

UNIVERSIDAD MAYOR DE SAN ANDRES
FACULTAD DE INGENIERIA
CARRERA DE INGENIERIA ELECTRONICA
P.E.T.E.N.G.



**“LABORATORIO VIRTUAL DE IMPLEMENTACION Y CONFIGURACION DE
SERVICIOS DE INTERNET EN UNA INTRANET EN LA FACULTAD DE
INGENIERIA – CARRERA INGENIERIA ELECTRONICA”**

Postulante: Univ. Jhonn Charles Bautista Humerez
Tutor : Ing. Hugo Balderrama Barrios

LA PAZ – BOLIVIA

2019



**UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE INGENIERIA**



LA FACULTAD DE INGENIERIA DE LA UNIVERSIDAD MAYOR DE SAN ANDRÉS AUTORIZA EL USO DE LA INFORMACIÓN CONTENIDA EN ESTE DOCUMENTO SI LOS PROPÓSITOS SON ESTRICTAMENTE ACADÉMICOS.

LICENCIA DE USO

El usuario está autorizado a:

- a) Visualizar el documento mediante el uso de un ordenador o dispositivo móvil.
- b) Copiar, almacenar o imprimir si ha de ser de uso exclusivamente personal y privado.
- c) Copiar textualmente parte(s) de su contenido mencionando la fuente y/o haciendo la cita o referencia correspondiente en apego a las normas de redacción e investigación.

El usuario no puede publicar, distribuir o realizar emisión o exhibición alguna de este material, sin la autorización correspondiente.

TODOS LOS DERECHOS RESERVADOS. EL USO NO AUTORIZADO DE LOS CONTENIDOS PUBLICADOS EN ESTE SITIO DERIVARA EN EL INICIO DE ACCIONES LEGALES CONTEMPLADAS EN LA LEY DE DERECHOS DE AUTOR.

Dedico este trabajo especialmente a Dios porque sin su amor y misericordia, esto no hubiera sido posible, a mi Esposa por sus palabras de aliento y apoyo incondicional, a mis padres y a la Universidad de Mayor de San Andrés por brindarme los conocimientos para llegar a obtener el título de Ingeniero Electrónico, mención sistemas de computación.

AGRADECIMIENTOS

A Dios todo poderoso que siempre nos dio sabiduría en la realización de este proyecto.

La familia porque siempre me brindó su apoyo y confianza en la realización de este proyecto y durante la carrera de Ingeniería Electrónica.

La Universidad que me dio la oportunidad de formarme profesionalmente y ayudó a crecer y a madurar con todas las situaciones vividas en el transcurso de mi carrera.

Expreso mi más profundo agradecimiento al Ing. Hugo Balderrama Barrios por su contribución como asesor y mentor durante el desarrollo de este informe, fundamental para el éxito de esta investigación.

Resumen

Los laboratorios virtuales son simuladores que modelan un laboratorio real o un experimento, se ejecuta en una computadora, se consideran simuladores porque físicamente no existen los equipos, los dispositivos ni los materiales necesarios para ejecutar el experimento, sin embargo, la simulación permite repetir múltiples veces los experimentos, con diversos parámetros, hasta que se comprenden los principios de funcionamiento del sistema.

Los laboratorios virtuales son programas especializados que permiten actuar virtualmente del mismo modo que en la realidad, facilitan la vinculación del conocimiento nuevo con la experiencia, además permiten cometer errores sin provocar daños en equipos costosos y tomar experiencia sobre el uso de equipo sofisticado a un precio asequible, estas ventajas, hacen de los laboratorios virtuales una herramienta muy útil para el aprendizaje de las ciencias experimentales y de las Ingenierías, las cuales requieren una constante capacitación en el uso de equipo, costoso, escaso y delicado.

El desarrollo del proyecto (memoria) comprende de cinco capítulos: 1) elaboración del perfil del proyecto donde se resume la idea del proyecto que se quiere realizar, 2) el fundamento teórico que es un conjunto de conocimientos que nos permitirá delimitar teóricamente los conceptos planteados, 3) La ingeniería de proyectos es aquella etapa en la que se definen los recursos necesarios para la ejecución de planes o tareas, 4) Costos donde analizamos el costo del software diseñado mediante metodologías propuestas en capítulos anteriores y 5) Conclusiones y recomendaciones del proyecto tanto para el docente y estudiante.

Abstract

Virtual labs are simulators that model a real laboratory or an experiment, run on a computer, are considered simulators because physically there are no equipment, devices or materials needed to run the experiment, however, the simulation allows multiple repetitions of the experiments, with various parameters, until the operating principles of the system are understood.

Virtual labs are specialized programs that allow you to act virtually in the same way as in reality, they facilitate the linking of new knowledge with experience, they also allow to make mistakes without causing damage to expensive equipment and gain experience on the use of sophisticated equipment at an affordable price, these advantages, make virtual laboratories a very useful tool for learning experimental sciences and engineering, which require constant training in the use of expensive, scarce and delicate equipment.

The development of the project (memory) comprises five chapters: 1) preparation of the profile of the project where the idea of the project is summarized, 2) the theoretical foundation that is a set of knowledge that will allow us to theoretically delimit the concepts presented , 3) Project engineering is that stage in which the necessary resources for the execution of plans or tasks are defined, 4) costs where we analyze the cost of the software designed by methodologies proposed in previous chapters and 5) the conclusions and recommendations of the project for both the teacher and the student.

INDICE

Resumen	iii
Abstract.....	iv
INDICE.....	v
Lista de Figuras	vii
Lista de Tablas.....	viii
Capítulo 1. Introducción.....	1
Introducción.....	1
1.1. Antecedentes.....	1
1.2. Descripción del problema.....	2
1.3. Objetivos.....	3
1.3.1. Objetivo general.....	3
1.3.2. Objetivos específicos.....	3
1.4. Justificación	4
1.5. Alcances y limites.....	4
Capítulo 2.Fundamento teórico	6
2.1. Introducción.....	6
2.2 Evolución de los medios de cómputo y su empleo como medio de enseñanza.....	7
2.3. La integración de recursos en el medio informático.....	8
2.4. El software educativo.....	11
2.4.1. Características esenciales del software educativo.....	11
2.4.2 Clasificación del software educativo.....	12
2.4.3 Estrategia para la evaluación y selección de un software educativo	17
2.4.4 Metodología para la elaboración del guion	25
2.5 Metodología XP	39
2.5.1 Alcance de XP	43
2.5.2. Planeación.....	44
2.6. Modelo Cocomo II.....	50
2.6.1 Características generales.....	51

2.6.2 Modelos de estimación	51
2.7. Herramientas de Desarrollo.	53
Capítulo 3. Ingeniería del proyecto	55
3.1. Búsqueda de Documentación.....	56
3.2. Análisis de requerimientos.....	56
3.3. Selección de software	57
3.4 Instalación y configuración de servicios LTSP	58
3.5. Metodología del laboratorio virtual.	61
3.4.1. Etapa I. Planificación.....	61
3.4.2. Etapa II. Diseño	62
3.4.3 Etapa III. Desarrollo	79
Capítulo 4 Costos	85
4.1. Introducción.	85
4.2. Estudio de factibilidad	85
4.2.1. Factibilidad técnica.....	86
4.2.2. Factibilidad Económica.	87
4.2.3. Factibilidad Operacional.....	89
4.3 Costos del Proyecto.	90
4.3.1. Costo de desarrollo del sistema.	90
4.3.2 Costos fijos.	97
4.3.3 Costos variables	98
4.3.4 Costos Totales.....	99
Capítulo 5. Conclusiones y Recomendaciones.....	102
Bibliografía.....	104

Lista de Figuras

Figura 1. Diagrama de diseño del Prototipo	55
Figura 7. Diagrama de Caso de Uso del Estudiante	63
Figura 8. Diagrama de Caso de Uso Ingreso al sistema	64
Figura 9. Diagrama de Caso de Uso Registro de Material	66
Figura 10. Administrar Usuarios	67
Figura 11. Caso de Uso Consultas y Reportes	69
Figura 12. Diagrama de Clases General	70
Figura 13. Diagrama de Secuencia del proceso de Ingreso al sistema	71
Figura 14. Diagrama de Secuencia del proceso de Registro del Cliente	72
Figura 15. Diagrama de Secuencia del proceso de Reportes	73
Figura 16. Diagrama de Actividades, Ingreso de Usuarios	75
Figura 17. Diagrama de Colaboración, Registro Cliente	76
Figura 18. Diagrama de Colaboración, Reportes	77
Figura 19. Diagrama de Componentes General	78
Figura 20. Diagrama de Despliegue	79
Figura 21. Pantalla de Inicio de Sesión	80
Figura 22. Pantalla de autenticación para el ingreso	80
Figura 23. Pantalla de los temas o módulos	81
Figura 24. Pantalla de avance de curso	82
Figura 25. Lista de estudiantes	82
Figura 26. Reporte de Estudiantes	83
Figura 27. Administración de usuarios	84
Figura 28. Inicio de instalación Ubuntu server	108
Figura 29. Esquema de tipos de software Educativo	126
Figura 30 . Pasos de la Metodología xp	127

Lista de Tablas

Tabla 1. Historia de Usuario "Autenticar Usuario"	61
Tabla 2. Historia de Usuario "Gestionar Cuenta Usuario"	62
Tabla 3. Historia de Usuario "Gestionar Producto"	62
Tabla 4. . Diagrama de Caso de Uso del Estudiante.....	63
Tabla 5. Caso de Uso Ingreso al Sistema	65
Tabla 6. Caso de Uso Registro de Módulos de aprendizaje	67
Tabla 7. Administrar Usuarios.....	68
Tabla 8. Caso de Uso Consultas	70
Tabla 9. Hardware Existente	86
Tabla 10. Software.....	87
Tabla 11. Equipo de Red	87
Tabla 12. Costo de Computadora	88
Tabla 13. Costo de Internet.....	88
Tabla 14. Costo de Software.....	89
Tabla 15. Recuento de Características.....	91
Tabla 17. Evaluación de Factores de Ajuste.....	92
Tabla 18. Selección de Valores de escala.....	94
Tabla 19. Selección de Drivers de Coste.....	95
Tabla 20. Software para el Desarrollo del Sistema	98
Tabla 21. Resumen de Costos variables	99
Tabla 22. Resumen de Costos Totales.....	100
Tabla 23. Comparación de sistemas operativos (Linux vs Windows)	107

Capítulo 1

Introducción

1.1. Antecedentes

En la actualidad, a nivel mundial existe un aumento en el desarrollo de Laboratorios Virtuales con el objetivo de cubrir las deficiencias de los cursos tanto a distancia como presenciales. En Bolivia los sistemas educativos virtuales se presentan en diferentes áreas de acuerdo a los requerimientos del usuario, el avance de las investigaciones en aspectos como el desarrollo de un Laboratorio virtual o programas interactivos, hace un campo medianamente explorado y los alcances logrados han utilizado herramientas de internet y programación que avanzan progresivamente con la tecnología tanto software como hardware. Por lo anterior es necesario diseñar cada vez nuevas y mejores aplicaciones vía internet.

Algunos proyectos de este tipo que se desarrollaron en la Facultad de Ingeniería Electrónica

Título: Laboratorio virtual de líneas de transmisión

Autor: Ayala Flores, Luis Marcelo

Universidad Mayor de San Andrés, Facultad de Ingeniería, Ingeniería Electrónica

Título: Laboratorio virtual para el aprendizaje de Electrónica

Autores: Calle Tarqui, Felisa, Laura Quispert, Marianela

Universidad Mayor de San Andrés, Facultad de Ingeniería, Ingeniería Electrónica.

1.2. Descripción del problema

A nivel mundial, el avance tecnológico representa un factor indispensable para el desarrollo de conocimientos, en donde la computadora es una fuerte herramienta que permite obtener y administrar información, su uso constante da como resultado el deterioro de hardware, lo cual genera un problema presente en todo el mundo, perjudicando el rendimiento de actividades en organizaciones o centros de formación académica, ya que la educación es el mayor contribuyente de futuras generaciones.

La reutilización de estas computadoras como servidores, o como terminales remotas se podrá utilizar. Debido a la inexistencia de un laboratorio virtual, resulta difícil desarrollar el proceso enseñanza-aprendizaje desaprovechando las capacidades de los estudiantes al no realizar las respectivas prácticas de los conocimientos adquiridos teóricamente.

La Facultad de Ingeniería electrónica, entre los objetivos principales de la Facultad está la formación de profesionales e investigadores de alto nivel, que puedan insertarse exitosamente en el mercado laboral o continuar con una carrera académica. De igual forma, en la Facultad se desarrollan diversos proyectos de investigación que aportan a la solución de problemas prácticos de interés para la sociedad, y que han sido difundidos a través de publicaciones científicas relevantes.

1.3. Objetivos

1.3.1. Objetivo general

Implementar un laboratorio virtual, de instalación y configuración de los servicios de internet, en un servidor Linux, en la Facultad de Ingeniería Electrónica entorno a una intranet, que permita optimizar los recursos tecnológicos para el fortalecimiento del aprendizaje académico.

1.3.2. Objetivos específicos.

- Analizar y diagnosticar el estado del hardware de las computadoras para verificar su disponibilidad de uso.
- Desarrollar la topología de red a implementarse para la creación del laboratorio virtual de la Facultad de Ingeniería, Ingeniería electrónica mención Sistemas de Computación.
- Configurar el laboratorio virtual con el servidor de redes LTSP y sus clientes para la carrera de Ingeniería electrónica.
- Realizar pruebas al sistema creado para un laboratorio virtual.
- Realizar pruebas en la interacción de cliente al servidor Linux Terminal Server Project.
- Realizar la documentación correspondiente al terminar el proyecto.

1.4. Justificación

El progreso actual en las redes tecnológicas, ha llegado al grado en que es fácil encontrar un computador conectado a una red, ya que en toda institución existe la necesidad de agilizar procesos y transportar la información, muchas empresas debido a la nueva generación de tecnologías se ven en la obligación de actualizar sus ordenadores, generando chatarras tecnológicas, muchas de estas instituciones no cuentan con los recursos necesarios para ir a la par con la tecnología, es por eso que se debe considerar la implementación de un laboratorio virtual usando un servidor, que nos permita reutilizar estos equipos con baja prestación de recursos.

La implementación de un laboratorio virtual ayudará al ahorro de recursos económicos, ya que al utilizar las computadoras que se encuentran fuera de servicio puedan ser reutilizadas, para la creación de una nueva sala de práctica y aportará el aprendizaje de los estudiantes que contribuirá con el desarrollo sustentable de la Universidad.

En el aspecto ambiental el proyecto se justifica ante la necesidad de proteger el entorno contra la descomposición de material tóxico que afecta al ecosistema, aprovechando los recursos considerados desechos con la finalidad de convertirlos en fuentes de trabajo académico, a su vez concientizar a los estudiantes de la importancia del cuidado del medio ambiente creando así un efecto socio-ambiental entre ellos.

1.5. Alcances y límites

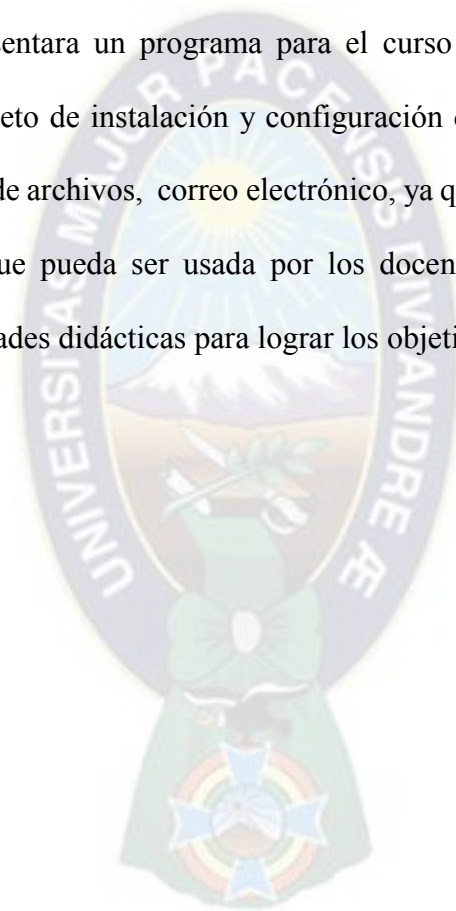
Con este proyecto se pretende lograr:

La instalación, configuración y utilización de servidores en base a Linux que se usarán como entorno de aprendizaje/enseñanza

La creación de una interfaz gráfica que permite configurar los servicios de internet.

La implementación de un software que busque el aprendizaje del estudiante.

Este proyecto, presentara un programa para el curso virtual conectado al servidor sin desarrollar un curso completo de instalación y configuración de servidores como base de datos, protocolo de transferencia de archivos, correo electrónico, ya que lo que se quiere es proponer una herramienta tecnológica que pueda ser usada por los docentes para crear sus propios cursos virtuales y formar sus unidades didácticas para lograr los objetivos de aprendizaje.



Capítulo 2.

Fundamento teórico

2.1. Introducción.

Los laboratorios virtuales son programas especializados que permiten actuar virtualmente del mismo modo que en la realidad, facilitan la vinculación del conocimiento nuevo con la experiencia, además permiten cometer errores sin provocar daños en equipos costosos y tomar experiencia sobre el uso de equipo sofisticado a un precio asequible, estas ventajas, hacen de los laboratorios virtuales una herramienta muy útil para el aprendizaje de las ciencias experimentales y de las Ingenierías, las cuales requieren una constante capacitación en el uso de equipo, costoso, escaso y delicado. (Juárez, 2013)

Para lograr Ingenieros en electrónica en la mención de Sistemas Computación bien capacitados, se requiere de un variado equipamiento que permita adquirir y fortalecer las habilidades y conocimientos de los estudiantes en disciplinas como electrónica, redes de telecomunicaciones, robótica, procesamiento de información y matemáticas. La inversión necesaria para cubrir las necesidades de uso, mantenimiento y actualización del equipamiento para dar servicio a todos los estudiantes inscritos en una institución es muy elevada y para las instituciones públicas, que generalmente están limitadas en recursos económicos, representa un compromiso difícil de cumplir.

2.2 Evolución de los medios de cómputo y su empleo como medio de enseñanza.

La computación viene a ser la tercera dentro de un grupo de creaciones tecnológicas que revolucionaron el universo de los medios de enseñanza. La primera de ellas fue el cine y la segunda, la televisión. Sin embargo, el antecedente directo del uso de la computación como medio de enseñanza puede ser encontrado en las máquinas de enseñar y en la teoría de la enseñanza programada. (Fernández, 2005)

En la primera mitad del siglo, investigadores como Pressey y Skinner desarrollaron máquinas mecánicas de enseñar y enunciaron los principios de la enseñanza por medio de las mismas, los cuales con muy pocas variaciones se encuentran presentes en la mayor parte de los medios de enseñanza computacionales actuales. Al aparecer en los años 40 y 50 las primeras computadoras fueron desechadas las máquinas mecánicas y se pasó a trabajar con la novedosa tecnología computacional. Estos proyectos dominaron totalmente el horizonte del uso de la computación como medio durante los años 50 y 60. (Fernández, 2005)

Los años 70 estuvieron marcados por el desarrollo de macroproyectos educativos por medio de computadoras, hasta llegar a los años 80, cuando el boom de las computadoras personales llegó hasta las escuelas motivando a todo un ejército de profesores - programadores a probar fuerzas en el terreno del software para la enseñanza, provocando una verdadera inundación de programas que se movían en un amplio espectro, desde la excelencia hasta la más pésima calidad. (Fernández, 2005)

2.3. La integración de recursos en el medio informático.

Las computadoras representan un nivel nuevo y cualitativamente superior dentro de la escala de los medios de enseñanza. Ellas aportan una forma nueva de interactividad, que las diferencia de todo lo antes empleado como medio y que debe ser considerado como el principal indicador de la necesidad de su uso.

Son capaces, además, de recibir y procesar información procedente del profesor o de los estudiantes y pueden, incluso, de acuerdo con esta información y a las características del programa que las controla, modificar convenientemente la secuencia de la información ofrecida de forma tal, que su exposición resulte lo más apropiada posible a la dinámica interna del proceso docente durante la clase. (Aguilera, 2011)

Posibilitan almacenar información sobre el desarrollo y los resultados de un estudiante en una sesión de trabajo con el objetivo de poder analizarla en sesiones de trabajo posteriores y actuar de acuerdo con esa información. Esta posibilidad representa una significativa mejora cualitativa, y es realmente una lástima el poco uso que hacen de ella la inmensa mayoría de los medios computacionales de hoy en día. (Aguilera, 2011)

Entre las principales características podemos enumerar las siguientes:

Permiten la individualización de la enseñanza y por tanto una atención diferenciada a los estudiantes.

Permiten la constante activación del estudiante, por lo que aumenta su actividad cognoscitiva, los estudiantes pasan de espectadores a actores en el aprendizaje.

Permiten la comprobación y/o corrección (retroalimentación) directa e inmediata de los resultados del aprendizaje.

Presentan amplias posibilidades de cálculo, velocidad de trabajo, versatilidad y flexibilidad, capacidad de almacenamiento de grandes masas de datos, efectos de animación, sonidos, uso de colores, así como capacidad de elección y decisión. (Aguilera, 2011)

Fácil reutilización del material de estudio, bajo distintas condiciones de trabajo, incluso sin la presencia del profesor.

Capacidad de modificación y perfeccionamiento del material a bajo costo de materiales.

Permiten transmitir mayores volúmenes de información en menor tiempo y en forma controlada.

Permiten dirigir el proceso de reforzamiento, auto aprendizaje y evaluación de los estudiantes en forma individual.

En medio del desarrollo tecnológico resulta casi imposible prescindir de un medio tan valioso como lo es la computadora, que indiscutiblemente puede conducirnos al logro de una mejor y mayor optimización del proceso de enseñanza - aprendizaje. Esto constituye uno de los objetivos más importantes de nuestro sistema educacional; es por ello que se plantea el logro de un aprendizaje activo y participativo donde cada estudiante sea capaz de descubrir nuevos conocimientos y aplicarlos a la obtención de otros, en un ambiente de trabajo en el que la computadora juegue un importante rol. (Aguilera, 2011)

En la actualidad en Informática se suele identificar como multimedia a la integración de dos o más medios de comunicación que pueden ser controlados o manipulados por el usuario en una computadora. O sea, es un sistema informático interactivo, controlable por el usuario, que integra diferentes medios como el texto, el vídeo, la imagen, el sonido y las animaciones.

La utilización de la computadora en el aula implica un mayor grado de abstracción de las acciones, una toma de conciencia y anticipación de lo que muchas veces hacemos "automáticamente". Desde los planos afectivo y social, el manejo de la computadora permite el trabajo en equipo, apareciendo así la cooperación entre sus miembros y la posibilidad de intercambiar puntos de vista, lo cual favorece también sus procesos de aprendizaje. (Rey., 2011)

Manejar una computadora permite a los estudiantes mejorar su autoestima, sintiéndose capaces de lograr cosas, realizar proyectos, crecer, entre otros. Aparece también la importancia constructiva del error que permite revisar las propias equivocaciones para poder aprender de ellas. Así el estudiante es un sujeto activo y participante de su propio aprendizaje que puede desarrollar usos y aplicaciones de la técnica a través de la inserción de las nuevas tecnologías. El método informático de razonar es concretamente el método de diseño descendente de algoritmos que es positivamente enriquecedor como método sistemático y riguroso de resolución de problemas y de razonamiento. De tal manera el docente, debe dominar una forma de trabajar metódicamente, que enseñe a pensar y que permita el aprendizaje por descubrimiento, el desarrollo inteligente y la adquisición sólida de los patrones del conocimiento. El estudiante, estará preparado entonces para distinguir claramente cuál es el problema y cuál es el método más adecuado de resolución. La computadora es además, para el docente, un instrumento capaz de revelar, paso a paso, el avance intelectual del estudiante. (Rey., 2011)

2.4. El software educativo.

Los software son los programas que utilizan las computadoras para trabajar o funcionar, por lo tanto, un software educativo es cualquier programa computacional cuyas características estructurales y funcionales le permiten servir de apoyo a la enseñanza, el aprendizaje y la administración educacional. El software educativo es uno de los recursos más interesantes, necesarios y motivadores que se pueden utilizar para ayudar al desarrollo del proceso enseñanza-aprendizaje. Al introducir las computadoras en la educación, se produjo una forma más amena de aprender, logrando además la retención a más largo plazo del aprendizaje. (Fernández, 2005)

2.4.1. Características esenciales del software educativo.

Dado que la adquisición de los conocimientos no es activa para la mayoría de los estudiantes, la personalización se hace entonces un poco más compleja, por lo que se impone por parte de los docentes el empleo de nuevos métodos y estilos de enseñanza y unido a esto, el empleo de nuevos métodos y estilos de aprendizaje por parte de los estudiantes, en los que juegan un papel importante el uso de la computadora y por tanto la utilización del software educativo. (Fernández, 2005)

Los programas educativos pueden tratar las diferentes materias (Matemáticas, Idiomas, Geografía, Dibujo, etc.), de formas muy diversas (a partir de cuestionarios, facilitando una información estructurada a los estudiantes, mediante la simulación de fenómenos) y ofrecer un entorno de trabajo más o menos sensible a las circunstancias de los estudiantes y más o menos rico en posibilidades de interacción; pero todos comparten cinco características esenciales:

Son materiales elaborados con una finalidad didáctica, como se desprende de la definición.

Utilizan la computadora como soporte en el que los estudiantes realizan las actividades que ellos proponen.

Son interactivos, contestan inmediatamente las acciones de los estudiantes y permiten un diálogo y un intercambio de informaciones entre la computadora y los estudiantes.

Individualizan el trabajo de los estudiantes, ya que se adaptan al ritmo de trabajo cada uno y pueden adaptar sus actividades según las actuaciones de los estudiantes.

Son fáciles de usar. Los conocimientos informáticos necesarios para utilizar la mayoría de estos programas son similares a los conocimientos de electrónica necesarios para usar un vídeo, es decir, son mínimos, aunque cada programa tiene unas reglas de funcionamiento que es necesario conocer. (Fernández, 2005)

El uso de software educativo en el proceso de enseñanza-aprendizaje puede ser:

a) *Por parte del docente:* se manifiesta cuando el docente opera directamente con el software y el estudiante actúa como receptor del sistema de información. La generalidad plantea que este no es el caso más productivo para el aprendizaje.

b) *Por parte del estudiante:* se evidencia cuando el estudiante opera directamente el software educativo, pero en este caso es de vital importancia la acción dirigida por el docente. (Novo, 2013)

2.4.2 Clasificación del software educativo.

Podemos definir tres grupos de la siguiente manera para realizar su clasificación:

En el primer grupo están todos aquellos medios, diseñados para intentar sustituir al docente y dirigir el proceso docente que tendrá un marcado carácter autodidacta. En este grupo se incluyen los:

- Tutoriales.
- Entrenadores.
- Repasadores.
- Evaluadores.

En el segundo se agrupan los medios que se desarrollan para ser empleados en una actividad docente, conducida por el docente, no pretendiendo sustituirlo. Se asemejan en este propósito a los medios de enseñanza tradicionales. Aquí se incluyen entre otros:

- Libros electrónicos
- Simuladores.

El tercer grupo considera a aquellos medios que el estudiante emplea sin el propósito consciente de aprender algo con ellos, pero que por sus características ejercen sutilmente su acción didáctica. En este grupo se encuentran:

- Juegos instructivos.

Tutoriales: Llamamos tutorial al software que trata de emular la acción del docente que imparte un contenido nuevo. Algunos autores emplean el término en forma más general

englobando en la categoría Tutorial a prácticamente todo el software de intención docente, sin embargo aquí solo incluiremos a aquellos que se proponen específicamente enseñar. Existen diferentes tipos de tutoriales que van desde los secuenciales, ramificados e inteligentes. (Toranzo, 2011)

Entrenador: Designamos con este nombre al software diseñados con el propósito de desarrollar una determinada habilidad, específicamente una habilidad manual o motora, en el estudiante que lo emplea. Muchos entrenadores utilizan la simulación de situaciones reales, en menor o mayor grado, con lo cual el estudiante puede entrenarse en la solución de tareas de diferentes grados de complejidad y los acerca a las formas cotidianas en que se presentan y resuelven determinados problemas. Existen diferentes tipos de entrenadores que van desde los más simples y lineales hasta aquellos que son capaces de identificar y caracterizar al estudiante que lo emplea y proponer una estrategia de entrenamiento de acuerdo a las características individuales de cada usuario. A este tipo de software algunos lo llaman repasador, especialmente cuando la habilidad a desarrollar es intelectual. Son usados fundamentalmente para desarrollar el tercer aspecto de la instrucción: la práctica. (Toranzo, 2011)

En forma general estos programas pueden tener incluidas simulaciones, como una forma poderosa de enseñar sobre algún aspecto del mundo imitándolo, el estudiante puede interactuar con el programa simulando situaciones reales; siempre y cuando no exista el medio “vivo”.

Categorías fundamentales de simulación.

Simulaciones físicas:

En ese caso un objeto físico es mostrado en pantalla y el estudiante puede usarlo o aprender sobre él. Ejemplos: máquinas que el estudiante tiene que aprender a usar, una experiencia física en un laboratorio, el vuelo de un aeroplano. Puede también operar controles simulados para ver cómo determinados instrumentos de lectura cambian en respuesta a una entrada de un parámetro dado. (Toranzo, 2011)

Simulaciones procedimentales:

La simulación física existe como un medio para enseñar un contenido procedimental, es decir, para que el estudiante aprenda acciones, o adquiera habilidades para operar una máquina, o los procedimientos para volar en un aeroplano, más que trabajar con los instrumentos en sí.

El propósito de la mayoría de las simulaciones procedimentales es enseñar una secuencia de acciones que constituyen un procedimiento, por ejemplo, para el diagnóstico del mal funcionamiento de un equipo. Como se ve, muchas simulaciones físicas son también procedimentales, ya que en ellas el estudiante puede imitar procedimientos reales de operación o manipulación. (Toranzo, 2011)

Simulaciones situacionales:

Tratan con las actitudes y comportamiento de las personas en diferentes situaciones, en lugar de con actuaciones hábiles. A diferencia de las simulaciones procedimentales, las cuales enseñan un conjunto de reglas, las situacionales, usualmente, permiten al estudiante explorar los efectos de diferentes aproximaciones o enfoques de una situación y a jugar diferentes roles en ellas.

En casi todas las simulaciones de este tipo el estudiante es una parte integral de la simulación, teniendo uno de los papeles principales.

Simulaciones de procesos:

Son diferentes de los otros tipos de simulación. En ellas el estudiante no participa en la simulación como en las situacionales, ni constantemente las manipula como en las físicas o procedimentales.

En lugar de esto, el estudiante selecciona parámetros con diferentes valores al inicio de la simulación y luego observa cómo el proceso ocurre sin intervenir en él.

Test o Evaluador: Este tipo de programa propone medir el nivel de los conocimientos o habilidades que posee un estudiante sobre una determinada materia. Las limitaciones en la comunicación en lenguaje natural reducen considerablemente la variedad de preguntas a realizar, por eso la mayor parte de los examinadores utilizan preguntas de verdadero o falso, de selección y de completamiento. (Toranzo, 2011)

Libro (o Guía) Electrónico: En este grupo se clasifican al software que brindan información sobre un determinado tema, pero donde el usuario selecciona libremente el tema sobre el que se va a informar y el orden en que abordará los diferentes tópicos. Los hipertextos son un caso particular de este grupo. (Toranzo, 2011)

Juegos: Pueden ser recreativos o instructivos. Estos últimos pretenden despertar, mediante el juego, el suficiente nivel de motivación y de predisposición para la asimilación del contenido

instructivo. En general los juegos se caracterizan porque establecen retos, competencia y por nivel de motivación y expectativa que crea en el usuario.

Los juegos, proporcionan un medio ambiente para facilitar el aprendizaje, sin embargo, la característica distintiva de los juegos es que, casi siempre, proporcionan al estudiante un reto entretenido, con un componente instructivo.

2.4.3 Estrategia para la evaluación y selección de un software educativo

La aplicación de esta metodología presupone un análisis anterior respecto a la necesidad de inserción de un software educativo en un momento dado y donde ya se ha definido las necesidades profesionales, el problema, sus causas, la consulta a fuentes especializadas y las alternativas de solución. (Cataldi, 2010)

En la interacción con educadores e informáticos que se interesan por el uso de software para la docencia es usual detenerse en el análisis de las cualidades y limitaciones de los mismos.

Cualquiera sea ella indudablemente tiene que ir encaminada a garantizar calidad. La calidad puede expresarse en diferentes formas:

a) Por la idoneidad o aptitud para su uso, entendiéndose por tal, las características que se pueden reconocer como beneficiosas, es decir, la estimación de que un software sirve con éxito a los fines del usuario durante su uso y de conformidad con las especificaciones establecidas.

b) Como medida de satisfacción de sus necesidades, entendiéndose de ello, que la calidad del proceso se alcanza cuando se satisfacen las expectativas del estudiante, del docente, de la familia y en definitiva de la sociedad.

Sea cual sea el análisis, se tendrá que necesariamente manejar la medida del cumplimiento de los fines, del cumplimiento de los requerimientos y de la propia existencia de una útil documentación.

Hoy por hoy este factor de calidad es una variable en el proceso con un gran peso, lo que se avala por el propio crecimiento del mercado de las tecnologías de la Información, el crecimiento del mercado de multimedia, el propio desarrollo del mercado latinoamericano, entre otros aspectos.

Todos estos datos nos obligan a pensar que el control de la calidad constituye un elemento importante en todo el trabajo de la Informática y que teniendo en cuenta las posibilidades y dificultades que se presentan en la elaboración de un software educativo, muchos profesionales han trabajado en la determinación de aspectos que deben servir de base para el proceso de selección y evaluación de sus productos. Insistimos en el factor didáctico por constituirse éste en el centro de atención de un software educativo. (Cataldi, 2010)

Consideremos que observar un sistema para evaluarlo y posiblemente seleccionarlo es algo que puede definirse de muchas formas, con muchos niveles de detalle y fundamentado en diferentes criterios. Luego, nos asociaremos a un esquema, como herramienta de trabajo, que contemple variables comunes para todo análisis de un producto educativo.

Difícilmente se pueden incluir todos los criterios evaluativos existentes y su selección debe ser modelada, en última instancia, teniendo en cuenta las individualidades locales y responder a las condiciones específicas según su grado de importancia. (Cataldi, 2010)

Dado todo lo anteriormente analizado, podemos plantear el siguiente guion como herramienta de trabajo para la observación y selección de un software educativo. Contempla un grupo de variables que resultan comunes para todo producto. (Cataldi, 2010)

Se han seleccionado para este guion cinco objetivos de análisis:

- 1) Objetivo general
- 2) Objetivo educativo
- 3) Objetivo técnico
- 4) Objetivo estético
- 5) Objetivo operacional

De manera breve se verá en qué consiste cada uno de ellos:

1) Objetivo General

Este es un objetivo que se refiere a la valoración de un grupo de aspectos tales como:

Consideración de factores humanos, se refiere al conocimiento del personal a utilizar el producto y de sus características físicas, psicológicas y mentales.

Valoración de la documentación. Existencia del manual de usuario y manual de referencia.

Costo. Su determinación y por tanto, si resulta costeable su uso a partir del aporte social que va a tener.

Potencialidad. Análisis en cuanto supera este software a otros existentes.

2) Objetivo Educativo

El corazón de un ambiente informático de apoyo a la enseñanza está en el ingrediente pedagógico que el mismo contemple. Por una parte, interesa establecer su alcance, punto de partida y contenido, que en definitiva está relacionado con el sistema de conocimientos y habilidades que deseamos desarrollar. Por otra parte interesa escudriñar la estrategia didáctica, el sistema de motivación, de evaluación que se diseña. Por ello una guía para el análisis de ello puede contemplar:

Valoración del tipo de software y su correspondencia con el uso educativo a que se destina.

Valoración de la necesidad del software, que problema resuelve, intereses, fundamentación de su necesidad educativa.

Valoración de su objetivo, es decir que sistema de conocimientos y habilidades se esperan del usuario en su acción con el software.

Valoración de los prerrequisitos para trabajar con el sistema, que conocimientos previos debe tener el estudiante y si el material lo asegura de alguna forma.

Si realmente el sistema de conocimientos y habilidades está acorde con el objetivo que se persigue y además está ajustado al nivel científico que se desea. Modelos matemáticos actuales a utilizar, el nivel de rigor del sistema de conocimientos, su actualización.

Valoración de la estrategia metodológica que se ha diseñado para lograr el conocimiento deseado en función de las teorías del aprendizaje.

Caracterización de los ejercicios que se presentan. Nivel de complejidad.

En qué medida se relaciona el modelo de caracterización del estudiante, del contenido y de la estrategia pedagógica con la aplicación.

Cómo es el sistema de motivación que oferta el sistema y su ajuste a la población que se destina.

Nivel de retroinformación y refuerzo que se puede lograr con el uso del sistema.

Forma en que establece y almacena los resultados de la evaluación.

3) Objetivo Técnico

Este objetivo está dirigido a valorar precisamente desde el punto de vista tecnológico la calidad del software teniendo en cuenta las herramientas de las que se dispone actualmente.

Valoración de cuál es la plataforma de trabajo que se utilizó y si está acorde con las exigencias actuales, análisis del sistema sobre que está hecho, ventajas técnicas que ofrece.

Análisis de los requerimientos técnicos que necesita, valoración de los periféricos que necesita, necesidad de memoria, necesidad de espacio en disco duro.

Existencia de instalador. Hoy todos los productos sobre tecnologías de avanzada requieren de los ficheros necesarios para poder llevar a cabo su instalación en el ordenador con vistas a su explotación.

Valoración de como se distribuye el sistema y su disponibilidad.

4) Objetivo estético

Este objetivo está relacionado con el correcto diseño de la interfaz gráfica y de comunicación con el usuario.

Calidad del diseño de la interfaz

Uso correcto del idioma. En este sentido se valora la no existencia de abreviaturas, que no existan errores gramaticales, así como la utilización correcta de la semántica.

Uso correcto de textos. Característica de los textos en lo que se refiere a la cantidad del mismo y de la fuente de letras que posibilite la rapidez y precisión en la lectura, evitar el scroll en pantalla, velocidad y selección adecuada, diseño de las entradas y salidas, etc.

Uso correcto de los colores. Se refiere al uso de las paletas de colores. Utilización de armonía de colores, matices y contraste.

Uso correcto de gráficos. Valoración del correcto uso de dibujos y esquemas para trabajar, dar o reforzar un concepto. Animaciones o vídeo para mostrar o ensayar el funcionamiento del algo, destacar elementos, motivar. Gráficos de tratamiento numérico para manipulación de cifras.

Uso correcto del sonido.

Uso de menús aceptables- textuales, gráficos, binarios, múltiples, extendidos (cuando no caben o son muchos), selección múltiple, permanente. Agrupación lógica por pantalla en zonas definidas.

Diseño de entradas y salidas. Entrada: fácil de llenar, segura, atractiva, que cumpla con un propósito y con una agrupación lógica. Salida: que cumpla un objetivo, salida en orden según periférico, tener en cuenta quién la va a utilizar, que sea precisa.

Tipos de redacción de mensajes, que no sean amenazantes ni humillantes, concisos, que brinde una posible opción para mayor información.

5) Objetivo Operacional

Este objetivo se refiere a las facilidades que puede brindar el sistema para el trabajo individual del usuario. Un poco se puede generalizar ello a través de las funciones de apoyo al estudiante y del propio profesor, este último, referido a posibilidades de edición de ejercicios, modificaciones de bases de datos, entre otros. Tiene en cuenta:

Control del sistema. Valoración del control del sistema en todo momento por el usuario a partir de órdenes, posibilidades de ir de un lugar a otro del programa, revocación de una orden, control de la secuencia por el usuario.

Valoración de ayuda operacional. Análisis del uso de teclas para el atajo, visualización de las opciones a utilizar, uso de línea de ayuda.

Valoración de ayuda de contenido. Valoración de la forma en que viene la ayuda, facilidades que le brinda al usuario. En este sentido es útil recordar las oportunidades que brinda el hipertexto al estudiante.

Tratamiento de errores. Valorar cómo reacciona el sistema ante posibles errores del estudiante, en la información de entrada, del operador, errores del hardware, errores por la documentación del sistema.

Conservación de la información

Interactividad con el usuario

Tiempo de ejecución

Facilidad de trabajo con independencia

Consistencia. Se refiere a que situaciones similares se presenten de igual forma y se resuelvan de forma similar.

Sistema de protección

2.4.4 Metodología para la elaboración del guion

El éxito de una aplicación educativa depende entre otros elementos de su concepción, la cual será materializada mediante el guion.

La metodología consta de dos partes. En la primera se indican los pasos constitutivos de la estructura del guion. En un segundo momento se ofrecen las orientaciones que incluyen desde la determinación del problema, hasta las relacionadas con la bibliografía del tema abordado. Esta metodología puede ser aplicada en la elaboración de presentaciones automáticas en Power Point y también en el diseño de páginas Web. (Fernández, 2005)

2.4.4.1 Parte 1: Metodología del guion

a). Nombre del software.

Se debe definir palabras claves que describan los principios básicos del software.

b). Objetivo.

Debe ser formulado con toda claridad y precisión. Su alcance estará en correspondencia con la función didáctica y con las habilidades profesionales.

c). Caracterización.

Incluye los siguientes aspectos:

- Temática
- Nivel y edad del usuario

- Conocimientos propedéuticos (conocimiento anterior)
- Función didáctica

d). Información.

Responderá a los criterios pedagógicos, psicológicos y socioculturales. La información puede ser gráfica, textual o sonora. Su contenido tendrá como requisitos:

- Nivel de actualidad
- Rigor científico
- Estar profesionalizado
- Responder a la interdisciplinariedad
- Ritmo
- Volumen
- Estructurado de acuerdo a los niveles de asimilación
- Definir núcleos de información para determinados tipos de software. En este aspecto se especificará la bibliografía consultada.

e). Diagrama de flujo.

Se refiere a la secuencia de escenas (textual o gráfica) del software. En este aspecto se definirán las figuras, videos, animaciones, efectos sonoros, etc.

f). Diseño general de las pantallas.

Concibe la forma de presentar el menú, los mensajes de error y la navegación. Para cada escena se tendrán en cuenta los siguientes elementos:

- Marcos/ventana
- Zonas sensibles
- Botones
- Iconos
- Animación
- Sonido
- Objetos estáticos: títulos, imágenes, zonas.
- Objetos dinámicos: mascotas, viñetas, animados
- Objetos interactivos: iconos, botones, listas
- Tiempo de duración de escenas y las acciones
- Contextualización profesionalizada de las escenas.

g). Diagrama de la ayuda.

- Flujo
- Contenido

- Diseño

h). Requerimientos para la manipulación.

Especificar para qué tipo de computadora se confeccionará la aplicación.

2.4.4.1 Parte 2: Orientaciones metodológicas

a). Caracterización del usuario.

- Incluye los siguientes aspectos:
- Nivel y edad de los estudiantes.
- Conocimientos propedéuticos.
- Deficiencias (conocimientos y habilidades) más generalizadas en la apropiación del contenido objeto de estudio.

b). Determinación del problema.

Una aplicación educativa es el resultado de un proceso cuyo paso inicial es la validación de la asignatura y de los resultados docentes alcanzados.

De este análisis se derivan las principales dificultades y cómo el software puede contribuir a eliminarlas.

c). Objetivo.

En el planteamiento del objetivo debe manifestarse tanto el aspecto cognitivo como el axiológico (estudio de valores).

d). Tipo de software.

Programa informático avanzado que funciona para gestionar las diferentes áreas de un centro educativo.

e). Función didáctica.

Se tendrá en cuenta la función didáctica predominante en el software, ya que a similitud con el proceso pedagógico puede estar presente más de una función. Para un tutorial o un libro electrónico se correspondería fundamentalmente con el tratamiento de la nueva materia, si es un evaluador la función didáctica sería el control y el repasador, entrenador y simulador se identificaría con la consolidación.

f). Contenido.

Debe garantizar que con él, los componentes conceptual, operacional y educativo se cumplan.

Deben valorarse los criterios pedagógicos, psicológicos y socioculturales a que debe responder el contenido.

g). Método.

Nos indica en qué forma se va a desarrollar el proceso para alcanzar el objetivo.

Las funciones del método (didáctica, psicológica, gnoseológica y cibernética) deben estar presentes de la siguiente forma.

- **Función didáctica.**

El método debe propiciar la búsqueda de otros contenidos o la profundización de los que se presentan. El estudiante aprende a buscar por sí mismo temas más profundos o más avanzados y llegar a ellos, lo que implica fortalecer la confianza y seguridad en su persona, con el consiguiente desarrollo de la independencia cognoscitiva y por ende el desarrollo de la personalidad.

- **Función psicológica.**

Con el método empleado se pueden despertar varias motivaciones en el estudiante como son:

Profundizar acerca del contenido que se estudia, estudiar o profundizar cuestiones relacionadas con la computación, los medios de enseñanza, la creatividad, etc.

La comunicación que establece con el objeto de estudio es mediante una computadora, por consiguiente, tiene que aprender o ejercitar su uso indispensable en la era moderna.

La actividad generada con este método es significativa. Las características de la aplicación deben obligar al estudiante a asumir una posición activa lo que generalmente es más fácil lograr con el software que con otros medios.

- **Función gnoseológica.**

Está dada por la potencialidad que posee el método de involucrar al estudiante y ofrecer razones para investigar, ya sea desde el punto de vista pedagógico, cognoscitivo y educativo, traducido esto en temas tales como los medios de enseñanza y el papel de la computación en ellos,

el propio contenido del software o los sistemas y lenguajes empleados, la posición activa y transformadora del sujeto al interactuar con los distintos tipos software, etc.

- **Función cibernética.**

Las funciones de organizar, planificar y controlar son inherentes a la función del método y cualquiera que sea el tipo del software, la función cibernética pertenece en alguna medida al docente. Aunque el estudiante, como en el ejemplo que tratamos, desarrolle su actividad independiente, la orientación inicial y el control final corresponden al docente. No obstante, la búsqueda de soluciones por parte del estudiante, el pensamiento alternativo, las tomas de decisiones, etc. son los resultados de estas funciones. (Fernández, 2005)

h). Medio.

El medio se comporta como el facilitador del proceso y la caracterización de sus cualidades precisa de las mismas funciones analizadas para el método.

i). Forma organizativa.

Como elemento integrador, muestra cómo se va a organizar la actividad. Esto significa que en dependencia del objetivo del software y las necesidades que motivaron su creación, el mismo podrá ser utilizado en la clase o fuera de ella. (Fernández, 2005)

Resulta de gran interés que en las actividades que se desarrollan en el contexto del uso del software se tenga en cuenta que las mismas complementan ese doble carácter de la enseñanza y el aprendizaje.

j). Evaluación.

Este es el proceso donde se comprueba y valora todo lo aprendido por el estudiante.

Incluye tanto el aspecto instructivo como el educativo.

Se deberá evaluar el grado de dominio de conocimientos, hábitos y habilidades, el nivel de independencia alcanzado, el nivel de actividad en la práctica social y los grados de responsabilidad individual y colectiva. El efecto educativo caracteriza la transformación que se espera en el estudiante, en qué medida ha interiorizado los valores y cómo los manifiesta en su conducta. (Fernández, 2005)

Con la utilización de una aplicación, la evaluación puede concebirse en la misma (de hecho existen los programas evaluadores), y también fuera de ella. Lo más acertado sería evaluar en ambos casos.

Si bien como parte del software, la evaluación le permite al estudiante autoevaluarse, una evaluación a través de un taller o seminario le brinda al profesor la oportunidad de evaluar el desarrollo personal del estudiante.

k). Diseño de la interfaz gráfica.

Respecto a la percepción de los mensajes visuales, el organismo recibe los siguientes efectos:

a) Equilibrio: El equilibrio se debe a la figura o a la conformidad o desviación con la verticalidad o la horizontalidad.

Consiste en la estabilidad óptica de los elementos, ubicación de la figura, relación entre volúmenes, colores, tonos. El equilibrio sugiere cosas estables, duraderas. La mayor estabilidad de una imagen coincide con el cuadrante inferior izquierdo. Una forma estable es estática y sugiere ser duradera, una inestable sugiere movimiento. (Fernández, 2005)

Requiere establecer un sistema de ejes vertical y horizontal, el primero es más importante, el horizontal es secundario. Depende del peso y la dirección visual. La zona de máxima estabilidad de una imagen coincide con el cuadrante inferior izquierdo, por lo que el mayor peso visual se logra en la zona superior derecha. En cuanto a la dirección visual (dirección de lectura de la imagen) debe considerarse el criterio de la diferencia entre las líneas de fuerza que van del ángulo superior derecho al inferior izquierdo y las líneas de interés que van del ángulo superior izquierdo al inferior derecho. Por otra parte, la dirección vertical provoca mayor impacto que la horizontal.

b) Ritmo: El ritmo es quien ofrece secuencia de regularidades. El estudiante espera en lo sucesivo analogía de las imágenes. Puede conseguirse el ritmo con gradaciones cromáticas, posición de las figuras en el plano, etc.

Este aspecto es el que guía la mirada sobre las imágenes. La forma de dirigir la atención hacia los elementos de interés se realiza mediante la priorización. Esta se logra por:

a) Diferencia del tamaño (escala)

b) Situación del elemento.

c) Color

d) Tono

e) Posición

f) Configuración

g) Iluminación

h) Enfoque.

c) Sorpresa: La sorpresa es utilizada para captar la atención. Es aquello cuya probabilidad de que ocurra es casi nula. Se puede utilizar la sorpresa por ejemplo, si en un procedimiento el estudiante incurre en una violación de las reglas de Protección e Higiene del Trabajo.

d) Contraste: El contraste es una forma de conseguir la diferencia entre la figura y el fondo. Este se establece respecto a elementos visuales (forma, medida, textura, color) y de relación (dirección, posición, espacio, gravedad).

· De figura Geométrica/orgánica

· De tamaño Grande/pequeño, largo/corto.

· De textura Suave/rugoso, pulido/tosco, opaco/satinado

· De dirección El máximo contraste se presenta cuando entre dos formas existe un ángulo de 90 grados.

· De posición La posición de la forma se reconoce por su relación con el marco.

En lo que respecta al contraste de color, el mismo se refiere a las combinaciones de los colores entre la figura o texto y el fondo. El negro sobre el amarillo es la que mejor se ve a mayor distancia y resulta ser la más agresiva. Las más visibles son aquellas que poseen fondo claro y la figura o el texto oscuro. Esta es la denominada polaridad positiva. Sus reflejos son menos perceptibles. El significado del color tiene carácter histórico, así por ejemplo el rojo es el color percibido con mayor impacto por el ojo humano. Provoca sensación de calor, temperatura y excitación.

El amarillo es el que puede distinguirse desde mayor distancia y junto con el naranja se asocian a la luz y al fuego. Los colores cálidos son el verde, el azul y el magenta. El verde lejos de provocar excitación da la sensación de paz y reposo. Por su parte el azul propicia sobrecogimiento y descanso. Es visible a corta distancia al igual que el magenta. Las combinaciones más atractivas a simple vista son las siguientes:

- Figura / Fondo
- negro/amarillo
- rojo/blanco
- azul fuerte/blanco
- negro/blanco
- blanco/negro
- rojo/verde

- verde/rojo

e) Tensión: Mientras más alejado se encuentre un elemento dentro de una figura del eje visual, provoca mayor stress y la atención se dirige hacia ese lado de la figura.

f) Nivelación y aguzamiento: La nivelación está dada por la cercanía del elemento a la estructura vertical y horizontal. Mientras más cerca esté de la misma mayor es la estabilidad. Si el elemento se aleja (incluso en el mapa estructural -diagonales-) es mayor el aguzamiento y genera por consiguiente mayor tensión. (Fernández, 2005)

g) Angulo inferior izquierdo: Primariamente la atención responde a los referentes verticales horizontales y de manera secundaria el impulso perceptivo responde a la zona inferior izquierda.

h) Atracción y agrupamiento: Mientras la distancia entre los elementos que componen la imagen es menor se percibe mayor atracción entre los mismos (los semejantes se atraen). El agrupamiento se percibe de acuerdo a la semejanza que tengan los elementos en cuanto a su contorno, tamaño, textura y tono.

i) Positivo y negativo: Lo positivo es aquello que domina la mirada (lo que llama más la atención).

Lo negativo es el elemento que actúa con mayor pasividad. También hay que tener en cuenta algunos hechos psicofísicos como son:

- Los elementos más anchos parecen más cercanos.

· La distancia relativa se percibe mejor al superponer los elementos. La superposición favorece el paralelismo.

· El elemento oscuro sobre fondo claro parece contraerse.

Los elementos visuales.

a) El punto: Cuanto más próximos están los puntos entre sí se intensifica la capacidad de ellos para guiar el ojo.

Ejerce gran fuerza de atracción sobre el ojo. Al situarlo sobre una superficie se crean determinadas tensiones visuales. Si se coloca en el centro visual (un poco por encima del geométrico) se estabiliza la composición y la sensación de equilibrio. (Fernández, 2005)

b) La línea: Sirve para visualizar lo que no puede verse. Es la historia del movimiento de un punto.

Las líneas rectas horizontales y verticales se identifican con el reposo, mientras que las oblicuas y curvas con el movimiento.

c) El contorno: La línea es quien lo describe. Los contornos básicos son el cuadrado, el triángulo y el círculo. Sus significados son:

Cuadrado: torpeza, honestidad, rectitud, esmero, robustez, dogmatismo.

Triángulo: acción, conflicto, tensión.

Círculo: infinitud, calidez, protección, perfección, equilibrio.

d) La dirección: Indica el desplazamiento visual de la imagen. Las direcciones básicas son expresadas por los contornos básicos.

Cuadrado: horizontal y vertical.

Triángulo: diagonal: fuerza direccional más inestable. Su significado es amenazador.

Círculo: su significado está asociado al encuadramiento, repetición y el calor.

e) El tono: Es la intensidad de oscuridad o claridad del objeto visto. El fondo tonal refuerza la apariencia de realidad. La sensibilidad tonal es importante porque a ella se debe que se vea la profundidad, el movimiento súbito, la distancia y otras referencias ambientales.

f) El color: Posee tres dimensiones: matiz, saturación y brillo.

Matiz: Se identifica más con las emociones.

Saturación: Es la pureza de un color respecto al gris. Mientras más saturada es la coloración de un objeto mayor es su expresión y emoción. Los colores extremos del espectro (rojo y azul saturados) no deben ser presentados simultáneamente en la pantalla.

Brillo: Va de la luz a la oscuridad. Se debe tener en cuenta que respecto a la reflexión directa, se ha demostrado que los colores oscuros brillan más que los claros.

g) Textura: Se refiere a las características de superficie de una figura (suave, rugosa, lisa, opaca, blanda, etc.) Sirve de doble de las cualidades del tacto. El significado se basa en lo que

vemos. Es la variación en la apariencia de consistencia. La textura visual es bidimensional. Es la clase de textura que puede ser vista por el ojo, aunque pueda evocar también sensaciones táctiles.

h) Escala: Permite relacionar el tamaño con el propósito y el significado. Su efecto se extiende a las manipulaciones del espacio.

i) Dimensión: La representación volumétrica de cualquier imagen se logra fundamentalmente con la perspectiva y sus efectos se intensifican con la variación del tono (claroscuro). La línea se utiliza para crear efectos de perspectiva.

j) Movimiento: Es el más intenso foco visual de atención. Implica cambio en las condiciones del medio. El movimiento de acercamiento o no, logrado con aumento o disminución del tamaño del objeto indica ir de lo general a lo particular o viceversa.

2.5 Metodología XP

XP resalta una serie de valores y principios que deben tenerse en cuenta y practicarlos durante el tiempo de desarrollo que dure el proyecto. Al final de este apartado se enuncian algunas de las características que deben tener los proyectos que se realicen con XP. (Echeverry Tobon, 2007, pág. 29)

Más que una metodología, XP se considera una disciplina, la cual está sostenida por valores y principios propios de las metodologías ágiles. Existen cuatro valores que cumplen su papel como pilares en el desarrollo de las metodologías livianas:

La comunicación. En la metodología XP es muy importante que exista un ambiente de colaboración y comunicación al interior del equipo de desarrollo, así como en la interacción de éste con el cliente. En XP la interacción con el cliente es tan estrecha, que es considerado parte del equipo de desarrollo. (Echeverry Tobon, 2007)

La simplicidad. Este valor se aplica en todos los aspectos de la programación extrema. Desde diseños muy sencillos donde lo más relevante es la funcionalidad necesaria que requiere el cliente, hasta la simplificación del código mediante la refactorización del mismo. La programación XP no utiliza sus recursos para la realización de actividades complejas, sólo desarrolla lo que el cliente demanda, de la forma más sencilla.

La retroalimentación. Se presenta desde el comienzo del proyecto, ayuda a encaminarlo y darle forma. Ésta se presenta en los dos sentidos, por parte del equipo de trabajo hacia el cliente, con el fin de brindarle información sobre la evolución del sistema, y desde el cliente hacia el equipo en los aportes a la construcción del proyecto. (Echeverry Tobon, 2007)

El coraje. El equipo de desarrollo debe estar preparado para enfrentarse a los continuos cambios que se presentarán en el transcurso de la actividad. Cada integrante debe tener el valor de exponer los problemas o dudas que halle en la realización del proyecto.

Aún con estas variaciones, las jornadas de trabajo deben proporcionar el máximo rendimiento.

A partir de los valores se plantea una serie de prácticas que sirven de guía para los desarrolladores en esta metodología. Uno de los aspectos más importantes para XP son las doce

reglas que se plantean, las cuales se caracterizan por su grado de simplicidad y por su enfoque en la practicidad, además de que cada regla se complementa con las demás. A continuación, se realizará una breve descripción de cada una de ellas.

El desarrollo está dirigido por pruebas. Antes de realizar una unidad de código, es necesario contar con su respectiva unidad de pruebas. El programador realiza pruebas dirigidas al funcionamiento de nuevas adiciones o módulos al sistema. El cliente con ayuda del tester se encarga de diseñar las pruebas de aceptación, cuyo propósito es verificar que las historias de usuario se hayan implementado correctamente.

El juego de la planificación. Desde el comienzo del desarrollo se requiere que el grupo y el cliente tengan una visión general y clara del proyecto, es decir, deben entender y estar de acuerdo con lo que el “otro” plantee. En el transcurso del proyecto se realizan diferentes reuniones, con el fin de organizar las tareas e ideas que surgen tanto por parte del cliente como por el equipo.

Cliente in-situ. El cliente, o un representante del mismo, deben estar en el sitio de desarrollo para solucionar las preguntas o dudas que se puedan presentar a medida que se realice el proyecto.

Programación en parejas. XP propone que exista una pareja de programadores por monitor y teclado, como medida para aumentar la calidad del código. Esta práctica busca reducir los errores de codificación, mientras uno de los programadores busca una forma de dar funcionalidad a un módulo, el otro programador aprueba dicho código y busca la forma de simplificarlo. (Echeverry Tobon, 2007)

Entregas pequeñas. En la programación extrema se realizan entregas constantes de módulos funcionales completos, de tal forma que en todo momento el cliente tiene una parte de aplicación funcionando. En XP no existe el desarrollo incompleto de una tarea, ésta se ejecuta en su totalidad o no se hace.

Refactorización sin piedad. El código se revisa de forma permanente para depurarlo y simplificarlo, buscando la forma de mejorarlo. La refactorización se realiza durante todo el proceso de desarrollo.

Integración continua del código. El código de los módulos debe ser integrado a cortos plazos de tiempo, preferiblemente no mayores a un día. Esto facilita la búsqueda y la corrección de errores de codificación e integración que se presenten en el proceso.

Diseño simple. Sólo se realiza lo necesario para que la aplicación cumpla con la funcionalidad requerida por el cliente. No es conveniente realizar diseños complejos que posiblemente no aporten soluciones claras al proyecto, y que a la hora de cambiar los requerimientos se conviertan en una gran barrera de tiempo.

Utilización de metáforas del sistema. Para el mejor entendimiento de los elementos del sistema por parte del equipo de desarrollo se acude a la utilización de metáforas, como una forma de universalizar el lenguaje del sistema.

Propiedad colectiva del código. El código no es conocido por una sola persona del grupo de trabajo, esto facilita implementar cambios al programa por parte de otros integrantes del equipo.

Convenciones de código. La aplicación de estándares de programación al código fuente de la aplicación, permite que todas las personas que conforman el grupo de trabajo puedan entender y realizar modificaciones al código del sistema.

No trabajar horas extras. Es preferible volver a estimar los tiempos de entrega. Con esta práctica se busca utilizar al máximo el rendimiento y energía del programador.

2.5.1 Alcance de XP

La programación extrema es conveniente en ciertas situaciones, pero también es necesario saber que presenta controversia en otras. Esta metodología es aplicable con resultados positivos a proyectos de mediana y pequeña envergadura, donde los grupos de trabajo no superan 20 personas.

Otro aspecto importante en la selección de esta metodología radica en el ambiente cambiante que se presenta en los requerimientos de la aplicación. La metodología XP está encaminada hacia los desarrollos que requieren de cambios continuos en el transcurso de un proyecto. La metodología es recomendada para proyectos en los cuales el costo de cambio no se incrementa a medida que transcurre vida del mismo. (Echeverry Tobon, 2007)

Los proyectos realizados bajo esta metodología cumplen con lo estrictamente necesario en su funcionalidad en el momento necesario: hacer lo que se necesita cuando se necesita. En XP no es conveniente precipitarse o adelantarse a las tareas que se han establecido previamente sin el consentimiento del cliente, estos hechos conllevan a inyectar complejidad al sistema, alejándolo del concepto de simplicidad. (Echeverry Tobon, 2007)

2.5.2. Planeación

La planeación es la etapa inicial de todo proyecto en XP. En este punto se comienza a interactuar con el cliente y el resto del grupo de desarrollo para descubrir los requerimientos del sistema. En este punto se identifican el número y tamaño de las iteraciones al igual que se plantean ajustes necesarios a la metodología según las características del proyecto.

En este apartado se tendrán en cuenta ocho elementos, los cuales son los siguientes.

Historias de usuario, velocidad del proyecto, iteraciones, entregas pequeñas, reuniones, roles en XP, traslado del personal y ajuste a XP.

2.5.2.1 Historias de usuario

El sistema es desarrollado para el cliente, por lo tanto, el usuario es quien decide que tareas realizará la aplicación. Este planteamiento se desarrolla a lo largo del proyecto: el cliente es quien decide que hacer. Como primer paso, se debe proporcionar una idea clara de lo que será el proyecto en sí.

Las historias de usuario son utilizadas como herramienta para dar a conocer los requerimientos del sistema al equipo de desarrollo. Son pequeños textos en los que el cliente describe una actividad que realizará el sistema; la redacción de los mismos se realiza bajo la terminología del cliente, no del desarrollador, de forma que sea clara y sencilla, sin profundizar en detalles. (Echeverry Tobon, 2007)

Se puede considerar que las historias de usuario en XP juegan un papel similar a los casos de uso en otras metodologías, pero en realidad son muy diferentes. Las historias de usuario sólo muestran la silueta de una tarea a realizarse. Por esta razón es fundamental que el usuario o un representante del mismo se encuentren disponibles en todo momento para solucionar dudas, estas no proporcionan información detallada acerca de una actividad específica. (Echeverry Tobon, 2007)

Las historias de usuario también son utilizadas para estimar el tiempo que el equipo de desarrollo tomará para realizar las entregas. En una entrega se puede desarrollar una o varias historias de usuario, esto depende del tiempo que demore la implementación de cada una de las mismas.

2.5.2.2 Velocidad del proyecto

Es una medida de la capacidad que tiene el equipo de desarrollo para evacuar las historias de usuario en una determinada iteración. Esta medida se calcula totalizando el número de historias de usuario realizadas en una iteración. Para la iteración siguiente se podrá (teóricamente) implementar el mismo número de historias de usuario que en la iteración anterior, cabe recordar que la velocidad del proyecto ayuda a determinar la cantidad de historias que se pueden implementar en las siguientes iteraciones, aunque no de manera exacta. La revisión continua de esta métrica en el transcurso del proyecto se hace necesaria, ya que las historias varían según su grado de dificultad, haciendo inestable la velocidad de la realización del sistema. (Echeverry Tobon, 2007)

2.5.2.3. Iteraciones

En la metodología XP, la creación del sistema se divide en etapas para facilitar su realización. Por lo general, los proyectos constan de más de tres etapas, las cuales toman el nombre de iteraciones, de allí se obtiene el concepto de metodología iterativa. La duración ideal de una iteración es de una a tres semanas.

Para cada iteración se define un módulo o conjunto de historias que se van a implementar. Al final de la iteración se obtiene como resultado la entrega del módulo correspondiente, el cual debe haber superado las pruebas de aceptación que establece el cliente para la verificar el cumplimiento de los requisitos. Las tareas que no se realicen en una iteración son tomadas en cuenta para la próxima iteración, donde se define, junto al cliente, si se deben realizar o si deben ser removidas de la planeación del sistema. (Echeverry Tobon, 2007)

2.5.2.4. Entregas Pequeñas

La duración de una iteración varía entre una y tres semanas, al final de la cual habrá una entrega de los avances del producto, los cuales deberán ser completamente funcionales. Estas entregas deben caracterizarse por ser frecuentes.

2.5.2.5. Reuniones

El planeamiento es esencial para cualquier tipo de metodología, es por ello que XP requiere de una revisión continua del plan de trabajo. A pesar de ser una metodología que evita la documentación exagerada, es muy estricta en la organización del trabajo.

Al comenzar el proyecto se realiza una reunión entre el equipo de trabajo y los clientes. En dicha reunión se define el marco temporal de la realización del sistema. El cliente expone las historias de usuario a los integrantes de grupo, quienes estimarán el grado de dificultad de la implementación de cada historia.

Las historias de usuario son asignadas a las diferentes iteraciones según su orden de relevancia para el proyecto. En el proceso de selección de las historias de usuario para cada iteración, se tiene en cuenta que la suma de las estimaciones sea aproximada a la velocidad del proyecto de la iteración pasada.

En esta reunión se predicen los tiempos que se utilizarán en la realización de las diferentes etapas del proyecto, los cuales no son datos exactos, pero proporcionan una base del cronograma.

Finalmente, a partir de las historias de usuario, el cliente plantea las pruebas de aceptación con las cuales se comprueba que cada una de estas ha sido correctamente implementada.

Al comenzar una iteración se realiza una reunión de la misma, donde se organizan las actividades de programación a realizar. Las historias de usuario son traducidas a tareas y asignadas a los desarrolladores.

Los desarrolladores estiman los tiempos para la realización de las tareas. Cada tarea se estima de uno a tres de días de programación ideales o sin distracciones. Estas estimaciones son más exactas que las realizadas en la planeación de entregas, por lo tanto no deben exceder la velocidad de proyecto de la iteración anterior. De ser así, se consulta con el cliente para determinar que historias de usuario se pospondrán para iteraciones futuras. (Echeverry Tobon, 2007)

Las reuniones Diarias o “stand-up meeting” se realizan al comenzar la jornada laboral. Todo el equipo de desarrollo se reúne para exponer los problemas e ideas que se estén presentando, esto con el fin que el equipo en conjunto construya una mejor solución.

Es de vital importancia evitar las discusiones largas, ya que se está utilizando tiempo laboral que puede ser destinado a la construcción del sistema. También debe evitarse las conversaciones separadas, las dudas que se presenten serán solucionadas por el equipo en conjunto.

2.5.2.6. Roles XP

En esta metodología se utiliza el concepto de roles para organizar quienes se encargarán de cada una de las actividades que deben realizarse en el transcurso del proyecto. Cada uno de estos papeles son desempeñados por uno o varios integrantes del grupo, sin descartar la posibilidad de rotar los roles entre el equipo durante la realización del sistema.

El jefe de proyecto tiene como responsabilidad la dirección y organización de las reuniones que se realizan durante el proyecto.

El usuario o cliente determina qué se va a construir en el sistema, además de decidir el orden en que se entregarán cada segmento del proyecto. Es parte fundamental del equipo XP (se menciona su importancia como una de las prácticas), en todo proyecto debe existir un cliente. Además, tiene como tarea establecer las pruebas de aceptación, las cuales determinan si el sistema cumple con los requerimientos del usuario.

En el grupo de los programadores se encuentran además los diseñadores y los analistas. Los programadores son quienes construyen el sistema y realizan las pruebas correspondientes a cada

módulo o unidad de código. Cuando surgen dudas o preguntas que afectan decisiones sobre la funcionalidad del sistema (las decisiones técnicas son solucionadas gracias a las habilidades de los programadores), el programador no debe hacer suposiciones acerca de lo que el cliente quiere; en este caso, debe dirigirse al mismo y aclarar la situación. (Echeverry Tobon, 2007)

El entrenador (coach) es el responsable de que el proceso se realice de forma correcta. Se asegura de que los conceptos de la metodología se apliquen al proyecto, además de brindar ayuda continua a los demás integrantes del equipo.

La persona quien realiza las pruebas, colabora en la realización de las pruebas de aceptación y es quien muestra los resultados de las mismas. En este proceso, ayuda al cliente a diseñar tales pruebas y a verificar que las pruebas sean aprobadas.

El rastreador (tracker) tiene como tarea observar la realización del sistema. Varias veces por semana cuestiona a los integrantes del equipo para anotar sus logros y avances. Mantiene datos históricos del proyecto.

2.5.2.7. Traslado de personal

Al mover el personal se evitan problemas relacionados con la pérdida de conocimiento y cuellos de botella. Todos los miembros del grupo deben tener suficiente conocimiento de la estructura del código de modo tal que se eviten las islas de conocimiento las cuales son susceptibles de generar pérdidas de información importante.

En la medida que todos los programadores entienden todas las partes del programa se evita que unos tengan una carga de trabajo muy alta mientras que otros no tengan mucho trabajo por hacer.

La programación en parejas se convierte en una herramienta muy importante para lograr el objetivo del traslado de personal sin que se pierda el rendimiento. Esto se logra haciendo que un miembro de la pareja se traslade mientras que el otro continúe el desarrollo con un nuevo compañero.

2.5.2.8. Ajustar XP

Todos los proyectos tienen características específicas por lo cual XP puede ser modificado para ajustarse bien al proyecto en cuestión. Al iniciar el proyecto se debe aplicar XP tal como es, sin embargo no se debe dudar en modificar aquellos aspectos en que no funcione.

Eso no quiere decir que los desarrolladores pueden hacer lo que se les antoje. Antes de implementarse un cambio, este debe ser discutido y aprobado por el grupo. (Echeverry Tobon, 2007)

2.6. Modelo Cocomo II

El Modelo Constructivo de Costos (o COCOMO, por su acrónimo del inglés CONstructive COSt MOdel) es un modelo matemático de base empírica utilizado para estimación de costos de software. Incluye tres submodelos, cada uno ofrece un nivel de detalle y

aproximación, cada vez mayor, a medida que avanza el proceso de desarrollo del software: básico, intermedio y detallado. (Fundación Wikimedia, 2018)

Este modelo fue desarrollado por Barry W. Boehm a finales de los años 70 y comienzos de los 80, exponiéndolo detalladamente en su libro "Software Engineering Economics" (Prentice-Hall, 1981)

2.6.1 Características generales

Pertenece a la categoría de modelos estimadores basados en estimaciones matemáticas. Está orientado a la magnitud del producto final, midiendo el "tamaño" del proyecto, en función de la cantidad de líneas de código, principalmente.

Se presentan tres niveles: básico, intermedio y detallado.

2.6.2 Modelos de estimación

Las ecuaciones que se utilizan en los tres modelos son:

- $E = a(Kl)^b * m(X)$, en persona-mes
- $Tdev = c(E)^d$, en meses

- $P = E/T_{dev}$, en personas

Donde:

E es el esfuerzo requerido por el proyecto, en persona-mes

T_{dev} es el tiempo requerido por el proyecto, en meses

P es el número de personas requerido por el proyecto

a, b, c y d son constantes con valores definidos en una tabla, según cada submodelo

Kl es la cantidad de líneas de código, en miles.

$m(X)$ Es un multiplicador que depende de 15 atributos.

A la vez, cada submodelo también se divide en **modos** que representan el tipo de proyecto, y puede ser:

modo orgánico: un pequeño grupo de programadores experimentados desarrollan software en un entorno familiar. El tamaño del software varía desde unos pocos miles de líneas (tamaño pequeño) a unas decenas de miles (medio). (Fundación Wikimedia, 2018)

modo semilibre o **semiencajado**: corresponde a un esquema intermedio entre el orgánico y el rígido; el grupo de desarrollo puede incluir una mezcla de personas experimentadas y no experimentadas.

modo rígido o empotrado: el proyecto tiene fuertes restricciones, que pueden estar relacionadas con la funcionalidad y/o pueden ser técnicas. El problema a resolver es único y es difícil basarse en la experiencia, puesto que puede no haberla.

2.7. Herramientas de Desarrollo.

Para este proyecto fue necesario utilizar diferentes herramientas indispensables para el desarrollo del software, entre las cuales se tiene:

Apache

Es un servidor web potente y flexible que pueda funcionar en la más amplia variedad de plataformas y entornos. Las diferentes plataformas y los diferentes entornos, hacen que a menudo sean necesarias diferentes características o funcionalidades, o que una misma característica o funcionalidad sea implementada de diferente manera para obtener una mayor eficiencia. Apache se ha adaptado siempre a una gran variedad de entornos a través de su diseño modular.

Este diseño permite a los administradores de sitios web elegir que características van a ser incluidas en el servidor seleccionando que módulos se van a cargar, ya sea al compilar o al ejecutar el servidor.

Apache nos brinda muchas ventajas como son:

Soportar de una forma más fácil y eficiente una amplia variedad de sistemas operativos.

El servidor puede personalizarse mejor para las necesidades de cada sitio web.

MySQL Server:

Es la base de datos de código fuente abierto más usada del mundo. Su ingeniosa arquitectura lo hace extremadamente rápido y fácil de personalizar. La extensiva reutilización del código dentro del software y una aproximación minimalística para producir características funcionalmente ricas, ha dado lugar a un sistema de administración de la base de datos incomparable en velocidad, compactación, estabilidad y facilidad de despliegue. La exclusiva separación del core server del manejador de tablas, permite funcionar a MySQL bajo control estricto de transacciones o con acceso a disco no transaccional ultrarrápido.

PHP

Es un lenguaje de programación usado generalmente para la creación de contenido para sitios web. PHP es el (acrónimo recursivo de "PHP: Hypertext Preprocessor", inicialmente PHP Tools, o, Personal Home Page Tools) es un lenguaje interpretado usado para la creación de aplicaciones para servidores, o creación de contenido dinámico para sitios web, y últimamente también para la creación de otro tipo de programas incluyendo aplicaciones con interfaz gráfica.

Capítulo 3

Ingeniería del proyecto

En este capítulo el objetivo general es resolver todo lo concerniente a la instalación y el funcionamiento del laboratorio virtual de implementación y configuración de servicios como se muestra en la siguiente figura.

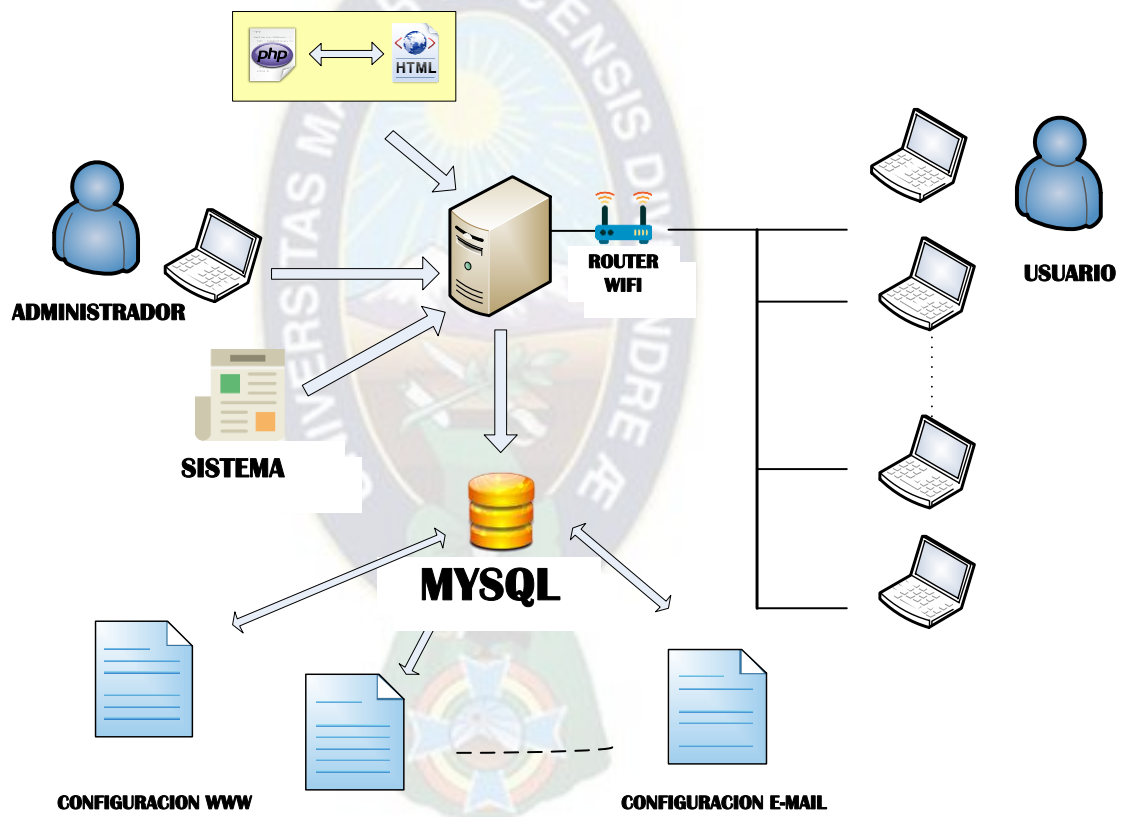


Figura 1. Diagrama de diseño del Prototipo Fuente. Creación propia

Para la realización de servidor LTSP (Linux Terminal Server Project) se llevaron a cabo las siguientes actividades:

3.1. Búsqueda de Documentación

La documentación fue obtenida casi en su totalidad por medio de internet siendo este el recurso con más información respecto al tema tratado y a su vez el más accesible dentro del área laboral.

3.2. Análisis de requerimientos

Los requerimientos fueron fundamentalmente basados con la prioridad de tener un servidor eficiente el cual pueda soportar adecuadamente la carga de los clientes ligeros por lo cual como anteriormente se especificó este dependerá del número de clientes a los que se les otorgará el servicio.

La definición de la estructura y topología de red es un requerimiento importante, para así poder definir adicionar más de una tarjeta de red en el servidor si esto llegara a ser necesario.

En cuanto al hardware de los clientes ligeros, deben contar con placa base, CPU, Memoria RAM, fuente de alimentación, teclado, ratón, pantalla y tarjeta de red con soporte PXE (Preboot eXecution Environment).

Según las necesidades ya expuestas anteriormente, el análisis de requerimientos se basó en específico en la funcionalidad del servidor cuantificando un 50% más de los usuarios actuales para con ello tomar en cuenta el crecimiento de tal proyecto en años futuros dentro de la organización.

La instalación del server LTSP (Linux Terminal Server Project) se llevó a cabo en VirtualBox 5.2, de igual manera se crearon a los clientes en máquinas virtuales en dicho software, las cuales sirvieron para probar el funcionamiento del software seleccionado.

3.3. Selección de software

Mediante una breve investigación sobre los componentes tanto de Ubuntu Server 19.04 y Ubuntu Server 18.04 estas versiones de Sistemas Operativos trabajan con PHP versión 7.0 o las últimas versiones, en nuestro sistema estructurado con el Framework CodeIgniter que utiliza PHP versión 7.0, se llegó a la conclusión de utilizar Ubuntu 16.04 por ser más estable y tener un tiempo 5 años de soporte en esta versión, donde se cuenta con actualizaciones de seguridad y en general sobre todo el funcionamiento del sistema, a continuación se menciona algunas de sus características:

- Entorno de escritorio GNOME 3.28, con algunos paquetes de GNOME 3.4.
- Interfaz Unity personalizable
- 100% accesible
- Centro de control más limpio
- Mejoras a Mozilla Thunderbird
- Reproductor de música Rhythmbox
- Proceso de arranque sin parpadeos
- Mejorada la velocidad del arranque
- Mejorado el soporte para múltiples monitores
- Gestión de la energía en la nube y proyecto Cloud-Live
- Libre Office actualizado a la versión 16.1
- Los mínimos requisitos recomendados para el óptimo funcionamiento del S.O Ubuntu Server 16.04 son los siguientes:

- RAM de 1 GB
- Procesador x64 a 1GHz
- HDD de 10GB

El software se descargó de la página oficial de Ubuntu (Canonical, 2018)

3.4 Instalación y configuración de servicios LTSP

Una vez instalado el servidor ver Anexo B, comenzamos con la instalación de los diferentes servicios.

3.4.1. Instalación y configuración de DNS (Sistema de Nombres de Dominio)

Un servidor DNS (Domain Name System) es un sistema que nos permite usar nombres de dominio en lugar de direcciones IP. Su principal ventaja es que para nosotros es mucho más fácil recordar un nombre que una dirección IP.

Instalación y configuración del servidor DNS más utilizado es Bind.

Los valores que debemos tener claros antes de comenzar son los siguientes:

Dirección IP del servidor: 192.168.0.101
Dominio que vamos a crear: www.intranet.com

Actualizamos la información de los repositorios con el siguiente comando:

```
# sudo apt-get update
```

Instalamos el servidor DNS Bind9:

```
# sudo apt-get install bind9
```

Editamos el archivo `/etc/bind/named.conf.local` con el siguiente comando:

```
# sudo nano /etc/bind/named.conf.local
```

y añadimos el siguiente contenido:

```
zone "www.intranet.com" {
    type master;
    file "db.intranet.com";
};

zone "0.168.192.in-addr.arpa" {
    type master;
    file "db.192.168.0";
};
```

Para comprobar la sintaxis del archivo de configuración ejecutamos el siguiente comando:

```
# named-checkconf
```

Creamos el archivo `/var/cache/bind/db.intranet.com`:

```
#sudo nano /var/cache/bind/db.intranet.com
```

Incluimos el siguiente contenido;

```
; BIND data file for local loopback interface
;
$TTL 604800
@ IN SOA www.intranet.com. root.intranet.com. (
    2      ; Serial
    604800 ; Refresh
    86400  ; Retry
    2419200 ; Expire
```

```
604800 ) ; Negative Cache TTL
;
@ IN NS www.intranet.com.
@ IN A 192.168.0.101
www IN A 192.168.0.102
```

A continuación creamos el archivo /var/cache/bind/db.192.168.0 para la zona inversa:

```
# sudo nano /var/cache/bind/db.192.168.0
```

Incluimos el siguiente contenido:

```
; BIND reverse data file for local loopback interface
;
$TTL 604800
@ IN SOA www.intranet.com. root.intranet.com. (
        1 ; Serial
        604800 ; Refresh
        86400 ; Retry
        2419200 ; Expire
        604800 ) ; Negative Cache TTL
;
@ IN NS www.intranet.com.
250 IN PTR intranet.com.
```

Reiniciamos el servicio

```
# /etc/init.d/bind9 restart
```

Instalación del servidor SSH.

```
# apt-get install ssh
```

Se instala el paquete LTSP.

```
# apt-get install ltsp-server-standalone
```

Este proceso se tardará de 10 a 30 minutos según los recursos del sistema.

Una vez terminada la instalación anterior crearemos la imagen del servidor LTSP.

sudo ltsp-build-client

Los demás servicios instalados se encuentran en Anexo B, Anexo C, Anexo D y Anexo E.

3.5. Metodología del laboratorio virtual.

Para la realización del sistema de simulación utilizaremos la metodología de Programación Extrema X.P., en el cual se presentan las historias de usuario para la planificación del sistema y el diseño de diagramas UML para reflejar el modelado de procesos que serán desarrolladas en el software, contribuyendo a una mejor representación de datos y flujo de la información.

3.4.1. Etapa I. Planificación

Se presentan las historias de los usuarios y los planes de realización más importantes.

3.4.1.1 Historia de Usuario.

Historia de Usuario	
Numero: 1	Usuario: Administrador, Usuario
Nombre historia: Autenticar Usuario	
Puntos Estimados:0.1	Iteración asignada:1
Riesgo en desarrollo: Media	
Descripción de Tarea: El usuario se autentica en el sistema para acceder a las funcionalidades del mismo.	

Tabla 1. Historia de Usuario "Autenticar Usuario"
Fuente: Elaboración Propia

Historia de Usuario	
Numero: 2	Usuario: Administrador
Nombre historia: Gestionar Cuenta Usuario	
Puntos Estimados: 0.1	Iteración asignada: 1
Riesgo en desarrollo: Media	

Descripción de Tarea: Se podrá crear, modificar y eliminar las cuentas de usuario en el sistema.
--

Tabla 2. Historia de Usuario "Gestionar Cuenta Usuario"

Fuente: Elaboración Propia

Historia de Usuario	
Numero: 3	Usuario: administrador
Nombre historia: Gestionar temas	
Puntos Estimados: 0.2	Iteración asignada: 1
Riesgo en desarrollo: Alto	
Descripción de Tarea: Permitir ingresar, modificar o eliminar la información relacionada con los temas.	

Tabla 3. Historia de Usuario "Gestionar Producto"

Fuente: Elaboración Propia

3.4.2. Etapa II. Diseño

3.4.2.1 Diagramas de Caso de Uso

Los casos de uso modelan el sistema desde el punto de vista del usuario, obteniendo la funcionalidad y operatividad del sistema.

A continuación, se realiza el diagrama de Caso de uso del estudiante en el sistema.

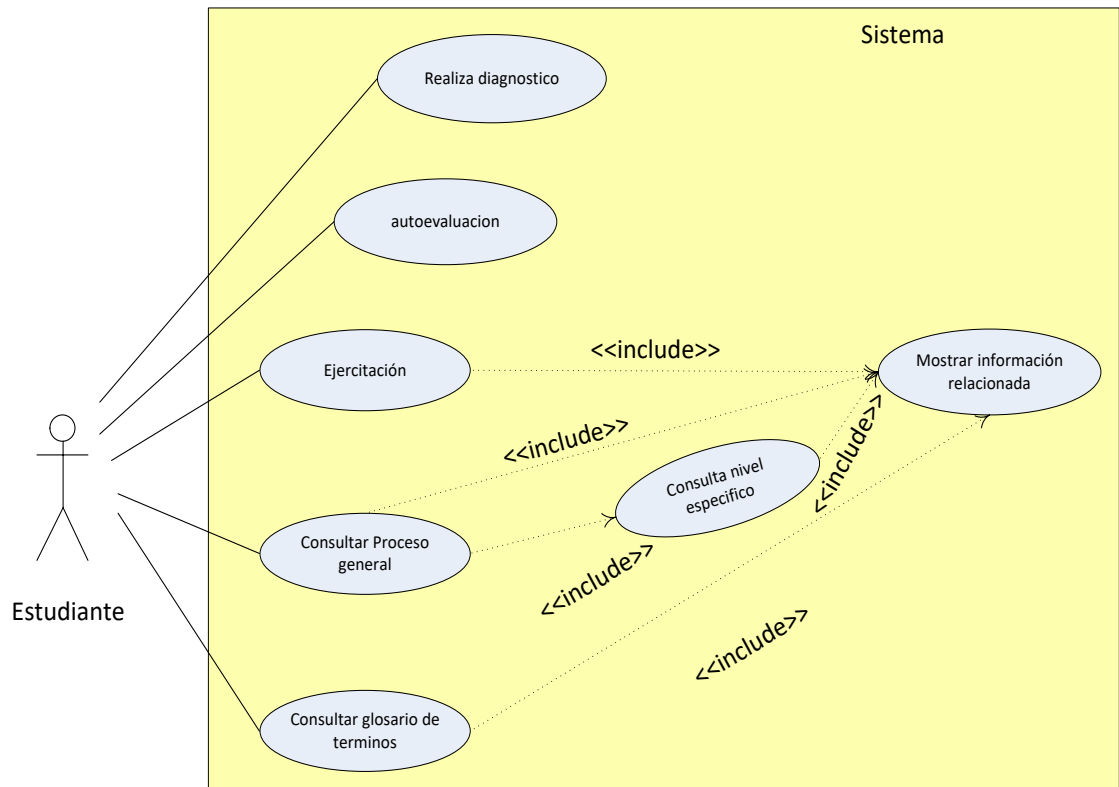


Figura 2. Diagrama de Caso de Uso del Estudiante
Fuente: Elaboración Propia

Caso de Uso: 1	Caso de uso Estudiante
Actores:	Estudiantes
Propósito:	Automatizar procesos de registros y administrar módulos
Tipo:	Primario
Curso Normal de Eventos	
Acción de los Actores	Respuesta del Sistema
El estudiante se registra al sistema.	
	El sistema guarda los datos para posteriormente generar pruebas de suficiencia a partir de estos datos.
El administrador gestiona y controla los módulos de aprendizaje.	
	El sistema guarda los cambios realizados por el administrador.

Tabla 4. . Diagrama de Caso de Uso del Estudiante

Fuente: Elaboración Propia

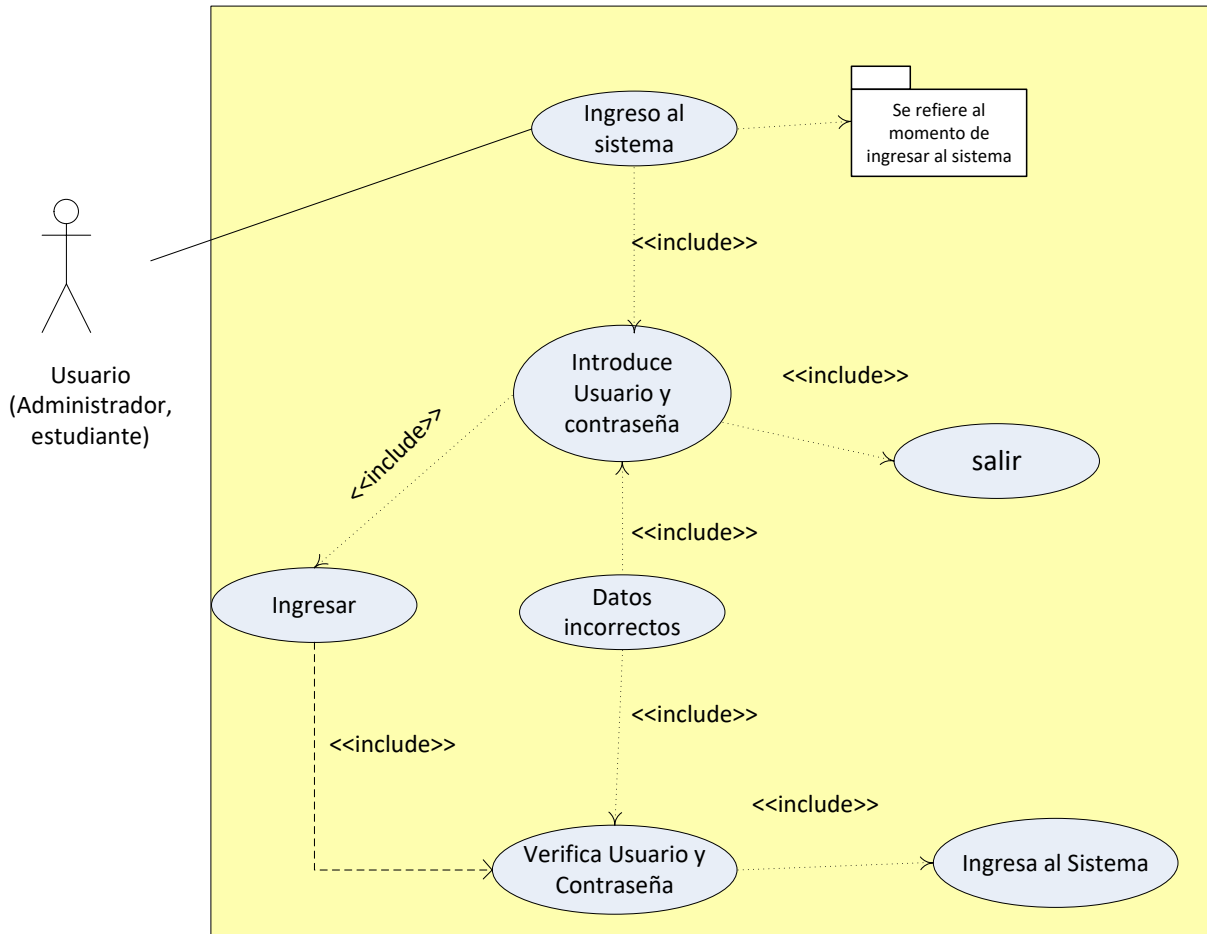


Figura 3. Diagrama de Caso de Uso Ingreso al sistema

Fuente: Elaboración Propia

Caso de Uso: 2	Ingreso al Sistema
Actores:	Usuarios (Administrador, estudiante)
Propósito:	Ingreso al Sistema
Resumen. Este Caso de uso comienza cuando el usuario ingresa los datos de usuario y contraseña para acceder al sistema.	
Tipo:	Primario
Curso Normal de Eventos	
Acción del Actor	Respuesta del Sistema
Selecciona ingresar al sistema.	
	El sistema pide Usuario y Contraseña para el acceso al sistema.
El usuario introduce los datos de usuario y contraseña.	
	Si los datos son correctos el sistema le habilita para acceder al sistema.
	Si los datos son incorrectos no permitirá el acceso al sistema.

Tabla 5. Caso de Uso Ingreso al Sistema

Fuente: Elaboración Propia

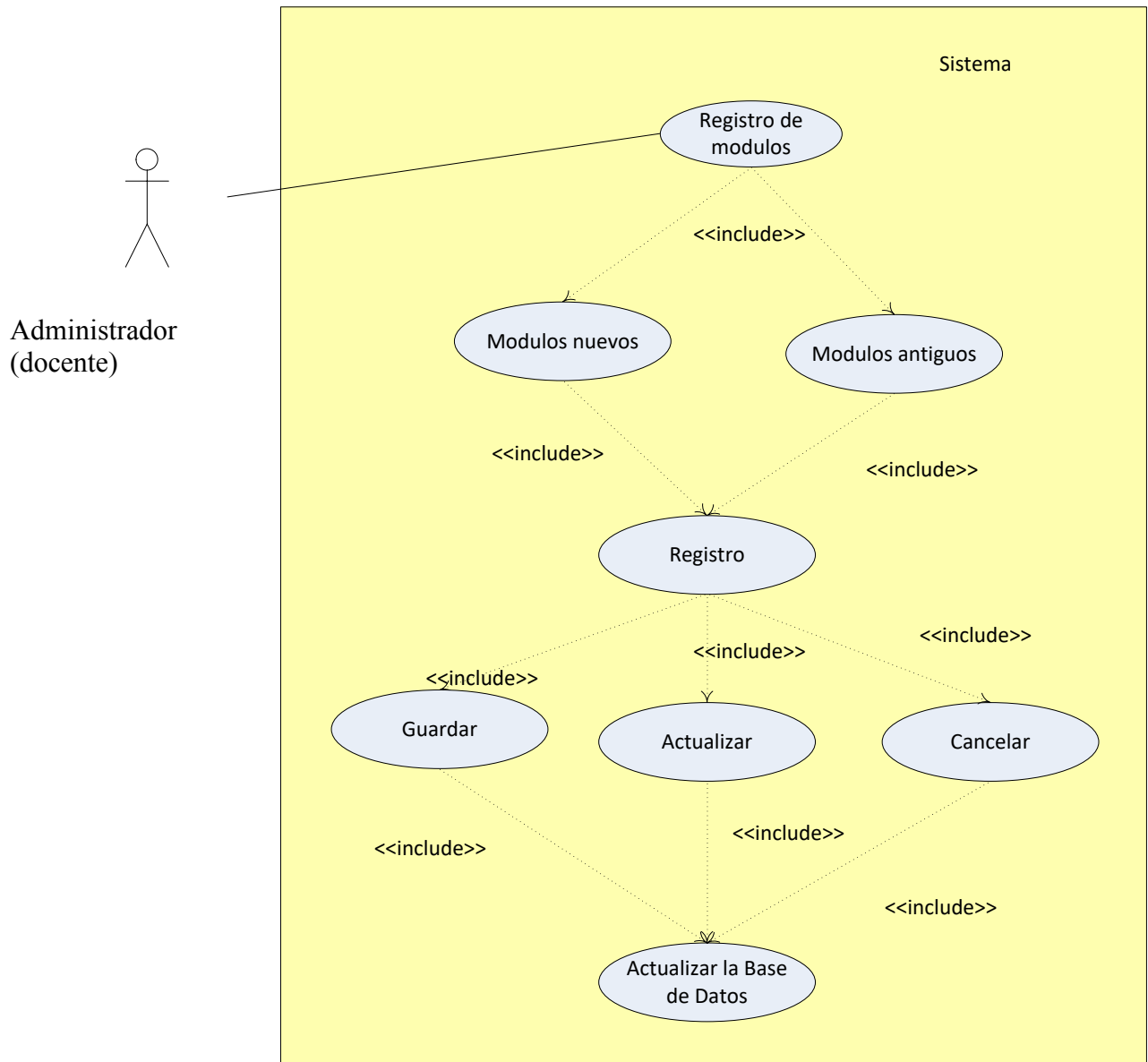


Figura 4. Diagrama de Caso de Uso Registro de Material
Fuente: Elaboración Propia

Caso de Uso: 3	Registro de Módulos de aprendizaje
Actores:	Administrador
Propósito:	Almacenar Módulos de aprendizaje
Resumen. Este Caso de uso comienza cuando el administrador accede a registrar o actualizar los datos de los módulos de aprendizaje	
Tipo:	Primario
Curso Normal de Eventos	
Acción del Actor	Respuesta del Sistema
El caso de uso comienza cuando el encargado ingresa al administrador para nuevo modulo	
	Muestra los módulos.
Elige la opción nueva modulo	
	Muestra el llenado de módulos a ingresar
Introduce la información de los nuevos módulos y luego presionar en guardar.	
	Valida los datos y se almacena en la base de datos.
	Muestra los módulos de aprendizaje.

Tabla 6. Caso de Uso Registro de Módulos de aprendizaje
Fuente: Elaboración Propia

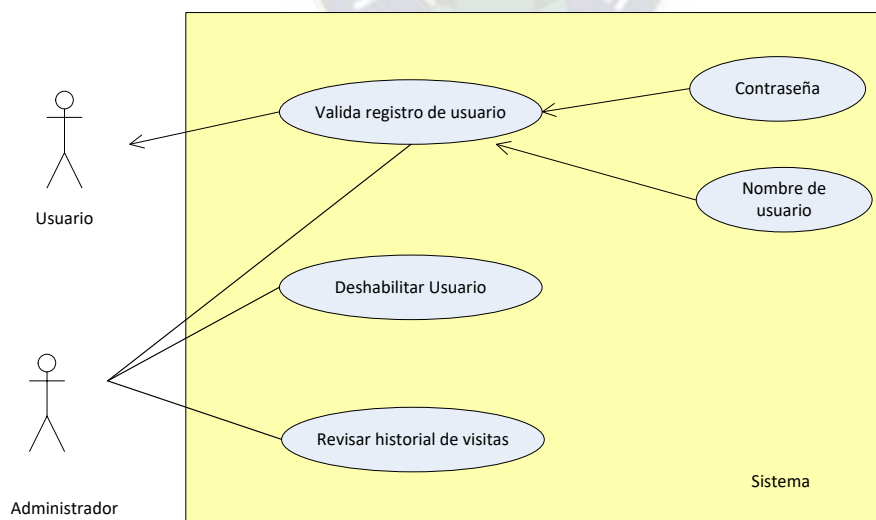


Figura 5. Administrar Usuarios
Fuente: Elaboración Propia

Caso de Uso: 4	Administrar Usuarios
Actores:	Administrador del sistema
Propósito:	Administrar los usuarios que tendrán acceso al sistema
<p>Resumen.</p> <p>Es el encargado de buscar, aprobar, adicionar, eliminar y actualizar a los usuarios, también podrá dar privilegios de uso desacuerdo al nivel que tenga cada usuario.</p>	
Tipo:	Primario
Curso Normal de Eventos	
Acción del Actor	Respuesta del Sistema
El caso de uso comienza cuando el administrador del sistema ingresa con su usuario y contraseña.	
	Verifica la autenticación de los datos del administrador.
Elige la opción usuarios.	
	Muestra un listado de los usuarios y su respectiva información.
Elige la opción de nuevo usuario.	
	Muestra el formulario de registro de nuevos usuarios.
Ingresa los datos del nuevo usuario y presiona guardar.	
	Valida y registra los datos en la base de datos.

Tabla 7. Administrar Usuarios

Fuente: Elaboración Propia

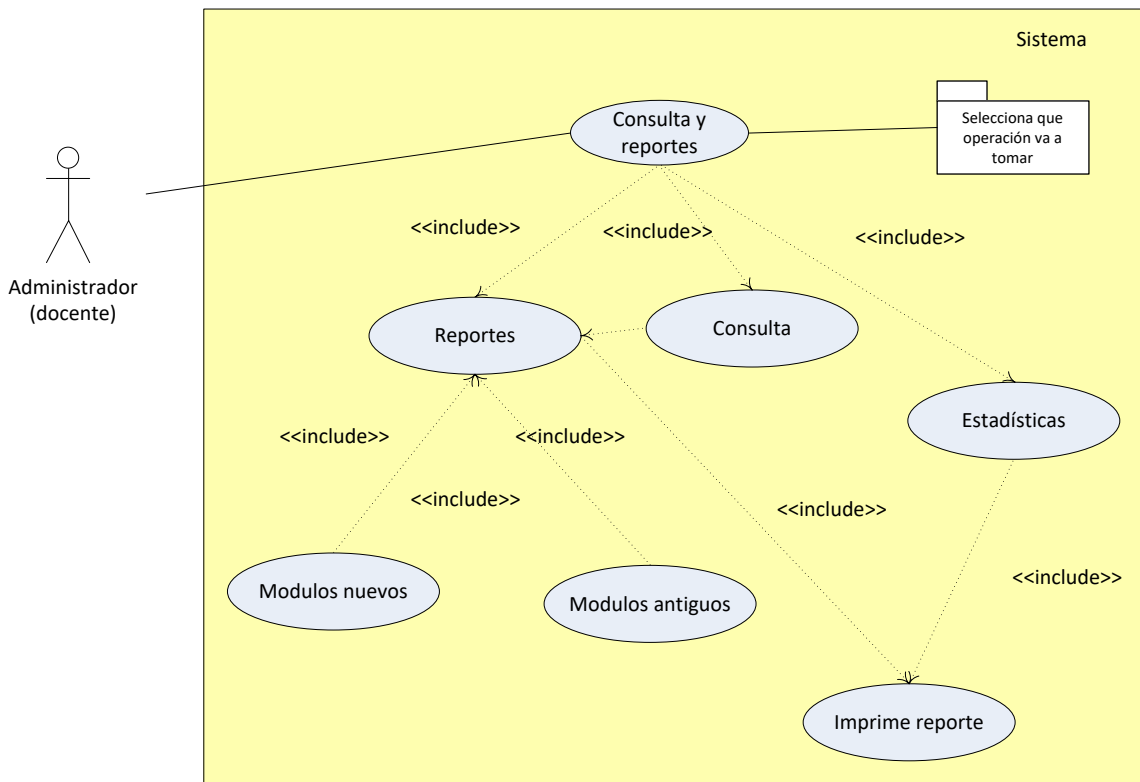


Figura 6. Caso de Uso Consultas y Reportes
Fuente: Elaboración Propia

Caso de Uso: 5	Consultas
Actores:	Usuario (Docente, Estudiante)
Propósito:	Realizar consultas o reportes con el sistema.
Resumen.	Este caso de uso comienza cuando el administrador (Docente) desea informarse acerca del rendimiento y cumplimiento de deberes de los estudiantes, para eso debe hacer la consulta y reporte.
Tipo:	Primario
Curso Normal de Eventos	
Acción del Actor	Respuesta del Sistema
El caso de uso comienza cuando selecciona la opción de consultas y reportes.	
	El sistema muestra el menú de opciones reporte y consultas.

El administrador (docente) realiza las operaciones de acuerdo a las órdenes del sistema en las diferentes opciones que se determine.	
	Una vez elegida la opción de consultas y reportes, la pantalla le muestra el menú.
	Confirma la operación realizada y puede hacer las diferentes consultas y reportes que ofrece el sistema.
Los usuarios (estudiantes) pueden verificar sus datos y diferentes consultas con el sistema.	

Tabla 8. Caso de Uso Consultas
Fuente: Elaboración Propia

3.4.2.2 Diagrama de Clase

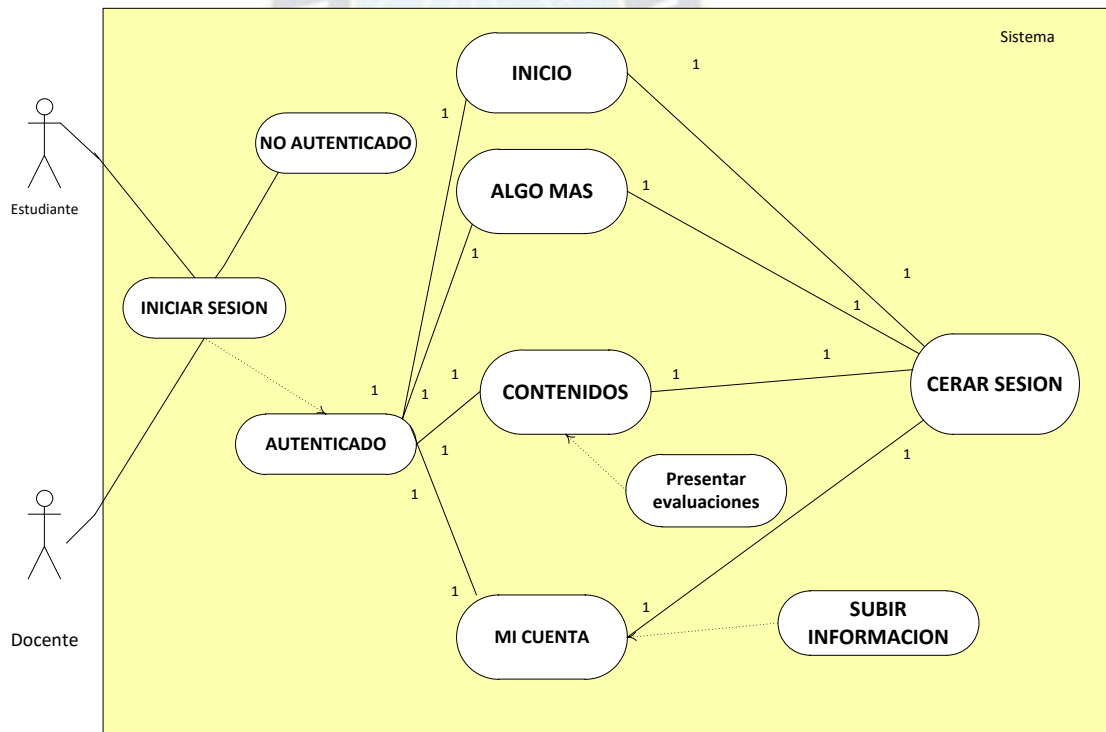


Figura 7. Diagrama de Clases General
Fuente: Elaboración Propia

3.4.2.3 Diagrama de Secuencia

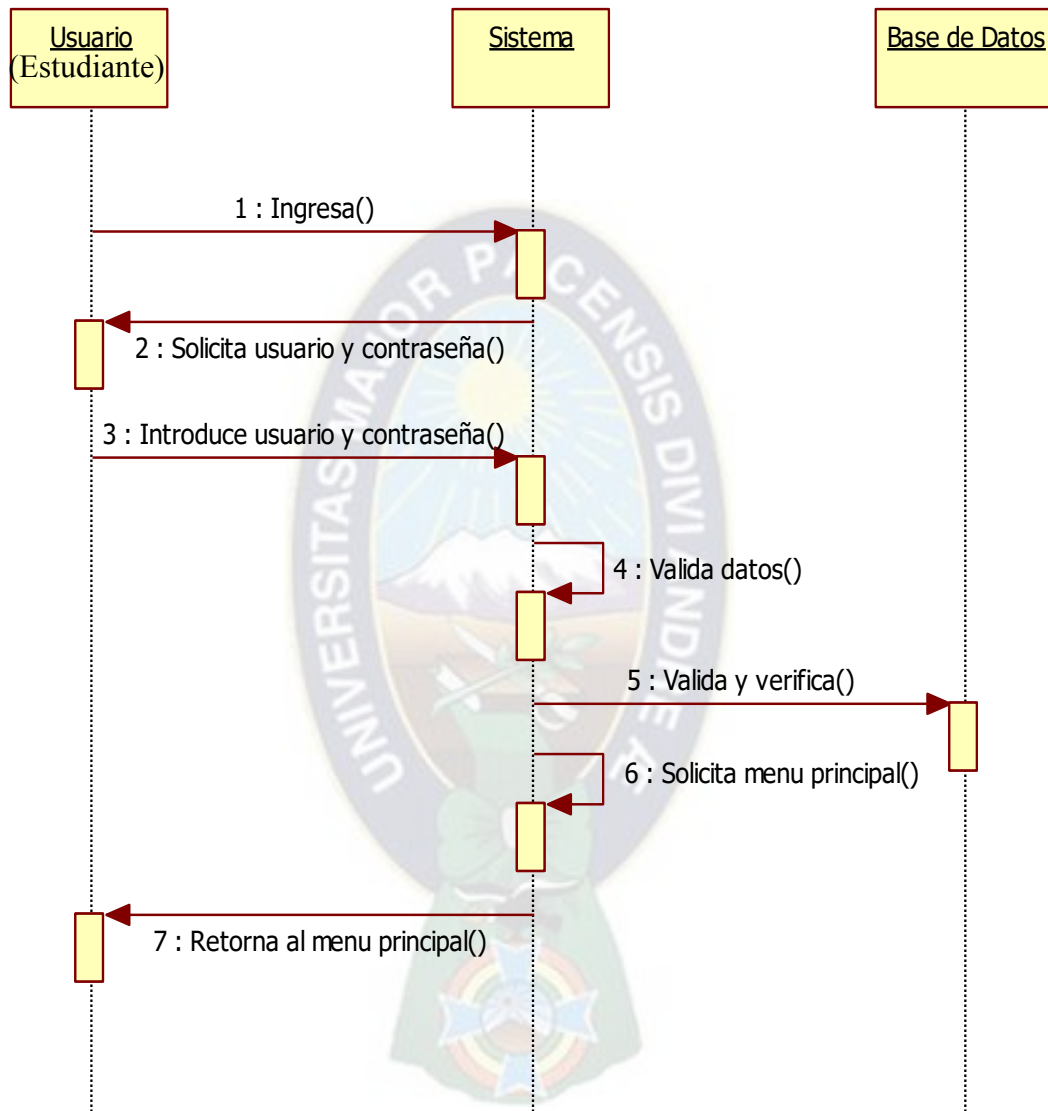


Figura 8. Diagrama de Secuencia del proceso de Ingreso al sistema
Fuente: Elaboración Propia

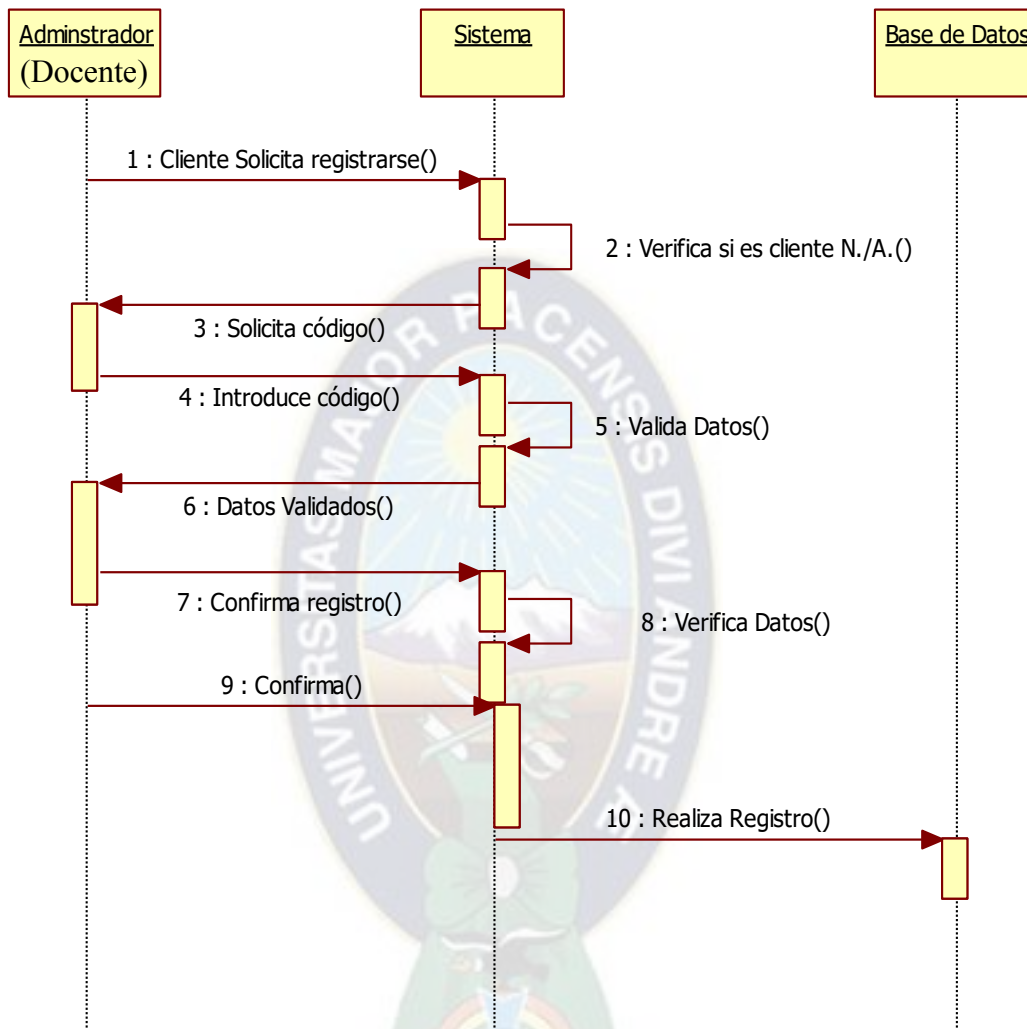


Figura 9. Diagrama de Secuencia del proceso de Registro del Cliente
Fuente: Elaboración Propia

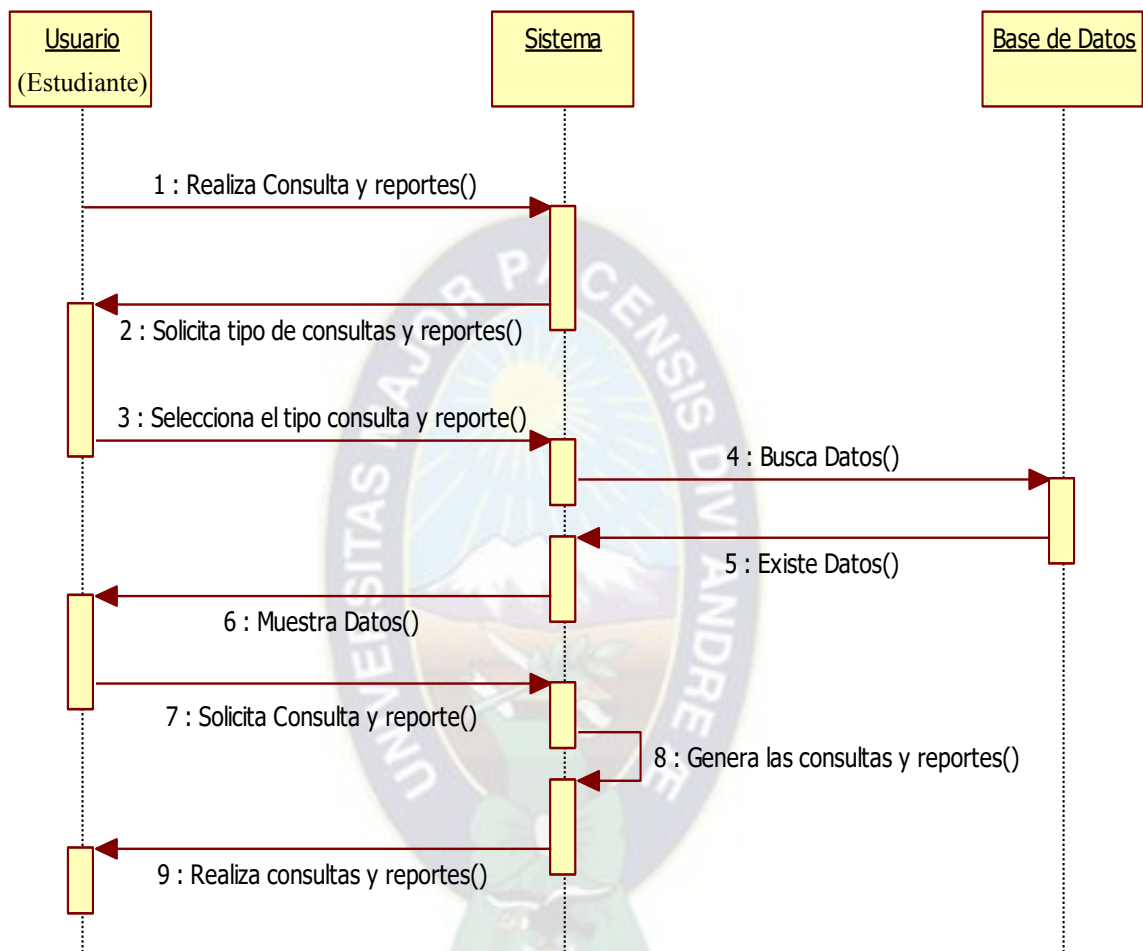


Figura 10. Diagrama de Secuencia del proceso de Reportes
Fuente: [Elaboración Propia]

3.4.2.3 Diagrama de Actividades

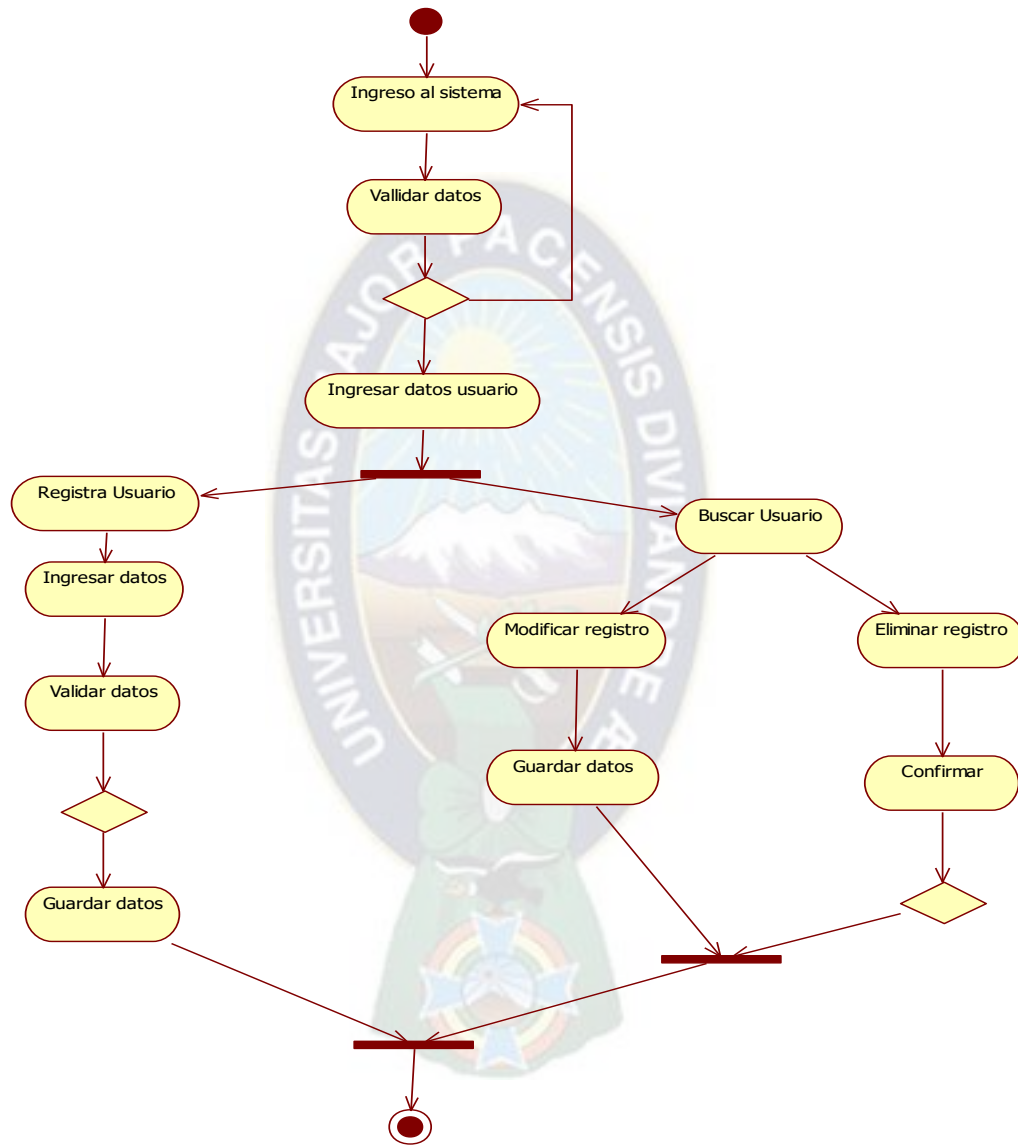




Figura 11. Diagrama de Actividades, Ingreso de Usuarios
Fuente: Elaboración Propia

3.4.2.4 Diagrama de Colaboración



Figura 12. Diagrama de Colaboración, Registro Cliente
Fuente: Elaboración Propia

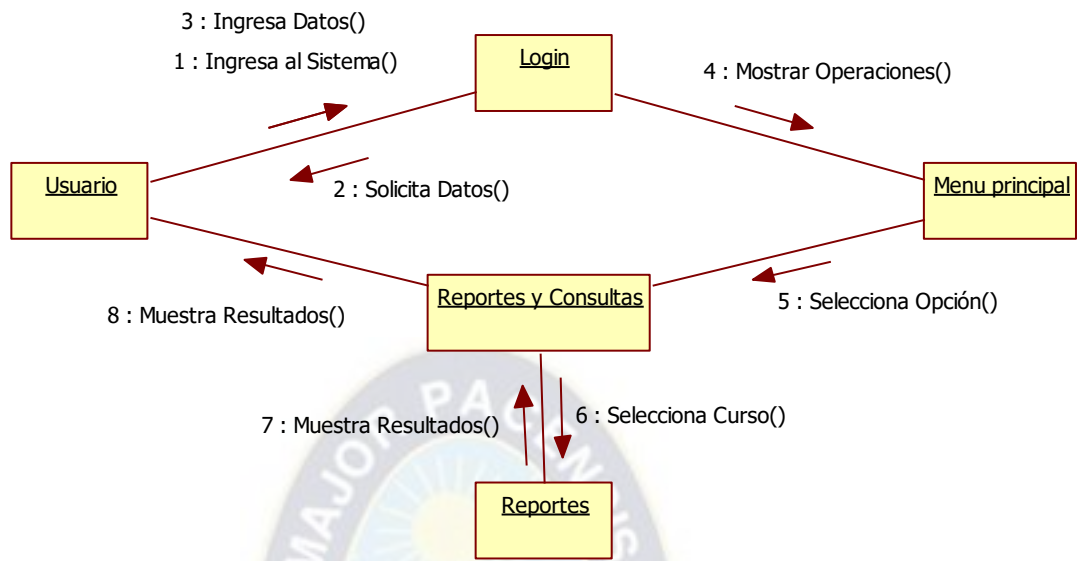


Figura 13. Diagrama de Colaboración, Reportes
Fuente: Elaboración Propia

3.4.2.4 Diagrama de Componentes

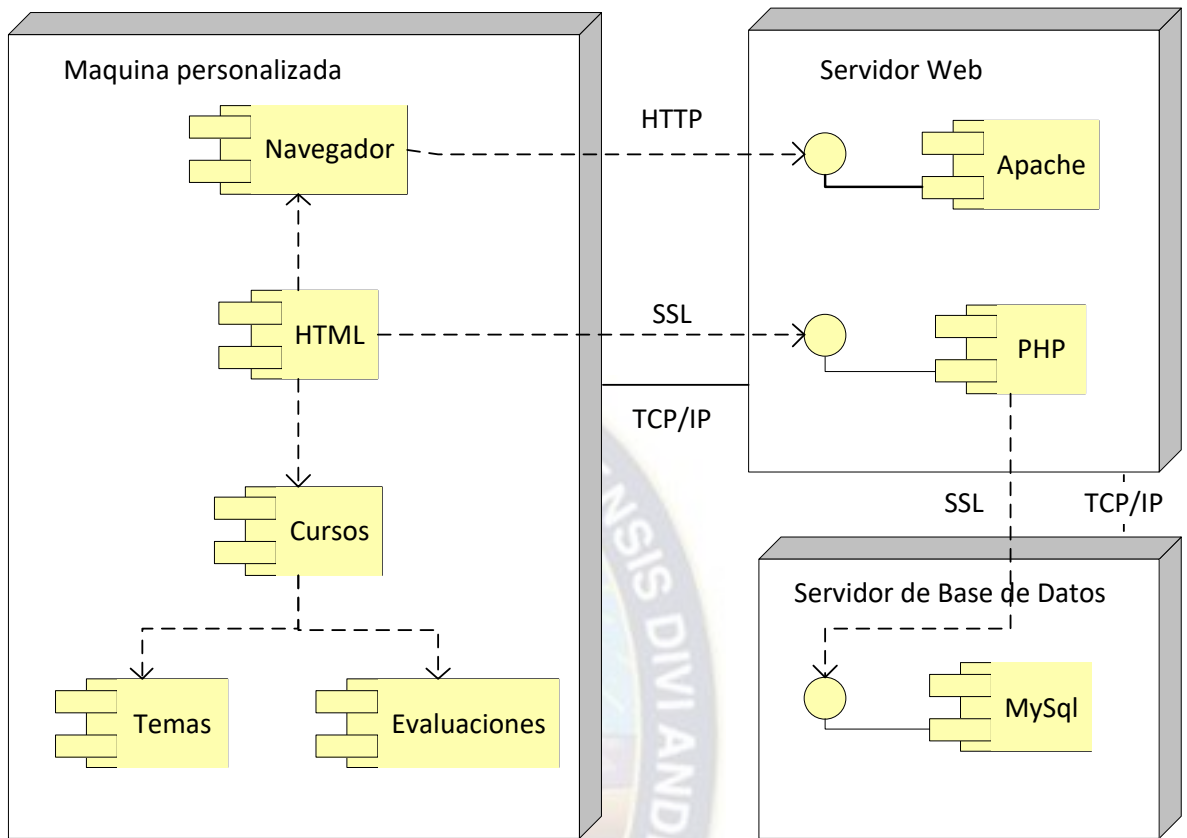


Figura 14. Diagrama de Componentes General
Fuente: Elaboración Propia

3.4.2.5 Diagrama de Componentes

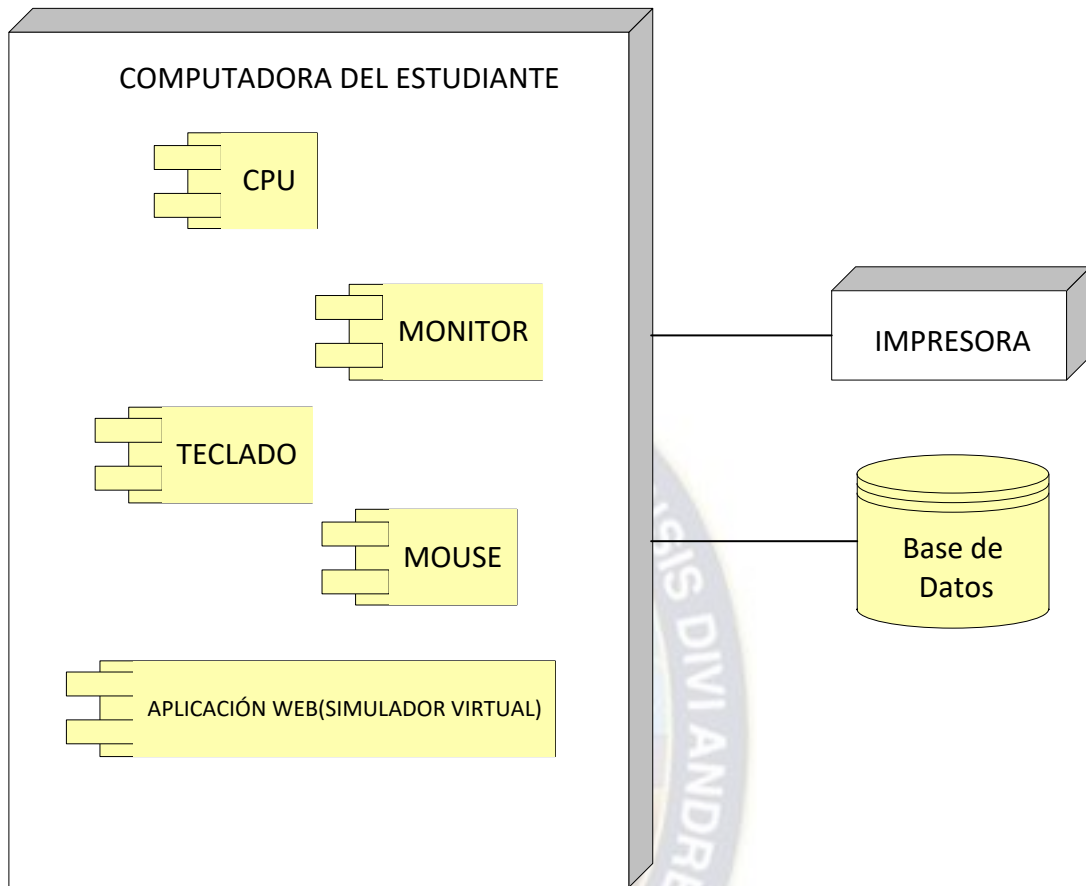


Figura 15. Diagrama de Despliegue
Fuente: Elaboración Propia

3.4.3 Etapa III. Desarrollo

Todo el proceso de modelado del sistema se ve reflejado en las siguientes capturas de pantallas del software, desarrollado con los lenguajes de PHP, Javascript y HTML apoyándose en el gestor de bases de datos de MySQL.

Interfaz del Sistema



Figura 16. Pantalla de Inicio de Sesión
Fuente: Elaboración propia

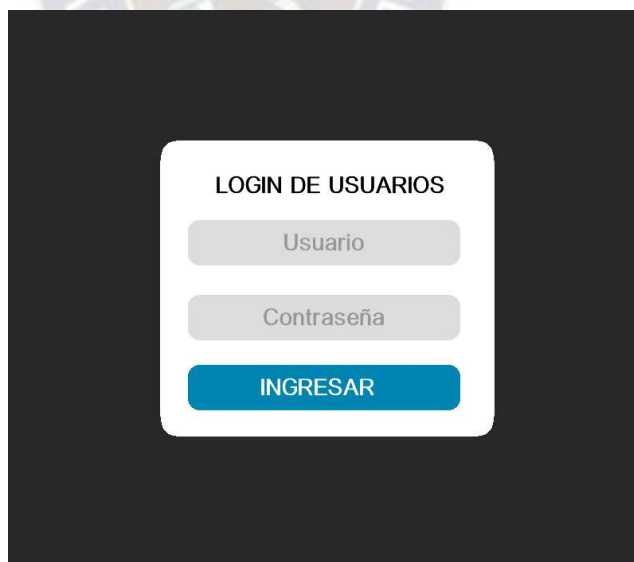


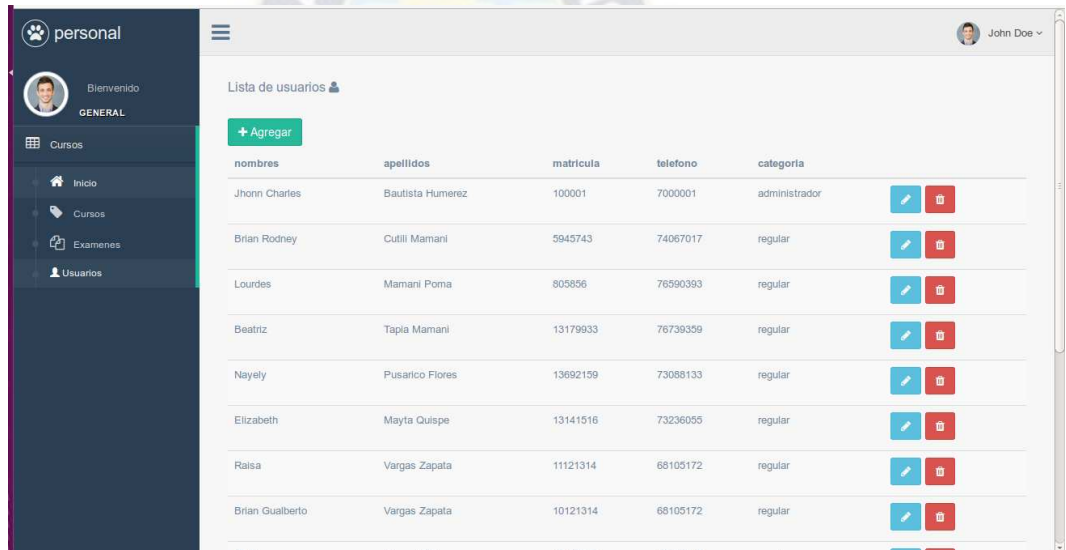
Figura 17. Pantalla de autenticación para el ingreso
Fuente: Elaboración propia



Figura 18. Pantalla de los temas o módulos
Fuente: Elaboración propia



Figura 19. Pantalla de avance de curso
Fuente: Elaboración propia



personal

Bienvenido
GENERAL

Cursos

Inicio

Cursos

Exámenes

Usuarios

John Doe

Lista de usuarios

+ Agregar

nombres	apellidos	matricula	telefono	categoria		
Jhonn Charles	Bautista Humerez	100001	7000001	administrador		
Brian Rodney	Cutlli Mamani	5945743	74067017	regular		
Lourdes	Mamani Poma	805856	76590393	regular		
Beatriz	Tipia Mamani	13179933	76739359	regular		
Nayely	Pusarico Flores	13692159	73088133	regular		
Elizabeth	Mayta Quispe	13141516	73236055	regular		
Raisa	Vargas Zapata	11121314	68105172	regular		
Brian Gualberto	Vargas Zapata	10121314	68105172	regular		

Figura 20. Lista de estudiantes
Fuente: Elaboración propia

Reporte de usuarios

Id	Nombre	Domicilio	Telefono	Email
1	Agustin Ramos	Mexico	+5219371331142	evlinapsis@gmail.com
2	Agustin 2 Ramos	Mexico	+5219371331142	xzcbnm@gmail.com
3	Agustin 4 Ramos	Mexico	+52178798798	nanana@gmail.com
4	Agustin 6 Ramos	Mexico	+521123456789	qertyu990@gmail.com

Figura 21. Reporte de Estudiantes
Fuente: Elaboración propia

personal

Bienvenido

GENERAL

Cursos

Inicio

Cursos

Exámenes

Usuarios

John Doe

Formulario de ingreso tabla

nombres:

apellidos:

matricula:

telefono:

Tipo de usuario:

usuario maquina virtual:

password maquina virtual:

Elizabeth	Mayta Quispe	13141516	73236005	regular	<input type="button" value="✎"/>	<input type="button" value="✖"/>
Raisa	Vargas Zapata	11121314	68105172	regular	<input type="button" value="✎"/>	<input type="button" value="✖"/>
Brian Gualberto	Vargas Zapata	10121314	68105172	regular	<input type="button" value="✎"/>	<input type="button" value="✖"/>

Figura 22. Administración de usuarios
Fuente: Elaboración propia



Capítulo 4

Costos

4.1. Introducción.

El estudio de costos y beneficios, consiste en la necesidad de demostrar la viabilidad del proyecto, hacer un estudio de todos los costos que involucra el desarrollo del sistema y los beneficios que este traerá para la institución.

Se detallan los costos (hardware, software), de operación y los de mantenimiento del nuevo sistema. Y por otro lado, los beneficios que acarreará el sistema, éstos pueden ser de dos tipos: tangibles e intangibles. Se denominan tangibles aquellos beneficios que pueden ser medidos en términos económicos y los beneficios intangibles son aquellos que difícilmente pueden ser medidos en términos económicos, pero que aportan una mejora a la institución.

En el presente capítulo, se pretende mostrar los costos que conlleva la implementación del sistema, que generara la inversión realizada para el mismo.

4.2. Estudio de factibilidad

Factibilidad, se refiere a la disponibilidad de los recursos necesarios para llevar a cabo los objetivos o metas señaladas, la factibilidad se apoya en tres aspectos básicos:

- Factibilidad Técnica
- Factibilidad Económica.
- Factibilidad Operacional.

El éxito de un proyecto está determinado por el grado de factibilidad que se represente en cada uno de los tres aspectos anteriores.

El estudio de factibilidad requiere ser implementado con todas la posibles ventajas para la organización, pero sin descuidar ninguno de los elementos necesarios para que el proyecto funcione. Este estudio debe representar los gastos y los beneficios que presentara la puesta en marcha del sistema.

4.2.1. Factibilidad técnica

En la parte tecnológica, la Facultad de Ingeniería Electrónica cuenta con tecnología (tanto en Hardware como en Software) necesaria y suficiente para la implementación del sistema. En la siguiente tabla se detallan las características de los equipos promedio con los que cuenta la institución:

Hardware	Característica
Procesador	Pentium Dual Core CPU 2.60Ghz
Placa Madre	Intel E-5300
Memoria Ram	Kingston 2 Gb
Disco Duro	Western Digital 500 Gb
Case Combo	Delux (Teclado, Mouse, Parlantes)
Monitor	Samsung 19 Pulgadas
Impresora	CANNON MP 250 Multifunción

Tabla 9. Hardware Existente

Fuente: Elaboración Propia

El sistema propuesto ofrecerá respuestas adecuadas a las peticiones de los usuarios sin importar la ubicación de los mismos, garantizando exactitud, facilidad de acceso, confiabilidad y

seguridad de los datos. El sistema puede ser ampliado, ya que cumple con la propiedad de escalabilidad.

SOFTWARE	APLICACIÓN	LICENCIA
LINUX (UBUNTU)	SISTEMAS OPERATIVO	LIBRE
MYSQL	GESTOR DE BASE DE DATOS	LIBRE
PHP	LENGUAJE DE PROGRAMACIÓN	LIBRE
APACHE	SERVIDOR	LIBRE
MOZILLA FIREFOX	NAVEGADOR WEB	LIBRE

Tabla 10. Software

Fuente: Elaboración Propia

HARDWARE	CARACTERÍSTICA	VELOCIDAD
Switch 10000 TPLINK	TP 8 Puertos	1024 Kbps

Tabla 11. Equipo de Red

Fuente: Elaboración Propia

La facultad de ingeniería cuenta con estos equipos computacionales, también con una conexión Internet.

4.2.2. Factibilidad Económica.

La factibilidad del presente proyecto (memoria) se justifica económicamente desde dos puntos de vista, desde el punto de vista de inversión la facultad de ingeniería cuenta con los recursos necesarios para poder llevar adelante el presente proyecto, donde el estudiante fortalecerá su aprendizaje académico. Como la facultad cuenta con los equipos adecuados para implementar el sistema, no será necesaria la adquisición de otros equipos tecnológicos.

4.2.2.1 Costo de computadora

Los costos de computadora se tomaron en cuenta obteniendo los precios de los equipos en el mercado informal de la ciudad de La Paz, como se detalla a continuación:

HARDWARE	CARACTERÍSTICAS	COSTO
Monitor	19 pulgadas	700 Bs
Microprocesador	Pentium dual i7 CPU 2.60Ghz	1950 Bs
Placa Madre	INTEL DG41RQ	480 Bs
Memoria RAM	4 Gb / Kingston	200 Bs
Disco Duro	Western Digital 1 Tb SATA	650 Bs
Tarjeta de Red	Realtek RTL8110sC 10/100/1000 Mbps	200 Bs
Lector Grabador	LG DVD 1 – 8 X	230 Bs
Case Combo	De lux completo	250 Bs
Impresora	CANNON MP 250 Multifunción	780 Bs
	Total	5440 Bs

Tabla 12. Costo de Computadora
Fuente: Mercado Informal (Uyustus)

4.2.2.2 Costo de Internet

El costo en el que se incurrirá para el uso del internet se refleja en la siguiente tabla:

PROVEEDOR	PLAN	VELOCIDAD	COSTOS
AXS Bolivia	Turbo 1 Mb	1024 Kbps	398 Bs

Tabla 13. Costo de Internet
Fuente: AXS Bolivia

4.2.2.3 Costo de Software

Los costos de software necesarios para el desarrollo del presente proyecto son de cero ya que el sistema se realizará con software sin licencia, lo que significa que son de distribución libre, como se muestra en la siguiente tabla:

SOFTWARE	APLICACIÓN	LICENCIA	COSTO
MYSQL	GESTOR DE BASE DE DATOS	LIBRE	0 Bs
PHP	LENGUAJE DE PROGRAMACIÓN	LIBRE	0 Bs
APACHE	SERVIDOR	LIBRE	0 Bs
FIREFOX	NAVEGADOR WEB	LIBRE	0 Bs

Tabla 14. Costo de Software

Fuente: Elaboración Propia

4.2.2.4 Costo de Operación

Para el desarrollo del proyecto(memoria), se considera el uso de herramientas y aplicaciones que son de distribución gratuita (Software Libre) como PHP como lenguaje de programación, MySQL como gestor de base de datos y un servidor Apache, lo cual implica que no se harán gastos en la compra de herramientas (software) para el desarrollo.

Considerando, además, que el sistema es multiplataforma, se ve por conveniente implementarlo en un sistema operativo libre, evitando las aplicaciones de pago, así abaratando los costos del mismo. Por otra parte, económicamente el proyecto es factible, ya que la tecnología (Hardware y Software) con la que cuenta la facultad, hace que no se tenga que hacer nuevas adquisiciones en equipos computacionales que sirvan para desarrollar el sistema o para el funcionamiento del mismo.

4.2.3. Factibilidad Operacional

El Sistema de simulación, presenta un manejo sencillo e intuitivo basándose en páginas web por las cuales se navegará a través de hipervínculos y botones de comando, muy fáciles de operar, y que permitirán realizar las acciones requeridas por los estudiantes.

El diseño navegacional, está de acuerdo a las necesidades y requerimientos de la Facultad; por otro lado, el diseño de las interfaces de usuario tiene un diseño gráfico acorde a las diferentes actividades o acciones que se van ejecutando en el sistema.

Debido a las características mencionadas anteriormente, la manipulación del sistema será fácilmente comprendida por los estudiantes o los operadores del sistema con una mínima capacitación para el correcto funcionamiento del sistema.

4.3 Costos del Proyecto.

Para establecer el costo total del proyecto (memoria), se deben considerar las tareas y las funciones que se desarrollan. Las tareas se refieren a los pasos o etapas que se siguen para el desarrollo del sistema.

El costo se establece de acuerdo al esfuerzo empleado por un equipo para el desarrollo del sistema, pudiendo estos tardar varios días, meses o años en hacer cada tarea y consecuentemente la función misma del proyecto. Además, se toma en cuenta los recursos (tanto en Hardware como en Software) que son utilizados para desarrollar el sistema, y los costos variables, que son aquellos que en una u otra medida afectan el desarrollo y la implementación del sistema.

4.3.1. Costo de desarrollo del sistema.

La determinación de costos de desarrollo de un sistema, es una parte importante para lograr el éxito del mismo. Con ella podemos conocer a tiempo si el precio nos permite lograr la obtención de beneficios, luego de cubrir todos los costos necesarios para el desarrollo y el funcionamiento del sistema.

4.3.1.1. Aplicación del Modelo Cocomo II.

Como se mencionó en el capítulo anterior la metodología escogida para realizar la estimación del costo del proyecto fue COCOMOII, específicamente el modelo de diseño anticipado, debido a que es un modelo que ofrece buena seguridad en la estimación sin mostrar toda la complejidad del modelo de post-arquitectura.

4.3.1.2. Determinación de puntos de función

Para el presente proyecto se ha realizado el recuento de sus características en base a su funcionalidad como resultado se han obtenido los siguientes datos, en la tabla 15:

Parámetros de medición	Cuenta	Factores de ponderación			Valor Obtenido
		Simple	Media	Complejo	
Nro. de entradas de usuario	5	-	4	-	28
Nro. de salidas de usuario	2	-	5	-	20
Nro. de peticiones de usuario	5	-	4	-	28
Nro. de archivos	9	-	6	-	48
Nro. de interfaces externas	0	-	0	-	0
Cuenta total					124

Tabla 15. Recuento de Características
Fuente: Elaboración Propia

Entonces tenemos:

$$PF = 124$$

En base a la funcionalidad del sistema se han evaluado los 14 factores de ajuste, en la Tabla 17:

Factores de complejidad	Sin influencia	Incidental	Moderado	Medio	Significativo	Esencial	Fi
	0	1	2	3	4	5	
¿Requiere el sistema copias de seguridad y de recuperación fiables?				x			3
¿Se requiere comunicación de datos?					x		4
¿Existen funciones de procesamiento distribuido?					x		4
¿Es crítico el rendimiento?				x			3
¿Se ejecutará el sistema con un entorno operativo existente y fuertemente utilizado?					x		4
¿Requiere el sistema entrada de datos interactiva?						x	5
Facilidad Operativa					x		4
¿Se actualiza los archivos maestros de forma interactiva?					x		4
¿Son complejas las entradas, las salidas, los archivos o las peticiones?						x	5
Procesamiento interno complejo				x			3
Diseño de código reutilizable						x	5
Facilidad de Instalación					x		4
¿Soporta múltiples instalaciones en diferentes sitios?						x	5
Facilidad de cambios						x	5
Factor de ajuste de complejidad							58

Tabla 16. Evaluación de Factores de Ajuste
Fuente: Elaboración Propia

Determinamos los valores de ajuste (VAF):

$$VAF = 0.65 + 0.01 \times \sum_{i=1}^{14} V_i$$

Entonces tenemos:

$$VAF = 0.65 + 0.01 \times 58$$

$$VAF = 1.23$$

Calculando los Puntos de Función Ajustados (PFA):

$$PFA = PF \times VAF$$

Entonces tenemos:

$$PFA = 124 \times 1.23$$

$$VAF = 152.52$$

Calculando las Líneas de Código (LDC):

A partir del lenguaje de programación PHP se tiene que los puntos de función asignados son de 29.

Entonces tenemos:

$$LDC = \text{Factor LDC} / PF * PF$$

$$LDC = 29 \times 124$$

$$LDC = 3596$$

$$KLDC = 3596 / 1000$$

$$KLDC = 3.596$$

Ahora, para hallar el esfuerzo reemplazamos los valores obtenidos hasta ahora:

$$E = 2.4 \times (3.596)^{1.05} = 9.2$$

$$D = 2.5 \times (3.596)^{0.38} = 4.06$$

Para el cálculo del número de programadores para el desarrollo:

$$P = 9.2/4.06 = 2.26 \cong 2$$

Calculando los factores de escala (SF):

En base a la funcionalidad del sistema se han evaluado los Valores escalares visualizados en la tabla 18 a continuación:

Driver de Escala	Descripción	Calibración	Valor
PREC	variable de precedencia u orden secuencial del desarrollo	Muy Alto	1.24
FLEX	variable de flexibilidad del desarrollo	Alto	2.03
RESL	indica la fortaleza de la arquitectura y métodos de estimación y reducción de riesgos	Extra Alto	0.00
TEAM	esta variable refleja la cohesión y madurez del equipo de trabajo	Extra Alto	0.00
PMAT	relaciona el proceso de madurez del software	Muy Alto	1.56
Total Drivers de Escala (SF)			4.83

Tabla 17. Selección de Valores de escala
Fuente: Elaboración Propia

Calculando Factor escalar (B):

$$B = 0.91 + 0.01 \times SF$$

$$B = 0.91 + 0.01 \times 4.83$$

$$B = 1.01$$

Calculando el Esfuerzo Nominal (Enominal):

$$E_{NOMINAL} = A \times (Size)^B$$

$$A = 2.45$$

$$B = 0.96$$

$$Size = 14122$$

$$E_{NOMINAL} = 2.45 \times (3.596)^{0.96}$$

$$E_{NOMINAL} = 9 \text{ personas / mes}$$

Calculando los Drivers de Coste (EM):

En base a la funcionalidad del sistema se han evaluado los Valores escalares, en la tabla 19 a continuación:

Driver de Coste	Descripción	Calibración	Valor
RCPX	Complejidad del producto.	Nominal	1
RUSE	Reusabilidad.	Alto	1.07
PDIF	Dificultad de la plataforma.	Nominal	1
PERS	Capacidad del personal.	Extra Alto	0.50
PREX	Experiencia del personal.	Muy Alto	0.71
FCIL	Facilidades.	Muy Alto	0.73
SCED	Calendario.	Alto	1
Total Drivers de Coste (EM)			0.28

Tabla 18. Selección de Drivers de Coste

Fuente: Elaboración Propia

Calculando el Esfuerzo (Eajustado):

$$E_{AJUSTADO} = E_{NOMINAL} \times EM$$

$$E_{AJUSTADO} = 9 \times 0.28$$

$$E_{AJUSTADO} = 3 \text{ personas / mes}$$

Calculando el tiempo total de desarrollo (T):

$$T = 3.67(E_{AJUSTADO}^{(0.28+0.2(B-0.9))})Meses$$

$$T = 3.67(3^{(0.28+0.2(0.96-0.9))})Meses$$

$$T = 5Meses$$

Calculando el número de personas (NP):

$$NP = \frac{E_{AJUSTADO}}{T} Person\text{as}$$

$$NP = \frac{3}{5} Person\text{as}$$

$$NP = 1Person\text{as}$$

Calculando el costo personal (CP):

$$CP = T \times NP \times Sueldo$$

Después de haber realizado entrevistas a grupos de consultoras (Megalink, GlobalNet, Holismatica) en fecha Mayo del 2018 se estimo un promedio de salario de 250 \$us mensuales por concepto de pago a los programadores.

$$CP = 5 \times 1 \times 250\$us$$

$$CP = 1250\$us$$

Calculando el costo por Línea de Código (CLDC):

$$CLDC = \frac{CP}{LDC}$$

$$CLDC = \frac{1250\$us}{3596}$$

$$CLDC = 0.34\$us$$

Luego de haber realizado el análisis de la factibilidad del presente proyecto, se llegó a las siguientes conclusiones respecto al costo del sistema:

El costo total del desarrollo del proyecto (memoria) es de 1250 \$us.

El tiempo estimado para la realización del proyecto es de 5 Meses.

El esfuerzo para la realización del proyecto es de 3 Personas/Mes.

La cantidad de participantes necesarios para la realización del proyecto es de 1 Persona.

4.3.2 Costos fijos.

Los Costos Fijos son aquellos cuyo monto total no se modifica de acuerdo con el desarrollo del sistema. En otras palabras, se puede decir que los Costos Fijos varían con el tiempo más que con la actividad; es decir, no cambian durante un periodo específico.

En el desarrollo del sistema, se toma en cuenta dos tipos de costos fijos, los costos de hardware y los costos de software que se necesitan para el desarrollo.

4.3.2.1 Costos de hardware

Considerando que la Facultad de Ingeniería cuenta con los equipos necesarios, como se menciona en el estudio de factibilidad económica anteriormente, y que el sistema no requiere de

equipos especiales para su desarrollo e implementación, no será necesario adquirir nuevos equipos computacionales por parte de la institución.

Para el desarrollo, no se requiere de hardware especial ya que se hace uso de software de desarrollo que no requieren de equipos adicionales a un ordenador. Para la etapa de implementación, tampoco se necesita hardware que no sea un ordenador con una conexión a internet y una impresora, equipos con que la facultad cuenta.

4.3.2.2 Costos de Software.

Los costos de software, están relacionados directamente con las aplicaciones o programas que se requieren para el desarrollo y la implementación del sistema.

A continuación se detalla el software requerido para el desarrollo del sistema, los cuales son de libre distribución y no requieren de pagos de licencias para su funcionamiento, por lo que ayuda a reducir los costos al mínimo.

SOFTWARE	APLICACIÓN	LICENCIA
MYSQL	GESTOR DE BASE DE DATOS	LIBRE
PHP	LENGUAJE DE PROGRAMACIÓN	LIBRE
APACHE	SERVIDOR	LIBRE
MOZILLA FIREFOX	NAVEGADOR WEB	LIBRE
KOMPOSER	EDITOR DE APLICACIONES WEB	LIBRE
VIRTUALBOX	SIMULACION DE SISTEMAS OPERATIVOS	LIBRE

Tabla 19. Software para el Desarrollo del Sistema
Fuente: Elaboración Propia

4.3.3 Costos variables

Los costos variables, son los que están directamente involucrados con la producción o el desarrollo del sistema, por lo que tienden a variar al momento de desarrollar o al momento de implementarlo en la facultad.

Se debe tomar en cuenta, los costos que son externos al sistema, pero que de una u otra forma, afectan económicamente el desarrollo o la implementación del sistema.

DESCRIPCIÓN	COSTOS \$US
Pago de servicios	70
Pago de Internet	30
Otros	40
TOTAL	140

Tabla 20. Resumen de Costos variables
Fuente: Elaboración Propia

4.3.4 Costos Totales

Los costos totales, del desarrollo y de la implementación del sistema en la institución, se calculan en base a los datos descritos en los puntos anteriores.

Con dichos datos se procede a realizar el cálculo:

$$\text{Costo Total} = \text{Costo del Sistema} + \text{Costos Fijos} + \text{Costos variables}$$

$$\text{Costo Total} = 1250 + 0 + 140$$

De donde, se obtiene que el costo total del Proyecto, tiene un valor de:

$$\text{Costo Total} = 1390 \text{ \$us.}$$

DESCRIPCIÓN	COSTOS \$US
Costo del Desarrollo del Sistema	1250
Costos Variables	140
TOTAL	1390

Tabla 21. Resumen de Costos Totales
Fuente: Elaboración Propia

4.4 Beneficios.

Los beneficios obtenidos por el Sistema de Simulación de aprendizaje de instalación y configuración de servicios Web, permitirá optimizar los recursos tecnológicos para el fortalecimiento del aprendizaje académico.

El presente sistema brindará beneficios que pueden clasificarse en beneficios tangibles y beneficios intangibles.

Los beneficios tangibles, son aquellos que son medibles en términos económicos mientras que los beneficios intangibles, son aquellos que no pueden ser cuantificables en términos económicos. Los mismos se detallan a continuación:

- **Beneficios tangibles.**
- Ahorro en servidores (hardware).
- Ahorro en configuraciones constantes.
- **Beneficios intangibles.**
- Reducción de errores de procesamiento.
- Incremento de flexibilidad.
- Mejoramiento en el seguimiento y control.
- Incremento en la velocidad de configuraciones de servidores.

- Información oportuna relacionada con el control y seguimiento de aprendizaje del estudiante.
- Impulso del uso de nuevas tecnologías en la institución.



Capítulo 5

Conclusiones y Recomendaciones

El diseño y desarrollo del laboratorio virtual permitió construir una herramienta tecnológica utilizando como medio de apoyo el aprendizaje, a partir de los diferentes procesos, lo cual permite al estudiante, docente complementar sus conocimientos sobre configuración e instalación de servicios de internet, interactuando de una forma mucho más amigable, lo que da como resultado una mejor motivación la realizar investigaciones relacionadas con estos temas.

Para la realización del diseño y desarrollo del simulador, se inició con un proceso de estudio y análisis de los diferentes temas que hacen parte de la Tecnología de información y comunicación (T.I.C.), en dicho proceso se determinó dividir en módulos (temas) donde se mostraran los temas más básicos de la configuración e instalación de servicios de Internet.

Luego de haber estudiado y analizado los módulos (temas) de los servicios que ofrece internet, se identificaron diferentes requerimientos, se determinó la parte del diseño gráfico de las interfaces que es muy importante para lograr capturar la atención de los estudiantes y así utilice esta herramienta, una interfaz gráfica que sea agradable a la vista, de fácil manejo y navegabilidad son aspectos que se tuvieron en cuenta al momento de realizar el laboratorio virtual.

En la etapa de desarrollo del software el lenguaje de programación que se utilizo es el PHP, ya que proporciono enlaces a la base de datos desarrollada en MySQL, lo que nos permitió afianzar más los conocimientos de esta herramienta tan importante en la programación Web. La utilización de MySQL como motor de base de datos permitió reforzar conocimientos acerca de esta

herramienta que es la más utilizada a nivel mundial cuando se trata de desarrollar aplicaciones Web, además de su gran compatibilidad con PHP.

La instalación de servicios es prácticamente cuestión de un minuto. La configuración sin embargo, en el Sistema Operativo Ubuntu Server carezca de interfaz gráfica le aporta dificultad y quizás un cierto temor a tocar algo sin darse uno cuenta y que todo deje de funcionar. Los puntos, comas y comillas tienen que estar en su sitio como indican las capturas, o de lo contrario nos dará error y nos volveremos locos buscándolo.

Recomendaciones

Universidad: La universidad debe seguir promoviendo la investigación sobre el campo de la tecnología y programación, para la innovación de nuevas herramientas de estudios que ayuden al mejoramiento del proceso Enseñanza-Aprendizaje, como es la creación de Software Educativos enfocados a las asignaturas de esta Universidad.

Estudiantes: Buscar nuevas metodologías para el proceso de enseñanza-aprendizaje, y vayan de la mano de las nuevas tecnologías de la programación, porque gracias a ellas se han realizado día a día laboratorios virtuales llevados más a la realidad.

Docentes: Seguir con ese ímpetu de mejorar la educación optimizando los recursos tecnológicos para el fortalecimiento del aprendizaje académico, continuando el desarrollo de experiencias sobre estos temas, hacerlas extensivas a otros centros de enseñanza.

Bibliografía

- Aguilera, Y. R. (junio de 2011). EL SOFTWARE EDUCATIVO COMO MEDIO DE ENSEÑANZA. (J. C. (CV), Ed.) *Cuadernos de Educación y Desarrollo*, 3(28), 2-30.
- Ayala Flores, L. M. Laboratorio virtual de líneas de transmisión. (*Proyecto de grado*). UNIVERSIDAD MAYOR DE SAN ANDRES, FACULTAD DE INGENIERIA, La Paz.
- Calle Tarqui, F. Q. Laboratorio virtual para el aprendizaje de Electrónica. (*proyecto de grado*). UNIVERSIDAD MAYOR DE SAN ANDRES, FACUTAD DE INGENIERIA.
- Canonical. (2018). *Ubuntu Downloader*. Obtenido de Ubuntu Server 18.04.1 LTS: <http://www.ubuntu.com/download/server>
- Cataldi, Z. L. (6 de enero de 2010). *INGENIERIA DE SOFTWARE EDUCATIVO*. Recuperado el 17 de mayo de 2018, de INGENIERIA DE SOFTWARE EDUCATIVO: <http://laboratorios.fi.uba.ar/lsi/c-icie99-ingenieriasoftwareeducativo.pdf>
- Echeverry Tobon, L. M. CASO PRACTICO DE LA METODOLOGIA AGIL XP AL DESARROLLO DEL SOFTWARE. (*Proyecto de Grado*). UNIVERSIDAD TECNOLOGICA DE PEREIRA, Pereira.
- Fernández, Y. B. (2005). Los Software Educativos. Metodología para la elaboración del guión de su diseño. El Software “Conociendo el Hardware”. *Forum de Base de Habana* (págs. 5-35). Cuba-La Habana: APPP Ciudad de La Habana.

Fundación Wikimedia, I. (31 de Agosto de 2018). *COCOMO*. Obtenido de Wikipedia-La Enciclopedia Libre: <https://es.wikipedia.org/wiki/COCOMO>

Juárez, I. A. (2013). Simuladores y laboratorios virtuales para Ingeniería en Computación. *Revista Iberoamericana para la Investigación y el Desarrollo Educativo*, 4-19.

Novo, L. V. (2 de julio de 2013). *Uso del software educativo en el proceso de enseñanza y aprendizaje*. Recuperado el 23 de mayo de 2018, de Gestipolis-conocimiento en negocios: <https://www.gestipolis.com/uso-del-software-educativo-en-el-proceso-de-ensenanza-y-aprendizaje/>

Rey., I. L. (1 de junio de 2011). *LA INSERCIÓN DE LA INFORMÁTICA EN LA EDUCACIÓN Y SUS EFECTOS EN LA RECONVERSIÓN LABORAL*. Recuperado el 15 de mayo de 2018, de Instituto de Formación Docente SEPA.: <http://www.c5.cl/ieinvestiga/actas/ribie96/Colombia.html>

Toranzo, L. Y. (2011). Software Educativo para fortalecer el aprendizaje de los productos en los estudiantes de 2do grado. *Ilustrados-Una Comunidad educativa Mundial*, 2-30.

ANEXOS

ANEXO A. Comparativa entre Windows y Linux.



	 Linux	 Windows
Precio	La mayoría de distribuciones de Linux están disponibles gratuitamente. ✓	Una licencia del sistema Windows puede costar entre \$190-\$230.
Uso	Linux ha mejorado dramáticamente, pero Windows sigue siendo un poco más fácil de usar.	Windows es un sistema operativo fácil de usar, con una interfaz amigable y conocida. ✓
Estabilidad	Linux es conocido por su estabilidad, puede correr por meses sin necesidad de reiniciarse. ✓	Windows ha mejorado en estabilidad, sin embargo sigue siendo inestable en ocasiones.
Software	Linux tiene una gran variedad de programas, utilitarios y juegos, pero Windows lo supera.	Debido a que Windows está instalado en la mayoría de PCs, tiene una mayor oferta de programas. ✓
Costo de Software	Muchos de los programas disponibles para Linux son gratuitos. Algunos muy avanzados como Gimp, OpenOffice, Wine, etc. ✓	Aunque Windows tiene una mayor selección de programas, la mayoría cuestan entre \$20 y \$200 por licencia.
Hardware	La compatibilidad de hardware ha avanzado notablemente en Linux, aunque algunas compañías de hardware aún no ofrecen soporte para el sistema.	Windows tiene mayor soporte de drivers para hardware debido a su mayor penetración en el mercado. ✓
Seguridad	Linux siempre ha sido un sistema operativo muy seguro. Aunque puede ser atacado ocasionalmente, la incidencia es muy baja comparada con Windows. ✓	En Vista la seguridad fue llevada al extremo tal que los usuarios casi no podían hacer nada sin autorización previa. Con todo y Windows sigue siendo un sistema vulnerable a los ataques, lo que lleva a un mayor costo en medidas de seguridad.
Código Abierto	Linux es de código abierto lo que significa que puede ser modificado a fin de adecuarse a las necesidades del usuario. ✓	Windows no es de código abierto y la mayoría de programas para Windows tampoco lo son.
Soporte	Las comunidades en torno a cada distribución de Linux son abundantes, con amplia documentación ofreciendo soporte valioso y gratuito. ✓	Windows también tiene un amplio soporte propio y ajeno a través de documentación online, libros y comunidades. ✓

Tabla 22. Comparación de sistemas operativos (Linux vs Windows)

ANEXO B. Instalación de Sistema Operativo Ubuntu 16.04 LTS.

Una vez ya corriendo el CD o DVD, primero se selecciona el idioma de nuestra preferencia para realizar la instalación.

A continuación, se selecciona instalar Ubuntu Server (Fig.2).

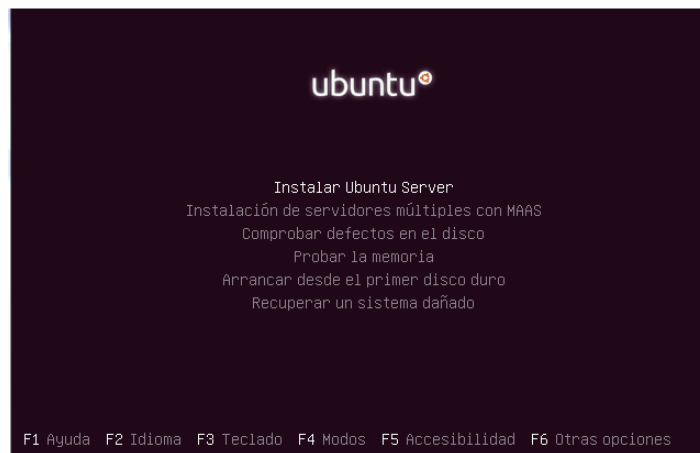


Figura 238. Inicio de instalación Ubuntu server

A continuación, se mencionan los pasos de instalación.

Como país de instalación le indicamos México, luego podemos cambiar a otro país.

Seleccionar el idioma del teclado en este caso será en español latinoamericano.

Comenzará la instalación de los primeros componentes y pedirá que seleccionemos nuestra interfaz de red principal.

Enseguida nos pedirá el nombre que tendrá nuestra máquina, así como la creación del primer usuario.

La instalación sigue y nos muestra la zona horaria (Fig. 2), verificamos que sea correcta y continuamos.

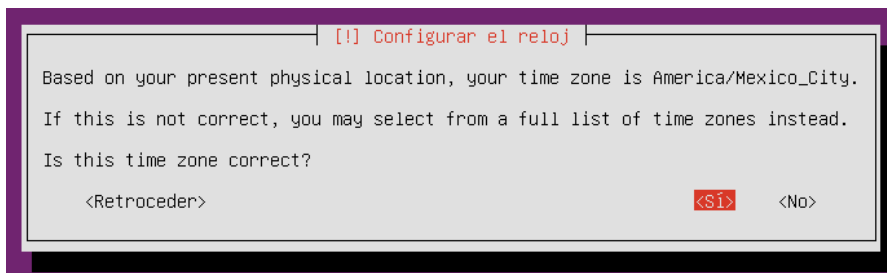


Figura 2. Verificación de Zona horaria

. Nos mostrará la forma de partición (Fig.3) que te ofrece para el disco, selección el deseado y continuamos.

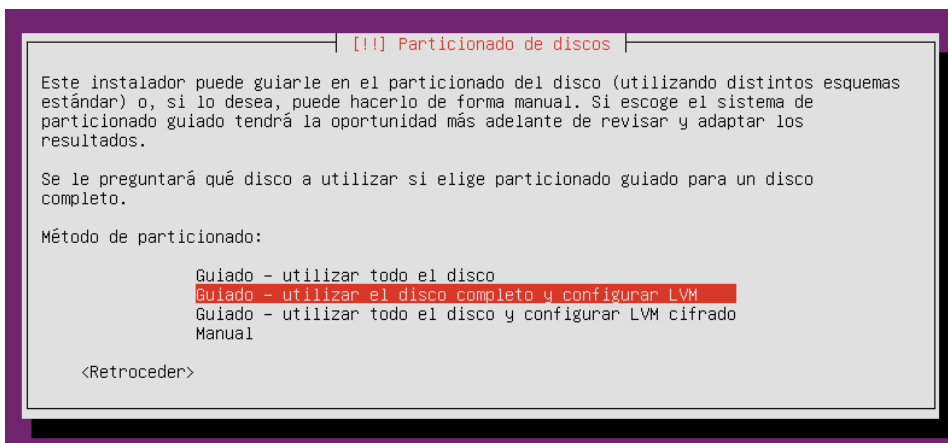


Figura 3. Particionado de discos.

Autorizamos la escritura en los discos. Ahora solo queda esperar que comience la instalación del sistema operativo (Fig.4) el tiempo puede variar según las características del sistema en el cual se está llevando a cabo la instalación.

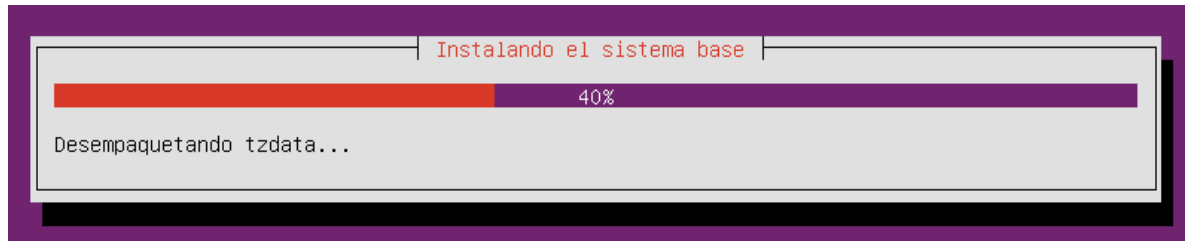


Figura 4. Carga de archivos.

La instalación se detiene para mostrarnos los paquetes que podemos instalar (Fig. 5), se realiza selección. En este caso no selecciono ninguno y posteriormente voy instalando las cosas conforme las necesito.

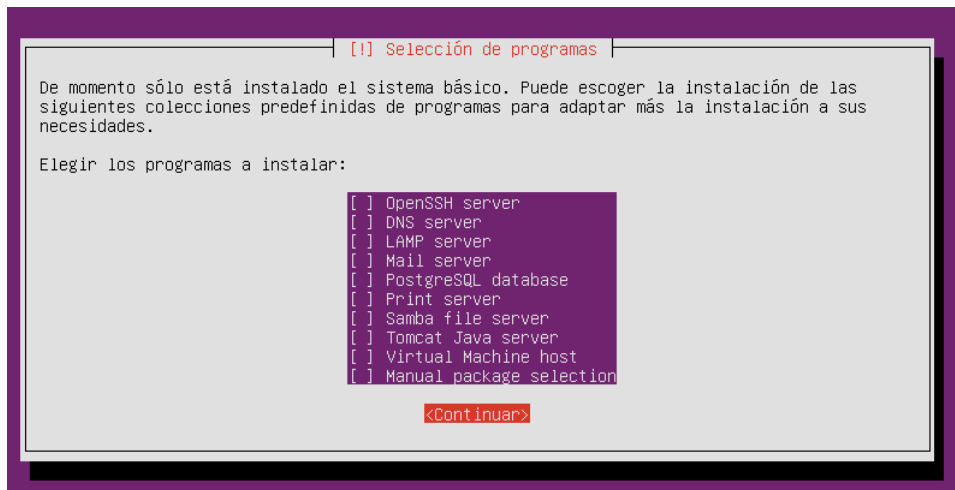


Figura 5. Preinstalación de software

Para finalizar nos pedirá el reinicio de la máquina una vez terminada la instalación, con lo cual ya tenemos Ubuntu Server funcionando, sin interfaz gráfica solamente con Kernel instalado.

ANEXO C. Instalación de un servidor FTP en Linux Ubuntu 16.04 Server

Instalaremos svFTPD, que es acrónimo de Very Secure FTP Daemon, es decir, un servicio FTP que también soporta FTPS

Partimos de una máquina con Linux Ubuntu 16.04 Server instalado. Vamos a la consola.

- Para instalar el servidor ftp, como con cualquier otro programa, en primer lugar, actualizamos la lista de paquetes del sistema a la última versión, con **sudo apt update** y si procede actualizarlos, lo haremos con **sudo apt upgrade**.
- A continuación, instalamos el servidor svFTPD con **sudo apt-get install vsftpd**. Podemos comprobar si hay ftp instalado haciendo **dpkg --get-selections |grep ftp**.
- Creamos un usuario para el ftp (por ejemplo *usuario*), haciendo **sudo adduser usuario**. Para evitar problemas, no utilizaremos ninguna mayúscula en el nombre de usuario. Nos pedirá una contraseña (por ejemplo *pass*) y algunos datos opcionales del usuario. Al aceptar, se crea una carpeta para el usuario, en */home/usuario*. Vamos a cambiar el dueño de esa carpeta para que sea el propio root, con **sudo chown root:root /home/usuario**. De esta manera reforzamos la seguridad ya que no podrá hacer cambios sobre “su” carpeta, en la que además luego le enjaularemos.
- Para que pueda subir archivos, dentro de su carpeta creamos otra, con **sudo mkdir /home/usuario/socket**, y le asignamos como propietario de esta segunda: **sudo chown usuario:usuario /home/usuario/socket**.
- Ahora mejoramos la seguridad quitando la Shell del usuario *usuario* y asignándole una personalizada que llamaremos *soloftp*. Preparamos la configuración de esta shell capada, creando un archivo con **sudo nano /bin/soloftp**. En su interior añadimos:

```
#!/bin/sh
echo "Cuenta solo con acceso por FTP."
```

Ahora damos permisos de ejecución para la shell *soloftp* con **sudo chmod a+x /bin/soloftp**. A continuación editamos el archivo de configuración de shells válidas con **sudo nano /etc/shells** y la añadimos al final de la lista:

```
/bin/soloftp
```

- Para obligarle a usar esta Shell, editamos el archivo de usuarios del sistema, haciendo **sudo nano /etc/passwd**, donde buscamos la línea correspondiente al usuario ftpuser. Al final de la misma se indica su Shell, que será la que hayapor defecto (*/bin/bash*). La cambiamos por la nueva:

```
ftpuser:x:1002:1002:,,,:/home/ftpuser:/bin/ftponly
```

- Con el usuario, su shell y sus carpetas preparados, editamos el archivo de configuración con **sudo nano /etc/vsftpd.conf**, y verificamos que las siguientes líneas estén activas, no comentadas:

```
# Allow anonymous FTP? (Disabled by default).
```

```

anonymous_enable=NO
#
# Uncomment this to allow local users to log in.
local_enable=YES

```

- Quitamos los comentarios de estas otras dos líneas, para permitir grabar información en el servidor a los clientes, y no solo descargarla:

```

# Uncomment this to enable any form of FTP write command.
write_enable=YES
# Default umask for local user is 077. You may wish to change this to 022,
# if your users expect that (022 is used by most other ftpd's)
local_umask=022

```

- Quitamos también el comentario de esta línea para hacer la carpeta de entrada al ftp el raíz de los usuarios. Nada que esté fuera de ella será accesible, pero también puede ser muy peligroso, si logran modificar la carpeta donde entran:

```

# You may restrict local users to their home directories. See the FAQ for
# the possible risks in this before using chroot_local_user or
# chroot_list_enable below.
chroot_local_user=YES

```

Además habilitamos a los usuarios locales del ftp para que puedan navegar por todo el árbol de directorios, aunque luego los restringiremos. Para restringir cuales son los usuarios que pueden navegar por todo el árbol de directorios, indicamos la ubicación del fichero donde está la lista de autorizados a hacerlo. Para ello descomentamos las tres líneas de instrucciones en el siguiente bloque:

```

# You may specify an explicit list of local users to chroot() to their home
# directory. If chroot_local_user is YES, then this list becomes a list of
# users to NOT chroot().
# (Warning! chroot'ing can be very dangerous. If using chroot, make sure
that
# the user does not have write access to the top level directory within the
# chroot)
chroot_local_user=YES
chroot_list_enable=YES
# (default follows)
chroot_list_file=/etc/vsftpd.chroot_list

```

- También añadiremos al final del fichero de configuración lo siguiente, para permitir aislar a los usuarios sin permiso de chroot, y configurar los puertos en modo pasivo:

```

allow_writeable_chroot=YES
pasv_min_port=30000
pasv_max_port=30100

```

Activamos también el uso de la variable `userlist_enable`, que permite negar o habilitar el acceso a los usuarios que incluyamos en un fichero definido en la ruta `userlist_file`. Además activamos la denegación de acceso por defecto, mediante la directriz `userlist_denny`. Para ello añadimos:

```
userlist_enable=YES
userlist_file=/etc/vsftpd.userlist
userlist_deny=NO
```

Para salir grabamos el archivo de configuración, con Ctrl+O y salimos con Ctrl+X.

- Ahora editamos el fichero **chroot_list** que contiene la lista de usuarios que no se enjaularán en su directorio raíz, mediante **sudo nano /etc/vsftpd.chroot_list**, y añadimos al menos el usuario root.

```
administrador
```

- Ahora editaremos el fichero **userlist**, con lista negra que permite o deniega el acceso a los usuarios locales especificados, según la variable `userlist_deny` esté en YES o NO. En nuestro caso denegará, haciendo **sudo nano /etc/vsftpd.userlist** añadimos el usuario del ftp

```
ftpuser
```

- Grabamos el archivo de configuración.
- Reiniciamos el servicio vsFTPD con la instrucción **sudo systemctl restart vsftpd**. Para arrancarlo o pararlo serían **sudo systemctl start vsftpd** y **sudo systemctl stop vsftpd**.
- Comprobamos que está funcionando con **service vsftpd status**. También podemos ver que por defecto se ha creado la carpeta `/srv/ftp`. Además podemos ver los usuarios con **cat /etc/passwd**, aparecerán `ftp` (para el servicio ftp) y `usuario` (que hemos creado).
- A partir de este momento, podremos utilizar el servidor introduciendo ftp://dirección_ip en el navegador o en el explorador. Si todo es correcto pedirá usuario y contraseña: `usuario / pass`, y podremos acceder. Veremos un archivo de ejemplo y la carpeta `socket` que hemos creado. No nos dejará salir hacia carpetas superiores. Con esta configuración podemos verificar que tenemos acceso para subir archivos, descargarlos y eliminarlos en el servidor ftp.
- Desde fuera del cortafuegos, para poder acceder con esta configuración, deberán estar abiertos los puertos 20, 21, 22, y del 30000 al 30100. En modo activo el puerto de entrada de comandos (enviados de nuestra máquina al servidor) es el 21, y el de salida de datos (que usará el servidor para enviarlos) el 20. En modo pasivo toda la iniciativa la toma nuestra máquina: el puerto de entrada de comandos es el 21 y el de datos el que le indique el servidor durante la negociación de la conexión, para que por él acceda a la información. En el cortafuegos solo debería permitir el paso a través de esos puertos, si solo tenemos el servidor ftp en nuestra red y queremos darle acceso desde fuera.

En ciertas aplicaciones, como un socket que reciba datos de estaciones remotas mediante ftp, la configuración realizada hasta ahora puede ser suficiente. Sin embargo, para otras funciones como el intercambio de archivos con datos personales puede ser necesaria una mayor seguridad.

Vamos ahora a ver como podemos configurar el servidor FTP para que trabaje con conexiones cifradas SSL. Previamente necesitaremos tener instalado OpenSSH en nuestro sistema operativo.

- Una vez tenemos OpenSSH, debemos crear certificados para el uso con vsftpd. Lo hacemos con **sudo openssl req -x509 -nodes -days 365 -newkey rsa:1024 -keyout /etc/ssl/private/vsftpd.pem -out /etc/ssl/private/vsftpd.pem**. A continuación, se nos preguntarán algunos datos para incluir en el certificado a crear.
- Con esto, habremos creado un certificado de 1024 bits en la carpeta */etc/ssl/private*, y será válido durante 365 días (o el número que demos). En esa misma ruta vemos que hay otro certificado, *ssl-cert-snakeoil.key*, que se crea al instalar OpenSSH para permitir la instalación segura del gestor de certificados, y no debe eliminarse.
- Debemos editar de nuevo el fichero *vsftpd.conf*, descomentando la línea que permite el uso de certificados ssl, o añadiéndola si no está:
ssl_enable=YES
- A continuación, añadimos los parámetros de gestión de certificados en la configuración, con las líneas que indican la carpeta donde se almacenan:
rsa_cert_file=/etc/ssl/private/vsftpd.pem
rsa_private_key_file=/etc/ssl/private/vsftpd.pem
- Además añadimos líneas para no permitir la conexión anónima con certificados, y forzar a todos los usuarios a usar certificados para conectarse:
allow_anon_ssl=NO
force_local_data_ssl=YES
force_local_logins_ssl=YES
- También añadimos líneas para definir la versión de conexiones SSL a utilizar:
ssl_tlsv1=YES
ssl_sslv2=NO
ssl_sslv3=NO
require_ssl_reuse=NO
ssl_ciphers=HIGH
- Grabamos el archivo y reiniciamos el servidor para hacer efectivos los cambios.
- Para conectarnos a nuestro ftp seguro no bastará el explorador o un navegador, salvo que tengamos las extensiones apropiadas. Debemos usar un cliente como FileZilla, capaz de manejar los certificados

ANEXO D. Instalación de VirtualBox

Debemos identificarnos como Administrador (root) y luego actualizar el sistema.

```
$ sudo apt-get update  
$ sudo apt-get upgrade
```

Importar la clave pública de Oracle .al sistema

```
$ wget -q https://www.virtualbox.org/download/oracle_vbox_2016.asc -O- | sudo apt-key add -  
$ wget -q https://www.virtualbox.org/download/oracle_vbox.asc -O- | sudo apt-key add -
```

Agregar al sistema Oracle Ubuntu PPA.

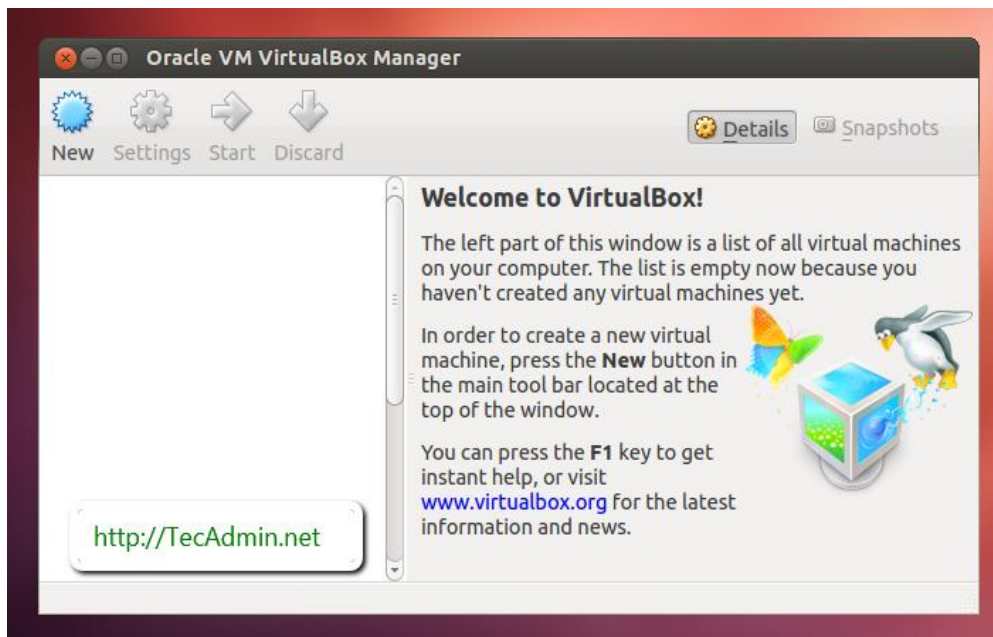
```
$ sudo add-apt-repository "deb http://download.virtualbox.org/virtualbox/debian xenial contrib"
```

Instalación del VirtualBox.

```
$ sudo apt-get update  
$ sudo apt-get install virtualbox-5.2
```

Para iniciar el VirtualBox

```
$virtualbox
```



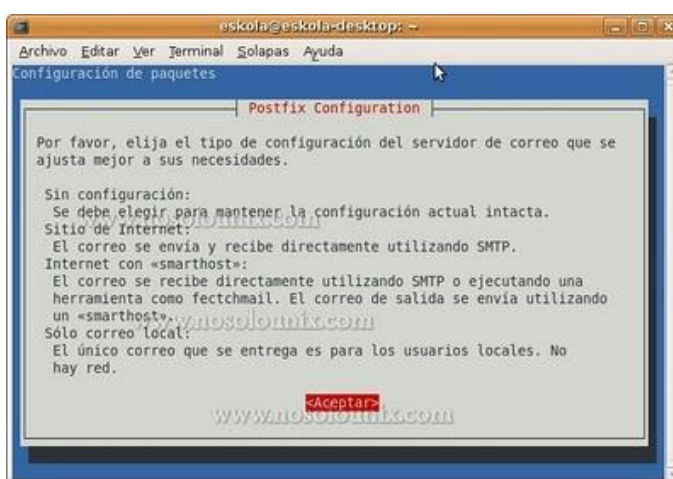
ANEXO E. Servidor de correo en Ubuntu

POSTFIX

Postfix es un agente de transporte de correo de manera que nos permite enrutar y transferir correo electrónico. Por ello, si queremos tener un servidor de correo en Ubuntu, deberemos instalar Postfix. Para instalar Postfix, solo tendremos que escribir lo siguiente en un terminal

```
sudo apt-get install postfix
```

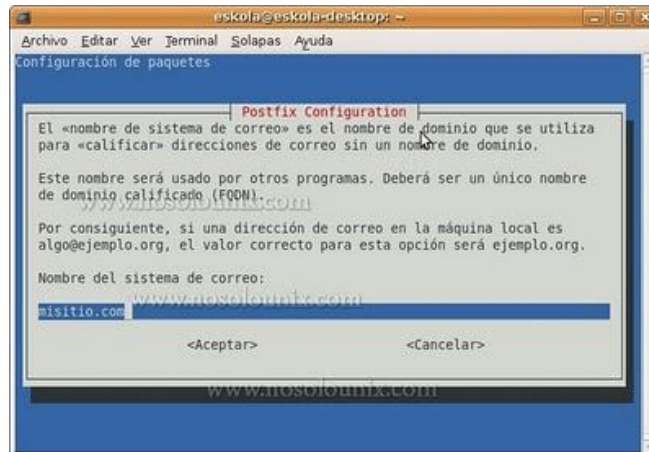
Durante la instalación nos aparecerán diferentes ventanas.



Seleccionaremos “Sitio de Internet”.

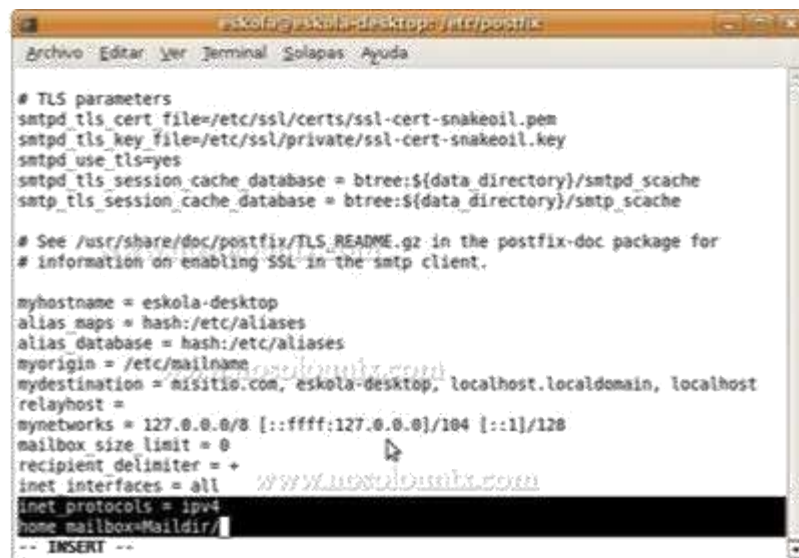


En la siguiente escribiremos el **dominio que hayamos escogido**. Para poder tener nuestro servidor de correo en Ubuntu, en nuestro caso lo que hemos puesto de la siguiente manera:



A continuación editaremos el siguiente archivo: `/etc/postfix/main.cf`

Al final del archivo le añadiremos las siguientes líneas de código que se ven en negro:



```
inet_protocols = ipv4
home_mailbox = Maildir/
```

Mediante estas líneas le indicaremos que **protocolo** vamos a usar y **donde se depositarán los emails**.

```
mydestination = misitio.com
```

Para que los cambios surtan efecto, escribiremos en terminal lo siguiente:

```
sudo /etc/init.d/postfix restart
```

COURIER POP Y COURIER IMAP

Para conseguir nuestro servidor de correo en Ubuntu, instalaremos **courier-pop** y **courier-imap** mediante los siguientes comandos en terminal:

```
sudo apt-get install courier-pop
```

En la instalación de courier-pop le indicaremos que **NO**.



Instalamos courier-imap:

```
sudo apt-get install courier-imap
```

MAILX

Instalaremos mailx, ya que más adelante lo utilizaremos en uno de los pasos para tener nuestro servidor de correo en Ubuntu. Esta aplicación nos permitirá **enviar emails mediante la línea de comandos**.

```
sudo apt-get install mailx
```

SQUIRRELMAIL

Squirrelmail es una **aplicación webmail en PHP**. El objetivo es instalar Squirrelmail para que gracias al servidor web que hemos instalado (Apache2), podamos ver los emails que nos

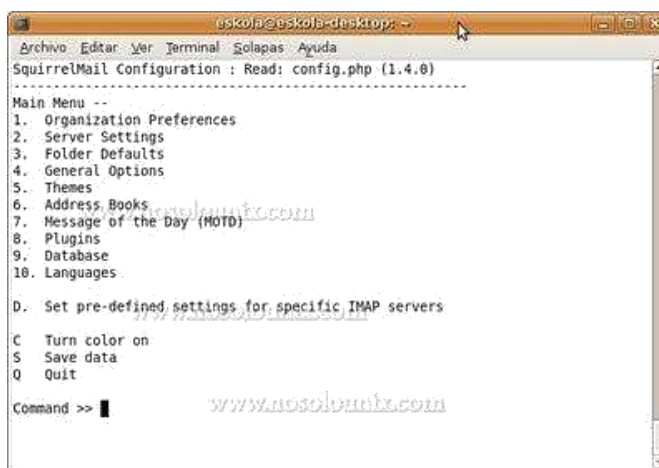
envien. De esta manera, podremos tener un servidor de correo en Ubuntu. Para instalar Squirrelmail, en un terminal introduciremos lo siguiente:

```
sudo apt-get install squirrelmail
```

Una vez instalado, escribiremos en la terminal lo siguiente para configurarlo:

```
squirrelmail-configure
```

Veremos el siguiente menú:



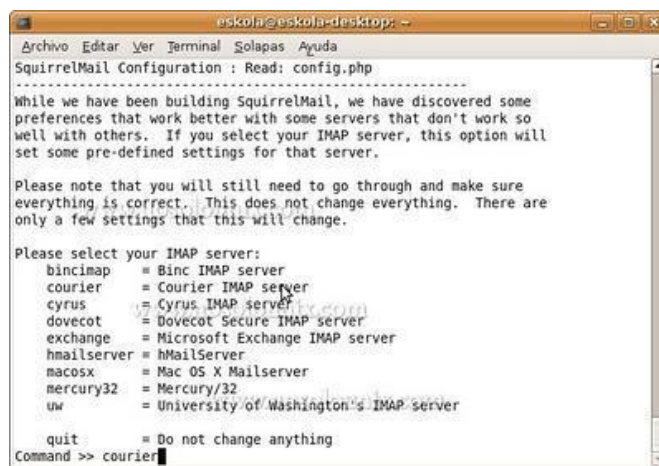
```
eskola@eskola-desktop: ~
Archivo Editar Ver Terminal Solapas Ayuda
SquirrelMail Configuration : Read: config.php (1.4.0)
-----
Main Menu --
1. Organization Preferences
2. Server Settings
3. Folder Defaults
4. General Options
5. Themes
6. Address Books
7. Message of the Day (MOTD)
8. Plugins
9. Database
10. Languages

D. Set pre-defined settings for specific IMAP servers

C Turn color on
S Save data
Q Quit

Command >> |
```

Elegiremos la opción D (Set pre-defined settings for specific imap servers). Acto seguido elegiremos el tipo de correo: courier:



```
eskola@eskola-desktop: ~
Archivo Editar Ver Terminal Solapas Ayuda
SquirrelMail Configuration : Read: config.php
-----
While we have been building SquirrelMail, we have discovered some
preferences that work better with some servers that don't work so
well with others. If you select your IMAP server, this option will
set some pre-defined settings for that server.

Please note that you will still need to go through and make sure
everything is correct. This does not change everything. There are
only a few settings that this will change.

Please select your IMAP server:
bincimap = Binc IMAP server
courier = Courier IMAP server
cyrus = Cyrus IMAP server
dovecot = Dovecot Secure IMAP server
exchange = Microsoft Exchange IMAP server
hmailserver = hMailServer
macosx = Mac OS X Mailserver
mercury32 = Mercury/32
uw = University of Washington's IMAP server

quit = Do not change anything
Command >> courier|
```

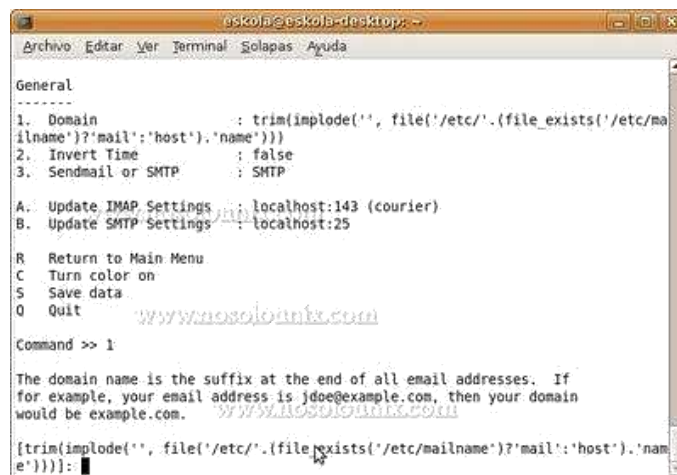
Obtendremos el siguiente resultado:

Lo siguiente será escribir 2 y pulsar la tecla ENTER. Veremos lo siguiente:



```
eskola@eskola-desktop: ~  
Archivo Editar Ver Terminal Solapas Ayuda  
SquirrelMail Configuration : Read: config.php (1.4.0)  
-----  
Server Settings  
  
General  
-----  
1. Domain : trim(implode('', file('/etc/'.(file_exists('/etc/mailname')?'mail':'host').'name')))  
2. Invert Time : false  
3. Sendmail or SMTP : SMTP  
  
A. Update IMAP Settings : localhost:143 (courier)  
B. Update SMTP Settings : localhost:25  
  
R Return to Main Menu  
C Turn color on  
S Save data  
Q Quit  
  
Command >> |
```

Lo siguiente será escribir 1 y pulsar ENTER de nuevo. Se verá algo como esto:



```
eskola@eskola-desktop: ~  
Archivo Editar Ver Terminal Solapas Ayuda  
-----  
General  
-----  
1. Domain : trim(implode('', file('/etc/'.(file_exists('/etc/mailname')?'mail':'host').'name')))  
2. Invert Time : false  
3. Sendmail or SMTP : SMTP  
  
A. Update IMAP Settings : localhost:143 (courier)  
B. Update SMTP Settings : localhost:25  
  
R Return to Main Menu  
C Turn color on  
S Save data  
Q Quit  
  
Command >> 1  
  
The domain name is the suffix at the end of all email addresses. If  
for example, your email address is jdoe@example.com, then your domain  
would be example.com.  
  
[trim(implode('', file('/etc/'.(file_exists('/etc/mailname')?'mail':'host').'name'))):
```

A continuación escribiremos el **dominio de nuestro sitio web** (en nuestro caso misitio.com) que más adelante utilizaremos para acceder a nuestro servidor de correo en Ubuntu. Lo dejaremos como se ve a continuación:

```
eskola@eskola-desktop: ~
Archivo Editar Ver Terminal Solapas Ayuda
-----
General
-----
1. Domain          : trim(implode('', file('/etc/'.(file_exists('/etc/mailname')?'mail':'host').'name')))
2. Invert Time     : false
3. Sendmail or SMTP : SMTP

A. Update IMAP Settings : localhost:143 (courier)
B. Update SMTP Settings : localhost:25

R Return to Main Menu
C Turn color on
S Save data
Q Quit

Command >> 1

The domain name is the suffix at the end of all email addresses. If
for example, your email address is jdoe@example.com, then your domain
would be example.com.

[trim(implode('', file('/etc/'.(file_exists('/etc/mailname')?'mail':'host').'name'))): misitio.com
```

Al pulsar la tecla ENTER el resultado final será el siguiente:

```
eskola@eskola-desktop: ~
Archivo Editar Ver Terminal Solapas Ayuda
SquirrelMail Configuration : Read: config.php (1.4.0)
-----
Server Settings
-----
General
-----
1. Domain          : misitio.com
2. Invert Time     : false
3. Sendmail or SMTP : SMTP

A. Update IMAP Settings : localhost:143 (courier)
B. Update SMTP Settings : localhost:25

R Return to Main Menu
C Turn color on
S Save data
Q Quit

Command >>
```

Por último, escribiremos Q para salir y guardaremos los datos:

```
eskola@eskola-desktop: ~
Archivo Editar Ver Terminal Solapas Ayuda
SquirrelMail Configuration : Read: config.php (1.4.0)
-----
Server Settings
-----
General
-----
1. Domain          : misitio.com
2. Invert Time     : false
3. Sendmail or SMTP : SMTP

A. Update IMAP Settings : localhost:143 (courier)
B. Update SMTP Settings : localhost:25

R Return to Main Menu
C Turn color on
S Save data
Q Quit

Command >> Q

You have not saved your data.
Save? [Y/n]: Y
```

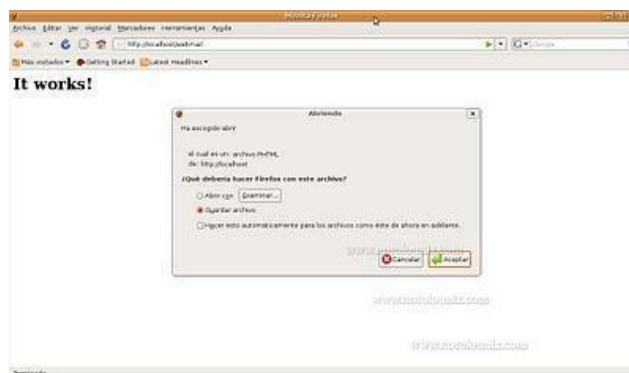
Lo siguiente que haremos será configurar el webmail de Squirrelmail para que podamos **acceder a él, vía web** mediante apache. Para ello, escribiremos lo siguiente en un terminal:

```
cd /var/www
```

```
sudo ln -s /usr/share/squirrelmail webmail
```

Una vez escrito, comprobaremos que todo se ha realizado correctamente. Para ello, entraremos en Firefox (o tu navegador favorito) y nos dirigiremos a la siguiente dirección: ***http://www.misitio.com/webmail*** (en vez de misitio.com introduciremos el dominio que hayamos configurado en pasos anteriores), donde podremos ver nuestro servidor de correo en Ubuntu vía web.

Es posible que veáis algo como lo que se ve en la siguiente imagen:



En caso de ver esto, borraremos las **cookies y toda la información privada** de nuestro navegador. Después, reiniciaremos Apache mediante el siguiente comando en la terminal:

```
sudo /etc/init.d/apache2 restart
```

Esperaremos a que Apache2 reinicie y volvemos a entrar a ***http://www.misitio.com/webmail*** y veremos lo siguiente:



Servidor de correo en Ubuntu:

Crear usuarios y probar que el servidor funciona

El siguiente paso para poder probar el servidor de correo en Ubuntu será **crear 2 usuarios**. Para ello, escribiremos en la terminal lo siguiente:

```
sudo adduser user1
```

Contestaremos a las diferentes preguntas que nos pide (lo más importante es el **password**, recordarlo ya que lo necesitaremos más adelante) y haremos lo mismo con otro usuario:

```
sudo adduser user2
```

Para probar que funcione el correo en Ubuntu, enviaremos **un email entre los usuarios**, nos logearemos como user1 mediante el siguiente comando en terminal:

```
su user1
```

Nos pedirá la contraseña de user1, lo siguiente será **enviar un email al user2**. Escribiremos lo siguiente en la terminal:

```
mail user2
```

En **Subject** escribiremos el asunto. Pulsaremos ENTER y ahí escribiremos el **texto del mensaje**. Cuando hayamos acabado de escribir el mensaje, introduciremos un salto de línea (pulsando la tecla ENTER), escribiremos un punto (es decir ".") y volveremos a pulsar ENTER. Acto seguido nos indicará si queremos enviar el mensaje a más destinatarios (Cc). Como no queremos enviar el mensaje a nadie más volveremos a pulsar ENTER. Puedes ver el resultado en la siguiente imagen:

```
user1@lalo-Aspire-one:~/home/lalo
lalo@lalo-Aspire-one:~$ su user1
Contraseña:
user1@lalo-Aspire-one:/home/lalo$ mail user2
Cc: user2
Subject: practica
esta es una practica DEL SERVIDOR DE CORREO para la materia de SISTEMAS OPERATIVO
S DE RED
```

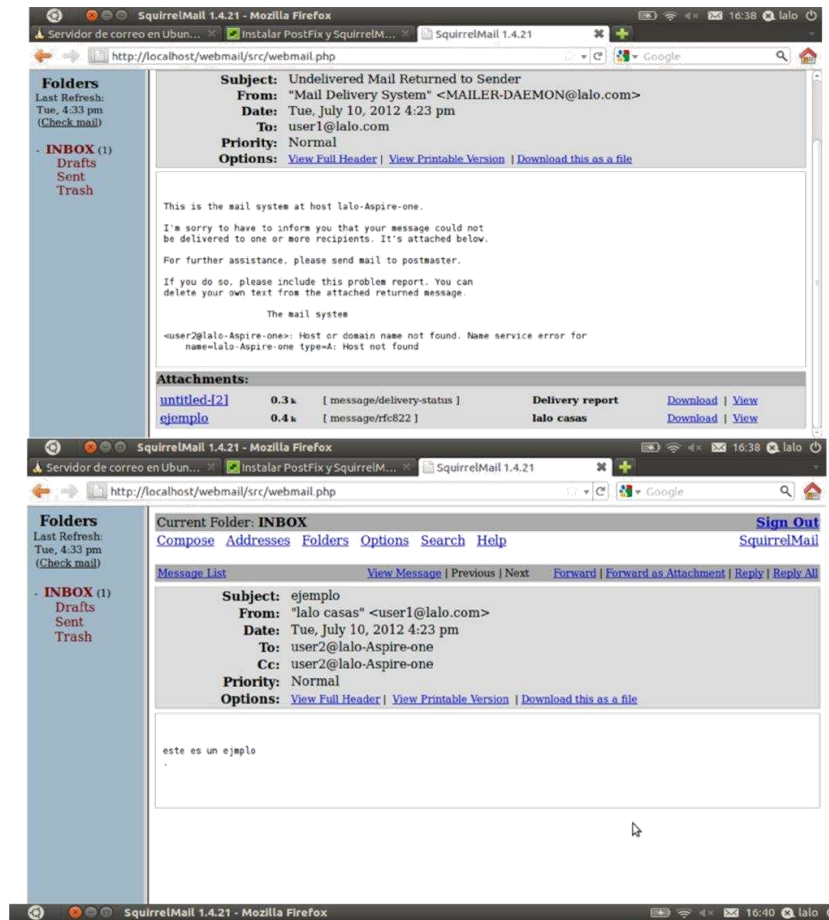
Lo siguiente será comprobar que ese email ha llegado correctamente. Para comprobarlo, abriremos nuestro navegador web, nos dirigiremos a *localhost/webmail* y nos logearemos:



Veremos la siguiente bandeja de entrada:



Si pulsamos sobre el email, veremos lo siguiente:



Ya hemos comprobado que nuestro **servidor de correo en Ubuntu** funciona a la perfección, por lo que hasta aquí llega este manual. Espero que este manual de cómo instalar y configurar un servidor de correo en Ubuntu os haya resultado de utilidad. Gracias a Marta por su colaboración en la elaboración de este manual.

ANEXO F. Tipos de Software Educativo.

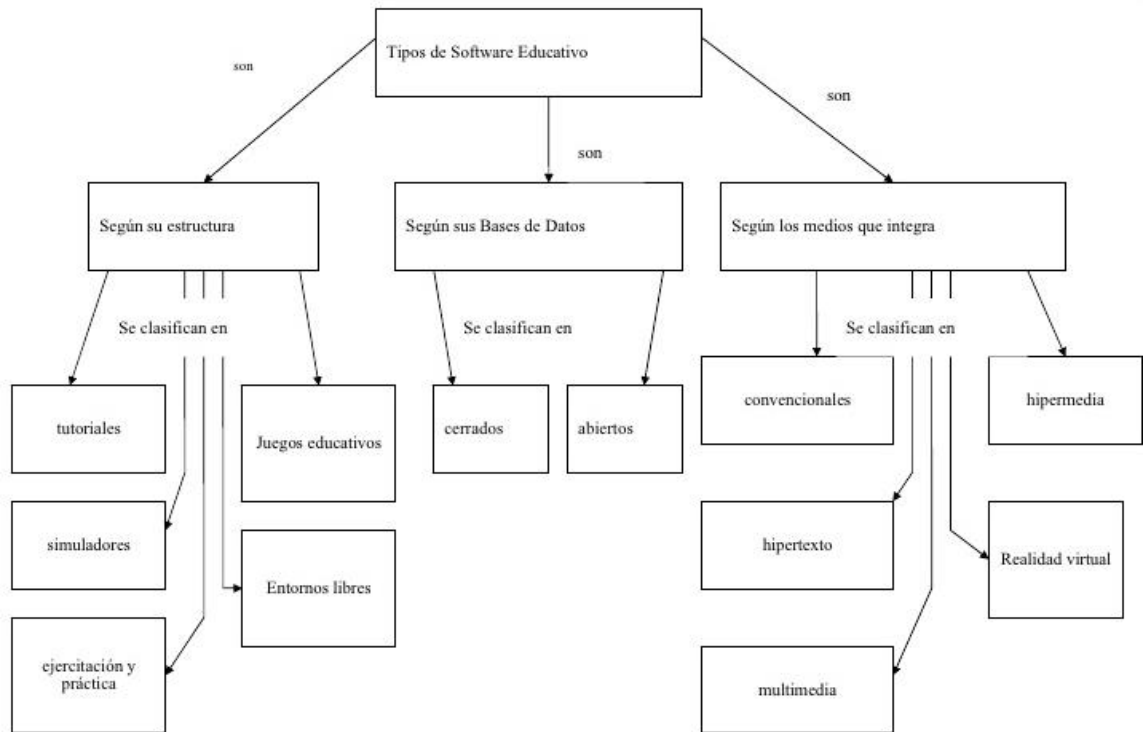


Figura 24. Esquema de tipos de software Educativo

ANEXO G. Metodología xp

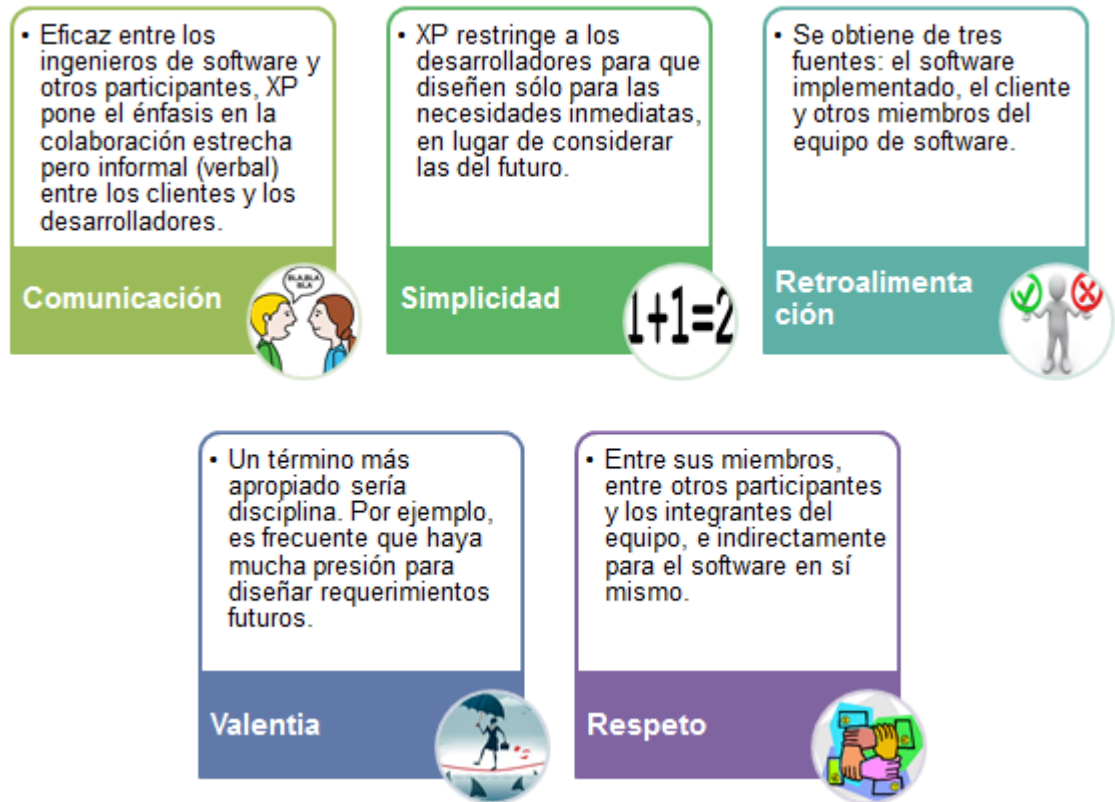


Figura 25 . Pasos de la Metodología xp