, 2017) el paso inicial para comenzar a trabajar en AppGini es aquel que permite “crear un proyecto” sobre ello, el autor explica que:

“Un proyecto de AppGini es el punto de inicio de tu trabajo, el proyecto es donde tu describes y diseñas tu base de datos y sus campos, y defines la apariencia de tu aplicación y su comportamiento, una vez que termines de trabajar con tu proyecto, solo presiona el botón de ‘generar’ en la barra de herramientas o presiona F5. Luego de esto app procede a generar todo el código PHP para tu aplicación y lo guarda en la carpeta de tu preferencia, tu puede entonces cargar los archivos a tu servidor web utilizando cualquier cliente FTP”

En virtud de lo anterior se esclarece entonces que el paso principal para comenzar a desarrollar es dar a la herramienta de modelo la instrucción de crear un nuevo archivo o proyecto e indicarle las características o requerimientos principales que posee nuestro sistema o aplicación web, en ello, por supuesto, van incluidos aspectos como lo son la estructura de nuestra base de datos junto con la naturaleza de cada uno de los campos que estos poseen, los diseños generales que darán vida y personalización al front-end de la aplicación y el comportamiento en líneas generales que esta tendrá.

También, y para aportar al orden de ideas anteriores, el tesista del presente proyecto de investigación considera pertinente la explicación de diferentes aspectos de AppGini que o bien permiten y facilitan el desarrollo eficiente de sistemas y aplicaciones web o son puntos y consideraciones importante a tener en cuenta al momento del uso de esta herramienta de modelado, como lo son:

- **Ventana de Proyecto:** es la ventana mediante el cual el desarrollador interactúa para lograr el modelado o personalización del sistema o aplicación, esta consta de dos partes como lo son: el panel navegador de los archivos del proyecto que permite la visualización organizada y detallada de cada uno de los archivos de código y componentes del proyecto que se está desarrollando y el otro al que se le puede llamar panel de características o propiedades que tiene como propósito la configuración y personalización de los diferentes archivos.
- **Manipulación de tablas:** si bien AppGini se caracteriza por no fijar como requisito obligatorio que el desarrollador posea un amplio conocimiento en el lenguaje mediante

el cual se está programando o el gestor de base de datos que utiliza, si requiere un conocimiento avanzado en la creación de estructuras de bases de datos y en la aplicación de las reglas habituales de normalización y optimización de tablas, pues de la correcta identificación de campos y comportamiento de cada una de las tablas y sus relaciones que conforman la base de datos depende el óptimo funcionamiento del sistema o aplicación en general, por ende, es considerado de vital importancia que al momento de crear un proyecto y comenzar a desarrollarlo no existan ya dudas acerca de la estructura de la base de datos que se posee y de que si esta cubre o no con la totalidad de las expectativas y requerimientos que se espera que el proyecto cumpla.

- **Optimización de los campos:** de la mano con el punto anterior, y, resaltando que para esta herramienta de modelo la correcta estructura de la base de datos es la piedra angular del desarrollo se procede a resaltar la importancia de optimizar cada uno de los campos que conforman nuestras tablas de la base de datos, esto no necesariamente antes de comenzar a desarrollar, puesto que AppGini provee al usuario de varias opciones que le permiten modificar, cambiar y configurar la naturaleza de cada uno de ellos incluso después de que la estructura ha sido creada, pero si, marcaría una diferencia en cuanto a la organización del proyecto y al factor tiempo si se comienza a modelar el sistema o aplicación con una estructura de base de datos asertiva desde el comienzo.
- **Herramientas de configuración y diseño:** Esta aplicación se caracteriza por ofrecer opciones de moldeado, diseño, estructura y configuración en lo relacionado a diversos aspectos del desarrollo de software. dichas opciones se encuentran desplegadas en distintos sectores de la herramienta de trabajo de la siguiente manera: en la barra de herramientas se pueden encontrar opciones básicas de la gestión del proyecto, como lo son por ejemplo, los botones de: crear un proyecto nuevo, abrir un proyecto existente, guardar los cambios realizados en el proyecto, seleccionar el tema del proyecto que se está desarrollando, etc.... Así como también, en este apartado encontraremos las opciones que nos permitirán crear una tabla nueva, crear un campo nuevo dentro de una tabla, eliminar una tabla o campo existente o también copiar o pegar una tabla a nuestra estructura de base de datos, mientras que en la ya antes nombrada sección de “características y propiedades” se pueden modificar aspectos más focalizados u objetivos de cada elemento que se esté desarrollando.

- **Trabajar con diferentes Estilos:** App Gini se caracteriza por ofrecer un gran alcance en cuanto al control de la apariencia de los proyectos desarrollados mediante la generación propia o inserción externa del CSS también conocidas como hojas en estilo de cascadas, y, si bien el mismo programa ofrece un numero vario de temar predeterminados a seleccionar permite completamente al desarrollador adaptarlos a conveniencia por mejor se ajusten a los requisitos acordados.
- **Niveles de Seguridad:** Otra consideración importante a tener en cuenta es la planificación previa un esquema de acceso divididos por anillos de seguridad, debía a que, no todos los usuarios de nuestro sistema o aplicación en desarrollo necesitaran acceder a todos los módulos que conforman el mismo, en ese orden de ideas, un usuario nivel uno, o aquel del que se espera la interacción más básica no necesita tener acceso o visualizar información de uno de nivel tres, también llamado administrador, por ende, y con la finalidad de proteger datos o configuración sensible que pueda afectar el funcionamiento eficaz de la aplicación o sistema en desarrollo es aconsejable fijar distintos niveles de acceso e interacción de acuerdo a las necesidades o posibilidades que se le otorgaran a cada tipo de usuario.

Finalmente, en la Figura 7 se puede resumir el funcionamiento de esta herramienta de modelado de la siguiente forma:



Figura 7 Funcionamiento de AppGini 1

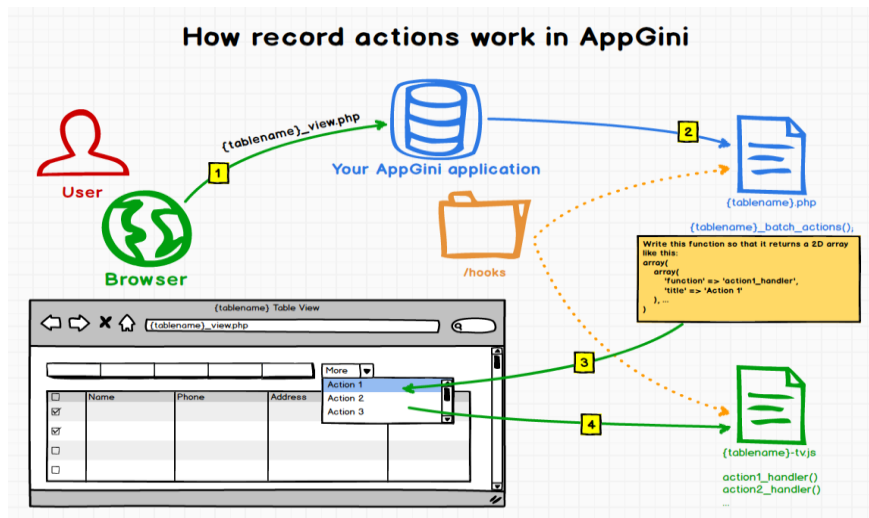


Figura 8 Funcionamiento de AppGini 2

Fuente: (BigProf, 2017)

Como se ha observado anteriormente en la Figura 8 en donde se pretende explicar de manera sencilla el funcionamiento de AppGini, se puede llegar a la conclusión de que consta de cuatro Fases:

1. El desarrollador inicia una vista que genera un archivo de nombre ‘tablename_view.php’ dentro de la carpeta seleccionada previamente para servirse de almacén por la aplicación web AppGini.
2. La herramienta de moldeado procede a llamar al archivo que recibe el nombre de ‘Hook’; en caso de que haya la existencia de código externo la aplicación procederá a aplicarla y ejecutará las acciones pertinentes que sean necesarias de lo contrario, el elemento será omitido.
3. De confirmarse la existencia de un código externo el programa entonces responderá a esto con un “array” realizando todos los procesos necesarios para optimizar la visualización de esto.
4. El desarrollador siempre podrá seguir trabajando, modificando y adaptando el proyecto de manera que los pasos anteriores se repetirán y tendrán repercusión en la vista de su aplicación o sistema.

2.2.2 MySQL.

Para ahondar en el tema de este importante gestor de datos cabe destacar ante todo que se ha seleccionado debido a que junto con el lenguaje de programación PHP forma parte del exclusivo grupo de herramientas mediante las cuales desarrolla AppGini. Teniendo en cuenta lo anterior, se procede a desglosar los temas referentes al concepto:

2.2.1 Origen de MySQL.

Para entender MySQL, es necesario puntualizar un poco de historia, su origen y el por qué surge y nace como sistema preferido para el manejo de bases de datos, en este sentido (Toledo, 2013) expone “MySQL surge en la década de 1990, cuando Michael Widenis comenzó a usar mSQL para conectar tablas utilizando sus rutinas propias de bajo nivel, tras lo cual, luego de numerosas pruebas, llegó a la conclusión de que mSQL no era lo bastante flexible ni rápido para lo que necesitaba, por lo que tuvo que desarrollar nuevas funciones. Esto resultó en una interfaz SQL a su base de datos, totalmente compatible a mSQL, el nuevo MySQL. El cual cabe destacar, era más rápido, y mejor estructurado en cuestión de sintaxis que el anterior mSQL.” Aunado a lo anterior, afirma que “en un principio, este gesto de base de datos los elementos básicos de configuración de relaciones entre base de datos, algunos de estos siendo: la integridad referencial y las transacciones propia de una DB. Sin embargo, buscaba contrarrestar esto ofreciendo una interfaz de interacción dinámica al usuario, debido a su simplicidad, de tal manera que los elementos faltantes fueron complementados por la vía de las aplicaciones que la utilizan”.

Lo cual quiere decir, que inicialmente a igual que el lenguaje de programación conocido como PHP, era distante de lo que se conoce hoy en día, pero sometido a cambios mediante su implementación constante a través de los años por parte de los desarrolladores de páginas web dinámicas de aquel entonces, que vieron potencial en él, además solucionar la problemática de los usuarios al momento de trabajar con niveles de seguridad.

2.3 DEFINICIÓN BÁSICA DE MYSQL

En cuanto al concepto de este gestor de datos se refiere podemos aportar que (García, 2007) explica: “MySQL es el sistema de administración de bases de datos (Database Management System, DBMS) más popular, desarrollado y proporcionado por MySQL AB. Es un sistema de gestión de base de datos relacional, multihilo y multiusuario.” Aparte, ahondando sobre el tema, el mismo autor explica que: “MySQL fue escrito en C y C++ y destaca por su gran adaptación a diferentes entornos de desarrollo, permitiendo su interacción con los lenguajes de programación más utilizados como PHP, Perl y Java y su integración en distintos sistemas operativos.”

En relación a lo anterior se puede acotar que el hecho de haber sido desarrollado en C y C++ le permite una mayor adaptabilidad y un incremento en su popularidad por sobre los gestores de datos rivales debido a que los desarrolladores con experiencia en esos lenguajes derivados le preferirán por poseer la capacidad de entenderle a un nivel más elemental, y, dejando en evidencia lo anterior el mismo autor (García, 2007) expone que: “Según las cifras del fabricante, existirían más de seis millones de copias de MySQL funcionando en la actualidad, lo que supera la base instalada de cualquier otra herramienta de bases de datos.”

2.4 COMPARATIVA DE MYSQL FRENTE A OTROS DBMS

Cuando el momento en que la inquietud de porque seleccionar MySQL se presenta, es importante destacar de que a pesar de que es la herramienta de gestión de base de datos más descargada en la actualidad posee un variado número de rivales, que en pequeña o gran medida contrarrestan ciertas características, ofrecen otro tipo de herramientas, están creadas para el desarrollo de sistemas con un fin específico o se adquieren a un precio más asequible.

Para ello, el investigador decidió esquematizar en la Tabla 6 una comparativa entre MySQL y dos de sus rivales más potentes:

Tabla 6*Comparativa de MySQL frente a otros DBMS*

PRESTACIÓN	ORACLE	MYSQL	SQL SERVER
Interfaz	GUI, SQL	SQL	GUI, SQL, Varios
Lenguaje soportado	Varios, C, C#, C++, Java, Ruby, Objective-C	Varios, C, C#, C++, D, Java, Ruby, Objective-C y PHP.	Java, Ruby, Python, VB, .Net, y PHP.
Sistema operativo	Windows, Linux, Solaris, HP-UX, OS X, z/OS, AIX	Windows, Linux, OS X, FreeBSD, Solaris	Windows
Licencia	Propietario	Código Libre	Propietario

El gestor de base de datos MySQL se caracteriza principalmente porque tiene como objetivo primordial el ofrecer un sistema eficiente y que proporcione opciones varias en cuanto a la manipulación de cada uno de los datos que el sistema maneja, por otro lado, a lo largo de las diferentes versiones que ha logrado ver la luz con el paso de los años se ha logrado cubrir un gran alcance en cuanto al soporte de todos los datos que manejan las diferentes columnas que conforman la estructura de la base de datos y, junto con esas versiones también le permitió poseer una gran adaptabilidad entre distintas plataformas y software, sirviéndose con ello de todo el potencial de los llamados sistemas multiprocesos gracias a la característica que posee a la cual se le conoce como multihilo.

2.5 VENTAJAS Y DESVENTAJAS DE MYSQL

Este gestor de base de datos, a pesar de haber sido seleccionado por su completa inherencia con AppGini posee además un variado número de puntos a favor y en contra por el cual debería o no utilizarse, a continuación, se describen en un cuadro aquellos considerados como de más peso por el investigador (Ver Tabla 7):

Tabla 7*Ventajas y Desventajas de MySQL*

VENTAJAS	DESVENTAJAS
Eficacia al gestionar las operaciones de manipulación de datos.	Poca investigación y documentación en lo que respecta a sus módulos y utilidades en general lo que podría ralentizar el aprendizaje.
La Elaboración de la estructura de una base de datos no supone un gran costo ya que el sistema gestor posee un bajo consumo de los recursos del sistema.	No es tan perceptivo y amigable como otros gestores rivales.
Simplicidad en su proceso de instalación y calibración para el desarrollo.	Posee un límite capacidad de almacenamiento.
Probabilidad mínima de pérdida y corrupción de datos.	Al momento de trabajar con un número de datos masivos pierde cierto porcentaje de su estabilidad, obligando al desarrollador a crear base de datos secundarios o migrar hacia otra alternativa.

2.5.1 PHP**Figura 9** Logo de PHP

Fuente: Imágenes de Google

Al momento de referirse a PHP (Ver Figura 9) conviene resaltar primero que dicho lenguaje de programación fue seleccionado por el investigador debido al hecho primordial de

que la herramienta principal de modelado y desarrollo (AppGini) trabaja exclusivamente con este lenguaje de programación, por lo que al hacer uso de dicho programa de desarrollo, se considera el punto actual inherente al mismo. Habiendo dejado en evidencia lo anterior, se procede a desglosar información de interés referente al termino PHP.

2.5.2 Origen de PHP.

Para comenzar a desglosar el significado y funcionamiento de PHP, es conveniente primero aclarar puntos importantes en cuanto a su origen y creación, para esos propósitos (Mariño, 2008) expone que “PHP es el heredero de un producto anterior, llamado PHP/FI. El cual fue creado por Rasmus Lerdorf en 1995 inicialmente como un conjunto de scripts de Perl para controlar los accesos a su trabajo online. Llamó a ese conjunto de scripts ‘Personal Home Page Tools’, según se requería más funcionalidad, se continuó escribiendo mayor implementación de C, que era capaz de comunicarse con bases de datos y permitía a los usuarios, desarrollar sencillas aplicaciones web dinámicas. Rasmus, eligió liberar el código de PHP/FI para que cualquiera pudiese utilizarlo, así como arreglar errores y mejorar el código.” así mismo el autor anterior expone que “PHP/FI, que se mantuvo para páginas personales y como intérprete de formularios, incluía algunas de las funcionalidades básicas de PHP tal y como lo conocemos hoy. Tenía variables como las de Perl, interpretación automática de variables de formulario y sintaxis embebida HTML (Hyper-Text Markup Language). La sintaxis por sí misma, era similar a la de Perl, aunque mucho más limitada, simple y algo inconsistente.”

Teniendo en cuenta lo anterior, podemos llegar a la conclusión de que PHP comenzó distante de su concepto actual, y a lo largo de los años fue atravesando una variado número de modificaciones e inserciones de códigos basados en otros lenguajes de programación con la finalidad de añadir funcionalidades de diversa índole de acuerdo a la visión del desarrollador en ese momento.

Agregando veracidad a la reflexión previa, (Mariño, 2008) explica también que: “Por 1997, PHP/FI 2.0, la segunda escritura de la implementación en C, tuvo un seguimiento estimado de varios miles de usuarios en todo el mundo, con aproximadamente 50.000 dominios informando que lo tenían instalado, sumando alrededor del 1% de los dominios de internet.

Mientras había mucha gente contribuyendo con bits de código a este proyecto, era todavía en su mayor parte el proyecto de una sola persona”, y luego, más adelante, agrega que: “PHP/FI 2.0 no se liberó oficialmente hasta noviembre de 1997, después de gastar la mayoría de su vida en desarrollos beta, fue sucedido en breve tiempo por las primeras versiones alfa de PHP 3.0”.

Basado en diversas investigaciones del tesista, se puede concluir que para cuando PHP 3.0 fue revelada, ya no contaba con la autoría del código del mismo Rasmus Lerdorf, pero era la primera versión fielmente parecida al concepto de PHP que se posee hoy en día, teniendo que atravesar para esto, una reescritura casi completa de su código, y con un largo camino por delante antes de poder considerarse un lenguaje de programación comercial

2.5.3 Definición de PHP.

En cuanto a definición se refiere el autor (García, 2007) afirma que: “PHP es un lenguaje de "código abierto" interpretado, de alto nivel, embebido en páginas HTML y ejecutado en el servidor. Es usado para la creación de aplicaciones para servidores, o creación de contenido dinámico para sitios web. Su interpretación y ejecución se da en el servidor web, en el cual se encuentra almacenado el script, y el cliente sólo recibe el resultado de la ejecución.”

De acuerdo con las afirmaciones anteriores entonces, se resume que PHP tiene la característica de código público, lo que proporciona libertad en su edición y uso a conveniencia por cualquier desarrollador en cualquier parte del mundo y al ser ejecutado por el servidor entra en la clasificación de lenguajes embebidos y de alto nivel en donde el cliente solo observa el resultado de la interpretación dada por el servidor web.

En cuanto a la compatibilidad y adaptabilidad se refiere (García, 2007) explica que PHP: “Otorga la Capacidad de conectarse de forma ágil y eficiente a diferentes servidores y gestores de base de dato. Por otro lado, posee una gran adaptabilidad al poder conectarse con un gran número de plataformas alrededor del mundo” Lo que, a criterio del investigador supone un plus en cuanto a niveles de compatibilidad se refiere y ahorro del factor tiempo por motivos de posibles errores y de adaptabilidad y compatibilidad al momento de la implantación.

2.5.4 Ventajas y puntos que considerar sobre PHP

Al momento de elegir un lenguaje de programación se considera que PHP fácilmente, se coloca sobre otros lenguajes, debido a los aspectos y prestaciones que ofrece, los cuales son explicados en la Tabla 8:

Tabla 8

Ventajas y Puntos a Considerar Sobre PHP

	PHP permite la implementación de códigos de seguridad mediante distintas librerías dedicadas especialmente al desarrollo de niveles de usuarios, encriptación de datos sensibles...etc.
Complejidad	En cuanto al nivel de conocimiento necesario para desarrollar utilizando este lenguaje de programación, PHP otorga la facilidad de que aquellos programadores que tengan conocimiento en lenguaje C y C++ puedan comenzar a utilizarlo rápidamente debido a las similitudes que guarda con ellos puesto que fueron parte de los lenguajes utilizados en sus tempranas versiones.
Estabilidad	Este lenguaje de programación se caracteriza por poseer soporte de una gran comunidad de desarrolladores a nivel mundial por lo que, si bien ningún lenguaje está libre de errores este en particular posee una singular capacidad de erradicación de estos debido al auge de personas trabajando diariamente en su desarrollo y mejoras, por otro lado PHP posee un sistema propio de gestión de Elementos y variables que le suponen los adjetivos de Surtido y estable.
Eficacia	PHP destaca por consumir muy pocos recursos del sistema, lo que evita los tiempos de espera excesivos en dispositivo, a su vez, cohabita muy bien con otros softwares, especialmente si estos poseen un núcleo de Unix.

Entre otras ventajas que este lenguaje de programación podría ofrecer se encuentran: se ejecuta en casi cualquier plataforma, posee un amplio soporte de interfaces de gestores de base de datos, se considera de alto rendimiento con un balance de muy bajo costo, posee la capacidad de extenderse mediante la instalación de diversas y surtidas librerías de contenido dependiendo de los requerimientos y preferencias de cada desarrollador.

2.5.4.1 *Funcionamiento de PHP.*

Cuando se define el comportamiento de este lenguaje de programación en específico se refiere. El tesista considera importante destacar que al ser un lenguaje script PHP (Ver Figura 10) trabaja fundamentalmente del lado del servidor, con su código insertado en archivos de naturaleza HTML que el servidor interpreta para luego mostrar únicamente el resultado del lado del cliente.

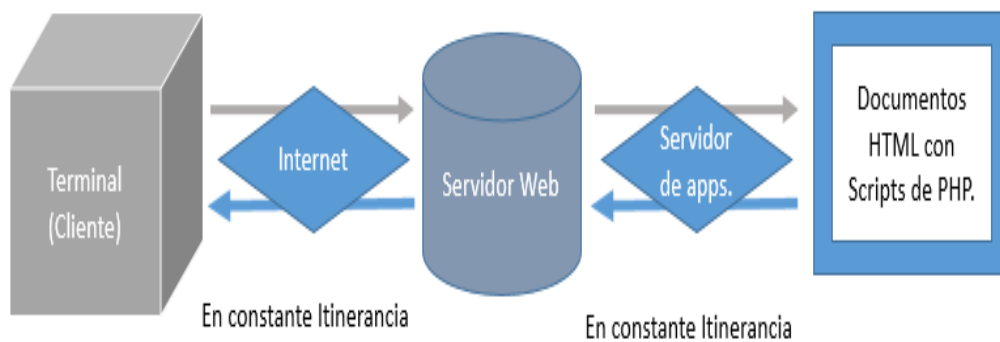


Figura 10 Funcionamiento de PHP

Visto lo anterior desde un enfoque más analítico, podríamos ejemplificar de manera más específica el funcionamiento de PHP y su consecuente interpretación de los scripts insertados en el HTML con la Figura 11 que hace referencia a lo mencionado:

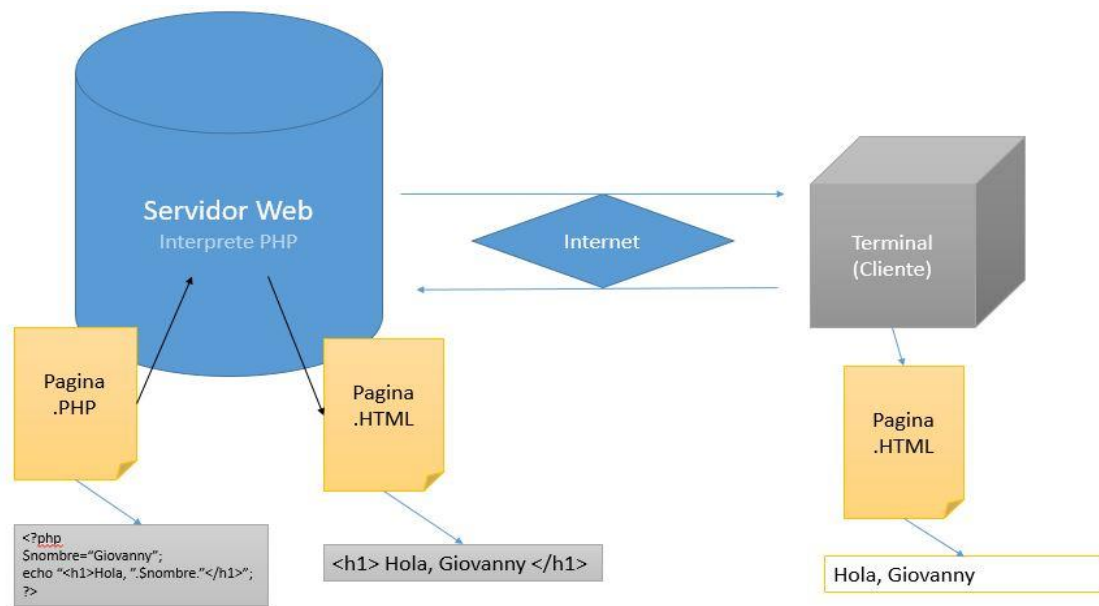


Figura 11 Interpretación mediante PHP

En el anterior gráfico, se evidencia la forma de trabajo de PHP, donde el Terminal (cliente) realiza una petición al servidor enviada mediante PHP, y obtiene en consecuencia una respuesta enviada desde PHP, que es recibida por el cliente ya pre procesado en HTML. Siendo el caso anterior el de la impresión de la variable “\$nombre”, que llega al cliente, como una simple cadena de texto con la variable impresa.

2.5.5 Git.



Figura 12 Logo de Git

Fuente: Git

De acuerdo con (Git, 2017) respecto a la aplicación que lleva su mismo nombre el aporta que: “lo que caracteriza a este programa de cualquier otro manejador de versiones, es la manera en la que Git (Ver Figura 12) almacena y gestiona la data puesto que mientras los demás

sistemas guardan la data repetidamente en forma de paquetes de archivos, pues estos sistemas (CVS, Subversion, Perforce, Bazaar, etc.) modelan la información que almacenan como un conjunto de archivos y las modificaciones hechas sobre cada uno de ellos a lo largo del tiempo.” Por otro lado, el mismo autor también expone que: “Git no estructura ni guarda sus datos de esta manera, sino que les cambia más en la forma en la que la memoria fotográfica funciona, convirtiéndole en una pequeña estructura de archivos y que se comporta de manera en la que cuando se realiza algún cambio o se le da la orden de un nuevo almacenamiento, literalmente se realiza una instantánea de toda la data almacenada en ese momento, y se almacena un enlace ellos, siempre cuidando que para la óptima utilización de los recursos del sistema, en un dado caso de que la data no presente ningún cambio no se almacena más nada sino que se sigue ofreciendo el enlace creado previamente.”

En ese orden de ideas, Git puede definirse como una aplicación que sirve como herramienta de control de versiones, es decir, que ayuda la manipulación del código durante el desarrollo controlando el almacenamiento de los cambios realizados en el proyecto y creando a su vez como una especie de instantáneas de cada una de las versiones que han sido almacenadas en él.

2.5.6 Características Generales de Git.

- Eficacia en la manipulación de las ramas, permitiendo el control real de los cambios mediante las advertencias de fusión que esta herramienta posee.
- Tanto las ramas locales como las secundarias pueden fusionarse.
- Facilita el control y el desarrollo óptimo de proyectos de gran tamaño.
- Permite un almacenamiento cíclico en modelo de paquetes de datos.

Finalmente, dicha herramienta de control de versiones fungió un papel de vital importancia a la hora del desarrollo del software con la finalidad de controlar que tanto del código se veía afectado en cada versión, y de esa manera disminuir los incidentes de corrupción que se podrían presentar por la alteración indiscriminada del código fuente.

2.5.7 BitBucket.



Figura 13 Logo de BitBucket

Fuente: Atlassian - BitBucket

Una vez descrito el tema anterior, se puede definir a BitBucket que en la Figura 13 se puede observar el logo, este es un servicio adicional a Git que proporciona la utilidad de alojamiento mediante la adquisición de planes comerciales y gratuitos de índole multiusuario, pero respetando la privacidad de cada cliente al no mostrar los repositorios dentro de su perfil si el usuario así lo desea.

2.5.8 PHPStorm



Figura 14 Logo de PHPStorm

Fuente: Google Imágenes

PHPStorm (Ver Figura 14) es un programa que facilita la redacción y creación de manera organizada de archivos que contienen sintaxis de lenguajes de programación en PHP y otros lenguajes afines con este, conocidos como scripts.

Así mismo, dicho editor de texto poseen entre sus características y herramientas motivos que le difieren de otros de sus rivales y que suponen razones por las cuales los desarrolladores le prefieren, como lo es, por ejemplo, el hecho de que posee resaltadores de colores para las diferentes variables, clases y funciones que la sintaxis de PhP contiene como

se puede visualizar en la Figura 15, así como también reconoce errores en la sintaxis de los procesos que se están desarrollando, a parte es famoso porque está desarrollado de manera que se utilice mediante combinaciones de teclado que agilizan el desarrollo pues evitan el uso de otros dispositivos periféricos como lo es el mouse .



Figura 15 Interfaz Inicial de PHPStorm

Fuente: Google Imágenes

2.6 METODOLOGÍA DE DESARROLLO SCRUM

2.6.1 Introducción.

La metodología utilizada en este trabajo de investigación fue la metodología Scrum. Para su entendimiento, es necesario puntualizar ciertos elementos de dicha metodología. Antes que nada, Scrum es una metodología basada en la satisfacción del cliente, ya que constituye una forma de trabajo en la que el feedback de comunicación continua con el cliente, es indispensable para el desarrollo del proyecto, garantizando así, la adaptación de cambios. De esta forma, agrega (Kniberg, 2007) que “scrum se enfoca en que el equipo debe presentar el producto potencialmente funcional y liberable al final de cada iteración, y que estas deben de ser de duración corta y fija”, de esta manera Kniberg, deja claro que scrum, es una metodología enfocada más en el desarrollo practico que teórico.

2.6.2 Principios de SCRUM.

En la metodología scrum, el equipo de trabajo se enfoca en construir software de calidad. La gestión para un proyecto hecho con la metodología scrum, debe estar centrado en definir y puntualizar cuáles son las características que debe tener el producto a construir (qué se va a construir, qué no se va a construir, además de en qué orden se va a construir) y en superar exitosamente cualquier obstáculo que pudiera presentarse y retrasar cualquier tarea del equipo de desarrollo.

2.6.3 Roles (Scrum Team).

El equipo de Scrum está formado por los siguientes roles:

- **Dueño del Producto (PO) Product Owner:** El cuál es el representante de los accionistas y clientes que usan el software. Se enfoca siempre en la parte de negocios y es el responsable del retorno de la inversión (parte lucrativa) del proyecto, es decir, entregar un monto superior al ya invertido. Este a su vez, traslada la visión del proyecto al equipo, tratando de ser lo más específico posible, formalizando así las prestaciones en historias a incorporar en el Product Backlog y re-priorizándolas de forma regular.
- **Equipo de Desarrollo (Team):** El equipo de desarrollo, está comprendido por todo aquel grupo de profesionales, técnicos con los conocimientos necesarios para llevar a cabo el desarrollo del proyecto, de manera conjunta, logrando llevar a cabo las historias a las que se comprometen al inicio de cada sprint, las cuales son dadas inicialmente por el Product Owner (PO).
- **Scrum Master:** Es la persona que lidera al equipo de desarrollo (Team) guiándolo a través del proyecto para que cumplan las reglas y procesos de la metodología. Este, además, gestiona la reducción de impedimentos y obstáculos del proyecto y trabaja conjuntamente con el Product Owner para maximizar el ROI (retorno de la inversión / ganancia).

2.6.4 Herramientas de Trabajo.

2.6.4.1 Fases.

Para definir las tareas de scrum, sus etapas, o fases (Schwaber, 2013) mencionan que la metodología scrum, se compone de cuatro eventos formales los cuales son detallados a continuación:

- **Reunión de planificación de Sprint (*Sprint Planning Meeting*):** en esta etapa, es donde se debe detallar el incremento resultante que se puede entregar del Sprint que inicia, además de mencionar y puntualizar que trabajo se va a realizar para entregar dicho incremento.
- **Scrum Diario (*Daily Scrum*):** En dicha etapa, se deben planificar y sincronizar las actividades en un plan estratégico para las siguientes 24 horas basadas en el último Scrum diario. En esta reunión, que debe realizarse a la misma hora y normalmente en el mismo lugar para evitar complejidad. Durante esta reunión, los miembros del equipo de desarrollo (team) deben explicar que tareas de las realizadas el día anterior, ayudaron a lograr el objetivo del Sprint, que actividades van a realizarse el presente día para cumplir el objetivo del Sprint y cuáles son los impedimentos visibles ante el cumplimiento del objetivo del Sprint. El Scrum diario sirve fundamentalmente para mejorar la comunicación entre el equipo, y es una reunión esencial de observación y adaptación.
- **Revisión del Sprint (*Sprint Review*):** En la etapa de revisión, el objetivo es obtener como resultado de la misma, una lista de productos posibles para el siguiente sprint, para ello, se analiza lo que se ha realizado durante el sprint actual y se detallan posibles mejoras a implementar, especificando los objetivos que se han alcanzado, así como los que no.
- **Retrospectiva de Sprint (*Sprint Retrospective*) :** Es la reunión que se debe ejecutar luego de la Revisión del Sprint (Sprint Review) y antes de que se ejecute la siguiente reunión de Planificación de Sprint (Sprint Planning), en esta reunión, el objetivo es crear un plan donde se plasmen las mejoras a implementar en el siguiente Sprint, detallando así las maneras para incrementar la calidad del producto final, para ello, se inspecciona las actividades del último Sprint en términos de personas, procesos, relaciones y herramientas.

2.6.5 Ciclo de Vida.

El proceso de desarrollo mediante Scrum se lo puede visualizar en la Figura 16 el cual realiza de forma iterativa e incremental, donde cada iteración, es denominada “sprint”, la cual tiene una duración preestablecida entre las 2 y 4 semanas, teniendo como objetivo dar como resultado una versión del software con nuevas características listas para ser usadas. A través de cada sprint, se va ajustando la funcionalidad ya construida y se añaden nuevas, priorizando siempre todas aquellas que aporten el mayor valor de negocio.

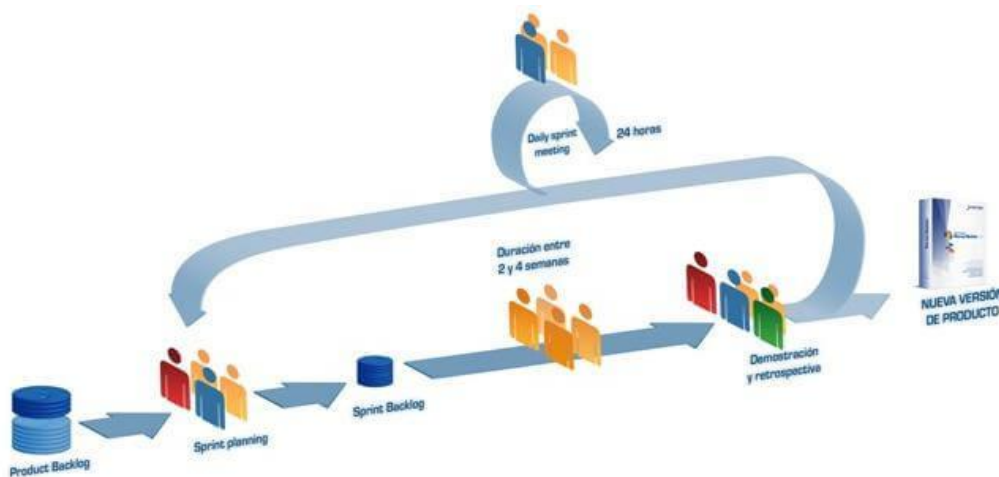


Figura 16 Ciclo de Vida Scrum

Fuente: Google Imágenes

- **Sprint:** Es un periodo de tiempo y medida fundamental dentro de la metodología scrum, que generalmente se encuentra comprendido en un mes de duración, en el que se debe crear una versión funcional del producto. Cada Sprint inicia a la culminación de un Sprint anterior, de manera inmediata. El Sprint contiene las etapas de Reunión de Planificación del Sprint (Sprint Planning Meeting), los Scrums Diarios también (Daily Scrums), el trabajo de desarrollo, la Revisión del Sprint (Sprint Review), y la Retrospectiva del Sprint (Sprint Retrospective).
- **Objetivos del Sprint:** este es creado en la Reunión de Planificación del Sprint (Sprint Planning Meeting), el objetivo del Sprint es la meta a ser alcanzada en la culminación

de cada Sprint, y dicho objetivo, fundamenta el propósito, el ¿por qué? del trabajo al equipo de desarrollo.

- **Lista de productos:** Es la lista de requerimientos que debe tener el producto una vez entregado totalmente al final de todos los sprint, dicha lista podrá irse modificando conforme los Sprint se terminen.

2.7 TRABAJOS RELACIONADOS

Entre los proyectos que el investigador ha considerado que guardan relación con el trabajo en actual desarrollo se encuentran:

- Tema: Center for Computational Thinking
- Autor: Jeanette M. Wing (Wing, 2006)
- Universidad: Carnegie Melon
- Año: 2012
- Resultados o conclusiones importantes: dicho proyecto tiene como propósito la difusión del conocimiento en relación al Pensamiento Computacional, Así como el suministro de información referente a los trabajos y antecedentes previos junto con los seminarios y cursos ofertados para la capacitación de su aplicación, es importante destacar que dicha página se encuentra apadrinada por el gigante de la tecnología Microsoft Corporation.
- Título: “El Pensamiento Computacional en la Enseñanza Obligatoria” (2017).
- Autor(es): Instituto Nacional de Tecnologías Educativas y Formación del Profesorado (INTEF) – Departamento de Proyectos Europeos
- Universidad: Universidad de Extremadura (España).
- Resultados o conclusiones importantes: En la sociedad educativa actual el concepto de Pensamiento Computacional supone un tema cuya investigación y aplicación se encuentra en constante auge, por ello, se busca orientar e informar a la colectividad de todos los temas relacionados y el tema central en general que permita su entendimiento a un nivel más profundo, y fundamentalmente que esclarezca las nociones de su aplicación no solamente para un ámbito profesional sino para su aplicación ni la vida.

CAPÍTULO III

ANÁLISIS, ARQUITECTURA Y DISEÑO DEL PROTOTIPO

2.8 ESPECIFICACIÓN DE REQUERIMIENTOS IEEE-830

2.8.1 Introducción.

En esta parte del documento se busca esclarecer todos los aspectos relacionados con los requerimientos y especificaciones del sistema planteado que lleva por nombre: DESARROLLO E IMPLANTACIÓN DE UN ECOSISTEMA INFORMÁTICO PARA LA ENSEÑANZA Y NORMALIZACIÓN DEL PENSAMIENTO COMPUTACIONAL PARA LA UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE UTILIZANDO LA HERRAMIENTA APPGINI, para ello el investigador se ha sustentado y orientado por el estándar IEEE830 la cual según (Castañeda Fuentes, p. 1): “es la sugerida para la documentación de la Especificación de Requerimientos IEEE Std. 830-1998”

2.8.2 Propósito.

La siguiente sección del documento tiene como finalidad definir la especificación de los requerimientos tanto funcionales como no funcionales que fueron planteados para el desarrollo del proyecto. Teniendo en cuenta todos los factores que interactúan con el proceso de producción, como lo es por ejemplo el del ecosistema informático que será utilizado por los alumnos y docentes de la Universidad de las Fuerzas Armadas ESPE.

2.8.3 Alcance.

La especificación de los requisitos posee un alcance enfocado en el estudio y evolución de los usuarios del sistema PECOMLMS, con ese propósito, las pruebas serán realizadas por un selecto grupo limitado grupo de estudiantes de primer nivel de tecnologías de la información quienes fungirán el rol de usuarios en una primera instancia logrando así la evaluación y emisión de criterios en lo que respecta a la funcionalidad del sistema en los distintos ámbitos que le conforman: (experiencia al navegar, funcionalidad, diseño...etc.).

En ese orden de ideas, el sistema informático tiene como propósito el cumplir con las siguientes actividades principales:

- El administrador tiene los privilegios de gestionar la totalidad de los aspectos de la aplicación, pudiendo al momento de realizar alguna modificación o cambio validar que la página se encuentre en mantenimiento o no; aparte tendrá acceso a bitácoras que indiquen los cambios creados o la data suministrada por los demás usuarios del sistema.
- El registro o adición de pruebas, las cuales tienen por finalidad ser aplicadas a cada uno de los alumnos, de manera que el profesor podrá tomar la libertad según considere conveniente la aplicación de pruebas de variadas naturalezas y estructuras.
- El estudiante podrá realizar un seguimiento continuo de todas las evaluaciones que le han sido aplicadas de acuerdo con cada una de sus asignaturas y analizar su desempeño en las mismas.

2.8.4 Definiciones, siglas y nomenclatura.

Tabla 8

Definición y Nomenclatura

USUARIO	Persona destinada a hacer uso del software planteado
ERS	Especificación de requerimientos de software
RF	Requerimiento funcional
RNF	Requerimiento no funcional
APPGINI	Herramienta de desarrollo web (Generador de código PHP)
SCRUM	Marco de desarrollo ágil

2.8.5 Personal involucrado.

Tabla 9

Detalles de Personal Involucrado

Nombre	Francisco Coral
Rol	Analista y programador
Categoría profesional	Estudiante de ingeniería en sistemas
Responsabilidad	Realizar la investigación pertinente, analizar la información recolectada y emitir criterios al respecto, diseñar y programar el sistema planteado
Información de contacto	francisco_coral@hotmail.com

2.8.6 Referencias.

Tabla 10

Referencias

TÍTULO	REFERENCIA
Std. IEEE 830 -1998	IEEE

2.8.7 Resumen.

El documento cuenta con tres principales tramos, el primero siendo introducción nos da una breve introducción en cuanto a la especificación de requisitos (ERS) la visión general de la utilidad de especificar los requisitos, las abreviaturas que se utilizaran a lo largo de la especificación.

En el segundo tramo se describe en forma general al sistema que se va a desarrollar, la funcionalidad del programa, los datos que se van a utilizar, las restricciones, los factores y dependencias.

El tercer tramo se define detalladamente los requisitos funcionales los cuales se encargan de satisfacer las funcionalidades del sistema y a su vez al cliente.

2.8.8 Descripción General.

Esta sección del capítulo de especificación de requerimientos trata sobre los factores que afectan en el desarrollo y progreso del software a realizar, pues trataremos acerca de los requisitos funcionales y no funcionales detallando cada uno de ellos.

De esta forma, los requisitos para la implementación del proyecto, a nivel de hardware y software, se encuentran bastante a favor, para brindar una excelente experiencia de usuario, y facilitar su desenvolvimiento dentro de la aplicación a la mayor cantidad de internautas interesados, ya que cuenta con unos requerimientos de hardware y software bastante comunes y presentes en la mayoría de dispositivos que se encuentran en circulación hoy en día.

Aunado a lo anteriormente expuesto, se expresa que el proyecto a implementar requiere por parte de los usuarios interesados, una conexión a internet de banda ancha de aproximadamente 512kbps de bajada, para el correcto funcionamiento del portal y la aplicación, lo cual corresponde al plan mínimo de internet en Ecuador, y supone un elemento bastante básico, presente en casi todos los hogares, oficinas e instituciones del Ecuador.

Asimismo, por parte del servidor de la aplicación, la aplicación se encuentra alojada en la plataforma basada en servicios web, AWS (Amazon Web Services), que ofrece de entre tantos servicios, los básicos para el funcionamiento de la aplicación, los cuales son Apache (HTTP), DBMS (MySQL), e Intérprete de PHP (PHP), además de ser rápida y eficiente para servir la aplicación del lado del cliente, es decir, internautas interesados.

2.8.9 Perspectiva del producto.

El aplicativo PECOMLMS es un sistema el cual trabajará en un entorno web, donde los módulos de hardware y software compatibles se juntarán colaborativamente respondiendo a los requisitos tanto funcionales como no funcionales del sistema, las interfaces de usuario deben estar sometidas al principio de consistencias para evitar confusiones en el momento de navegación.

Debido al auge que tiene en los últimos años, el concepto de pensamiento computacional, y su constante crecimiento, aporte, e implementación, cada vez en más países del mundo, se estima una recepción totalmente positiva por parte de los usuarios, ya que supone un monitoreo del crecimiento constante de esta filosofía para las personas y la posibilidad de dar un vistazo a todos sus grandes aportes a la sociedad, a través del porfolio electrónico que se implementa en conjunto con el LMS.

2.8.10 Interfaces del sistema.

Aquí se detallada a fondo los aspectos y características que posee el sistema planteado en lo que respecta a sus especificaciones de usuario que en la Figura 17 y Figura 18 se puede observar la interfaz del usuario administrador, de hardware, de software y de comunicación.



Figura 17 Interfaz del aplicativo (administrador)



Figura 18 Interfaz del aplicativo (administrador) 2

2.8.11 Interfaces del Usuario.

Al momento de definir la modalidad e interfaz de usuario del presente sistema, cabe destacar que ello se refiere a la itinerancia de datos que existe entre un computador o dispositivo y su utilidad, es decir, la conexión continua entre la parte física y lógica de una máquina. En ese orden de ideas, al momento de ingresar al sistema debe hacerse por medio de un navegador web, en el que, al acceder en una primera instancia, los usuarios tendrán acceso y privilegios de acuerdo al nivel de usuario en el que fueron registrados en la base de datos, esto mediante un nombre de usuario y contraseña; esta última que puede ser cambiada por parte del usuario si este lo considera conveniente.

Por otro lado, Para asegurar un acceso más perdurable a los datos que cada usuario almacena en el sistema, esta interfaz ofrece la posibilidad de indicarle al sistema que recuerde la contraseña y aunado a eso, existe también la posibilidad de iniciar un procedimiento para recuperarla en caso que el usuario la olvide, esto mediante una nueva ventana donde se es introducido su nombre de usuario o dirección de correo electrónico, todo lo anterior con la finalidad de generar una interfaz que se caracterice por ser amigable, intuitiva y de fácil uso para los usuario, de la mano con los requerimientos solicitados por el cliente y los aspectos que el desarrollador ha considerado importante.

2.8.12 Interfaces de hardware.

Cuando se habla de ‘hardware’ es importante destacar, que el hardware, son todos los dispositivos físicos que intervienen en este caso, en la funcionalidad del proyecto. Entendiéndose por hardware todos aquellos elementos físicos y tangibles que funcionan juntamente con el software, la parte intangible. El hardware es parte fundamental de la interacción digital en todos los aspectos, ya que fundamenta la principal interfaz mediante la que se accede al programa en este caso, el hardware viene dado por los elementos básicos tangibles de un computador, Tablet o teléfono inteligente, y son esos elementos que mantienen el óptimo funcionamiento del equipo, interactuando entre ellos de forma física, manteniendo el software en función.

De este modo, tomando en cuenta lo anteriormente expuesto, por parte del servidor web, se necesita un equipo para mantener la aplicación que cuente con al menos las siguientes características, las cuales son básicamente las mínimas presentes en todos los equipos de hoy en día.

- MEMORIA RAM 3GB
- PROCESADOR: Intel Celeron 1.60 GHz
- CACHE 3MB

Por parte del cliente, la aplicación puede ser consultada mediante básicamente cualquier equipo en circulación hoy en día, es una de las ventajas que también proporcionan las web apps, al ser tan livianas, rápidas y optimizadas. Permiten una fluidez en todos los dispositivos, sin pérdida de calidad del producto. En este sentido, para utilizar la aplicación en dispositivos del lado del cliente, en el caso de una PC, lo mínimo recomendado y es, además, lo presente en todos los equipos activos hoy en día, es:

- MEMORIA RAM 2GB
- PROCESADOR: Intel Celeron 1.60 GHz
- CACHE 2MB

2.8.13 Interfaces de software.

Se puede entender por 'software', toda la parte intangible del proyecto, es decir el sistema, y los requerimientos que a simple vista son invisibles e intangibles para las personas, el software forma parte fundamental de todo equipo, ya que se encuentra compuesto por todos los elementos de procesamiento lógico que se encargan de llevar la interacción del hardware a datos del sistema. De esta manera, se puede decir entonces que el software es la parte no tangible y menos perceptible físicamente del proyecto, a diferencia del hardware que además de ser dentro de lo que funciona el software, es también su contraparte, tangible y perceptible físicamente.

En el presente caso, dentro de los requerimientos que implementa el software del servidor para funcionar la aplicación del proyecto, se encuentran los siguientes:

- Apache 2.4.29
- MySQL 10.1.28
- PHP 5.6.32
- PhpMyAdmin 4.7.4
- XAMPP Control Panel 3.2.2
- Webalizer 2.23-04
- Mercury Mail Transport System 4.63
- Filezilla FTP server 0.9.41
- Tomcat 7.0.56
- Strawberry Perl 7.0.56 portable

2.8.14 Interfaces de comunicación.

Cuando conviene referirse a la interfaz de comunicación, cabe definir con anterioridad que esto se refiere a la modalidad de comunicación en intercambio de protocolos, data y procedimientos mediante el cual se comunica el dispositivo o computador con el servidor web.

Habiendo aclarado lo anterior, se procede a afirmar entonces que el sistema planteado se desarrolló para intercambiar protocolos y data con el servidor web mediante el conocido protocolo de control de transmisión también conocido como (TCP/IP) por sus siglas en inglés

desarrollado inicialmente por Vint Cerf y Robert Kahn. Y que tiene como propósito establecer un vínculo o conexión de intercambio de datos seguros entre distintos sistemas y aplicaciones mediante el envío de los datos exclusivamente a la IP o código único de identificación de red de cada dispositivo a la que se ha dirigido el paquete de datos, por otro lado es importante también resaltar que el requisito mínimo exigido para el correcto funcionamiento del sistema es un ancho de banda de 512 Mbps.

2.8.15 Restricciones de memoria.

Las restricciones de memoria son aquellos elementos que suponen si bien no un obstáculo, al menos un punto en contra para las tecnologías a implementar. Todo ello sin ser, determinadamente malo o llevando por mal camino el proyecto.

Por parte del usuario, las restricciones no representan un gran obstáculo, y vienen dadas por los factores siguientes:

- Se necesita internet para funcionar y acceder a la App
- Se necesita un navegador web para acceder a la App

Por otro lado, para el admin, desarrollador, en cuanto a restricciones se refiere, se pueden evidenciar las siguientes:

- PHP es aún un lenguaje en desarrollo por lo tanto tiene ciertas carencias.
- PHP al ser software libre tiene mucha documentación, pero posee un soporte netamente basado en el desarrollo de sus usuarios
- La herramienta AppGini tiene ciertos bugs o errores a la hora de regenerar el modelo.
- MySQL posee un cierto límite para la creación de tabla y el almacenamiento de los datos.

2.8.16 Operaciones.

Para el correcto funcionamiento del sistema se debe poseer una conexión a internet y se debe acceder a él mediante un navegador web de la preferencia del usuario, una vez dentro del sistema es posible observar los cambios que han sido realizados.

Esto se lo realiza mediante el acceso al módulo de administrador y mediante BitBucket se puede tener acceso a la última versión estable conocida y reestablecerla en caso de haber realizado algún cambio que alterara la estabilidad del sistema.

2.8.17 Funcionalidad del producto.

Conexión de la base de datos con el aplicativo

Este sistema al tener un entorno web se utiliza a través de internet maneja una base de datos en donde se aloja toda la información que el sistema maneja y produce y que a su vez permite el acceso a los usuarios mediante el alojamiento de nombres de usuario y contraseña que permiten acceder mediante el cotejamiento previo con dicha base de datos, pero limitando la información dependiendo el rol que cumpla dentro de la universidad.

2.8.18 Características del usuario.

En la Tabla 11 se puede visualizar las características por cada tipo de usuario:

Tabla 11

Características de los usuarios

Administrador General	El grupo de innovación y educación. Quien maneja la plataforma de InnovaEdu
Usuario (Profesor)	Interactúa con el sistema, realizando la gestión para la publicación de preguntas a su grupo de alumnos
Usuario (Alumno)	Interactúa con el sistema, visualiza información y responde a peticiones

2.8.19 Restricciones del sistema.

- El sistema no será accesible a cualquier usuario web, pues debe poseer una contraseña con su respectivo perfil de acceso para poder tener la información correspondiente a los permisos asignados

2.8.20 Requisitos Específicos.

- **REQ001:** Base de datos para poder almacenar la información que se requiere y de la misma manera generar los resultados que se necesitan.
- **REQ002:** El sistema debe permitir el control de acceso al sistema, haciendo uso de perfiles de usuario el cual tendrá nombre de usuario y clave de usuario. Esto evitará que los usuarios tengan permisos que no correspondan a su perfil.
- **REQ003:** El sistema debe permitir la administración de los usuarios creados para el uso de la aplicación. En él se podrá verificar los estados de los usuarios, las actualizaciones y las actividades que cada usuario realice; así como el ingreso, modificación, edición y lectura de los mismos.
- **REQ004:** El sistema debe permitir la administración de grupos, en él se podrá agregar, modificar, eliminar y observar; así también conocer las actividades de cada grupo.
- **REQ005:** El sistema debe permitir la creación de perfiles de alumnos, los cuales podrán acceder al sistema bajo su nivel de permisos concedidos por el administrador.
- **REQ006:** El sistema debe permitir matrículas de alumnos a las diferentes materias que existen. Esto se podrá realizar bajo el perfil de administrador y profesor. todo eso con la implementación de usuarios según OWASP.
- **REQ007:** El sistema debe permitir realizar comentarios con cualquier perfil de usuario, pero estos tendrán diferentes perspectivas de verlos. Estos comentarios servirán para retroalimentación del sistema y así pueda ir mejorando con el paso de los años.
- **REQ008:** El sistema debe permitir realizar preguntas de diferentes tipos como selección múltiple, verdadera o falsa, preguntas abiertas; para la posible realización de pruebas, con la implementación de usuarios según OWASP.

- **REQ009:** El sistema debe permitir realizar respuestas las cuales serán asignadas a las preguntas para realizar una posible prueba, con la implementación de usuarios según OWASP
- **REQ010:** El sistema genera los siguientes reportes:
 - Se mostrará el porcentaje y número de alumnos por cada tipo de aprendizaje en forma global de todos los usuarios existentes.
 - Se mostrará el histórico de calificaciones de una prueba (previamente filtrada por materia).
 - Se mostrará nuevos registros por parte de los usuarios del sistema, lista de los usuarios a modo general, estadísticas de usuarios (cuantos, activos, bloqueados, número de grupos)
 - Estadística de cual o cuales son los usuarios (todos los roles) que interactúan más con el sistema
 - Estadísticas de cumplimiento de Talleres
 - Estadísticas de cumplimiento de Wikis
 - Estadísticas de cumplimiento de Ensayos
 - Se mostrará el histórico de calificaciones de una prueba (previamente filtrada por materia).
 - Progreso del estudiante se verá la relación entre la prueba inicial de una materia y su evaluación final; se verificará progreso o no progreso

2.8.21 Clasificación de requerimientos funcionales.

Funcionalidad

- **REQ1:** Construir la base de datos, la aplicación no permitirá hacer algún cambio en la estructura de la base de datos, este requerimiento es esencial no se alterará ni eliminará la base de datos. Requerimiento prioritario
- **REQ2:** Ingresar al sistema, bajo los perfiles de usuario el cual tendrá nombre de usuario y clave. Esto evitará que los usuarios se den permisos y solo tendrán acceso a los permisos que correspondan a su perfil. Requerimiento prioritario
- **REQ3:** Administrar usuarios creados para el uso de la aplicación. En él se podrá verificar los estados de los usuarios, las actualizaciones y las actividades que cada usuario realice, se puede enviar mensajes de correo electrónico. Requerimiento prioritario

- **REQ4:** Administrar grupos creador para el uso de la aplicación. En él se podrá verificar las actividades, descripción de cada grupo. Requerimiento prioritario.
- **REQ5:** Administrar alumnos, creación de perfiles de alumnos los cuales podrán acceder al sistema bajo su nivel de permisos concedidos por el administrador. Requerimiento prioritario
- **REQ6:** Administrar matrículas, permitir matrículas de alumnos a las diferentes materias que existen. Esto se podrá realizar bajo el perfil de administrador y profesor. todo eso con la implementación de usuarios según OWASP. Requerimiento prioritario
- **REQ7:** Administrar cometarios, permitir realizar comentarios con cualquier perfil de usuario. Estos comentarios servirán para retroalimentación del sistema y así pueda ir mejorando con el paso de los años. Requerimiento esencial.
- **REQ8:** Administrar preguntas, permitir realizar preguntas de diferentes tipos como selección múltiple, verdadera o falsa, preguntas abiertas; para la posible realización de pruebas, con la implementación de usuarios según OWASP. Requerimiento prioritario
- **REQ9:** Administrar respuestas, permitir realizar respuestas las cuales serán asignadas a las preguntas para realizar una posible prueba, con la implementación de usuarios según OWASP
- **REQ10:** Administrar reportes. Requerimiento prioritario

2.8.22 Requerimientos no funcionales

2.8.23 Propósito.

En esta parte del documento se define y describe las operaciones, interfaces, rendimiento y requerimientos externos que necesita el sistema para ser de calidad.

2.8.24 Alcance.

Se definen los requerimientos no funcionales para tener un sistema confiable, estable, robusto, compatible, usabilidad y de esta manera hacer factible su uso.

2.8.25 Referencias.

- Especificación de requerimientos
- IEEE

2.8.26 Suposiciones y dependencias.

El sistema propuesto pretende facilitar el proceso de aprendizaje de los usuarios mediante la aplicación que un test que permita el diagnóstico del tipo de aprendizaje propicio para cada individuo y así lograr facilitar el desenvolvimiento académico de los estudiantes, para ello, también pretende otorgar la posibilidad a los profesores de diseñar pruebas académicas y matricular al grupo deseado para que las tome y por otro lado permite evaluar la veracidad y el impacto de la aplicación del pensamiento computacional en base a diferentes estadísticas que muestran el porcentaje de alumnos que ha tomado la prueba y el porcentaje que las ha aprobado.

2.8.27 Usabilidad.

Para la facilidad de navegación por parte del usuario se ha realizado una interfaz pensando en que su diseño sea de ayuda en el uso de la aplicación y así el usuario a utilizar no requiera de entrenamiento previo al uso.

2.8.28 Fiabilidad.

El sistema debe estar disponible para los usuarios las 24 horas a día los 7 días de la semana.

2.8.29 Acceso.

El sistema tendrá un acceso bajo la norma de perfiles de usuario con 100% de confiabilidad.

2.8.30 Rendimiento.

En cuanto a los tiempos de respuesta:

- El sistema no tardará más de 3 segundos en presentar su interfaz principal al usuario, en caso de tardar se deberá tener en cuenta la conexión a internet.
- En el menú principal se podrá realizar las búsquedas para organizar alguna tarea requerida. Dentro de cada actividad se registrará la actividad a realizar.

2.8.31 Capacidad.

Al ser un sistema escalable tendrá toda la libertad de acoplarse a las necesidades del de InnovaEdu.

2.8.32 Compatibilidad.

El sistema al ser desarrollado para la web será compatible con cualquier navegador.

2.8.33 Mantenimiento.

Dependerá absolutamente del administrador de la aplicación y del contrato que se maneje con los mismos.

2.8.34 Diseño de restricciones.

- Plataforma: para el cliente es necesario un computador con procesador de 1 Ghz o mayor con un mínimo de 50 Mb de espacio en disco y 1 Gb de RAM.
- Navegadores: podrá ser usado en cualquier navegador pues se encuentra orientado para la web

2.9 ESPECIFICACIÓN DE CASOS DE USO

2.9.1 Introducción.

Este capítulo se basa en entender el problema, conocer más a fondo quienes intervienen y cuáles son los requerimientos funcionales, para así saber cómo se va desarrollando la

aplicación. Al final si todos los requisitos se cumplen es porque el sistema está listo para entregarse.

2.9.2 Descripción general de los actores.

- Administrador: puede ser el administrador de dirección, administrador de innova o a quien se designe el cargo.
- Usuario: persona quien podrá revisar notificaciones, interactuar con las actividades planificadas en el sistema.
- Sistema: aplicación educativa para INNOVAEDU ESPE
- Base de Datos: repositorio donde se almacenarán los datos referentes a la aplicación.

2.9.3 Caso de uso: Ingreso al sistema.

En la Tabla 12 se encuentra descrito el caso de uso de ingreso al sistema y con su respectivo diagrama que se encuentra en la Figura 19.

Tabla 12

Caso de Uso I: Ingreso al Sistema

Caso de uso	Ingresar al Sistema
Id. Requerimiento	REQ2
Actor	Administrador, Usuario
Descripción	Mediante nombre de usuario y contraseña de acceso. El usuario se autenticará en la interfaz de Ingreso, el aplicativo identifica los perfiles que maneja el usuario. Ingresa al aplicativo solamente con acceso a los módulos que su perfil de usuario le permite.
Entradas	Nombre de usuario y contraseña

CONTINÚA 

Salidas	Ingreso al sistema
Proceso	<ol style="list-style-type: none"> 1- Ingreso del nombre de usuario 2- Ingreso de la contraseña de usuario 3- Valida los dos campos ingresados 4- Inicio de sesión del usuario 5- Muestra la página inicial de la aplicación
Precondiciones	<ul style="list-style-type: none"> • Tener un nombre de usuario habilitado • Tener una contraseña de usuario
Pos condiciones	NA
Efectos colaterales	NA
Prioridad	Alta
Excepciones	

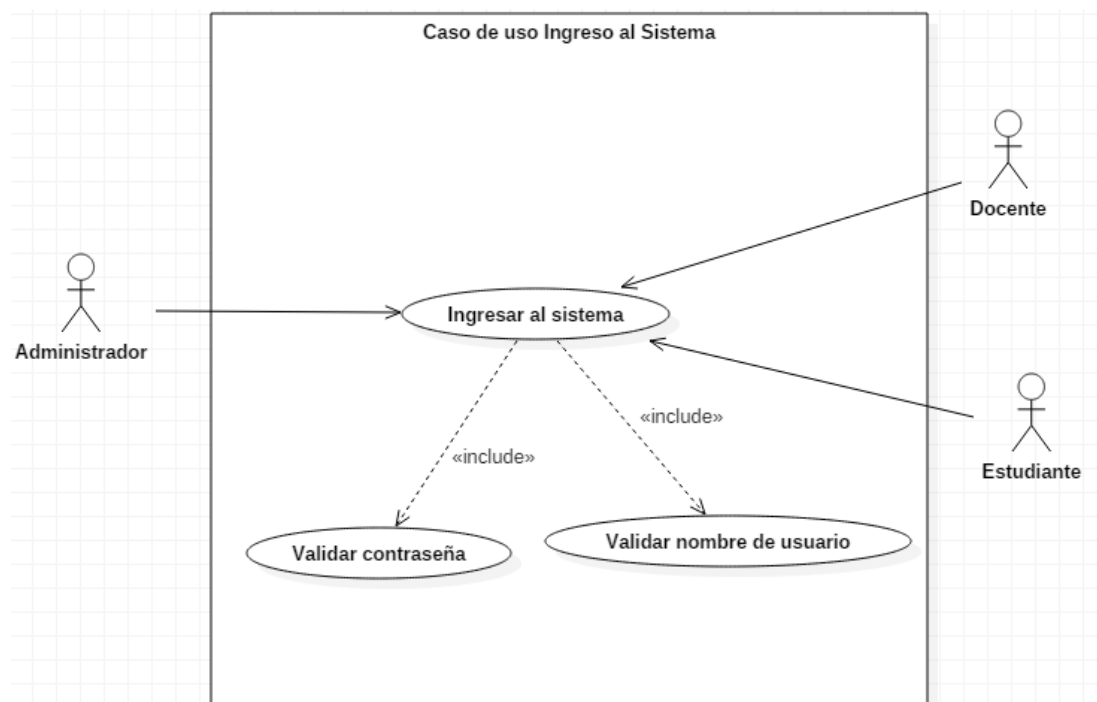


Figura 19 Caso de Uso I: Ingreso al Sistema

2.9.4 Caso de uso II: Administración de usuarios.

En la Tabla 13 se encuentra descrito el caso de uso de administración de usuarios y con su respectivo diagrama que se encuentra en la Figura 20.

Tabla 13

Caso de Uso II: Administrador de Usuario

Caso de uso	Administración de usuarios
Id. Requerimiento	REQ3
Actor	Administrador
Descripción	El administrador después de ingresar al sistema puede señalar en usuarios y puede seleccionar que desea hacer con ellos.
Entradas	Datos de los usuarios
Salidas	Almacenamiento de datos en la base de datos

CONTINÚA →

Proceso	<ol style="list-style-type: none"> 1. Selecciona de la opción que desea realizar: <ul style="list-style-type: none"> • Ver usuarios • Agregar usuarios • Ver registros de Usuarios 2. Al ver los usuarios podemos conocer los datos que posee cada uno de ellos. 3. Podremos buscar un usuario específico. 4. Podemos eliminar y resetear a los usuarios que deseemos. 5. Podremos agregar más usuarios, es necesario llenar los campos especificados. 6. Al seleccionar Ver registro de usuarios podremos ver los usuarios registrados por sí mismo.
Precondiciones	Estar en el perfil de administrador
Pos condiciones	NA
Efectos colaterales	NA
Prioridad	Media
Excepciones	

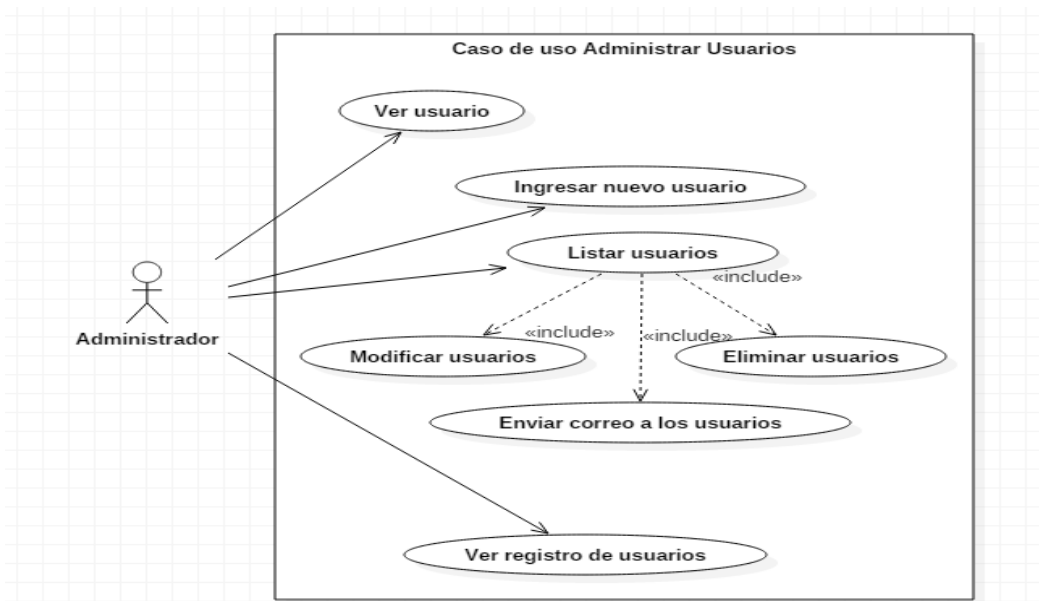


Figura 20 Caso de uso II: Administrador de Usuarios

2.9.5 Caso de uso III: Administrador de grupos.

En la Tabla 14 se encuentra descrito el caso de uso de administración de grupos y con su respectivo diagrama que se encuentra en la Figura 21.

Tabla 14

Caso de Uso III: Administrador de Grupos

Caso de uso	Administración de grupos
Id. Requerimiento	REQ4
Actor	Administrador
Descripción	El administrador después de ingresar al sistema puede señalar en grupos y puede seleccionar que desea hacer con ellos.
Entradas	Datos de los grupos
Salidas	Almacenamiento de datos en la base de datos
Proceso	<ol style="list-style-type: none"> 1. Selecciona de la opción que desea realizar: <ul style="list-style-type: none"> • Ver grupos • Agregar grupo • Editar permisos anónimos 2. Al ver los grupos podemos conocer los datos que posee en cada uno de ellos. 3. Podremos buscar un grupo en específico. 4. Podemos eliminar y resetear a los grupos que deseemos. 5. Podremos agregar más grupos, es necesario llenar los campos especificados. 6. Al seleccionar Editar tiene todos los permisos.
Precondiciones	Estar en el perfil de administrador
Pos condiciones	NA
Efectos colaterales	NA
Prioridad	Media
Excepciones	

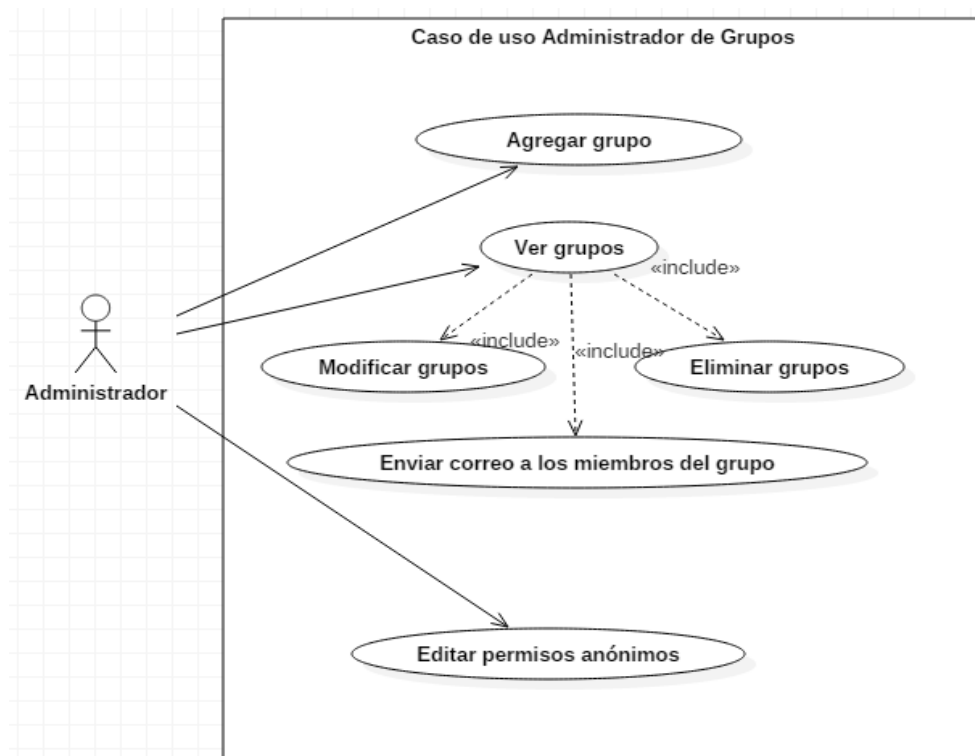


Figura 21 Caso de uso III: Administrador de Grupos

2.9.6 Caso de uso IV: Cambio de contraseña.

En la Tabla 15 se encuentra descrito el caso de uso de cambio de contraseña.

Tabla 15

Caso de Uso IV: Cambio de Contraseña

Caso de uso	Cambio de Contraseña
Id. Requerimiento	REQ5
Actor	Administrador
Descripción	El administrador después de ingresar al sistema puede cambiar su contraseña y editar los campos que cree necesario para mantener una seguridad en su perfil.
Entradas	Datos del administrador
Salidas	Mensaje de configuración: Configuración de administrador guardada correctamente.

CONTINÚA →

Regresar a Configuración de Administrador	
Proceso	<ol style="list-style-type: none"> 1. Seleccionar herramientas 2. Ir a configuración del administrador 3. Cambiar la contraseña o nombre del administrador 4. Editar los cambios que desee. 5. Guardar cabios realizados. 6. Llegada de correo al correo del administrador.
Precondiciones	Estar en el perfil de administrador
Pos condiciones	NA
Efectos colaterales	NA
Prioridad	Media
Excepciones	

2.9.7 Caso de Uso V: Administrar Estudiantes.

En la Tabla 16 se encuentra descrito el caso de uso de la administración de estudiantes con su respectivo diagrama que se visualiza en la Figura 22.

Tabla 16

Caso de Uso V: Administrar Estudiantes

Caso de uso	Administrar Estudiantes
Id. Requerimiento	REQ5
Actor	Administrador
Descripción	El administrador puede crear perfiles de estudiantes y administrar cada uno de ellos.
Entradas	Datos de los alumnos
Salidas	
Proceso	<ol style="list-style-type: none"> 7. Seleccionar Estudiantes dentro de la primera pantalla de presentación. 8. Luego selecciona Estudiantes en el despliegue de la información. 9. En Añadir podemos ingresar un nuevo registro de Estudiante.

CONTINÚA 

	<p>10. Si seleccionamos en uno de los alumnos, se nos desplegara una nueva ventana.</p> <p>11. En ella podemos, eliminar, editar o hacer una copia del registro.</p> <p>12. Guardamos la información.</p>
Precondiciones	Estar en el perfil de administrador
Pos condiciones	NA
Efectos colaterales	NA
Prioridad	Media
Excepciones	

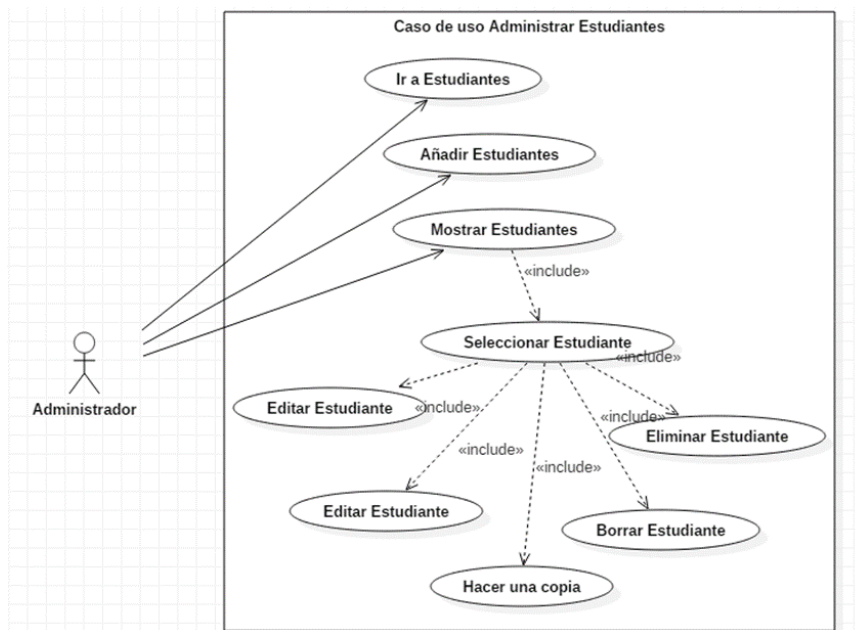


Figura 22 Caso de Uso V: Administrar Estudiantes

2.9.8 Caso de Uso VI: Administrar Materias.

En la Tabla 17 se encuentra descrito el caso de uso de la administración de materias con su respectivo diagrama que se visualiza en la Figura 23.

Tabla 17

Caso de Uso VI: Administrar Materias

Caso de uso	Administrar Materias
Id. Requerimiento	REQ6
Actor	Administrador
Descripción	El administrador puede gestionar las materias que desee.
Entradas	Datos de la materia
Salidas	
Proceso	<ol style="list-style-type: none"> 1. Seleccionar Materias dentro de la primera pantalla de presentación. 2. Luego selecciona Materia en el despliegue de la información. 3. En Añadir podemos ingresar un nuevo registro de Materia. 4. Ponemos un nombre a la materia y la descripción. 5. Guardamos la información correspondiente. 6. Al seleccionar una materia ya creada podemos editar, eliminar o hacer una copia. 7. Guardamos la información y verificamos.
Precondiciones	Estar en el perfil de administrador
Pos condiciones	NA
Efectos colaterales	NA
Prioridad	Media
Excepciones	

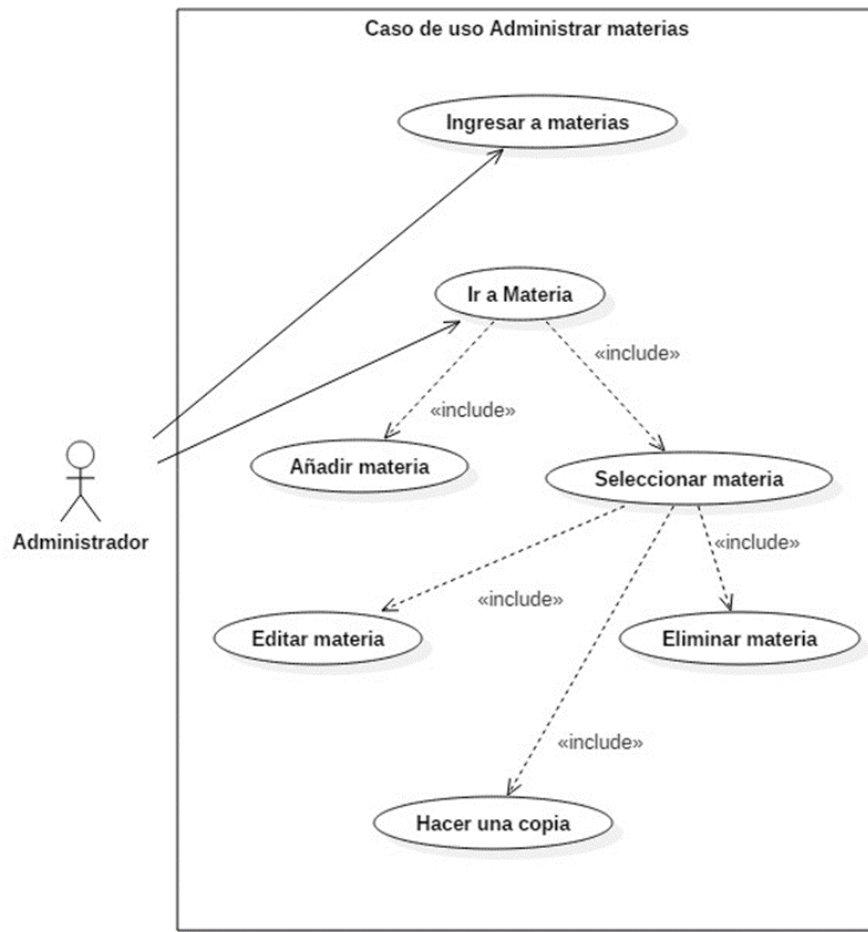


Figura 23 Caso de Uso VI: Administrar Materias

2.9.9 Caso de Uso VII: Administrar Matrículas.

En la Tabla 18 se encuentra descrito el caso de uso de la administración de matrículas con su respectivo diagrama que se visualiza en la Figura 24.

Tabla 18

Caso de Uso VII: Administrar Matrículas

Caso de uso	Administrar Matriculas
Id. Requerimiento	REQ7
Actor	Administrador

CONTINÚA →

Descripción	El administrador puede administrar las matrículas de los estudiantes o asignar pruebas a los mismos.
Entradas	Datos de la matricula
Salidas	
Proceso	<ol style="list-style-type: none"> 1. Seleccionar Matriculas dentro de la primera pantalla de presentación. 2. Luego selecciona Matriculas en el despliegue de la información. 3. En Añadir podemos ingresar un nuevo registro de Matricula. 4. Seleccionamos la materia y el alumno a quien queremos matricular. 5. Grabamos la información y observamos los detalles. 6. Dentro de matrículas también tenemos Asignación de P-E. 7. Podemos asignar las pruebas a los estudiantes. 8. Grabamos la información deseada y observamos el despliegue.
Precondiciones	Estar en el perfil de administrador
Pos condiciones	NA
Efectos colaterales	NA
Prioridad	Media
Excepciones	

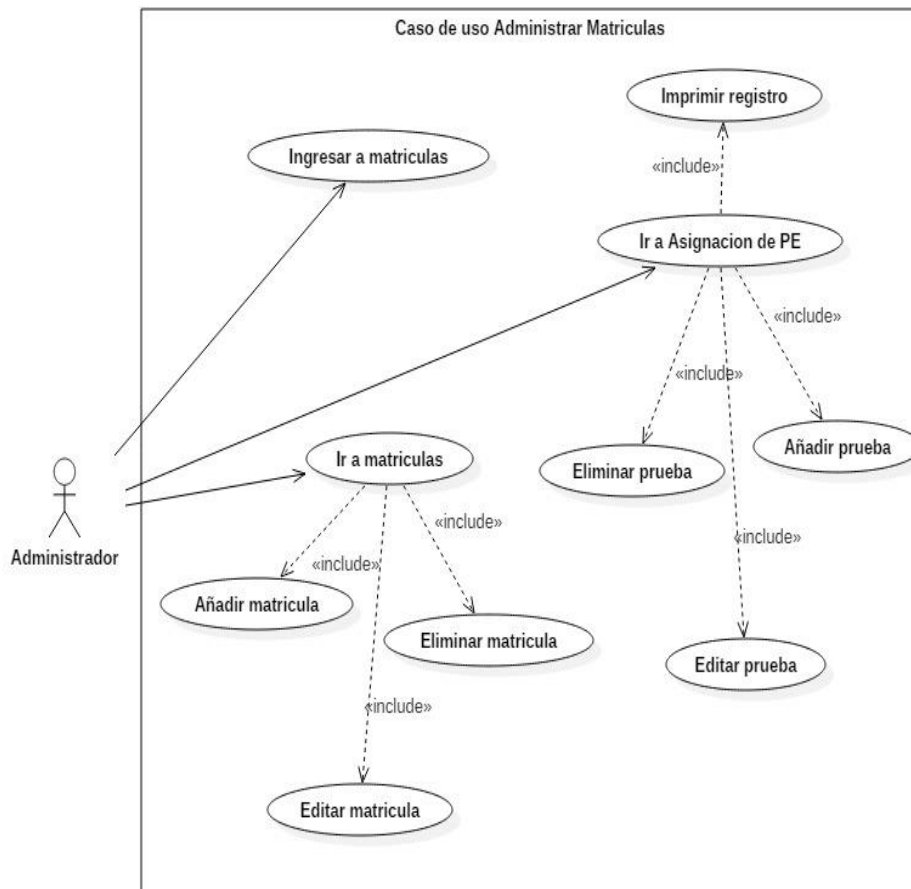


Figura 24 Caso de Uso VII: Administrar Matrículas

2.9.10 Caso de Uso VIII: Administrar Comentarios.

En la Tabla 19 se encuentra descrito el caso de uso de la administración de comentarios con su respectivo diagrama que se visualiza en la Figura 25.

Tabla 19

Caso de Uso VIII: Administrar Comentarios

Caso de uso	Administrar Comentarios
Id. Requerimiento	REQ8
Actor	Administrador, estudiantes, profesores
Descripción	Los usuarios pueden gestionar los comentarios de cada uno de ellos, estos comentarios se guardarán para que sean observables por el administrador.

CONTINÚA

Entradas	Comentarios de los usuarios
Salidas	
Proceso	<ol style="list-style-type: none"> 1. Seleccionar Comentarios dentro de la primera pantalla de presentación. 2. Luego selecciona Comentario en el despliegue de la información. 3. En Añadir podemos ingresar un nuevo Comentario. 4. Seleccionamos un comentario y podemos editar, eliminar o hacer una copia. 5. Grabamos la información y observamos los detalles.
Precondiciones	Estar en el perfil de usuario
Pos condiciones	NA
Efectos colaterales	NA
Prioridad	Media
Excepciones	

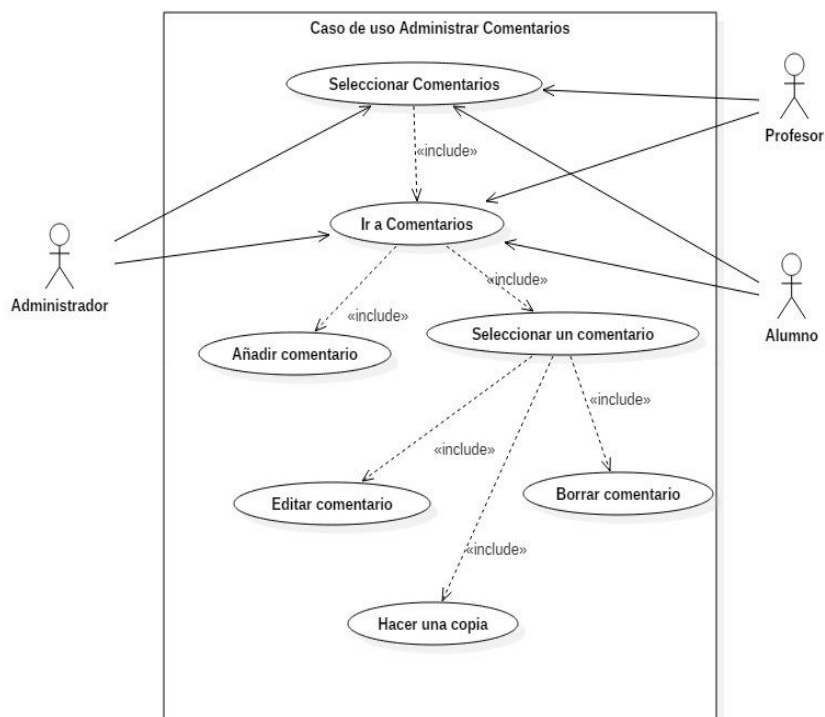


Figura 25 Caso de Uso VIII: Administrar Comentarios

2.9.11 Caso de Uso IIX: Administrar Pruebas.

En la Tabla 20 se encuentra descrito el caso de uso de la administración de pruebas con su respectivo diagrama que se visualiza en la Figura 26 y en la Figura 27 se muestra un diagrama del caso de uso: prueba y en la Figura 28 muestra la respuesta.

Tabla 20

Caso de Uso IIX: Administrar Pruebas

Caso de uso	Administrar pruebas
Id. Requerimiento	REQ9
Actor	Administrador, profesores
Descripción	El administrador y los profesores pueden gestionar las pruebas para los alumnos. Esta actividad va subdividida en diferentes gestiones.
Entradas	Datos para las pruebas
Salidas	
Proceso	<ol style="list-style-type: none"> 1. Seleccionar Pruebas dentro de la primera pantalla de presentación. 2. Luego selecciona Prueba en el despliegue de la información. 3. En Añadir podemos ingresar una nueva prueba <ol style="list-style-type: none"> 3.1. Seleccionamos la materia 3.2. Ponemos un nombre a la prueba 3.3. Colocamos una descripción 4. Seleccionamos pregunta <ol style="list-style-type: none"> 4.1. Añadimos nueva pregunta 4.2. Seleccionamos una pregunta específica y podemos editar, eliminar o hacer una copia. 4.3. Grabamos la información y observamos. 5. Tipos de preguntas <ol style="list-style-type: none"> 5.1. Añadimos el tipo de pregunta 5.2. Ponemos la descripción. 5.3. Podemos editar, eliminar o hacer una copia. 6. Seleccionamos respuesta <ol style="list-style-type: none"> 6.1. Añadimos nuevas respuestas dependiendo de la pregunta que ya hemos llenado

CONTINÚA 

	<p>6.2. Seleccionando una respuesta en particular podemos editar, eliminar, hacer una copia.</p> <p>6.3. Grabamos la información</p> <p>7. Asignación de preguntas</p> <p>7.1. Añadimos una asignación</p> <p>7.2. Seleccionamos la materia</p> <p>7.3. Seleccionamos la prueba</p> <p>7.4. Seleccionamos la pregunta</p> <p>8. Grabamos la información y observamos los detalles.</p>
Precondiciones	Estar en el perfil de usuario
Pos condiciones	NA
Efectos colaterales	NA
Prioridad	Media
Excepciones	

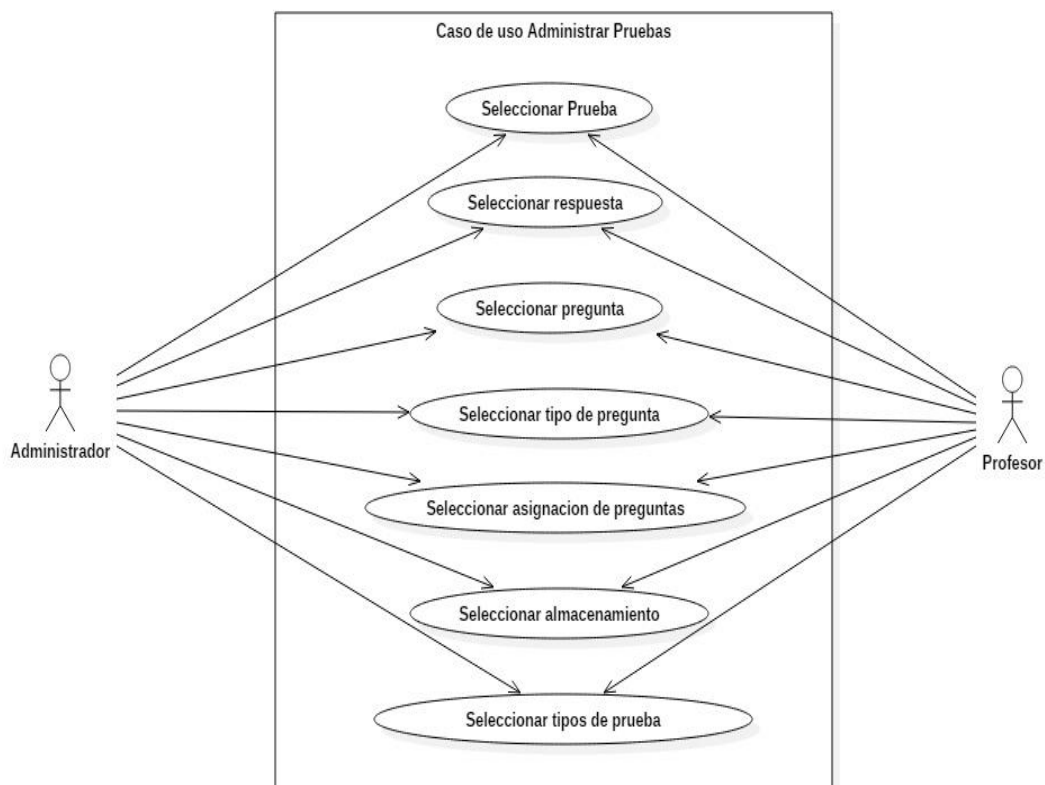


Figura 26 Caso de Uso IIX: Administrar Pruebas

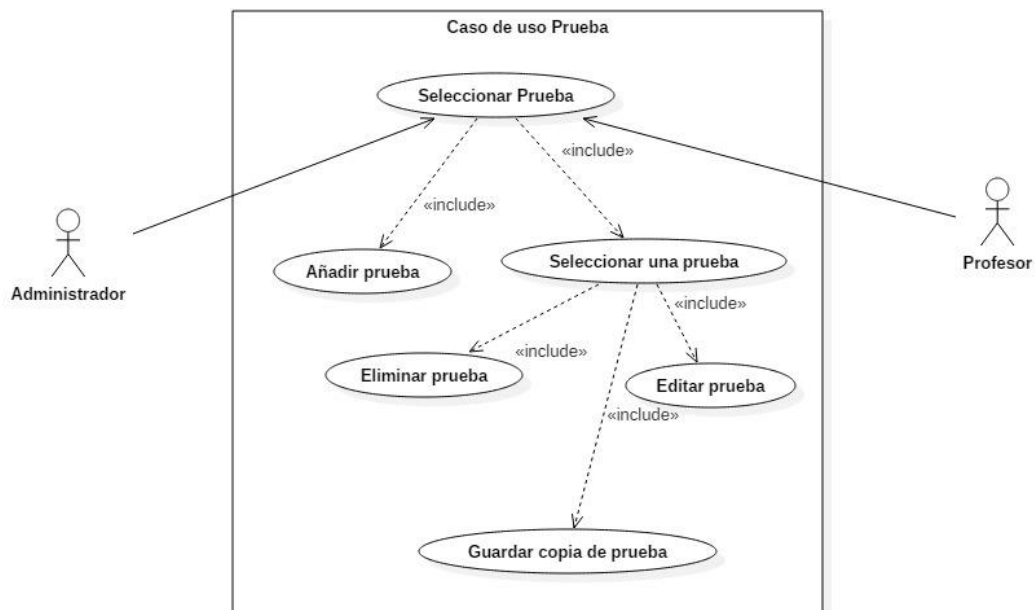


Figura 27 Caso de Uso IIX: Prueba

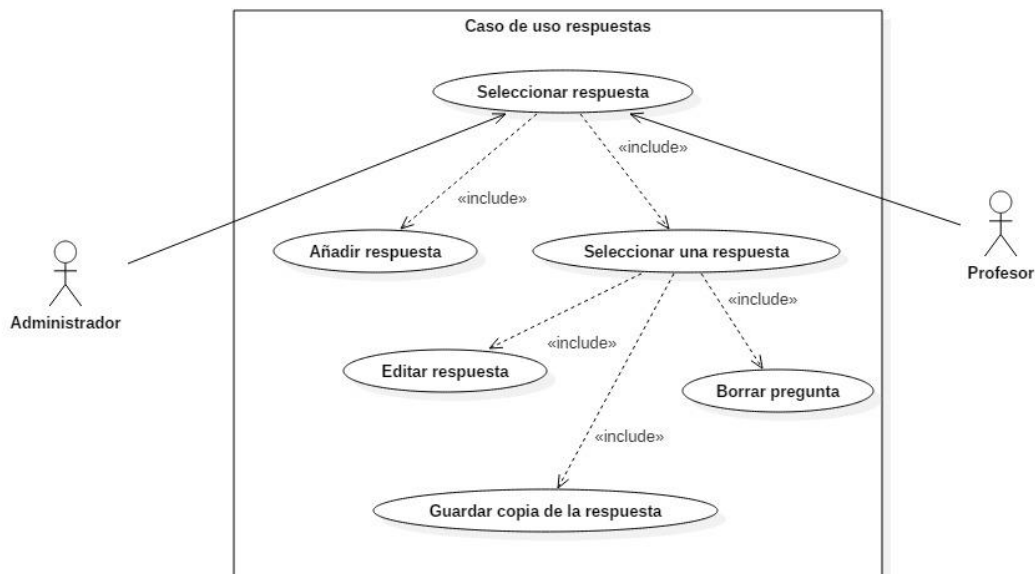


Figura 28 Caso de Uso IIX: Respuestas

2.9.12 Caso de Uso IX: Administrar Talleres.

En la Tabla 21 se encuentra descrito el caso de uso de la administración de talleres con su respectivo diagrama que se visualiza en la Figura 29, de la misma manera en la Figura 30 se muestra de manera más detallada.

Tabla 21

Caso de Uso IX: Administrar Talleres

Caso de uso	Administrar talleres
Id. Requerimiento	REQ10
Actor	Administrador, profesores
Descripción	El administrador y los profesores pueden gestionar los talleres para los alumnos. Esta actividad va subdividida en diferentes gestiones.
Entradas	Datos para los talleres
Salidas	
Proceso	<ol style="list-style-type: none"> 1. Seleccionar Talleres dentro de la primera pantalla de presentación. 2. Luego selecciona Taller en el despliegue de la información. 3. En Añadir podemos ingresar un nuevo taller <ol style="list-style-type: none"> a. Seleccionamos la materia b. Ponemos la fecha de vencimiento del taller c. Colocamos el tema del Taller d. Damos una descripción al Taller e. Seleccionamos algún archivo f. Y colocamos la hora de finalización del Taller 4. Seleccionamos algún Taller <ol style="list-style-type: none"> a. Dentro de él podemos editar, eliminar, hacer una copia. b. Grabamos la información y observamos. 5. Al seleccionar Taller <ol style="list-style-type: none"> a. Observamos las respuestas de los talleres establecidos 6. En Taller - Estudiante

CONTINÚA 

	<ul style="list-style-type: none"> a. Asignamos los talleres a los estudiantes b. Podemos añadir c. Editar, eliminar, grabar una copia del taller <p>7. Grabamos la información y observamos los detalles.</p>
Precondiciones	Estar en el perfil de administrador o profesor
Pos condiciones	NA
Efectos colaterales	NA
Prioridad	Media
Excepciones	

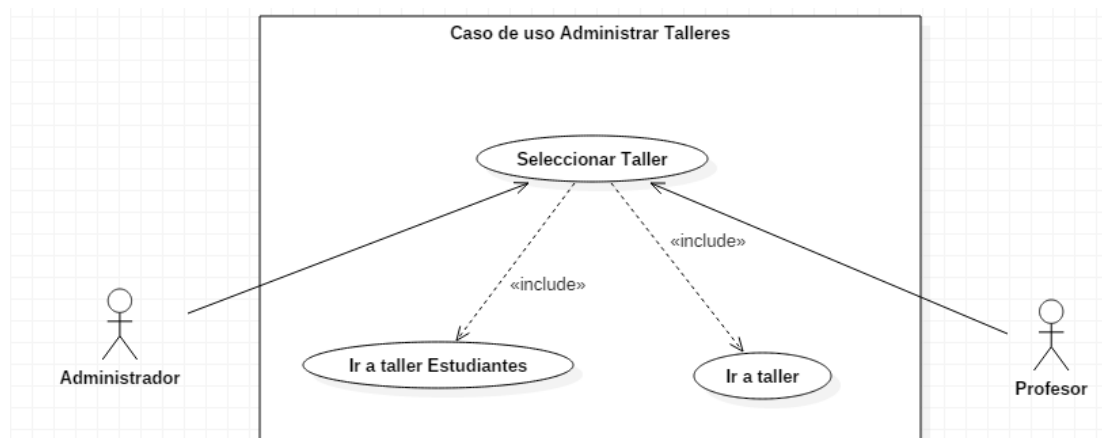


Figura 29 Caso de Uso IX: Administrar Talleres

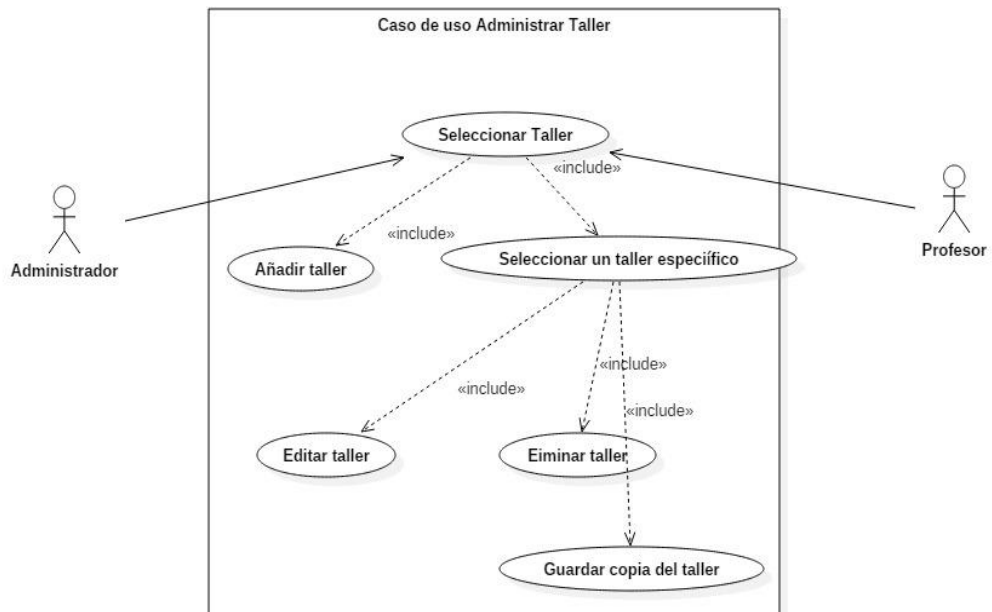


Figura 30 Caso de Uso IX: Administración de Talleres Detallado

CAPÍTULO IV

DESARROLLO DEL PROTOTIPO

3.1 DESARROLLO DE LA PLATAFORMA

El presente proyecto, como se ha mencionado anteriormente, se encuentra fundamentalmente desarrollado a través de la herramienta AppGini, la cual funciona soportando lenguaje de programación PHP, generando el código mediante la lectura de la base de datos MySQL diseñada para el proyecto.

Teniendo en cuenta lo anteriormente planteado, se procedió entonces a en primer lugar y como paso esencial para iniciar el proyecto, el diseño de las tablas que contendrá la base de datos relacional MySQL, desde la cual AppGini genera el código PHP correspondiente para la interacción de datos, es decir, crear, consultar, actualizar y eliminar (CRUD), la cual se encuentra estructurada (Ver Figura 31) para el momento de la redacción de este texto de la siguiente manera:

Una vez realizado lo anterior, y teniendo normalizado y bien esquematizado el modelo de la base de datos a implementar ya cargada en el servidor local, se procedió con el uso e implementación de la herramienta AppGini (Ver Figura 32), para continuar con el desarrollo del código PHP de la aplicación, la cual consta de varios elementos esenciales de un generador de módulos CRUD.

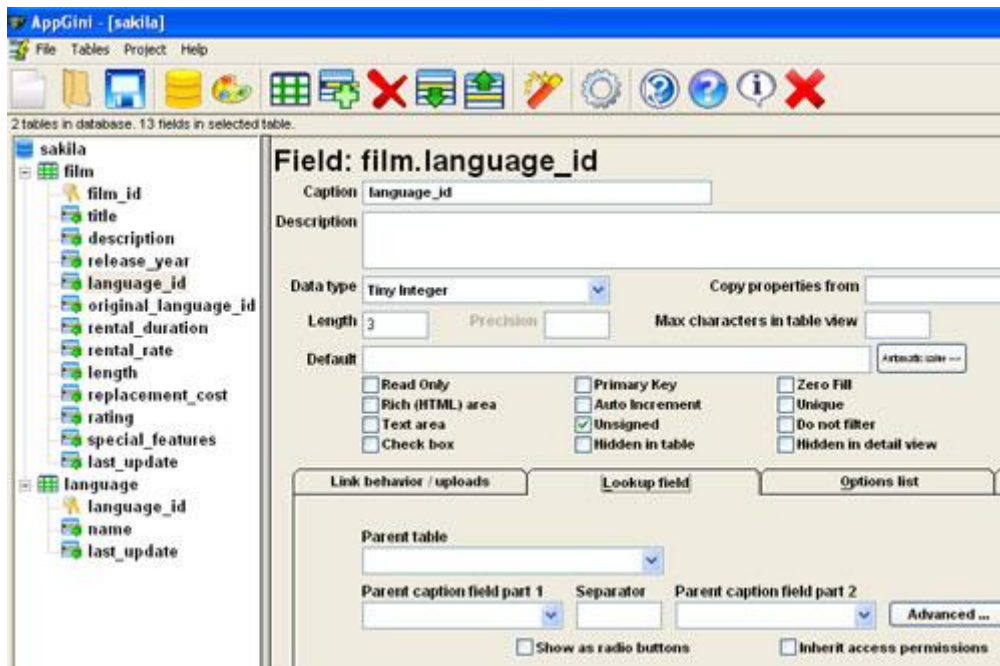


Figura 32 Entorno de Desarrollo de AppGini

De esta forma, con la herramienta anteriormente expuesta, AppGini. Se procede a generar el código base sobre el cual se trabajara a modo de framework, para generar la aplicación con los módulos de Insertar, Consultar, Actualizar y Eliminar en cada una de las tablas de la base de datos.

Ya con lo anterior, se obtiene código PHP funcional que interactúa con la base de datos, pero carente de cálculo de datos y proceso lógico. Para ello, se utilizó como editor de código, la herramienta de pago. PHPStorm. La cual es un IDE (Entorno de Desarrollo Integrado / Integrated Development Environment), especial para desarrollo con PHP (Ver Figura 33), y que además de facilitar el desarrollo, mediante herramientas como corrector de código y autocomprobación de sintaxis, también posee otros aspectos básicos y esenciales de un IDE como lo son, depurador de código y compilador.

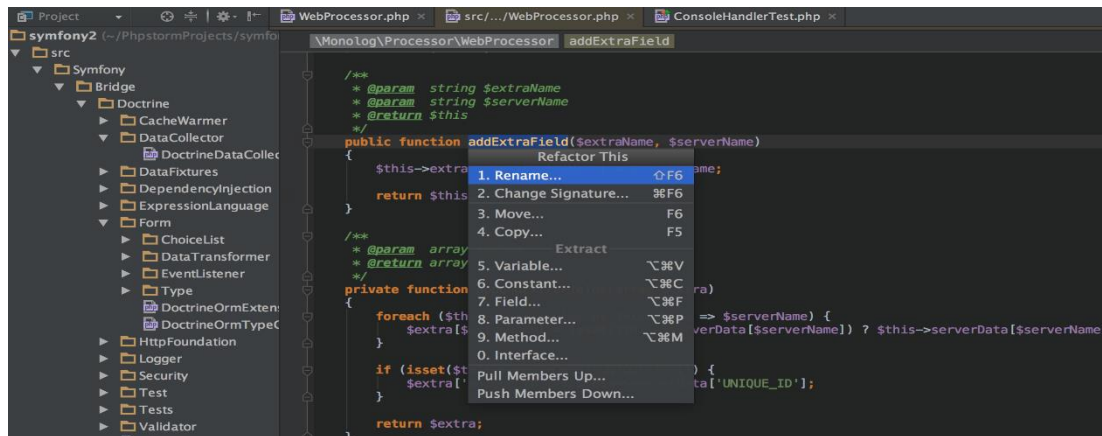


Figura 33 Entorno de Desarrollo de PHPStorm

Una vez realizado el framework o código base sobre el cual se realizó la aplicación, se procedió a editarlo mediante el IDE como se visualiza en la Figura 34, PHPStorm para agregar su funcionalidad y proceso lógico para el manejo de los datos.

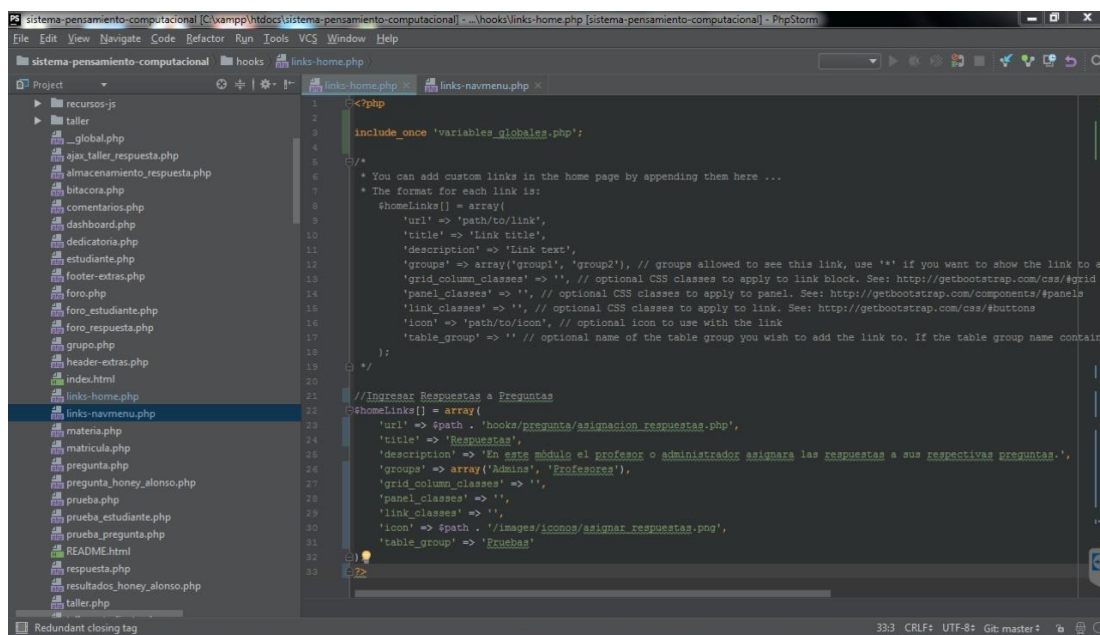


Figura 34 Archivos de Sistema siendo Editados

3.2 PRUEBAS TÉCNICAS

3.2.1 Prueba de Performance / Desempeño.

Tabla 22

Prueba de Performance

Módulo de la Aplicación	Tiempo					
	Respuesta					
	Esperado Script (segundos)	Usuarios Simultáneos	Insertar Datos	Consultar Datos	Actualizar Datos	Eliminar Datos
Gestión de Usuarios	2 – 3,5	1-20	✓	✓	✓	✓
Asignación de Pruebas	2 – 3,5	1-20	✓	✓	✓	✓
Creación de Grupos	2 – 3,5	1-20	✓	✓	✓	✓
Matriculas	2 – 3,5	1-20	✓	✓	✓	✓
Presentación de Pruebas	2 – 3,5	1-20	✓	✓	✓	✓

Teniendo en cuenta los resultados de la prueba de desempeño aplicada al sistema (Ver Tabla 22), se puede decir que cumple con el criterio de completitud, siendo que un usuario, una transacción completaron las pruebas sin ninguna falla y dentro del tiempo esperado o requerido por transacción (entre 2 y 3,5 segundos por tiempo de ejecución de script, consulta SQL) y múltiples usuarios, con los cuales se hizo prueba con 20, para múltiples transacciones simultaneas, llevaron a cabo y completaron las pruebas del sistema sin ninguna falla y dentro del tiempo esperado, requerido por cada transacción. (2 - 3,5 segundos).

3.2.2 Prueba de Stress.

Cálculos Basados en el Testeo de un servidor estándar (16 GB RAM, INTEL CORE I7, 3 TB HDD)

Tabla 23

Prueba de Stress 1

Módulo de la Aplicación	Tiempo	Usuarios Simultáneos	Insertar Datos	Consultar Datos	Actualizar Datos	Eliminar Datos
	Respuesta Esperado Script (segundos)					
Gestión de Usuarios	2 – 3,5	1-20	✓	✓	✓	✓
Asignación de Pruebas	2 – 3,5	1-20	✓	✓	✓	✓
Creación de Grupos	2 – 3,5	1-20	✓	✓	✓	✓
Matriculas	2 – 3,5	1-20	✓	✓	✓	✓
Presentación de Pruebas	2 – 3,5	1-20	✓	✓	✓	✓

Servidor Reducido (512 mb ram, dual core, 80gb hdd)

Tabla 24

Prueba de Stress 2

Módulo de la Aplicación	Tiempo	Usuarios Simultáneos	Insertar Datos	Consultar Datos	Actualizar Datos	Eliminar Datos
	Respuesta Esperado Script (segundos)					
Gestión de Usuarios	2 – 3,5	1-20	✓	✓	✓	✓
Asignación de Pruebas	2 – 3,5	1-20	✓	✓	✓	✓

CONTINÚA 

Creación de Grupos	2 – 3,5	1-20	✓	✓	✓	✓
Matriculas	2 – 3,5	1-20	✓	✓	✓	✓
Presentación de Pruebas	2 – 3,5	1-20	✓	✓	✓	✓

3.3 DISCUSIÓN DE LA EVALUACIÓN

A fin de discutir la naturaleza de los resultados arrojados por la evaluación realizada y finalmente llegar a las conclusiones que permitan reflexionar los diferentes aspectos que buscar determinar si el sistema planteado resuelve la problemática presentada o no, es importante explicar primero cuales fueron los aspectos seleccionados para dar forma a los instrumentos de evaluación aplicados y, junto con eso, esclarecer los detalles del enfoque de investigación que se consideró propicio adoptar, para ello, se elaboraran primero una serie de puntos a delimitar que permitirán dar forma al estudio realizado.

3.3.1 Tipo de investigación utilizada.

En primer lugar cuando es momento de referirse al tipo de investigación utilizado, se está haciendo alusión en síntesis a la estrategia o técnica que adopta el investigador para responder a la interrogante, inquietud o problemática planteado en el estudio.

En ese orden de ideas, cabe resaltar que en el actual trabajo de grado se evidencia que la investigación realizada converge en naturalezas de diferentes tipos como lo son de: aplicada, de campo, experimental y descriptiva, dichas investigaciones de diferentes índoles permitieron establecer las bases de desarrollo del sistema que se está presentando.

Siguiendo el mismo orden de pensamientos entonces se proceder a explicar que:

La investigación es de tipo aplicada pues se según (Egg, 2007, pág. 33) esta se describe como: “La respuesta efectiva y fundamentada a un problema ya detectado, descrito y analizado. La investigación aplicada concentra su atención en las posibilidades fácticas de llevar a la práctica las teorías generales, y destina sus esfuerzos a resolver los problemas y necesidades

que se plantean los hombres en la sociedad en un corto, mediano y largo plazo, es decir, se interesa fundamentalmente en la propuesta de solución en un contexto físico-social específico.” Habiendo plasmado lo anterior, la investigación realizada se considera de esa índole pues consiste en la obtención y producción propia de conocimiento, para su posterior reflexión y aplicación de las teorías existentes de manera que se concrete la solución a la problemática en un corto, mediano o largo plazo una vez que se lleve a cabo.

Por otro lado, en lo que a la investigación de campo se refiere, los autores (Palella & Martins, 2010) define que es: “aquella que consiste en la recolección de datos directamente de la realidad donde ocurren los hechos, sin manipular o controlar las variables. Estudia los fenómenos sociales en su ambiente natural” se considera que la investigación realizada aplica como de esta índole porque ésta se apoyó en informaciones derivadas de entrevistas, cuestionarios, encuestas y observaciones que permitieron el estudio directo de la realidad donde se presentan los hechos

Cuando es momento de hablar acerca de la investigación experimental cabe citar a (Arias, 2012, pág. 31) que expone que: “es aquel proceso que consiste en someter a un objeto o grupo de individuos a determinadas condiciones, estímulos o tratamientos, para observar los efectos o reacciones que se producen” y por ende se considera que es de carácter experimental ya que los datos fueron recogidos mediante experimentos y analizados mediante la disección de los estratos en los individuos y fueron luego usados para poder realizar la ejecución y puesta en funcionamiento del proyecto.

Por último, cuando se hace mención a la investigación descriptiva, se procede a tomar la palabra del autor (Arias, 2012, pág. 31) en la que afirma que es: “aquella que consiste en la caracterización de un hecho, fenómeno, individuo o grupo, con el fin de establecer su estructura o comportamiento” y se considera que la investigación realizada califica como descriptiva debido a que se logró caracterizar el objeto de estudio concreto, identificando sus particularidades, señales destacadas y propiedades. Lo que permitió agrupar y organizar los objetos y protagonistas involucrados en el presente trabajo de investigación.

3.3.2 Enfoque metodológico de la investigación.

Al momento de delimitar los enfoques metodológicos es importante conocer su funcionalidad, de acuerdo con (Palazzolo & Vidarte Asorey, 2013) los enfoques metodológicos: “establecen una primera idea general relacionada a la metodología de investigación., donde el investigador establece su posición respecto al proceso y entorno a investigar. Generalmente se utilizan tres (3) enfoques: el cuantitativo, el cualitativo y la triangulación. En la mayoría de las investigaciones se utilizan los dos primeros. La diferencia fundamental entre estos dos es que el cuantitativo analiza la asociación o relación entre variables cuantificadas, mientras que la investigación cualitativa lo efectúa en contextos estructurales y situacionales.”

En relación a lo anterior cabe destacar que (Jimenez, 2017) ante la mención de las diferencias entre el enfoque cuantitativo y cualitativo afirma que: “la investigación es un proceso metódico y sistemático dirigido a la solución de problemas o preguntas científicas, mediante la producción de nuevos conocimientos, los cuales constituyen la solución o respuesta a tales interrogantes”. Así mismo expone que “la investigación cuantitativa es la que analiza diversos elementos que pueden ser medidos y cuantificados. Toda la información se obtiene a base de muestras de población, y sus resultados son extrapolables a toda la población, con un determinado nivel de error y nivel de confianza.” Por lo que, en resumen: “se sirve de números y métodos estadísticos. Parte de casos concretos para llegar a una descripción general o comprobar hipótesis causales. Por lo que se le llama cuantitativa-sistemática-generalizadora”

En cambio, al momento de referirse a la investigación cualitativa (Jimenez, 2017) expresa que: “Da cuenta de la credibilidad de la comunicación, los conceptos, o de los beneficios ofrecidos. Los estudios cualitativos observan al individuo encuestado, por un lado, y por otro observan el producto, u objeto de la investigación sobre la que se encuesta y que es el estímulo generador de determinadas conductas entre los individuos. Estas observaciones adquirirán un carácter probabilístico, y, por ende, proyectarle al universo en una segunda etapa de la investigación que ya será cuantitativa.”

Por otro lado, para ahondar el tema de la triangulación metodológica es importante destacar que según (Arias Valencia, Volumen 18, Número 01, Marzo 2000, págs. 13-26) esta “proviene

del término usado en los círculos de navegación donde se utilizan múltiples puntos de referencia para localizar una posición desconocida.”

Teniendo en cuenta lo anterior al momento de desglosar los pensamientos y reflexiones en cuanto al proceso y el entorno a investigar, se pudo llegar a la conclusión de que ambos enfoques de investigación presentan ventajas y desventajas dependiendo de la problemática que se plantee solucionar, por ende, es indispensable el entendimiento a fondo de su aplicación, del alcance y veracidad de los datos obtenidos mediante esta, pues, mientras que en los métodos cualitativos la propensión va dirigida a “comunicarse con” los sujetos del estudio y por ende el investigador está limitado a preguntar y confiar en la veracidad de la información basándose en la credibilidad de la comunicación, en el caso de los métodos cuantitativos la propensión va dirigida a “servirse de” los sujetos del estudio y por ende el investigador se limita a responder con métodos científicos medibles y cuantificables las incógnitas que se le plantean.

Habiendo racionalizado lo anterior se procedió a analizar las diferentes herramientas y técnicas de recolección de datos que los distintos enfoques metodológicos ofrecían, para así consecuentemente seleccionar aquellos que se adaptaran de manera más asertiva a la problemática actual definida en el presente trabajo de investigación.

3.3.3 Técnicas específicas Aplicadas.

Ante todo, al momento de hablar de técnicas de investigación, se habla según (Puente, 2017) que son aquellas que: “consisten en observar atentamente el fenómeno, hecho o caso, tomar la información y registrarla para su posterior análisis. Siendo la observación un elemento fundamental de todo proceso investigativo; en ella se apoya el investigador para obtener el mayor número de datos y siendo está sustentada y apoyada con el resto de las herramientas de recolección y análisis de datos” lo que explica que, a pesar de que existen diversas técnicas y herramientas que permiten la recolección, análisis y cotejo de cada uno de los datos que conforman la información que se obtiene como resultado, la observación vendría siendo la técnica fundamental y primitiva para cualquier proceso investigativo.

Dejando en evidencia lo anterior, para la realización de la investigación y consecuente recolección de datos que hicieron posible el desarrollo del proyecto se utilizaron técnicas y procedimientos pertenecientes a los tipos de investigación nombrados anteriormente.

3.3.3.1 Investigación Bibliográfica.

Al hacer mención a la investigación bibliográfica o documental es primordial que se defina, para estos propósitos entonces, (Rodríguez, 2013) explica que: “La investigación bibliográfica y documental es un proceso sistemático y secuencial de recolección, selección, clasificación, evaluación y análisis de contenido del material empírico impreso y gráfico, físico y/o virtual que servirá de fuente teórica, conceptual y/o metodológica para una investigación científica determinada.” Al mismo tiempo, el autor resalta que: “En el proceso de recolección de información para la construcción de un objeto de investigación o de un proyecto de tesis, la investigación bibliográfica y documental ocupa un lugar importante, ya que garantiza la calidad de los fundamentos teóricos de la investigación.”

De ahí que se considere que antes de reflexionar sobre los enfoque metodológicos y teóricos a utilizar para la estructuración, esquematización y desarrollo de un proyecto de grado, sea importante el indagar, interpretar y presentar datos e informaciones sobre un tema determinado de cualquier ciencia basado en documentos empíricos y producidos por autores entendidos en cada uno de los tópicos, de modo que, el investigador tenga noción y entendimiento de la aplicación de los métodos, técnicas y herramientas de investigación para dar como resultado un trabajo con bases propias y adecuadas a los procesos científicos debidos.

En resumen, se considera que por ser un trabajo investigativo se requiere documentar, analizar y profundizar la investigación sobre la base del uso de libros, artículos, sitios web, informes y otras fuentes bibliografías de web válidas. Todo el material consultado es referenciado en la sección “Bibliografía”, en el presente trabajo.

3.3.3.2 Método Inductivo-Deductivo.

Al momento de hablar de método inductivo-deductivo (Julian Pérez Porto & María Merino, 2012) explican que: “Es aquel método científico que obtiene conclusiones generales a partir de premisas particulares. Se trata del método más usual, en el que pueden distinguirse cuatro pasos esenciales: la observación de los hechos para su registro; la clasificación y el estudio de estos hechos; la derivación inductiva que parte de los hechos y permite llegar a una generalización; y la contrastación” también los autores exponen que: “esto supone que tras una primera etapa de observación, análisis y clasificación de los hechos, se logra postular una hipótesis que brinda una solución al problema planteado. Una forma de llevar a cabo el método inductivo es proponer, mediante diversas observaciones de los sucesos u objetos en estado natural, una conclusión que resulte general para todos los eventos de la misma clase.”

Se utilizó este método ya que mediante la observación directa y la racionalización de lo que el investigador percibe se considera que se describe de mejor manera el fenómeno estudiado y mediante esta técnica se producirán las mejores recomendaciones y conclusiones a beneficio del sistema.

3.3.4 Técnicas e Instrumentos de Recolección de Información.

Visto desde la perspectiva de (Rodríguez P. , 2008, pág. 8) las técnicas son: “Los medios empleados para recolectar información, entre las que se destacan la observación, cuestionarios, entrevistas y encuestas”.

Por otro lado (Rojas, 2006, pág. 197) destaca que: “El volumen y el tipo de información (bien sea cualitativa o cuantitativa) que se recaben en el trabajo de campo deben estar plenamente justificados por los objetivos e hipótesis de la investigación, o de lo contrario se corre el riesgo de recopilar datos de poca o ninguna utilidad para efectuar un análisis adecuado del problema.”

Habiendo esclarecido lo anterior, se procede a desglosar que las técnicas de recolección de información para preparar las bases de los requerimientos del sistema web fueron las siguientes:

3.3.4.1 Encuesta.

Según (Puente, 2017) “la encuesta es una técnica destinada a obtener datos de varias personas cuyas opiniones interpersonales interesan al investigador. Para ello, a diferencia de la entrevista, se utiliza un listado de preguntas escritas que se entregan a los sujetos, a fin de que las contesten igualmente por escrito, y dicho listado recibe el nombre de cuestionario.”

Por ende, en vista que se necesitaba determinar cuáles eran las opiniones acerca del uso y rendimiento del sistema de Pensamiento Computacional así como también de las opiniones de los estudiantes que hicieron uso de él y del impacto que este tuvo en su desempeño académico se decidió aplicar encuestas, tanto a los profesores expertos y conocedores del área que hacen vida en la Universidad de las Fuerzas Armadas ESPE como a un selecto grupo de estudiantes de la misma institución para poder obtener información acerca de los diferentes puntos de vistas tanto desde la parte estudiantil, como desde el área docente.

Al mismo tiempo (Puente, 2017) afirma también que la encuesta es “impersonal porque el cuestionario no lleva el nombre de la persona que lo responde, ya que no interesan estos datos.” Y que “es una técnica que se puede aplicar a sectores más amplios del universo de manera mucho más económica que mediante entrevistas” y resaltó el hecho de que “varios autores llaman cuestionario a la técnica misma. Los mismo u otros unen en un mismo concepto a la entrevista y al cuestionario, denominándolo encuesta, debido a que en los dos casos se trata de obtener datos de personas que tienen alguna relación con el problema que es materia de investigación.”.

Cabe resaltar lo anterior para tener en cuenta que ambas técnicas guardan una relación estrecha, pero diferencias importantes a considerar teniendo en cuenta las características y particularidades a resaltar de cada individuo estudiado, pues mientras que en uno la identidad de la persona no guarda mayor importancia, en la otra dicha identidad y su relación directa con la problemática a tratar supone el factor más relevante o uno de los más relevantes en el estudio.

3.3.4.2 *Cuestionario.*

Este concepto va de la mano con la definición de encuesta anteriormente desglosado, habiendo hecho referencia ello, se determina que, aunque las encuestas proporcionan un instrumento muy importante en la obtención de información, las preguntas que se usan en las mismas deben ser de carácter abiertas de modo de lograr la participación objetiva de un mayor número de personas. En este caso en específico para intentar obtener la información más completa posible, se elaboraron cuestionarios compuestos por preguntas cerradas que fueron aplicados y un número determinado de estudiantes guardando que la cantidad sea afín con la regla estadística que explica que la muestra debe ser proporcional a la población total.

3.3.5 **Población.**

La población según (Tamayo & Tamayo, 2013) es “La totalidad de un fenómeno de estudio, incluye la totalidad de las unidades de análisis que integran dicho fenómeno y que debe cuantificarse para un determinado estudio integrando un conjunto “N” de entidades que participan en una determinada característica y se le denomina “la población” por constituir la totalidad de un fenómeno adscrito a una investigación”. Así mismo el autor también explica que la población “es el conjunto total de individuos, objetos o medidas que poseen algunas características comunes observables en un lugar y en un momento determinado, en la Tabla 25 se indica la población donde se desarrollará la investigación.

Tabla 25

Población de estudio

Población de Estudio (ESPE)	Cantidad	Porcentaje sobre el total (%)
Profesores del Departamento de Ciencias de la Computación	5	0,4 %
Estudiantes ESPE	230	99,6 %
TOTAL	235	100%

Por otra parte, para determinar a cuantas personas se deben aplicar la encuesta se requiere de los siguientes parámetros: tamaño de la población, margen de error (intervalo de confianza), nivel de confianza y la desviación estándar. La fórmula para calcular el tamaño de muestra cuando se conoce el tamaño de la población es la indicada en la ecuación siguiente (Pickers, 2015):

$$n = \frac{Nz_a^2 p x q}{d^2 x (N - 1) + Z_a^2 p x q}$$

Donde:

- n : Tamaño de la muestra
- N : Tamaño de la población (230)
- Z_a : Nivel de confianza (95 %)
- p : Probabilidad de éxito (50 %)
- q : Probabilidad de fracaso (50 %)
- d : Precisión (5 %)

Siendo el caso: 17 individuos de muestra, representados por 5 profesores y 12 estudiantes. Así que, habiendo plasmado lo anterior, se procede a estructurar el análisis de los datos obtenidos mediante las diversas técnicas de investigación aplicadas.

3.3.6 Resultados del Test de Usabilidad Aplicado a Docentes y Estudiantes.

ENCUESTA A PROFESORES

Tabla 26

Pregunta 1 de Profesores

1.- ¿Es la aplicación web adaptable a todos los dispositivos para su cómodo uso desde cualquier plataforma?	
Si	5
No	0

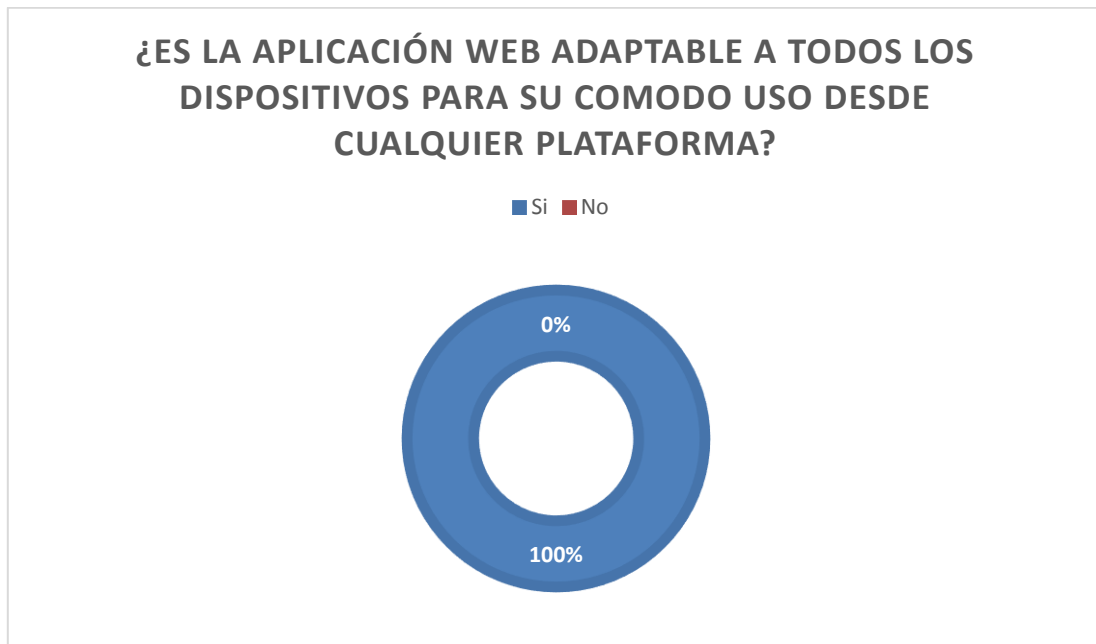


Figura 35 Pregunta 1 Profesores

Tabla 27

Pregunta 2 de Profesores

2.- ¿Considera que, desde el punto de vista de la usabilidad, la aplicación presenta una interfaz amigable al usuario y de fácil uso?	
Si	3
No	2

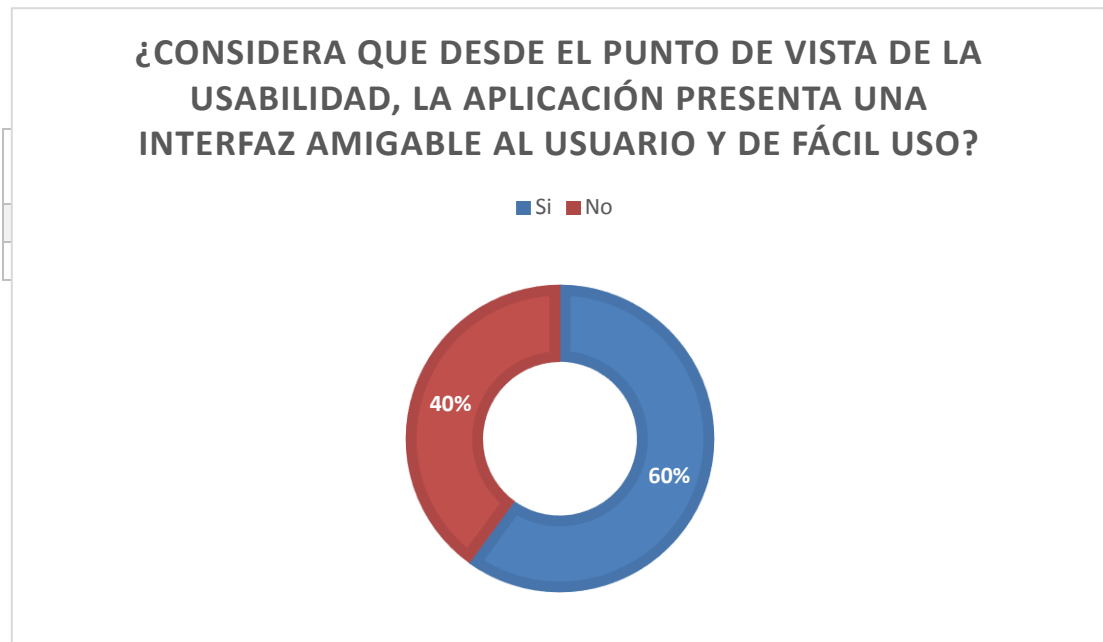


Figura 36 Pregunta 2 Profesores

Tabla 28

Pregunta 3 de Profesores

3.- ¿Representa la aplicación una herramienta significativa en su modelo de impartir conocimiento?	
Si	3
No	2

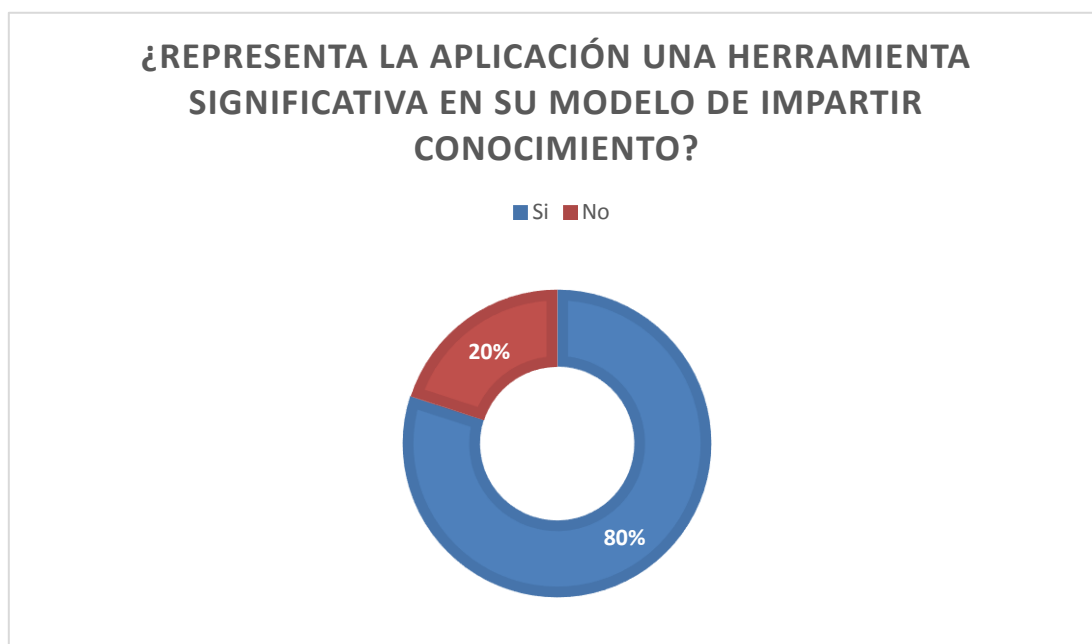


Figura 37 Pregunta 3 Profesores

Tabla 29

Pregunta 4 de Profesores

4.- ¿Ha tenido un impacto positivo en usted la implementación del LMS?	
Si	4
No	1

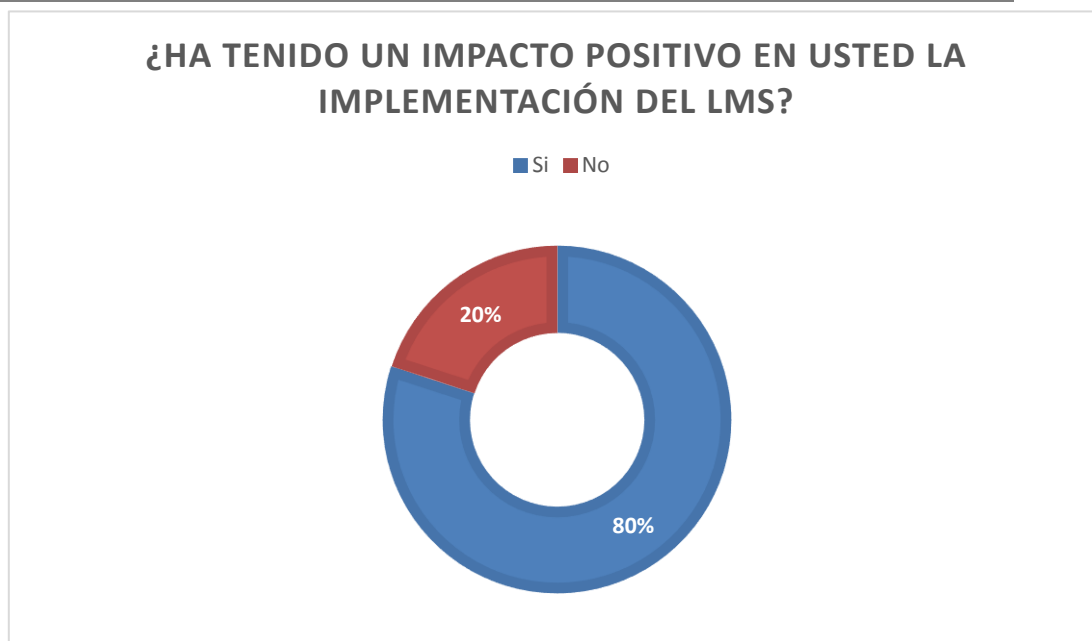
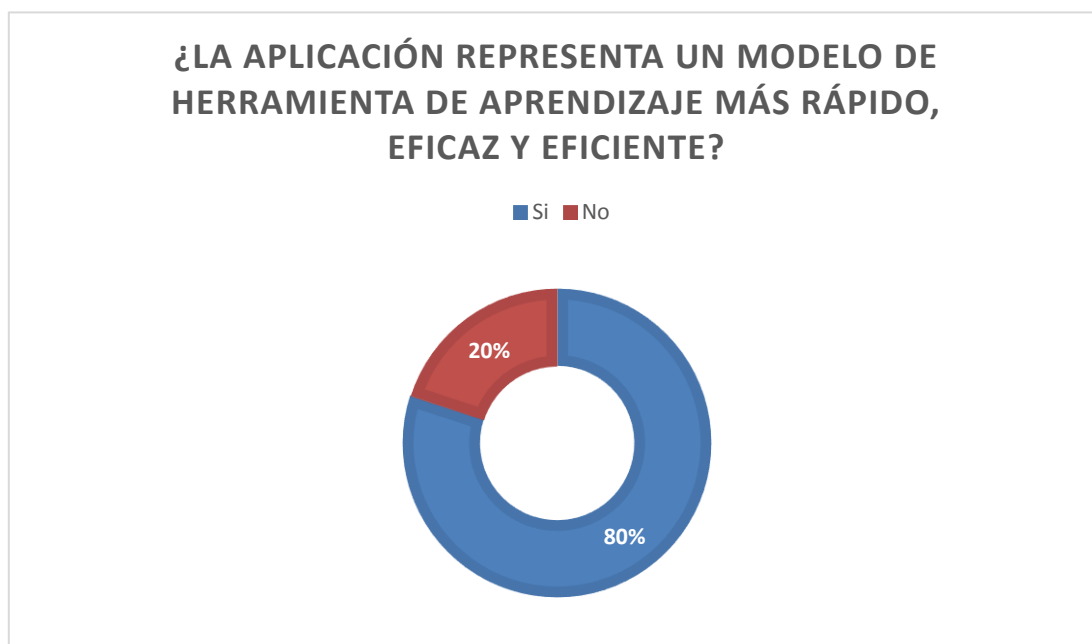


Figura 38 Pregunta 4 Profesores

Tabla 30*Pregunta 5 de Profesores*

5.- ¿La aplicación representa un modelo de herramienta de aprendizaje más rápido, eficaz y eficiente?	
Si	4
No	1

**Figura 39** Pregunta 5 Profesores**Tabla 31***Pregunta 6 de Profesores*

6.- ¿Considera INDISPENSABLE la implementación de un software LMS como el del presente proyecto en cada institución educativa?	
Si	3
No	2

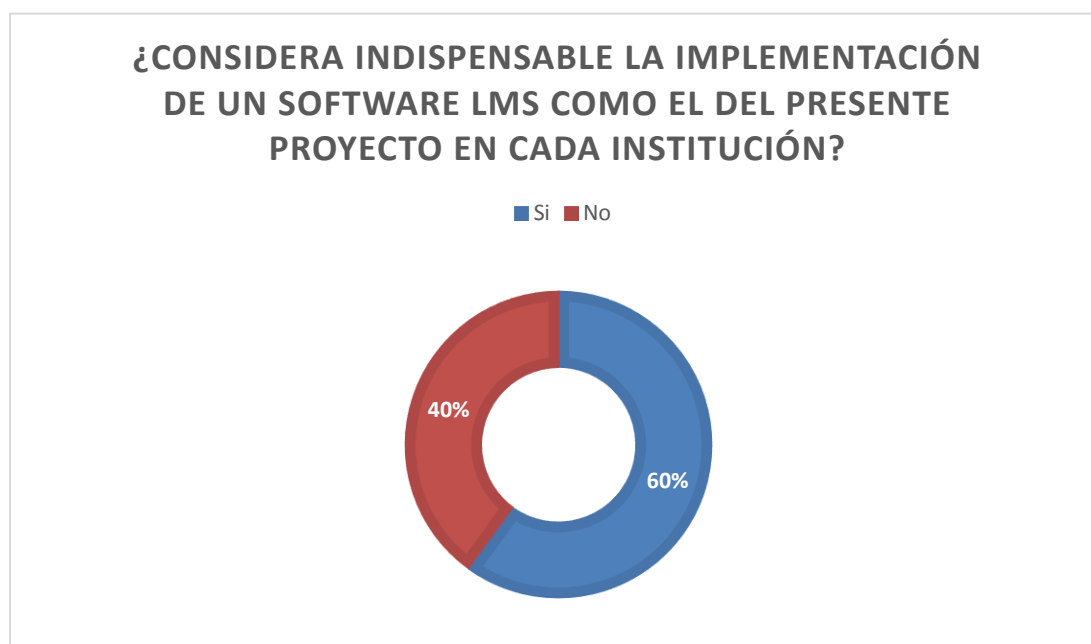


Figura 40 Pregunta 6 Profesores

Tabla 32

Pregunta 7 de Profesores

7.- ¿Suponen los módulos de pruebas y de preguntas y respuestas para estudiantes una mejor forma de evaluación académica?	
Si	5
No	0

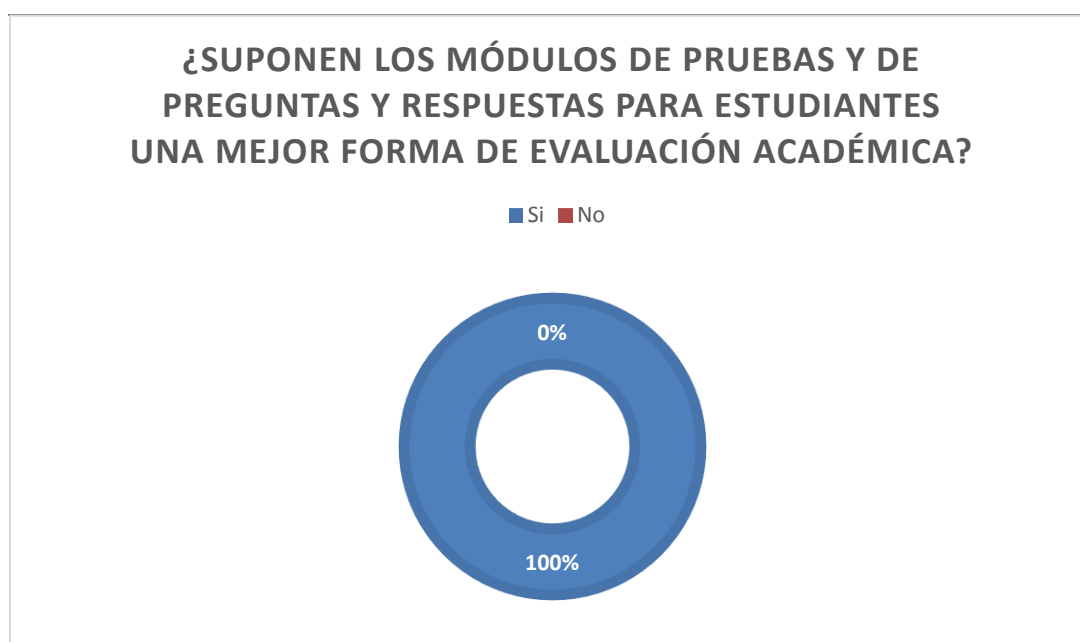
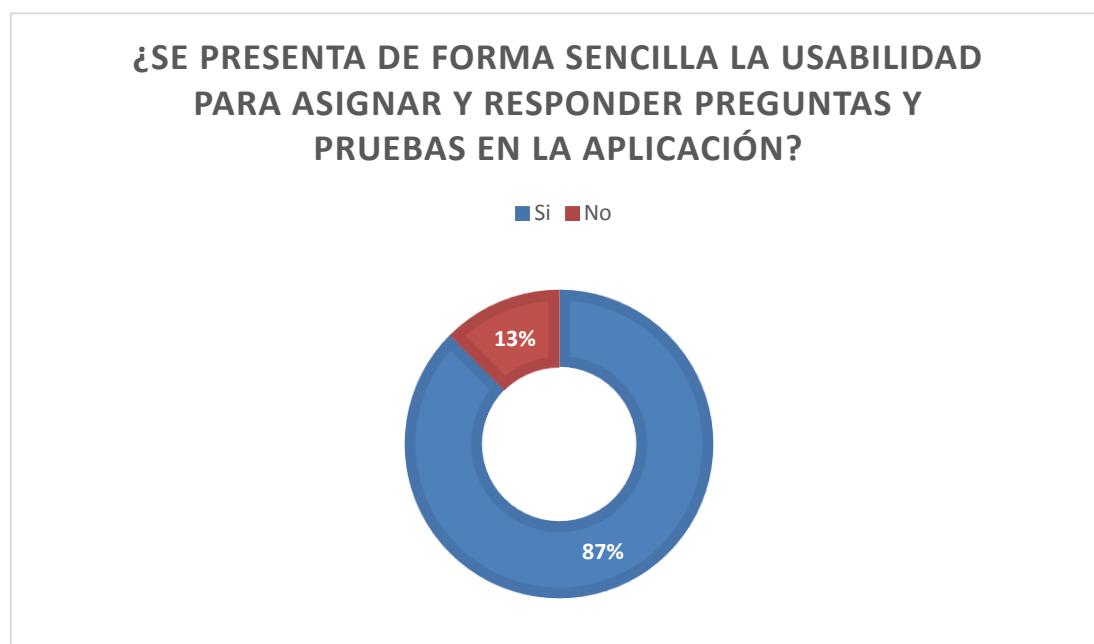


Figura 41 Pregunta 7 Profesores

Tabla 33*Pregunta 8 de Profesores*

8.- ¿Se presenta de forma sencilla la usabilidad para asignar pruebas en la aplicación?	
Si	4
No	1

**Figura 42** Pregunta 8 Profesores**Tabla 34***Pregunta 9 de Profesores*

9.- ¿Es entendible para su uso el Dashboard / Panel de Tareas de la aplicación?	
Si	5
No	0

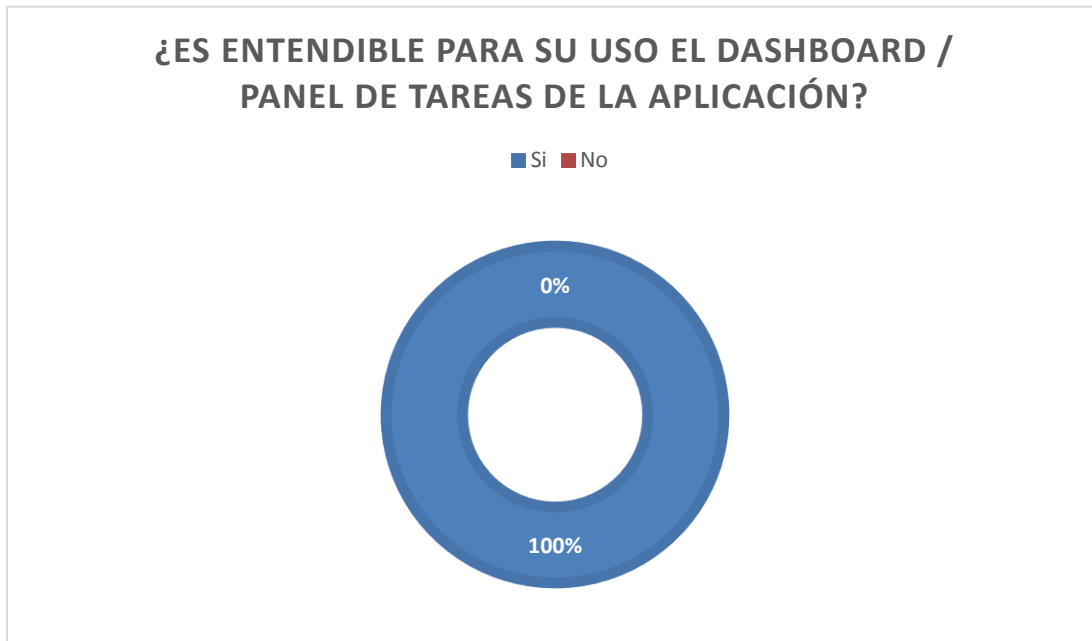


Figura 43 Pregunta 9 Profesores

ENCUESTA A ESTUDIANTES

Tabla 35

Pregunta 1 de Estudiantes

1.- ¿considera usted que el hecho de conocer su tipo de aprendizaje más efectivo ha tenido un impacto en su proceso de aprendizaje?	
Si	12
No	0

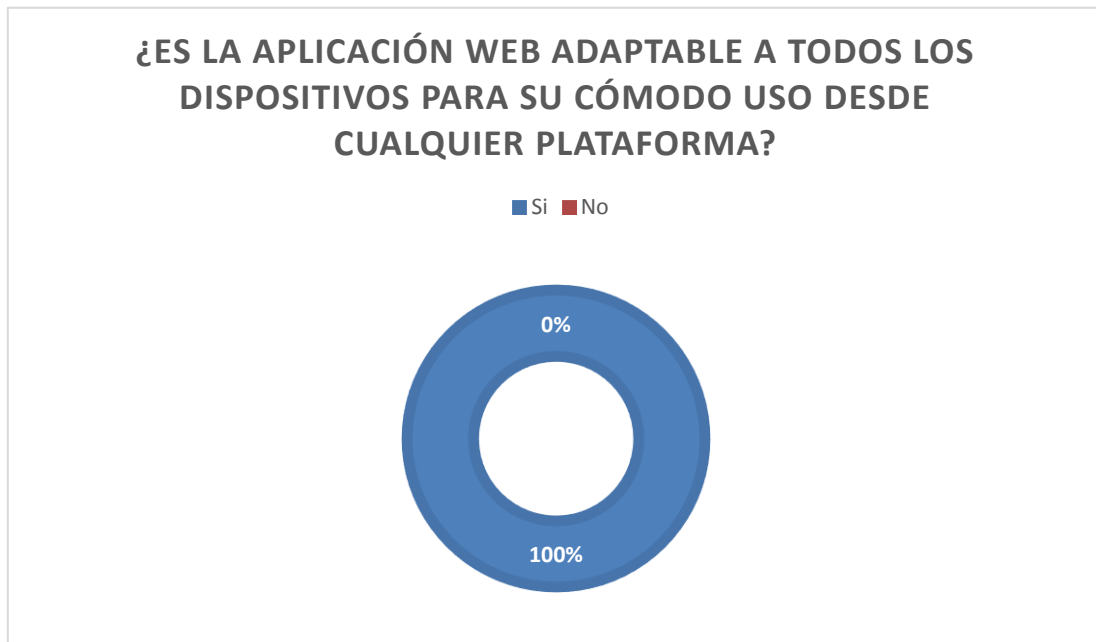


Figura 44 Pregunta 1 Estudiantes

Tabla 36

Pregunta 2 de Estudiantes

2.- ¿Considera que, desde el punto de vista de la usabilidad, la aplicación presenta una interfaz amigable al usuario y de fácil uso?	
Si	9
No	3

¿CONSIDERA QUE DESDE EL PUNTO DE VISTA DE LA USABILIDAD, LA APLICACIÓN PRESENTA UNA INTERFAZ AMIGABLE AL USUARIO Y DE FÁCIL USO?

■ Si ■ No

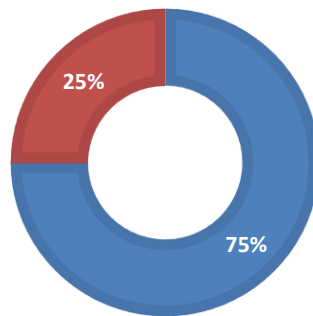


Figura 45 Pregunta 2 Estudiantes

Tabla 37

Pregunta 3 de Estudiantes

3.- ¿Representa la aplicación una herramienta significativa en su modelo de adquirir conocimiento?	
Si	8
No	4

¿REPRESENTA LA APLICACIÓN UNA HERRAMIENTA SIGNIFICATIVA EN SU MODELO DE ADQUIRIR CONOCIMIENTO?

■ Si ■ No

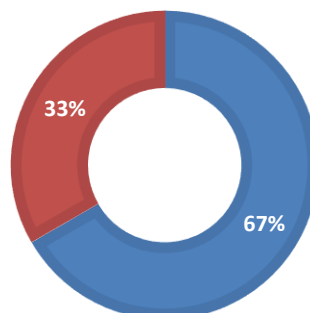
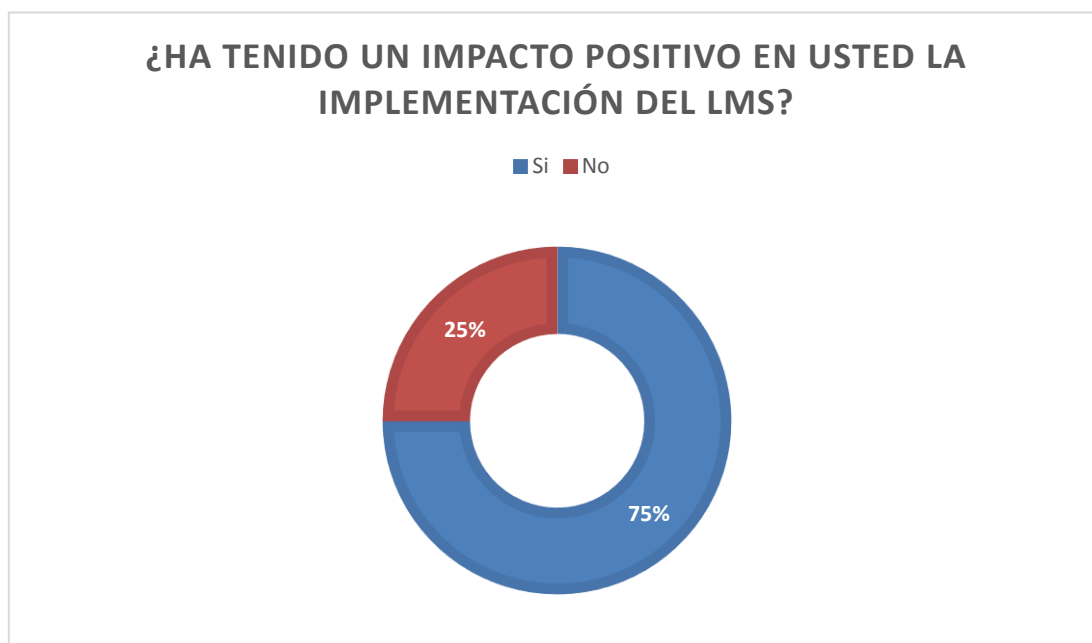


Figura 46 Pregunta 3 Estudiantes

Tabla 38*Pregunta 4 de Estudiantes*

4.- ¿Ha tenido un impacto positivo en usted la implementación del LMS?	
Si	9
No	3

*Figura 47* Pregunta 4 Estudiantes**Tabla 39***Pregunta 5 de Estudiantes*

5.- ¿La aplicación representa un modelo de herramienta de aprendizaje más rápido, eficaz y eficiente?	
Si	12
No	0

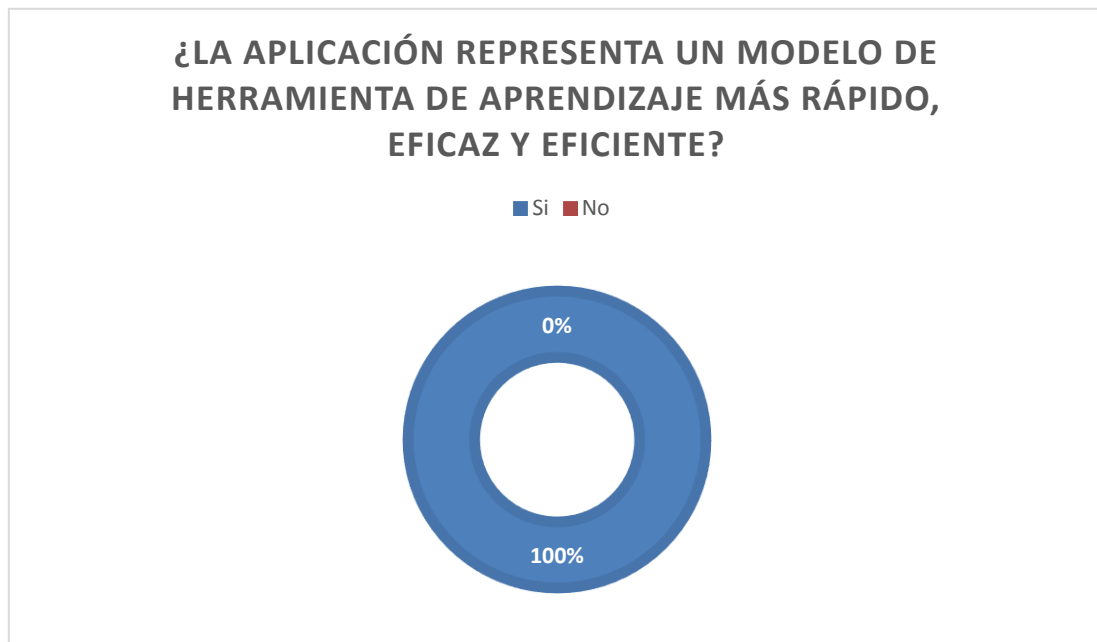


Figura 48 Pregunta 5 Estudiantes

Tabla 40

Pregunta 6 de Estudiantes

6.- ¿Se presenta de forma sencilla la usabilidad para responder preguntas y pruebas en la aplicación?	
Si	10
No	2

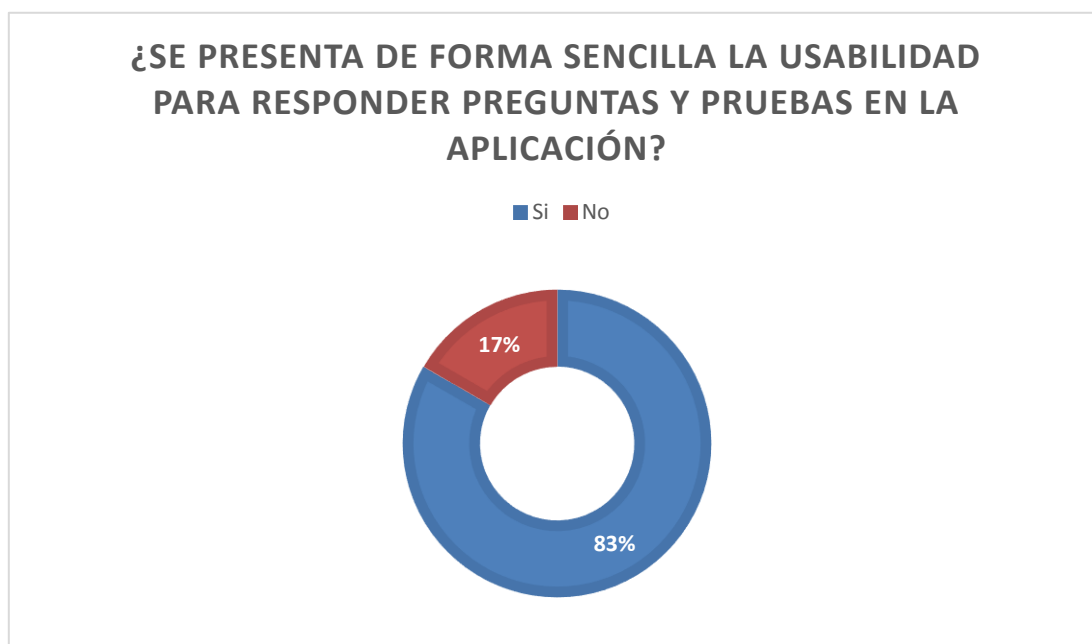


Figura 49 Pregunta 6 Estudiantes

Tabla 41

Pregunta 7 de Estudiantes

7.- ¿Su índice académico ha aumentado desde que descubrió cuál era su método de aprendizaje más efectivo?	
Si	10
No	2

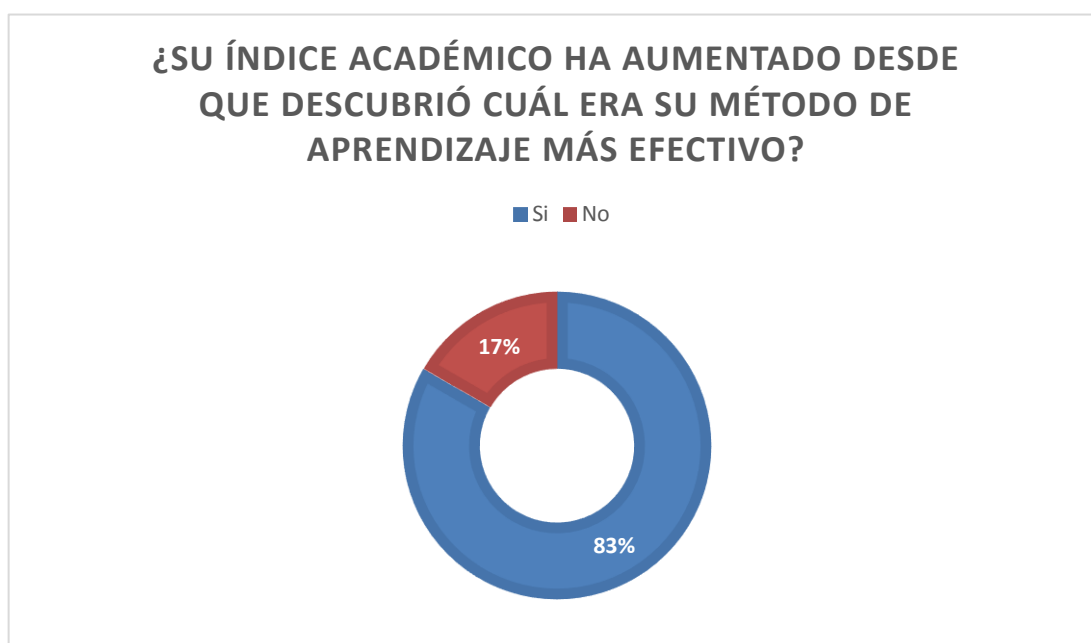


Figura 50 Pregunta 7 Estudiantes

ENCUESTA AL ADMINISTRADOR DE LA APLICACIÓN

Tabla 42

Pregunta 1 al administrador

1.- ¿Es la aplicación web adaptable a todos los dispositivos para su cómodo uso desde cualquier plataforma?	
Si	1
No	0

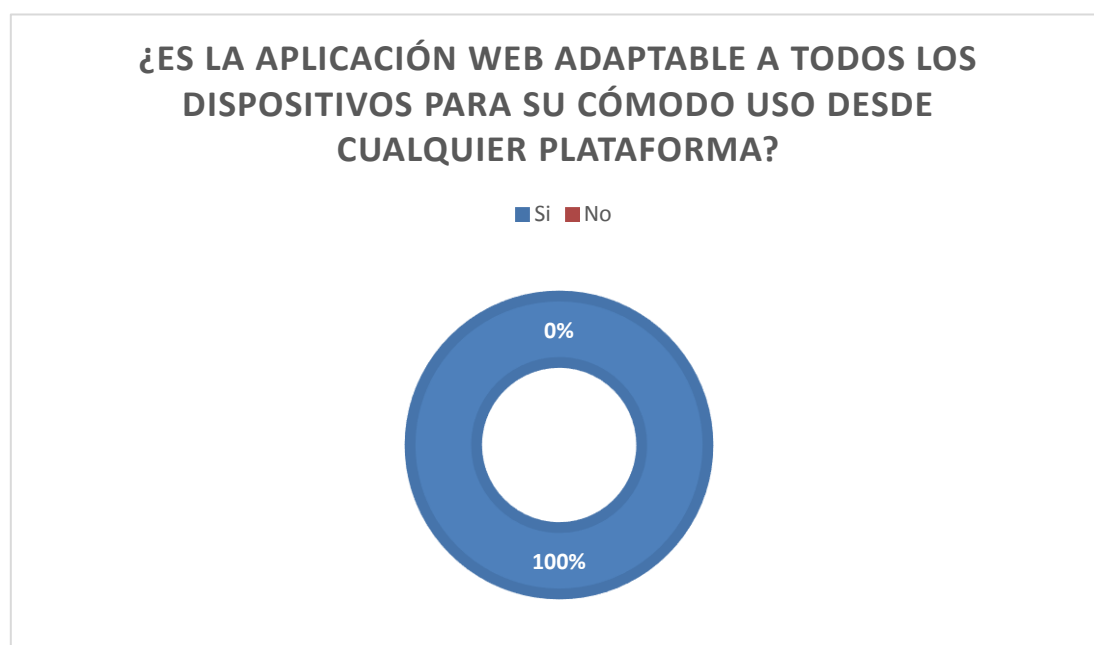


Figura 51 Pregunta 1 Administrador

Tabla 43

Pregunta 1 al administrador

2.- ¿Considera que, desde el punto de vista de la usabilidad, la aplicación presenta una interfaz amigable al usuario y de fácil uso?	
Si	1
No	0

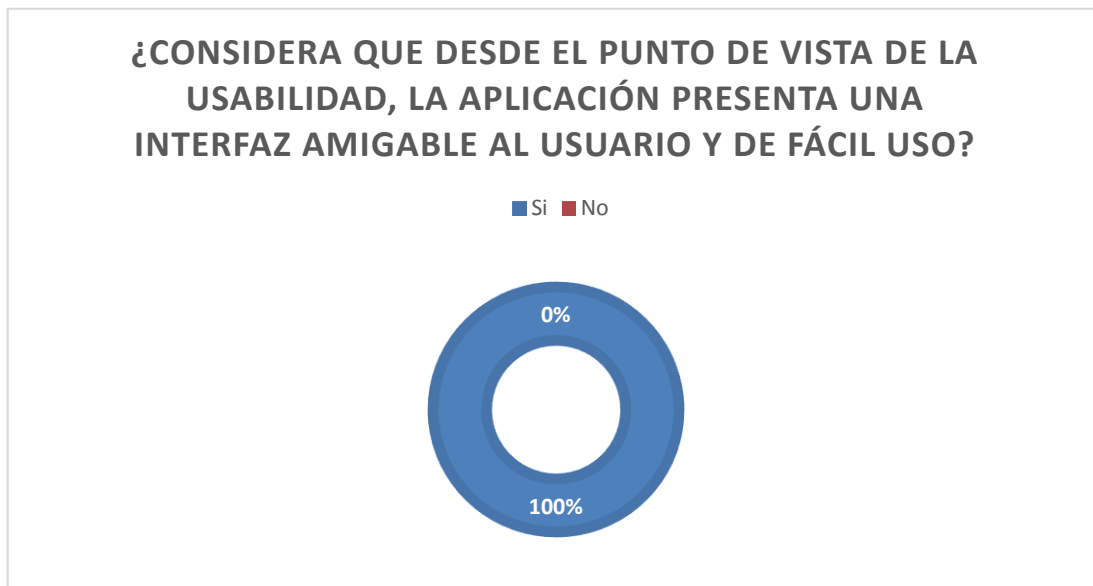


Figura 52 Pregunta 2 Administrador

Tabla 44

Pregunta 3 al administrador

3.- ¿Es la aplicación realmente efectiva a la hora de realizar cambios en aspectos técnicos y de mantenimiento de la misma?	
Si	1
No	0

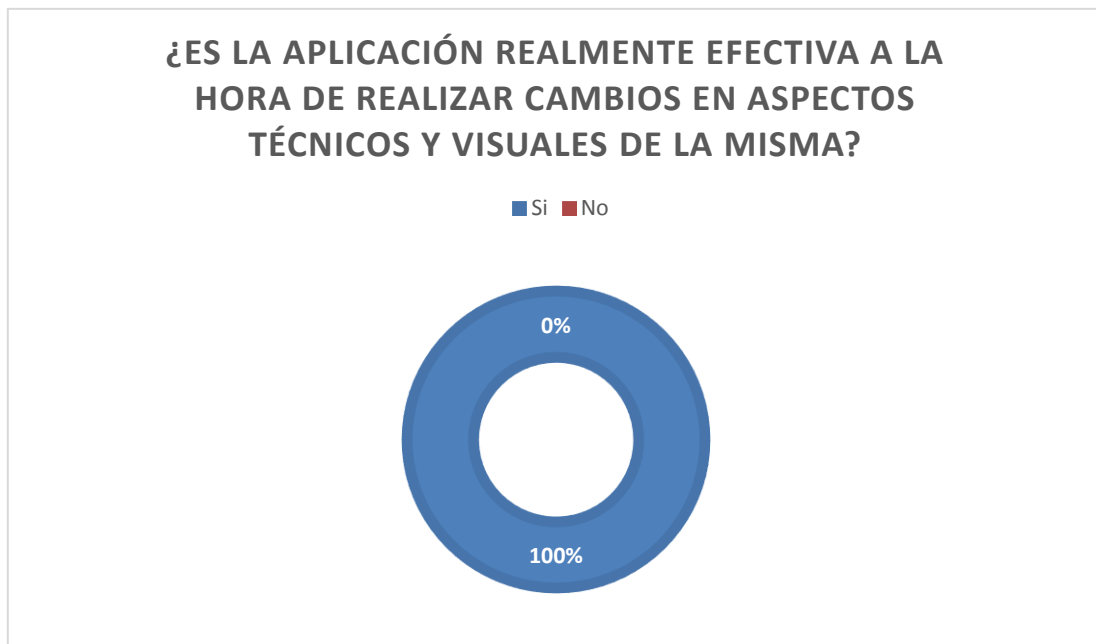


Figura 53 Pregunta 3 Administrador

Tabla 45

Pregunta 4 al administrador

4.- ¿El dashboard / panel administrador es de fácil entendimiento y usabilidad?	
Si	1
No	0

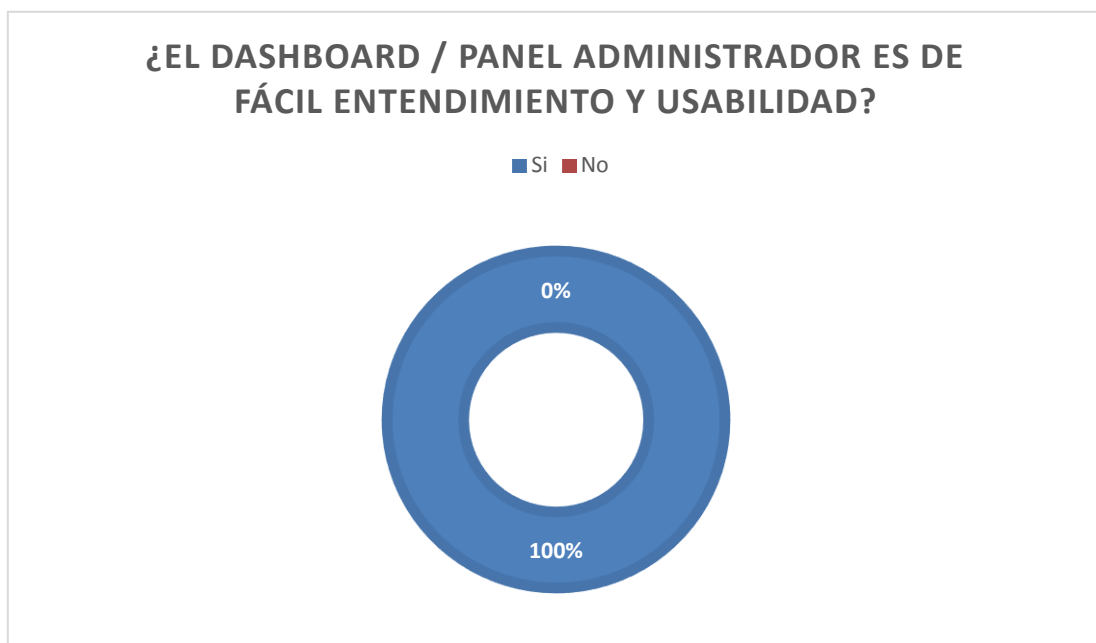
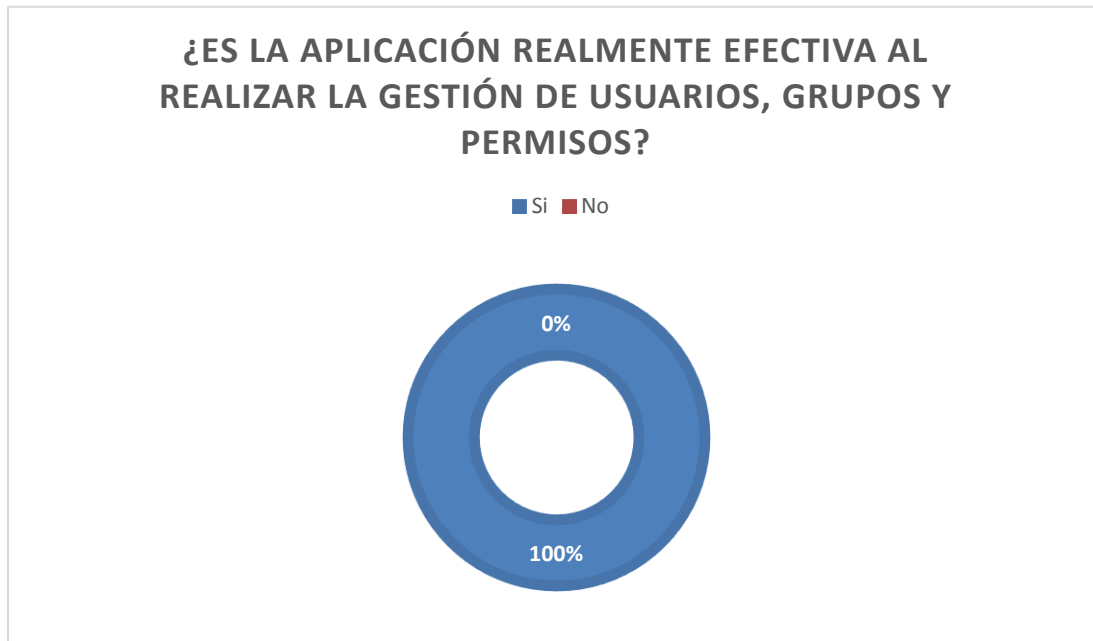


Figura 54 Pregunta 4 Administrador

Tabla 46*Pregunta 5 al administrador*

5.- ¿Es la aplicación realmente efectiva al realizar la gestión de usuarios, grupos y permisos?	
Si	1
No	0

**Figura 55** Pregunta 5 Administrador**Tabla 47***Pregunta 6 al administrador*

6.- ¿La aplicación cumple con los estándares de rapidez y procesamiento de peticiones?	
Si	1
No	0

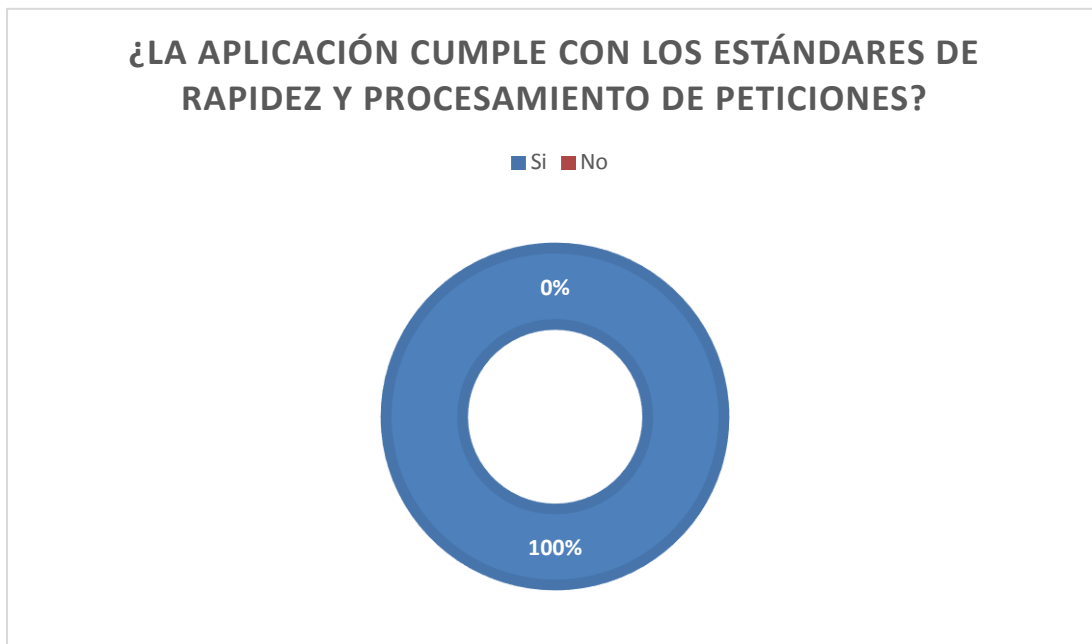


Figura 56 Pregunta 6 Administrador

Interpretación de los Resultados

De acuerdo con los resultados de los instrumentos aplicados, se puede llegar a las siguientes conclusiones:

Respecto a las preguntas aplicadas a los profesores, se evidencia comodidad al momento de usar la aplicación, Y en lo referente a la receptividad de su interfaz los resultados arrojan que más de la mitad de los encuestados se encuentran conformes con las características de usabilidad que posee y consideran su interfaz amigable y de practico uso, por otro lado, al momento de evaluar el impacto que la puesta en práctica del LMS desarrollado ha tenido en la enseñanza de los encuestados cabe arrojar que un 80% de los encuestados afirma que representa una herramienta significativa en su modelo de impartir conocimiento y ha tenido un impacto positivo en esta, a su vez, alrededor de un 80% afirma también que la herramienta supone un método eficiente y eficaz a la hora de evaluar el nivel de aprendizaje obtenido de los estudiantes y cuando conviene evaluar su nivel de adaptabilidad más de la mitad de los encuestados expresó que le consideraría una herramienta indispensable de enseñanza en cualquier institución educativa.

Por otro lado, en lo que respecta a las funcionalidades del sistema y a su entendimiento los datos arrojan que los encuestados no tuvieron ningún obstáculo al momento de asignar pruebas ni para servirse del módulo del panel de tareas ni de los datos presentados por los dashboards.

En otro orden de ideas, en lo que respecta a la encuesta aplicada a los estudiantes, previa a su interacción con el sistema y a la aplicación de la prueba que determina su tipo de aprendizaje un 100% de los encuestados manifiesta que le ha sido de utilidad para su proceso de aprendizaje mientras que un 75% afirma que le ha resultado útil en el modo de adquirir conocimiento, a su vez, en lo referente al entendimiento del funcionamiento del LMS y su usabilidad un 75% afirma en base a su experiencia vivida que el sistema posee una interfaz perceptiva y amigable y alrededor de ese mismo porcentaje de encuestados manifiesta que ha tenido un impacto positivo el uso de un LMS, en esa línea de argumentos, un 100% expresa que en general el método de evaluación del sistema le supone un método eficaz, practica y eficiente.

Así mismo, alrededor de un 83% de los encuestados manifiesta que el proceso de responder a las preguntas que componen a cada prueba se ha caracterizado por ser sencillo, y acerca del mismo porcentaje expresa que el sistema ha incidido en un aumento favorable de su índice académico.

Finalmente, en lo que respecta a la encuesta aplicada al administrador del sistema, este manifiesta que la aplicación posee adaptabilidad y versatilidad en cuanto a los dispositivos que pueden utilizarse para acceder al sistema y que se puede acceder a este mediante cualquier navegador web que sea de la preferencia del usuario, además, el mismo afirma que la aplicación web presenta una interfaz amigable para con el Usuario y aunado a eso que la aplicación resulta efectiva al momento de gestionar cambios varios en el sistema y realizar el mantenimiento del mismo.

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

Teniendo en cuenta todo lo anteriormente planteado se puede concluir que el ecosistema informático para la enseñanza y normalización del pensamiento computacional ha tenido un impacto positivo en cuanto al índice académico de los estudiantes y supone una herramienta valiosa mediante la cual los profesores asignan y diseñan los contenidos a ser evaluados, pero a través de los instrumentos más propicios al tipo de aprendizaje de cada individuo.

Se utilizó la herramienta AppGini que permitió desarrollar el proyecto de manera adecuada; mejorando tiempos, costes y procesos que hubieran sido más extensos sin la utilización de la herramienta.

La falta de conocimiento sobre el uso apropiado de las herramientas de versionamiento de código es determinante en la planificación de un proyecto relacionado a la escalabilidad y mantenibilidad del mismo.

La metodología SCRUM es apropiada, por su agilidad y su historial de desarrollo, se puede lograr un producto funcional en base a los requerimientos correctamente levantados.

La implantación y el desarrollo del Pensamiento Computacional son relevantes en la vida cotidiana ya que es un concepto que forja una metodología de análisis para ofrecer un sinnúmero de beneficios educativos e intelectuales para los estudiantes.

Teniendo en cuenta todo lo anteriormente planteado se puede concluir que el ecosistema informático para la enseñanza y normalización del pensamiento computacional ha tenido un impacto positivo en cuanto al índice académico de los estudiantes y supone una herramienta valiosa mediante la cual los profesores asignan y diseñan los contenidos a ser evaluados, pero a través de los instrumentos más propicios al tipo de aprendizaje de cada individuo.

4.2 Recomendaciones

Habiendo planteado lo anterior se acota como recomendación la implantación de más ecosistemas a fines que permitir diagnosticar los procesos más asertivos de enseñanza y solucionar las falencias en el sistema de educación que a la par le proporcione herramientas a los educadores que signifiquen una mejor toma de decisiones que puedan beneficiar al sistema educativo para mejorar el proceso de aprendizaje.

La comparación de las herramientas de desarrollo de sistemas debe ser minuciosa y tomar su debido tiempo, a fin de tener la mejor opción para el desarrollo de un sistema en base a los requerimientos funcionales y no funcionales.

Para el desarrollo de proyectos se recomienda utilizar herramientas como GIT o SVN en el proyecto desde cero; a fin de evitar inconvenientes en el mismo.

Se deben establecer normativas para el correcto desarrollo de software; establecer una metodología desde 0 y terminar el proyecto con la misma.

Se recomienda llevar un registro de investigaciones o proyectos relacionados con el Pensamiento Computacional aplicados en todos los niveles educativos.

Habiendo planteado lo anterior se acota como recomendación la implantación de más ecosistemas a fines que permitir diagnosticar los procesos más asertivos de enseñanza y solucionar las falencias en el sistema de educación que a la par le proporcione herramientas a los educadores que signifiquen una mejor toma de decisiones que puedan beneficiar al sistema educativo para mejorar el proceso de aprendizaje.

REFERENCIAS BIBLIOGRÁFICAS

- ACM. (Marzo de 2006). *ACM*. Obtenido de ACM: <https://www.cs.cmu.edu/~15110-s13/Wing06-ct.pdf>
- Amazon. (2017). *Amazon Web Services*. Recuperado el 20 de Agosto de 2017, de Amazon Web Services: <https://aws.amazon.com/es/>
- AppGini. (2017). *AppGini*. Obtenido de AppGini: <https://bigprof.com/appgini/>
- Bitbucket. (2017). *Bitbucket*. Recuperado el 21 de Agosto de 2017, de Bitbucket: <https://bitbucket.org/product/features>
- Bocconi, S., Chiocciariello, A., Dettori, G., Ferrari, A., & Engelhardt, K. (2016). *El Pensamiento Computacional en la Enseñanza Obligatoria (Computhink)*. doi:10.2791/792158
- Borrega, J. (2016). El Pensamiento Computacional en la educación obligatoria, Una revisión sistemática de la literatura. *El pensamiento computacional en la educación obligatoria*. Cáceres, España: Universidad de Extremadura.
- Borrega, J. A. (2016). *El Pensamiento Computacional en la educación obligatoria*. Cáceres.
- Canaza, B. (20 de agosto de 2009). *Ingeniería de Software*. (U. U. Bolivariana, Productor, & Universidad Unión Bolivariana) Recuperado el 24 de junio de 2015, de Metodologías Ágiles RUP: http://ingenieriadesoftware.mex.tl/images/18149/P5_METODOS%20AGILLES%20RUP_2_.pdf
- Charoula, A., Voogt, J., Fluck, A., Webb, M., Cox, M., Malyn-Smith, J., & Zagami, J. (2016). A K-6 Computational Thinking Curriculum Framework: Implications for Teacher Knowledge. *Journal of Educational Technology & Society*, 3.
- Code.org. (2013). *code.org*. Obtenido de www.code.org
- CodeAcademy. (2017). *CodeAcademy*. Recuperado el 18 de 08 de 2017, de CodeAcademy: <https://www.codecademy.com/es>
- Codejobs. (2013). *Codejobs*. Obtenido de Codejobs: <https://www.codejobs.biz/>
- Csizmadia. (2015).
- Donnelly, M. (2000). *An introduction to LDAP*. Obtenido de An introduction to LDAP.: http://redeslinux.com/manuales/openldap/introduccion_ldap.pdf
- Ecuador, S. d. (2017). *Quipux - Sistema de Gestión Documental*. Obtenido de Quipux - Sistema de Gestión Documental: <https://www.gestiondocumental.gob.ec>

- Europa Ec. (2016). *Europa Ec*. Obtenido de Europa Ec: <http://ec.europa.eu/social/>
- García, A. A. (2007). *Desarrollo de herramientas web de gestión docente*. Cartagena: Escuela Técnica Superior de Ingeniería de Telecomunicación.
- Git. (2017). *Git*. Obtenido de Git: <https://git-scm.com/>
- Git. (2017). *Git*. Recuperado el 09 de Septiembre de 2017, de Git: <https://git-scm.com/book/es/v1/Empezando-Una-breve-historia-de-Git>
- Git. (2017). *Git --distributed-is-the-new-centralized*. Obtenido de <https://git-scm.com/book/es/v1/Empezando-Fundamentos-de-Git>
- Inc., A. (2017). *Alfresco - ECM and BPM Software*. Obtenido de Alfresco - ECM and BPM Software: <https://www.alfresco.com>
- InnovaEDU. (2017). *InnovaEDU*. Recuperado el 22 de Agosto de 2017, de InnovaEDU: <http://innovaedu.espe.edu.ec/>
- Kniberg. (2007). *Scrum y XP desde las Trincheras*. C4Media.
- Kolodner. (2010). Taller de Alcance y Naturaleza del Pensamiento Computacional. (C. N. (NRC), Entrevistador) EE.UU.
- Lee. (2011).
- M. J., & M. J. (2003). Recuperado el 05 de Febrero de 2017, de <http://www.enterprisearchitecture-info.com/Images/MDA/>
- M. J., & M. M. (2014). La Usabilidad Web en el Desarrollo de Software Dirigido por Modelos. *WICC 2014 XVI Workshop de Investigadores en Ciencias de la Computación*, 597-600.
- Mariño, C. V. (2008). *Programacion en PHP5. Nivel Básico*. Ferrol.
- Nipro. (2017). *Bushido Ecuador*. Obtenido de Bushido Ecuador: bushidoecuador.com
- Nipro. (2017). *Nipro-OCD Pedidos*. Obtenido de Nipro-OCD Pedidos: nipro-ocd.com/pedidos
- Olabe, X. B. (2015). Pensamiento Computacional a través de la Programación : Paradigma de Aprendizaje. *Revista de Educación a Distancia*.
- Papert, S. (1980). *Mindstorms: Children, Computers, and Powerful Ideas*.
- Pfleeger, S. (2002). *Ingeniería de Software, Teoría y Práctica*. Buenos Aires: Prentice Hill.
- Piano Nazionale Scuola Digitale. (PNSD). (2017). *Piano Nazionale Scuola Digitale. (PNSD)*. Obtenido de Piano Nazionale Scuola Digitale. (PNSD): http://www.istruzione.it/scuola_digitale/index.shtml

- ProyectosAgiles. (2017). *ProyectosAgiles*. Obtenido de ProyectosAgiles: <https://proyectosagiles.org/que-es-scrum/>
- RoyalSociety. (2012).
- Schwaber, K. &. (2013). *La Guía de SCRUM*.
- Scratch. (2017). Recuperado el 18 de 08 de 2017, de Scratch: <https://scratch.mit.edu/about>
- ScratchEd Team, M. (2015). *Computational Thinking webinars*.
- Silapachote, P., & Srisuphab, A. (9 de Diciembre de 2016). Teaching and Learning Computational Thinking through Solving Problems in Artificial Intelligence. *IEEE International Conference on TALE*, 4. doi:978-1-5090-5598-2/16/
- Synacor, I. (2017). *Zimbra*. Obtenido de <https://www.zimbra.com>
- Toledo, A. E. (2013). *MySQL*.
- WebRatio. (01 de Enero de 2012). *WebRatio*. Recuperado el 05 de Febrero de 2017, de WebRatio: http://www.webratio.com/site/CMSMobileStyle/img/webratio_official_logo.png
- Wing, J. (2006). *Computational thinking*. *Communications of the ACM*.
- Zhu, B. C. (2010). *U.S. Patent No. 7,690,026*. Washington, DC: U.S. Patent and Trademark Office. Obtenido de U.S. Patent No. 7,690,026. Washington, DC: U.S. Patent and Trademark Office.: <https://www.google.com/patents/US7690026>