

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

**DESARROLLO DE UN SISTEMA WEB PARA LA RECEPCIÓN DE
REQUERIMIENTOS Y CONTROL DE TIEMPO EMPLEADO POR LOS
DESARROLLADORES EN LA EJECUCIÓN DE PROYECTOS EN LA
EMPRESA GAMEDASOFT**

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO EN
ANÁLISIS DE SISTEMAS INFORMÁTICOS**

TUSO GUAYTA ALVARO DAVID

davidanalisis@hotmail.es

DIRECTOR: ING. EDWIN SALVADOR

Quito, Septiembre 2016

DECLARACIÓN

Yo, Tuso Guayta Alvaro David, declaro que el trabajo aquí descrito es de mi total autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias que se incluyen en este documento.

La Escuela Politécnica Nacional puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido por la Ley de Propiedad Intelectual, por su reglamento y por la normativa institucional vigente.

CERTIFICACIÓN

Certificamos que el presente trabajo fue desarrollado por Tuso Guayta Alvaro David, bajo nuestra supervisión.

Ing. Edwin Salvador Pesantes

DIRECTOR DE PROYECTO

Ing. Cesar Gallardo Carrera

CO-DIRECTOR DEL PROYECTO

AGRADECIMIENTO

Primero agradezco a Dios por la oportunidad de estar en este mundo y por darme la sabiduría y perseverancia para alcanzar la meta propuesta en mi vida.

Agradezco a mis padres por el apoyo que me brindaron durante mis estudios, gracias por su paciencia en los momentos que lo necesité, ya que ellos son un pilar fundamental en mi vida.

A mis hermanos, que con su apoyo incondicional he logrado alcanzar mis metas propuestas.

Al Ing. Edwin Salvador, por su paciencia, apoyo y enseñanza para alcanzar esta meta que se refleja en este proyecto.

Agradezco a la Empresa Gamedasoft y a todo su personal, quienes me brindaron su confianza para implementar el sistema.

Finalmente, a la Escuela Politécnica Nacional por brindarme la oportunidad de culminar mis estudios en sus aulas.

DEDICATORIA

Esta meta cumplida es para mi familia ya que en el camino de mi formación me apoyaron en los momentos buenos y malos en los tropiezos y logros siempre estuvieron ahí, me han demostrado que con perseverancia siempre se alcanza el éxito.

Alvaro

ÍNDICE DE CONTENIDOS

LISTA DE FIGURAS	VIII
LISTA DE TABLAS	X
RESUMEN	XII
CAPITULO I	1
1. ASPECTOS GENERALES	1
1.1. INTRODUCCIÓN	1
1.2. PLANTEAMIENTO DEL PROBLEMA.....	2
1.3. OBJETIVOS.....	3
1.3.1. OBJETIVO GENERAL.....	3
1.3.2. OBJETIVOS ESPECÍFICOS.....	3
1.4. ALCANCE	3
1.5. PRESUPUESTO	4
CAPITULO II.....	5
2. MARCO TEÓRICO	5
2.1. INGENIERÍA DE SOFTWARE	5
2.2. ETAPAS DEL DESARROLLO DE LOS SISTEMAS INFORMÁTICOS ..	6
2.3. METODOLOGÍA DE DISEÑO HIPERMEDIA ORIENTADO A OBJETOS (OOHDM).....	8
2.3.1. DISEÑO CONCEPTUAL	9
2.3.2. DISEÑO NAVEGACIONAL.....	10
2.3.3. DISEÑO DE INTERFAZ ABSTRACTO	10
2.3.4. IMPLEMENTACIÓN.....	10
2.4. LENGUAJE DE MODELADO UML	10
2.4.1. BENEFICIOS QUE PROPORCIONA EL UML	11
2.5. MODELADO DE BASE DE DATOS.....	12
2.5.1. DISEÑO CONCEPTUAL	12
2.5.2. DISEÑO LÓGICO	12
2.5.3. DISEÑO FÍSICO	12
2.6. PARADIGMA DE PROGRAMACIÓN ORIENTADA A OBJETOS.....	12
2.6.1. CONCEPTOS IMPORTANTES DE LA PROGRAMACIÓN ORIENTADA A OBJETOS.....	13
2.6.2. CARACTERÍSTICAS DE LA PROGRAMACIÓN ORIENTADA A OBJETOS	13
2.7. PLATAFORMA WEB.....	14
2.7.1. OPEN SOURCE	14

2.7.2.	XAMPP	14
2.7.3.	SYMFONY	15
2.8.	DISEÑO WEB	18
2.8.1.	HTML 5	18
2.8.2.	CSS	18
2.8.3.	JAVASCRIPT	18
2.8.4.	JQUERY	18
2.8.5.	BOOTSTRAP	18
2.8.6.	AJAX	19
CAPITULO III		20
3.	DISEÑO Y DESARROLLO DE LA APLICACIÓN WEB	20
3.1.	ENCUESTAS Y REQUERIMIENTOS PARA EL DEL DESARROLLO DEL SISTEMA	20
3.1.1.	OBTENCIÓN DE REQUERIMIENTOS	20
3.1.2.	LISTA DE REQUERIMIENTOS	20
3.2.	ANÁLISIS DE REQUERIMIENTOS	21
3.2.1.	REQUERIMIENTOS FUNCIONALES SEGÚN OOHDM	21
3.3.	DISEÑO DE BASE DE DATOS	49
3.4.	DISEÑO DE NAVEGACIONAL	51
3.5.	DISEÑO DE LA APLICACIÓN WEB	51
3.5.1.	DISEÑO DEL INICIO SESIÓN DEL SISTEMA	52
3.5.2.	DISEÑO PRINCIPAL DEL SISTEMA	52
3.6.	DESARROLLO DEL SISTEMA	53
3.6.1.	COMPONENTES QUE FORMAN EL PROYECTO	54
3.6.2.	ESTRUCTURA DE DIRECTORIOS DEL PROYECTO	54
3.6.3.	CONFIGURACIÓN DEL SISTEMA	55
3.6.4.	CREACIÓN DE BUNDLE	55
3.6.5.	CREACIÓN DE ENTIDADES	56
3.6.6.	CREACIÓN DE CONTROLADORES	57
3.6.7.	CREACIÓN DE LA VISTA	59
3.7.	IMPLEMENTACIÓN	63
3.7.1.	REQUISITOS PREVIOS PARA LA IMPLEMENTACIÓN	63
3.7.2.	INTERFACES DEL SISTEMA	65
3.8.	PRUEBAS DE FUNCIONAMIENTO	68
3.8.1.	PRUEBAS DE ACEPTACIÓN	69

CAPITULO IV	79
4. CONCLUSIONES Y RECOMENDACIONES	79
4.1. CONCLUSIONES	79
4.2. RECOMENDACIONES	80
BIBLIOGRAFÍA.....	81

LISTA DE FIGURAS

FIGURA 1- Capas de la ingeniería de software	6
FIGURA 2- Etapas del desarrollo de los sistemas informáticos	7
FIGURA 3- Relación entre el esquema conceptual, navegacional y los objetos de interfaz.....	9
FIGURA 4- Etapas de la metodología OOADM.....	10
FIGURA 5- Diagramas UML.....	11
FIGURA 6- Popularidad de Symfony en proyectos personales.....	15
FIGURA 7- MVC en Symfony	16
FIGURA 8- Funcionamiento de AJAX.....	19
FIGURA 9- Diagrama del caso de uso del rol administrador.....	34
FIGURA 10- Diagrama de caso de uso del rol empleado.....	43
FIGURA 11- Diagrama del caso de uso del rol cliente.....	46
FIGURA 12- Diseño de la base de datos.....	50
FIGURA 13- Diseño navegacional del sistema.....	51
FIGURA 14- Estructura inicio sesión	52
FIGURA 15- Diseño plantilla base	53
FIGURA 16- Relación que existe con elementos MVC.....	53
FIGURA 17- Estructura de directorios del proyecto.....	54
FIGURA 18- Estructura de directorios del Bundle.....	56
FIGURA 19- Entidades de sistema.....	57
FIGURA 20- Controladores del sistema.....	57
FIGURA 21- Estructura de directorios Twig.....	59
FIGURA 22- Directorio plantilla base Twig.....	60
FIGURA 23- Código fuente plantilla base Twig.....	60
FIGURA 24- Integración de Layout con plantilla base.....	61
FIGURA 25- Archivos y directorios de la vista Twig.....	61
FIGURA 26- Directorio públicos de nuestro sistema.....	62
FIGURA 27- Archivos Bootstrap.....	62
FIGURA 28- Instalación de XAMPP.....	64
FIGURA 29- Prueba de funcionamiento servidor web Apache.....	64
FIGURA 30- Página de ingreso al sistema.....	65
FIGURA 31- Página de principal Administrador.....	65

FIGURA 32- Página para actualizar datos del administrador.	66
FIGURA 33- Página de asignación de requerimientos.	66
FIGURA 34- Página de principal Empleados.	67
FIGURA 35- Página ingreso tiempos invertidos en la solución de requerimientos.....	67
FIGURA 36- Página de principal cliente.	68
FIGURA 37- Página de ingreso de requerimientos.....	68
FIGURA 38- Esquema de pruebas del sistema.	69
FIGURA 39- Flujo principal del sistema.	69
FIGURA 40- Iniciar sesión cliente.....	71
FIGURA 41- Prueba inicio sesión cliente.	71
FIGURA 42- Ingreso de requerimiento.....	72
FIGURA 43- Registro de requerimiento.	72
FIGURA 44- Inicio sesión Administrador.	73
FIGURA 45- Prueba de inicio de sesión Administrador.....	74
FIGURA 46-Selección de requerimiento a asignar.	75
FIGURA 47- Seleccionamos el empleado a asignar.	75
FIGURA 48- Requerimiento asignado.....	75
FIGURA 49- Inicio sesión empleado.....	76
FIGURA 50- Prueba de inicio de sesión empleado.....	77
FIGURA 51- Seleccionar cliente ingreso tiempos.	78
FIGURA 52- Seleccionar requerimiento.....	78
FIGURA 53- Registro de tiempo del requerimiento.	78

LISTA DE TABLAS

Tabla 1- Identificación de actores y su descripción.	21
Tabla 2- Identificación de tareas por actor	22
Tabla 3- Escenario inicio de sesión administrador.	23
Tabla 4- Escenario actualizar datos administrador.	23
Tabla 5- Escenario cambiar contraseña administrador.	23
Tabla 6- Escenario administrar empleados.	24
Tabla 7- Escenario administrar clientes.	24
Tabla 8- Escenario asignación requerimientos.	24
Tabla 9- Escenario administrar módulos.	25
Tabla 10- Escenario administrar menús.	25
Tabla 11- Escenario administrar Submenús.	25
Tabla 12- Escenario consulta estado del requerimiento por empleado.	26
Tabla 13- Escenario consultar el tiempo invertido en la solución de cada requerimiento por cliente.	26
Tabla 14- Cerrar sesión administrador.	27
Tabla 15- Escenario inicio de sesión empleado.	27
Tabla 16- Escenario actualizar datos empleado.	27
Tabla 17- Escenario cambiar contraseña administrador.	28
Tabla 18- Escenario ingreso tiempos invertidos en la solución de requerimientos.	28
Tabla 19- Escenario consulta requerimiento asignado.	28
Tabla 20- Escenario cerrar sesión empleado.	29
Tabla 21- Escenario iniciar sesión cliente.	29
Tabla 22- Escenario actualizar datos cliente.	29
Tabla 23- Escenario cambiar contraseña cliente.	30
Tabla 24- Escenario ingresar requerimiento.	30
Tabla 25- Escenario consultar estado requerimiento.	30
Tabla 26- Escenario cerrar sesión del cliente.	31
Tabla 27- CU-01 Actualizar datos.	32
Tabla 28- CU-02 Cambiar contraseña.	33
Tabla 29- CU-03 Cerrar sesión.	34
Tabla 30- CU-04 Iniciar sesión administrador.	35
Tabla 31- CU-05 Administrar empleados.	36

Tabla 32- CU-06 Administrar clientes.	36
Tabla 33- CU-07 Asignar requerimientos.	37
Tabla 34- CU-08 Administrar módulos.	38
Tabla 35- CU-09 Administrar menús.	39
Tabla 36- CU-10 Administrar Submenús.	40
Tabla 37- CU-11 Consulta estado de requerimiento por cliente.	41
Tabla 38- CU-12 Consulta de tiempos invertidos en la solución de requerimiento por cliente.	42
Tabla 39- CU-13 Iniciar sesión empleado.	44
Tabla 40- CU-14 Ingreso de tiempo invertido en la solución del requerimiento por cliente.	45
Tabla 41- CU-15 Consulta requerimientos asignados.	46
Tabla 42- CU-16 Iniciar sesión cliente.	47
Tabla 43- CU-17 Ingreso requerimientos.	48
Tabla 44- CU-18 Consulta estado de requerimientos.	49
Tabla 45- Requerimientos mínimos hardware para la implementación del sistema.	63
Tabla 46- Requerimientos mínimos de software para la implementación del sistema. .	63
Tabla 47- P.A.-01-Prueba de inicio de sesión cliente.	70
Tabla 48- P.A.-02-Prueba de Ingreso requerimiento.	72
Tabla 49- P.A.-04- Prueba de inicio de sesión Administrador.	73
Tabla 50- P.A.-04- Prueba de aceptación asignación de requerimiento.	74
Tabla 51- P.A.-05- Prueba de inicio de sesión empleado.	76
Tabla 52- P.A.-04- Prueba de inicio de sesión Administrador.	78

RESUMEN

El presente proyecto está enfocado al desarrollo de una aplicación para realizar la recepción de requerimientos y control de tiempo empleado por los desarrolladores en la ejecución de proyectos en la empresa Gamedasoft vía web.

Se ha considerado utilizar la metodología de diseño de hipermedia orientado a objetos (OOHDM) para el desarrollo de esta aplicación, ya que esta metodología es ágil la cual permite desarrollar aplicaciones web.

Con esta aplicación web la empresa Gamedasoft mejorará el control de requerimientos de los clientes. Cabe recalcar que se han cumplido con los objetivos planteados para este sistema web.

El presente documento se ha dividido en 4 capítulos, que se detallan a continuación.

El capítulo 1, realiza una breve introducción a los requerimientos obtenidos de la empresa Gamedasoft para el desarrollo del software. También se realiza el planteamiento del problema y se detalla porqué y para qué se realizó el sistema, se indica el objetivo general y los específicos, el alcance del sistema y el presupuesto a utilizar al desarrollar el sistema.

En el capítulo 2, se presenta el marco teórico sobre ingeniería software, etapas del desarrollo de los sistemas informáticos, arquitectura y el patrón de desarrollo del sistema web, y una descripción de las etapas comprendidas en la metodología OOHDM, etc.

El capítulo 3, describe la parte principal del proyecto como es el desarrollo del sistema web y el proceso realizado en cada etapa del desarrollo en base a la metodología OOHDM.

En el capítulo 4, se presentan las conclusiones obtenidas durante la elaboración del proyecto, y se indican las recomendaciones que ayudarán a utilizar el sistema web y guiará el desarrollo de futuras mejoras.

CAPITULO I

1. ASPECTOS GENERALES

1.1. INTRODUCCIÓN

En la actualidad el uso y desarrollo de aplicaciones web ha crecido notablemente desde que tuvo sus inicios, que se dio en el año 1987 donde se crea el primer lenguaje de programación para aplicaciones web denominado “Perl” la cual fue inventado por Larry Wall cuando el internet no era accesible para el público en general teniendo básicamente paginas estáticas. En el año 1995 el programador Rasmus Lerdorf pone en marcha el lenguaje procesador pre hipertexto (PHP), desde entonces el desarrollo de aplicaciones web tomo auge, teniendo en la actualidad aplicaciones web dinámicas ya que posee varias ventajas tanto para el desarrollador como para el usuario final, entre ellos se puede destacar la ejecución de varias tareas simultáneamente, por ejemplo, solo basta tener acceso a internet para tener información en un tiempo extremadamente mínimo, utilizando diferentes dispositivos móviles como son: Smartphone, Tablets ,etc., que ayudan a los sistemas modernos a conectarse desde cualquier lugar.

En la empresa Gamedasoft cada día los requerimientos son más frecuentes y son solicitados vía telefónica, lo que conlleva al desarrollo del sistema, considerando que esta empresa está dedicada al desarrollo de software y no cuenta con un sistema para recepción de requerimientos.

Cabe recalcar que los requerimientos son realizados por los usuarios que están en continuo uso del sistema Gameda y saben cómo realizar sus solicitudes en virtud al problema planteado, por ejemplo el personal que lleva la contabilidad de la empresa, en consecuencia ellos serán los que realicen las solicitudes de requerimientos.

Con las ventajas que nos ofrece actualmente el internet se planteó el desarrollo del sistema de recepción de requerimientos y control de tiempo empleado por los desarrolladores en la ejecución de proyectos en la empresa Gamedasoft, utilizando metodologías que nos proponen ciertos autores como son: Sommerville y Pressman las cuales ayudan al desarrollo de aplicaciones web como se requiere en la actualidad.

La arquitectura del software es algo indispensable en la actualidad ya que el objetivo principal de la arquitectura consiste en tener un sistema eficaz y de calidad.

En este proyecto se utilizó una tecnología moderna en el desarrollo de sistemas web donde se sigue los procesos de la ingeniería de software utilizando una arquitectura modelo vista controlador (MVC).

1.2. PLANTEAMIENTO DEL PROBLEMA

La empresa Gamedasoft, es una empresa que ha venido desarrollando desde hace varios años un producto de software para el apoyo en la gestión de las operaciones de las empresas ecuatorianas; el cual ha ido evolucionando hasta convertirse en un eficiente Enterprise Resource Planning (ERP).

De la experiencia laboral diaria en la empresa Gamedasoft se puede evidenciar dificultades en la comunicación al momento de la recepción de requerimientos, en la actualidad recepta los requerimientos vía telefónica. Estos requerimientos son administrados en una hoja electrónica (Excel), en la misma hoja, el desarrollador registra el tiempo dedicado a la solución de estos requerimientos.

Los tiempos registrados por los desarrolladores son utilizados para la facturación mensual ya que Gamedasoft ofrece un tiempo determinado para el mantenimiento de los sistemas de cada empresa. Por consiguiente, si los usuarios exceden el tiempo máximo de requerimientos se facturará un valor adicional.

Para solucionar estos problemas se plantea el desarrollo de la aplicación que permita:

- Registrar el tiempo empleado en un proyecto (requerimiento).
- Acceder a la información del tiempo que un desarrollador ha invertido en un proyecto.
- Generar reportes dinámicos con parámetros específicos enfocados en programadores vs tiempo.

1.3. OBJETIVOS

1.3.1. OBJETIVO GENERAL

Mejorar el proceso administrativo de registro de requerimientos de los usuarios mediante el desarrollo de un sistema web para recepción de requerimientos y control de tiempo empleado por los desarrolladores en la ejecución de proyectos en la empresa Gamedasoft.

1.3.2. OBJETIVOS ESPECÍFICOS

- Permitir al usuario el registro del requerimiento via web.
- Permitir al usuario la visualización del estado de la solicitud.
- Permitir la administración de requerimientos de los usuarios en forma eficiente via web.
- Controlar el acceso al sistema manejando perfiles de usuarios.
- Obtener reportes dinámicos que permitan visualizar el estado actual de las solicitudes de requerimientos.
- Obtener reportes de tiempos empleados en la ejecución de las solicitudes de requerimientos de los usuarios.

1.4. ALCANCE

El proyecto tiene como finalidad el desarrollo de un sistema web para el apoyo administrativo en la recepción de requerimientos y control de tiempo empleado por los desarrolladores en la ejecución de proyectos en la empresa Gamedasoft. La solución que se plantea incluirá los siguientes módulos.

- **Manejo de perfiles.**
Generación de perfiles a cada usuario para controlar el acceso a los módulos.
- **Administración de usuarios.**
Ingreso
Modificación
Control de acceso y uso del sistema web mediante el número de cédula o Ruc y una contraseña según corresponda el perfil.
- **Administración de requerimientos**

Ingreso de requerimientos por los clientes.

Cambio de estados de los requerimientos que incluye: Pendiente, asignado, en proceso, solucionado.

Registro de tiempos en proyectos: Permitir el ingreso de los tiempos invertidos por los desarrolladores en la solución de requerimientos.

- **Generación de reportes en general.**

Generar reportes solicitados por la empresa Gamedasoft como por ejemplo número de requerimientos por empresa y tiempo invertido por los desarrolladores en la ejecución de requerimientos.

La aplicación web será desarrollada utilizando un framework PHP open source como es Symfony.

1.5. PRESUPUESTO

	COSTO POR HORAS	HORAS	TOTAL
Diseñador	10	100	1000
Programador	10	100	1000

	HORAS	TOTAL
Software		
Apache	--	--
PHP	--	--
JavaScript	--	--
MYSQL	--	--
Hardware		
PC Escritorio	850	850
Otros		
Impresiones	50	50
Internet	150	150
Suministros de oficina	30	30
TOTAL		\$ 3.800
EL COSTO ASUMIRÁ GAMEDASOFT		

CAPITULO II

2. MARCO TEÓRICO

2.1. INGENIERÍA DE SOFTWARE

A continuación, se presentan las definiciones dadas por autores destacados en el campo:

“Software son programas de cómputo y documentación asociada. Los programas de software se desarrollan para un cliente en particular o para el mercado en general.” (Sommerville, 2011)

“El software es: 1) instrucciones (programas de cómputo) que cuando se ejecutan proporcionan las características, función y desempeño buscados; 2) estructuras de datos que permiten que los programas manipulen en forma adecuada la información, y 3) información descriptiva tanto en papel como en formas virtuales que describen la operación y uso de los programas.” (Pressman, 2010)

Antes de mencionar que es la ingeniería de software, se debe tener claro que software no solo son programas, sino abarca todas las características y configuraciones necesarias para el funcionamiento del programa.

La ingeniería de software es un proceso que comprende la producción de software desde la fase inicial que es la planificación hasta llegar a una fase de mantenimiento del software, en sus fases intermedias comprende fases que ayudarán a complementar la creación del software.

La ingeniería de software según el Standard Glossary of Software Engineering (IEEE), dice que es una aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo y mantenimiento del software. (IEEE, 2006)

El autor Roger Pressman, expone que la ingeniería de software es una tecnología que comprende capas que son herramientas, métodos, proceso y enfoque de calidad ver FIGURA 1- Capas de la ingeniería de software.

- **Herramientas.**

Sirven como soporte para los procesos y métodos el desarrollo del sistema web, para esto hay la posibilidad de utilizar herramientas computacionales denominadas Computer-Aided Software Engineering (CASE).

- **Métodos.**

Los métodos indican los pasos que se deben seguir utilizando diversas metodologías para construir técnicamente el software.

- **Proceso.**

El proceso es el componente que agrupa las actividades, métodos y tecnologías aplicadas al desarrollo del software.

- **Compromiso con la calidad.**

Es la base y la que gestiona la calidad de la ingeniería de software.



FIGURA 1- Capas de la ingeniería de software
(Pressman, 2010)

2.2. ETAPAS DEL DESARROLLO DE LOS SISTEMAS INFORMÁTICOS

Las etapas del desarrollo de los sistemas informáticos comprenden un conjunto de actividades que se debe seguir cuando se va a crear un proyecto, a lo largo de su vida se debe aplicar metodologías que pueden ser seleccionadas dependiendo del problema o grado de complejidad que se presente. Las etapas del desarrollo de sistemas informáticos aplican un proceso evolutivo involucrando a participantes que están relacionados al proyecto, con el fin de entender los requerimientos que debería tener el proyecto y así ayudar a definir las características y funcionalidades más eficientes del software.

Las etapas que comprenden el desarrollo de los sistemas de información son: planificación, análisis, diseño, implementación, pruebas, instalación o despliegue

y mantenimiento, ver FIGURA 2- Etapas del desarrollo de los sistemas informáticos.

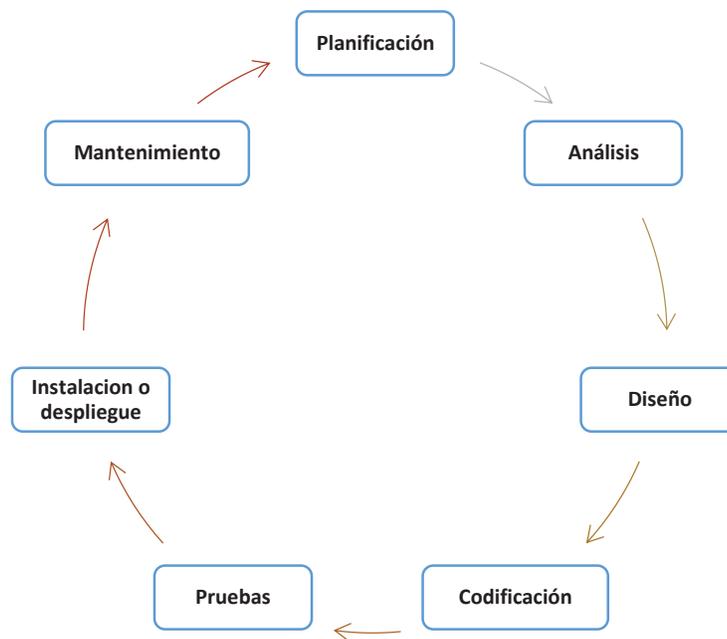


FIGURA 2- Etapas del desarrollo de los sistemas informáticos

A continuación, se presenta a una breve descripción en que consiste cada etapa del desarrollo de los sistemas informáticos como son:

- **Planificación.**

Se identifican las tareas iniciales del proyecto como son los objetivos que se persigue alcanzar y se debe cumplir, realizar el estudio de viabilidad, estimación de costos análisis de riesgo que podrían presentar antes, durante y después asociados con el proyecto.

- **Análisis.**

Se especifican los requerimientos que tendrá el proyecto, en la cual se determina las características que el sistema debería poseer. Para el análisis hay la posibilidad de aplicar técnicas entre la persona analista y las personas involucradas en el proyecto, como son la entrevista, aplicando cuestionarios o desarrollando prototipos. También se aplican herramientas de modelado de sistemas que no son más que un compendio del entorno.

- **Diseño.**

Al igual que la fase de análisis se crean diferentes modelos en base al aspecto que se requiere tener del sistema. Aquí se define las interfaces que deberían tener cada módulo y el diseño de la base de datos, seleccionar las herramientas de entorno de desarrollo adecuado y el lenguaje de programación que se va utilizar.

- **Codificación.**

La etapa comprende la programación del código fuente ósea la programación en función a los objetivos planteados.

- **Pruebas.**

Tiene como objetivo identificar los errores, en caso de existir, corregirlos antes que llegue al usuario final, con esto se reduce el costo ya que después del despliegue resultaría más costoso, para hacer pruebas se debería hacer con la colaboración de los analistas del sistema.

- **Instalación o despliegue.**

Luego de pasar las etapas anteriores, hay que buscar la planificación adecuada para que el sistema entre en funcionamiento como pueden ser configuraciones, tanto en hardware y software por ejemplo configuraciones de red, ya que las aplicaciones residen en ella.

- **Mantenimiento.**

El mantenimiento son los cambios que podrían presentar después de la etapa de instalación esto empieza desde el momento de su instalación y puesta en marcha, esto se lleva a lo largo de la vida del sistema.

2.3. METODOLOGÍA DE DISEÑO HIPERMEDIA ORIENTADO A OBJETOS (OOHDM).

La metodología (OOHDM) tiene como principal característica el uso de objetos para el desarrollo de aplicaciones web e hipermedia.

Características

- Está basada en la orientación a objetos.
- Propone un proceso predeterminado para el que indica las actividades a realizar y los productos que se deben obtener en cada fase del desarrollo. (Guerra Velez, 2011)

Esta metodología se divide en varios métodos para el análisis de los requisitos los cuales son el modelado conceptual, diseño navegacional, diseño de interfaz abstracto e implementación.

En la FIGURA 3- Relación entre el esquema conceptual, navegacional y los objetos de interfaz se muestra las relaciones que existe entre el esquema conceptual, navegacional y los objetos de interfaz en OOHDM.

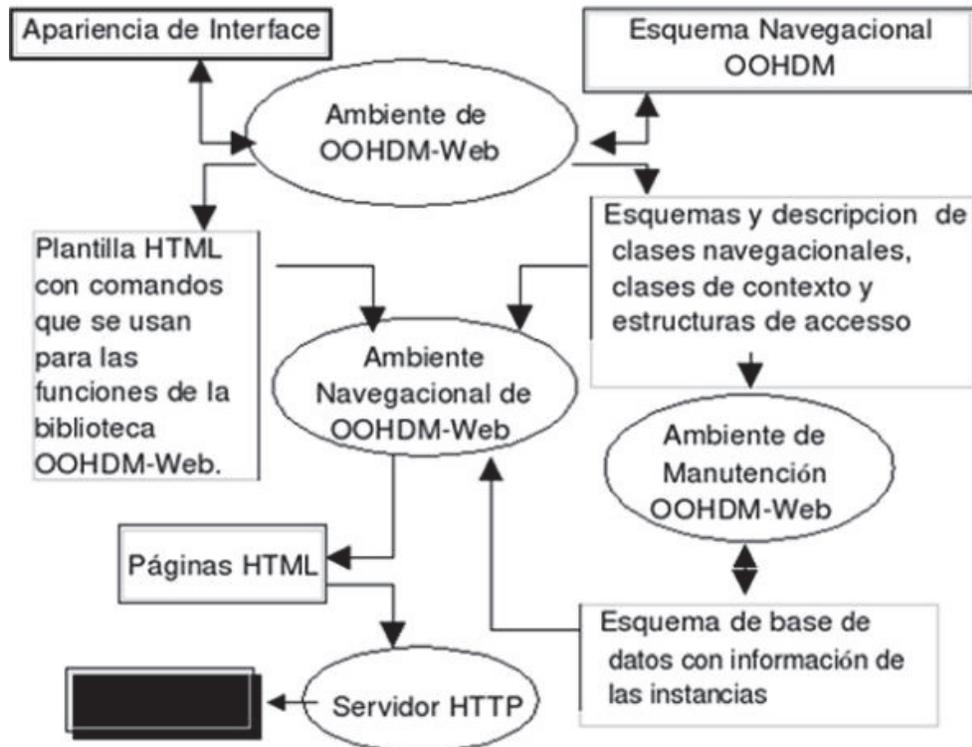


FIGURA 3- Relación entre el esquema conceptual, navegacional y los objetos de interfaz (documents.mx, 2015)

2.3.1. DISEÑO CONCEPTUAL

En esta fase se construyen un modelado conceptual representado por los objetos del dominio, las relaciones son creadas entre ellos creando técnicas propias de la orientación de objetos. La finalidad principal en esta fase es obtener el dominio semántico del sistema teniendo en cuenta el papel de los usuarios que intervienen y las tareas que les corresponde a cada uno de ellos.

En esta fase se puede hacer el uso de diagramas de lenguaje unificado de modelado (UML) que se explicará más adelante. Aquí se puede utilizar diagramas de clases y objetos.

2.3.2. DISEÑO NAVEGACIONAL

En esta fase se identifica un conjunto de objetos que son procedentes de la fase conceptual y la cuales son definidos en esta fase como nodos o denominadas clases de navegación.

Según como se amplíe el diseño se establecen un conjunto de clases de navegación como son nodos, vínculos y estructuras de acceso.

2.3.3. DISEÑO DE INTERFAZ ABSTRACTO

Después de pasar las dos fases y haber determinado la estructura del diseño navegacional, en esta fase se debe determinar el aspecto que tendrá la interfaz, es decir, definir los diferentes aspectos de la navegación que aparecerán y que transformaciones de la interfaz tendrán que actuar en un determinado proceso.

2.3.4. IMPLEMENTACIÓN

En esta fase se integran todas las fases indicadas anteriormente se aplican características relevantes para la ejecución del problema, como son llevar a un lenguaje concreto de programación, es decir, es el proceso final del proyecto.



FIGURA 4- Etapas de la metodología OOADM.

2.4. LENGUAJE DE MODELADO UML

El lenguaje unificado de modelo (UML) es un lenguaje gráfico que adoptan empresas dedicadas al desarrollo de software, para crear diferentes diagramas al momento de construir programas informáticos. Se podría decir que es un lenguaje

simbólico para expresar modelos y no una metodología para el desarrollo de sistemas, cabe destacar que no es un lenguaje de programación.

Este modelo permite modelar problemas de diferentes ámbitos y trasladar a un lenguaje gráfico para la especificación, visualización, construcción y documentación de procesos ocurridos durante el desarrollo del software.

En conclusión, se podría decir que UML es un compendio de notación gráfica, útil al momento de desarrollar sistemas informáticos, particularmente sistemas que se desarrollarán en técnicas de orientación a objetos.

2.4.1. BENEFICIOS QUE PROPORCIONA EL UML

- Provee a los desarrolladores de software un modelamiento visual listo para usarse.
- Es independiente de los lenguajes de programación del desarrollo del software.
- El vocabulario se centra en la representación conceptual y física del sistema a desarrollar.
- Definen conjunto de reglas claras y precisas con el fin de facilitar la comunicación.
- Permite mejorar técnicas para el desarrollo de software.

Se puede realizar varios diagramas que están en la versión UML 2.x, que se muestran en la FIGURA 5- Diagramas UML.

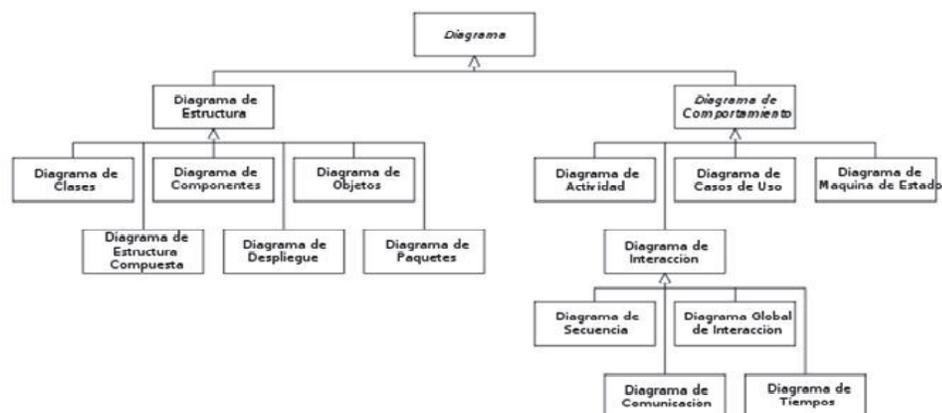


FIGURA 5- Diagramas UML.
(Yeison, 2011)

2.5. MODELADO DE BASE DE DATOS

El modelado de datos se utiliza para describir el espacio de información que será construido o manipulado por el software. El modelo de datos empieza con la representación de los objetos de datos información compuesta que debe ser entendida por el software. (Pressman, 2010)

2.5.1. DISEÑO CONCEPTUAL

El diseño del modelo conceptual es la primera etapa para el desarrollo de la base de datos de un sistema. En esta etapa se representa la información coherente obtenida del mundo real a un esquema conceptual, dicha información se obtendrá de entrevistas a usuarios, en esta etapa se busca una representación normalizada y evitar redundancias en los datos. Para representar este diseño se puede utilizar el modelo entidad relación.

2.5.2. DISEÑO LÓGICO

En esta etapa se parte del esquema conceptual, consiste en transformar el modelo entidad relación donde las entidades pasan a ser tablas, los atributos se convierten en campos, etc. El objetivo principal es definir correctamente las tablas, campos, claves primarias, claves foráneas, etc.

2.5.3. DISEÑO FÍSICO

Después del diseño lógico, la base de datos puede trasladarse a un modelo físico, el cual puede ser utilizado por un sistema gestor de base de datos (SGBD) y determinará como se guardarán los datos, tipos de datos, etc.

2.6. PARADIGMA DE PROGRAMACIÓN ORIENTADA A OBJETOS

La programación orientada a objetos es un método de desarrollo en el cual los programas se organizan como colecciones de objetos que cooperan para resolver un problema. En general, los objetos pueden corresponderse a entidades del mundo real, acciones o a procesos. (Vélez Serrano, 2011)

2.6.1. CONCEPTOS IMPORTANTES DE LA PROGRAMACIÓN ORIENTADA A OBJETOS

- **CLASE**

Es una plantilla que a partir de esta se crean objetos, dichos objetos tienen un comportamiento y estructura común.

- **OBJETO**

Un objeto es algo que puede ser identificado y caracterizado.

- **MÉTODOS**

Son aquellas operaciones que pueden realizar los objetos.

- **PROPIEDADES**

Son características o atributos que poseen los objetos.

2.6.2. CARACTERÍSTICAS DE LA PROGRAMACIÓN ORIENTADA A OBJETOS

- **ABSTRACCIÓN**

Es la que permite hacer referencia a las características principales que distinguen de los demás objetos dejando en segundo plano los detalles. Con estos elementos se puede definir la comunicación con objetos sin permitir realizar cambios directamente.

- **ENCAPSULAMIENTO**

La capacidad que permite mantener oculta la implementación y mantener al mismo nivel a los elementos que se consideran de la misma entidad para tener una adecuada organización.

- **POLIMORFISMO**

Es la capacidad que tienen los objetos de una clase para responder a diferentes llamadas a través de métodos a un mismo mensaje, de acuerdo al comportamiento que estos tengan.

- **HERENCIA**

Esta característica es la que organiza el encapsulamiento y el polimorfismo permitiendo a los objetos compartir sus atributos y operaciones las cuales están basadas en una relación jerárquica. En herencia una clase se define en forma general y de esta se puede ir refinando en otras subclases.

2.7. PLATAFORMA WEB

2.7.1. OPEN SOURCE

En español código abierto, en la actualidad la mayor cantidad de desarrolladores optan por la utilización de lenguajes de programación que se basan en soluciones de software libre y ofrecer a los clientes soluciones tecnológicas de punta con mayores beneficios. Entre estos beneficios tenemos la independencia de plataformas y el costo de licenciamiento.

2.7.2. XAMPP

Es un paquete de software libre que sirve como ayuda para el desarrollo de sistemas web, generalmente utilizado para pruebas en forma local permite realizar configuraciones necesarias para levantar un sistema web.

Este paquete contiene el servidor web Apache, lenguaje de programación PHP y un servidor de base de datos MYSQL.

- **APACHE**

Es un servidor web de libre distribución compatible con Linux, Windows y Mac con excelentes características, permite realizar configuraciones acordes a las necesidades del sistema a levantar sobre este y cuenta con una mayor seguridad y robustez.

- **PHP**

PHP es un lenguaje de programación que los desarrolladores utilizan en la actualidad para la programación de sitios web dinámicos, ya que tiene varias ventajas entre ellos es libre y de código abierto, un lenguaje orientado a objeto. Hay que mencionar que PHP es un lenguaje que se ejecuta en el lado del Servidor

- **MYSQL**

MySQL es un sistema de administración de bases de datos relacionales rápido, sólido y flexible es ideal para crear bases de datos con acceso desde páginas web dinámicas, para la creación de sistemas de transacciones online o para cualquier otra solución profesional que implique almacenar datos, teniendo la posibilidad de realizar múltiples y rápidas consultas.

Las ventajas que tiene MySQL son muchas, la principal es que es libre, es un sistema cliente-servidor, es portable, esto quiere decir que se puede llevar a cualquier plataforma.

2.7.3. SYMFONY

Symfony es un framework diseñado para optimizar el tiempo y realizar aplicaciones web en forma rápida, basada en lenguaje de programación PHP, las características que nos ofrecen son muchas gracias a la arquitectura modelo, vista y controlador (MVC) separando la lógica del negocio, lógica del servidor y presentación.

PHP Framework Popularity in Personal Projects - SitePoint, 2015

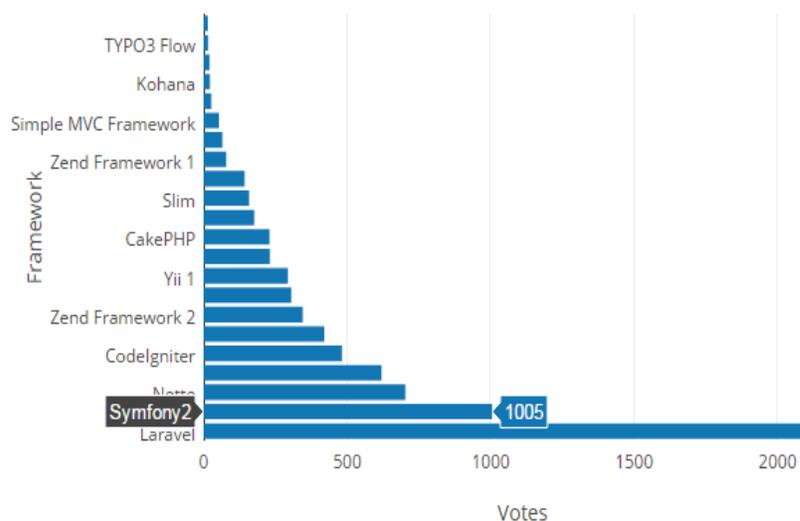


FIGURA 6- Popularidad de Symfony en proyectos personales.
(SitePoint, 2015)

En la FIGURA 6- Popularidad de Symfony en proyectos personales. Podemos ver que Symfony se encuentra en segundo lugar de popularidad con respecto a otro framework como es Laravel.

- **CARACTERÍSTICAS DE SYMFONY**

- Programación de código es de software libre.
- La seguridad es muy importante en los proyectos de desarrollo de software bajo framework Symfony.
- Utiliza componentes que pueden ser utilizados en cualquier momento con garantía de funcionamiento.
- Es independiente de los sistemas de gestores de base de datos.

- **MODELO VISTA CONTROLADOR EN SYMFONY**

La arquitectura MVC separa la lógica de negocio (el modelo) y la presentación (la vista) por lo que se consigue un mantenimiento más sencillo de las aplicaciones. Si por ejemplo, una misma aplicación debe ejecutarse tanto en un navegador estándar como un navegador de un dispositivo móvil, solamente es necesario crear una vista nueva para cada dispositivo; manteniendo el controlador y el modelo original. El controlador se encarga de aislar al modelo y a la vista de los detalles del protocolo utilizado para las peticiones (HTTP, consola de comandos, email, etc.). El modelo se encarga de la abstracción de la lógica relacionada con los datos, haciendo que la vista y las acciones sean independientes de, por ejemplo, el tipo de gestor de bases de datos utilizado por la aplicación. (Potencier & Zaninotto, 2008). FIGURA 7- MVC en Symfony

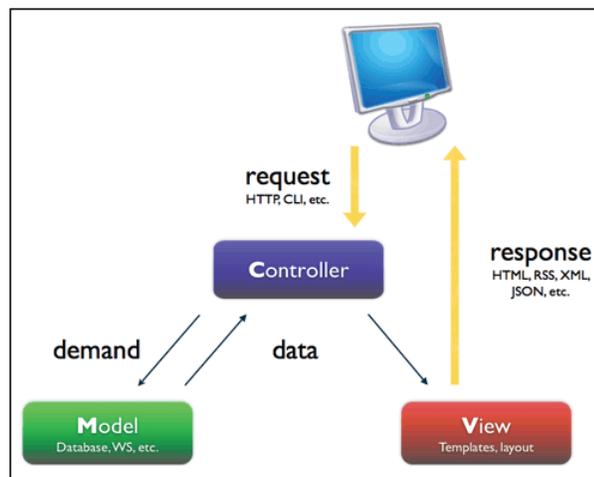


FIGURA 7- MVC en Symfony
(Symfony.com, 2016)

- **ELEMENTOS IMPORTANTES EN SYMFONY**

- ✓ **EL CONTROLADOR**

Es la que se encarga de interpretar la información de una petición, procesar y devolver la respuesta al usuario a través de un navegador.

El principal objetivo del controlador es crear y devolver un objeto de respuesta. El controlador representa una función creada con código PHP.

- ✓ **DOCTRINE**

Esto representa la capa del modelo en la creación del sistema, doctrine es el mapeo de objeto-relacional la cual consiste en la transformación de las tablas de una base de datos relacional a entidades, los registros a objetos y los campos a propiedades.

- ✓ **TWIG**

Es un motor de plantillas que es rápido, seguro y confiable, ya que son las páginas que el sistema da respuesta a una petición del usuario cuando se ingresa una acción en el sistema, esto representa la vista en el modelo MVC.

- ✓ **SISTEMA DE ENRUTAMIENTO**

El enrutamiento es un mecanismo que reescribe la URL para simplificar su aspecto en ingles el routing.

- ✓ **BUNDLE**

Es una carpeta que agrupa los archivos necesarios para un grupo de funcionalidades del sistema. El Bundle está compuesto por controladores, archivos Twig, archivos de enrutamiento y archivos de configuración.

2.8. DISEÑO WEB

2.8.1. HTML 5

Lenguaje de marcas de hipertexto versión 5 (HTML 5), por sus siglas inglés (Hypertext Markup Language), es una de las últimas versiones de HTML aplicando nuevos elementos, con el fin de tener mayor alcance en los sitios web modernos es creado por desarrolladores open web.

2.8.2. CSS

Hojas de estilo en cascada (CSS), por sus siglas en inglés (Cascading style sheets), es utilizada por los diseñadores de páginas web para dar estilos a documentos escritos en formato HTML, es esencial al momento de establecer formatos tipográficos utilizando reglas establecidas por el mecanismo que tiene CSS.

2.8.3. JAVASCRIPT

Es un lenguaje de programación orientada a la web, permite tener aplicaciones web dinámicas utilizando lenguaje de programación interpretado. JavaScript funciona en entorno del cliente ya que los navegadores modernos soportan este lenguaje.

2.8.4. JQUERY

Es una librería de JavaScript la cual permite simplificar código de JavaScript y permite interactuar con el documento HTML. Con esta librería se puede realizar una gran cantidad de soluciones de manera sencilla, por ejemplo, animaciones, manipulación del Modelo de Objetos del Documento (DOM) y CSS y manejo de eventos.

2.8.5. BOOTSTRAP

Bootstrap es un framework que nos ayuda a crear interfaces amigables para la vista del usuario, fue creado por Twitter y posteriormente liberado. Este framework facilita la obtención de adaptables a cualquier tipo de dispositivo ya sea de escritorio o móvil.

Bootstrap es compatible con la mayoría de navegadores modernos que se encuentran en el mercado. Es un framework basado en herramientas de diseño actual como son CSS, JavaScript y JQuery.

2.8.6. AJAX

Es una tecnología para el desarrollo de páginas web dinámicas que permite, la comunicación con el servidor en forma asincrónica. Esto reduce la necesidad de recargar las páginas al momento de hacer solicitudes al servidor, como puede ser la obtención de datos de una base de datos.

FIGURA 8- Funcionamiento de AJAX

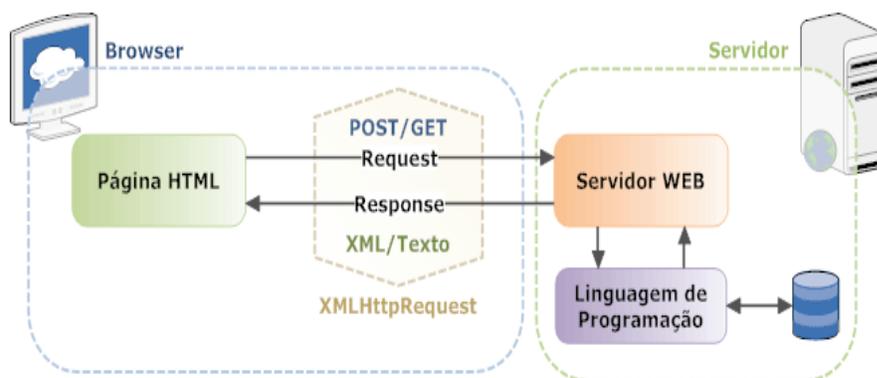


FIGURA 8- Funcionamiento de AJAX
(Rafale, 2008)

CAPITULO III

3. DISEÑO Y DESARROLLO DE LA APLICACIÓN WEB

3.1. ENCUESTAS Y REQUERIMIENTOS PARA EL DEL DESARROLLO DEL SISTEMA

3.1.1. OBTENCIÓN DE REQUERIMIENTOS

Preguntas que se propuso por parte del entrevistador hacia la persona encargada de administrar los requerimientos en la empresa Gamedasoft.

- ¿Cómo es el proceso de registro de requerimientos?
- ¿Con que frecuencia se presentan los requerimientos?
- ¿Cómo se presentan los requerimientos?
- ¿Qué herramientas informáticas utiliza para registrar los requerimientos?
- ¿Cómo es la asignación de requerimientos a los programadores?
- ¿Qué datos registran los programadores?
- ¿Cómo se comunica a los clientes el estado del requerimiento?
- ¿Qué tipo de reporte obtiene en la herramienta informática?

3.1.2. LISTA DE REQUERIMIENTOS

- Permitir ingreso de requerimientos por los clientes.
- Permitir asignar requerimientos a programadores según la dificultad del mismo.
- Permitir el ingreso de tiempos invertidos en la solución de requerimientos por los programadores.
- Permitir ver el estado actual del requerimiento tanto a clientes como a la persona encargada de asignación.
- Emisión de reportes de los requerimientos solicitados por los clientes.
- Reporte de solicitud de requerimientos por módulo.
- Reporte de solicitud de requerimiento por fecha.
- Reporte del tiempo invertido por programadores por cliente.
- Registrar usuarios nuevos (clientes, Empleados).
- Permitir actualizar datos de clientes, empleados y administrador.
- Permitir el ingreso al sistema con claves.

3.2. ANÁLISIS DE REQUERIMIENTOS

3.2.1. REQUERIMIENTOS FUNCIONALES SEGÚN OOHDM

Esta metodología divide a la obtención de requerimientos en las siguientes etapas:

- **Identificación de actores y tareas.**
- **Especificación de escenarios.**
- **Especificación de casos de uso.**

DEFINICIÓN DE ACTORES Y TAREAS

En esta etapa se identifican los actores y sus respectivas tareas, ya que son el punto de partida para seguir a la siguiente etapa como es el diseño de los escenarios. Los actores que se han definido para el sistema se muestran en la Tabla 1- Identificación de actores y su descripción.

ACTOR	DESCRIPCIÓN
 ADMINISTRADOR	Es el usuario que tiene acceso total al sistema, el cual tendrá que asignar los requerimientos.
 EMPLEADO	Es aquel usuario que tiene permisos para actualizar sus datos y registrar tiempos en la solución de requerimientos.
 CLIENTE	Es el usuario que tendrá la posibilidad de ingresar requerimientos para su posterior solución y; actualizar sus datos.

Tabla 1- Identificación de actores y su descripción.

ACTOR	TAREAS
Administrador	<ol style="list-style-type: none"> 1. Iniciar sesión desde el perfil administrador. 2. Actualizar datos. 3. Cambiar contraseña. 4. Administrar empleados. 5. Administrar clientes. 6. Asignar requerimientos. 7. Administrar módulos. 8. Administrar menús. 9. Administrar submenús. 10. Consultar el estado del requerimiento por cliente. 11. Consultar el tiempo invertido en la solución de cada requerimiento por cliente. 12. Cerrar sesión.
Empleado	<ol style="list-style-type: none"> 1. Iniciar sesión desde el perfil empleado. 2. Actualizar datos. 3. Cambiar contraseña. 4. Ingresar tiempo invertido en la solución de requerimientos. 5. Consultar requerimiento asignado. 6. Cerrar sesión.
Cliente	<ol style="list-style-type: none"> 1. Iniciar sesión desde el perfil cliente. 2. Actualizar datos. 3. Cambiar contraseña. 4. Ingresar requerimiento. 5. Consultar estado requerimiento. 6. Cerrar sesión.

Tabla 2- Identificación de tareas por actor

ESPECIFICACIÓN DE ESCENARIOS

A continuación, se describen los escenarios para los usuarios del sistema.

- **Usuario Administrador**

Escenario	E1	Iniciar sesión	Usuario	Administrador
Contexto	El administrador debe autenticarse para poder realizar las tareas asignadas.			
Objetivo	Ingresar al sistema.			
Acciones	<ol style="list-style-type: none"> 1. Ingresar a la parte privada del sistema. 2. Seleccionar el perfil administrador. 3. Ingresar su número de cédula y contraseña correspondiente. 			

Tabla 3- Escenario inicio de sesión administrador.

Escenario	E2	Actualizar datos	Usuario	Administrador
Contexto	El administrador tendrá la posibilidad de actualizar sus datos.			
Objetivo	Tener datos actualizados.			
Acciones	<ol style="list-style-type: none"> 1. Seleccionar menú correspondiente. 2. Cambiar datos que desea actualizar. 			

Tabla 4- Escenario actualizar datos administrador.

Escenario	E3	Cambiar contraseña	Usuario	Administrador
Contexto	El administrador podrá cambiar la contraseña cuantas veces quiera.			
Objetivo	Cambiar contraseña por seguridad.			
Acciones	<ol style="list-style-type: none"> 1. Seleccionar menú correspondiente. 2. Ingresar clave anterior. 3. Ingresar clave nueva. 4. Confirmar clave nueva. 			

Tabla 5- Escenario cambiar contraseña administrador.

Escenario	E4	Administrar Empleados	Usuario	Administrador
Contexto		El administrador deberá crear empleados que cumplirán una determinada función.		
Objetivo		Tener el control de todos los empleados del sistema.		
Acciones		<ol style="list-style-type: none"> 1. Crear empleados nuevos. 2. Modificar datos del empleado. 		

Tabla 6- Escenario administrar empleados.

Escenario	E5	Administrar Clientes	Usuario	Administrador
Contexto		El administrador deberá crear clientes y actualizar datos.		
Objetivo		Tener el control de todos los clientes del sistema.		
Acciones		<ol style="list-style-type: none"> 1. Crear clientes nuevos. 2. Modificar datos del cliente. 		

Tabla 7- Escenario administrar clientes.

Escenario	E6	Asignar Requerimientos	Usuario	Administrador
Contexto		El administrador deberá analizar el requerimiento y asignar al empleado para su resolución.		
Objetivo		Asignar requerimiento a empleados.		
Acciones		<ol style="list-style-type: none"> 1. Seleccionar empleado. 2. Asignar requerimiento. 		

Tabla 8- Escenario asignación requerimientos.

Escenario	E7	Administrar módulos	Usuario	Administrador
Contexto	El administrador deberá crear módulos y cambiar el estado.			
Objetivo	Mantener actualizado los módulos que cuenta el otro sistema.			
Acciones	<ol style="list-style-type: none"> 1. Crear módulos. 2. Modificar estado del módulo. 			

Tabla 9- Escenario administrar módulos.

Escenario	E8	Administrar Menús	Usuario	Administrador
Contexto	El administrador deberá crear menús y asignar al módulo que pertenece, también puede cambiar el estado del menú.			
Objetivo	Mantener actualizado los menús que cuenta el otro sistema.			
Acciones	<ol style="list-style-type: none"> 1. Crear menús. 2. Modificar estado del menú. 3. Asignar al módulo que pertenece el menú. 			

Tabla 10- Escenario administrar menús.

Escenario	E9	Administrar Submenús	Usuario	Administrador
Contexto	El administrador deberá crear Submenús asignar al módulo y menú correspondiente, también puede cambiar el estado del Submenú.			
Objetivo	Mantener actualizado los Submenús que cuenta el otro sistema.			
Acciones	<ol style="list-style-type: none"> 1. Crear Submenús. 2. Modificar estado del Submenú. 3. Asignar al módulo y menú que pertenece el submenú. 			

Tabla 11- Escenario administrar Submenús.

Escenario	E10	Consultar estado del requerimiento por cliente	Usuario	Administrador
Contexto		El administrador podrá realizar consulta del estado en que se encuentra el requerimiento.		
Objetivo		Permitir consultar el estado del requerimiento de cierto cliente.		
Acciones		<ol style="list-style-type: none"> 1. Seleccionar el respectivo menú de opciones. 2. Seleccionar el cliente al que se requiere realizar la consulta. 		

Tabla 12- Escenario consulta estado del requerimiento por empleado.

Escenario	E11	Consultar el tiempo invertido en la solución de cada requerimiento por cliente.	Usuario	Administrador
Contexto		El administrador podrá realizar consultas de los tiempos que invirtió el empleado en la solución del requerimiento, esto lo puede realizar por cliente.		
Objetivo		Permitir al administrador consultar tiempos invertidos por requerimientos.		
Acciones		<ol style="list-style-type: none"> 1. Seleccionar el respectivo menú de opciones. 2. Seleccionar el cliente al que se requiere realizar la consulta de tiempos. 		

Tabla 13- Escenario consultar el tiempo invertido en la solución de cada requerimiento por cliente.

Escenario	E12	Cerrar sesión	Usuario	Administrador
Contexto	El administrador al terminar de realizar su actividad en el sistema deberá terminar la sesión.			
Objetivo	Evitar que personas mal intencionadas no autorizadas modifiquen la información que tiene el sistema.			
Acciones	1. Cerrar sesión.			

Tabla 14- Cerrar sesión administrador.

- **Usuario Empleado**

Escenario	E13	Iniciar sesión	Usuario	Empleado
Contexto	El empleado debe autenticarse para poder realizar las tareas asignadas.			
Objetivo	Ingresar al sistema.			
Acciones	<ol style="list-style-type: none"> 1. Ingresar a la parte privada del sistema. 2. Seleccionar el perfil empleado. 3. Ingresar su número de cédula y contraseña correspondiente. 			

Tabla 15- Escenario inicio de sesión empleado.

Escenario	E14	Actualizar datos	Usuario	Empleado
Contexto	El empleado tendrá la posibilidad de actualizar sus datos.			
Objetivo	Tener datos actualizados.			
Acciones	<ol style="list-style-type: none"> 1. Seleccionar menú correspondiente. 2. Cambiar datos que desea actualizar. 			

Tabla 16- Escenario actualizar datos empleado.

Escenario	E15	Cambiar contraseña	Usuario	Empleado
Contexto	El empleado podrá cambiar la contraseña cuantas veces quiera.			
Objetivo	Cambiar contraseña por seguridad.			
Acciones	<ol style="list-style-type: none"> 1. Seleccionar menú correspondiente. 2. Ingresar clave anterior. 3. Ingresar clave nueva. 4. Confirmar clave nueva. 			

Tabla 17- Escenario cambiar contraseña administrador.

Escenario	E16	Ingreso tiempos invertidos en la solución de requerimientos.	Usuario	Empleado
Contexto	El empleado podrá ingresar tiempos que se invirtió en la solución de requerimientos.			
Objetivo	Registrar tiempos invertidos en la solución de requerimientos.			
Acciones	<ol style="list-style-type: none"> 1. Registrar actividad que se realizó para solución de requerimientos. 2. Registra el tiempo en minutos que utilizo en la solución del requerimiento. 			

Tabla 18- Escenario ingreso tiempos invertidos en la solución de requerimientos.

Escenario	E17	Consultar requerimiento asignado	Usuario	Empleado
Contexto	El empleado podrá realizar consulta de los requerimientos que le fueron asignados por el administrador del sistema.			
Objetivo	Consultar requerimientos que se le asignaron.			
Acciones	<ol style="list-style-type: none"> 1. Consultar requerimientos asignados. 			

Tabla 19- Escenario consulta requerimiento asignado.

Escenario	E18	Cerrar sesión.	Usuario	Empleado
Contexto	El empleado al finalizar todas las actividades tendrá la obligación de cerrar la sesión.			
Objetivo	Evitar que personas no autorizadas manipulen los datos.			
Acciones	<ol style="list-style-type: none"> 1. Seleccionar la opción correspondiente. 2. Salir cerrando la sesión. 			

Tabla 20- Escenario cerrar sesión empleado.

- **Usuario Cliente**

Escenario	E19	Iniciar sesión	Usuario	Cliente
Contexto	El Cliente debe autenticarse para poder realizar las tareas requeridas.			
Objetivo	Ingresar al sistema.			
Acciones	<ol style="list-style-type: none"> 1. Ingresar a la parte privada del sistema. 2. Seleccionar el perfil cliente. 3. Ingresar su número de Ruc y contraseña correspondiente. 			

Tabla 21- Escenario iniciar sesión cliente.

Escenario	E20	Actualizar datos	Usuario	Cliente
Contexto	El cliente tendrá la posibilidad de actualizar sus datos.			
Objetivo	Tener datos actualizados.			
Acciones	<ol style="list-style-type: none"> 1. Seleccionar menú correspondiente. 2. Cambiar datos que desea actualizar. 			

Tabla 22- Escenario actualizar datos cliente.

Escenario	E21	Cambiar contraseña	Usuario	Cliente
Contexto	El cliente podrá cambiar la contraseña cuantas veces quiera.			
Objetivo	Cambiar contraseña por seguridad.			
Acciones	<ol style="list-style-type: none"> 1. Seleccionar menú correspondiente. 2. Ingresar clave anterior. 3. Ingresar clave nueva. 4. Confirmar clave nueva. 			

Tabla 23- Escenario cambiar contraseña cliente.

Escenario	E22	Ingresar requerimiento	Usuario	Cliente
Contexto	El cliente registrará un nuevo requerimiento en el sistema.			
Objetivo	Registrar requerimientos en el sistema.			
Acciones	<ol style="list-style-type: none"> 1. Seleccionar menú correspondiente. 2. Crear un nuevo requerimiento. 			

Tabla 24- Escenario ingresar requerimiento.

Escenario	E23	Consultar estado requerimiento	Usuario	Cliente
Contexto	El cliente podrá consultar estado del requerimiento.			
Objetivo	Consultar estado requerimiento			
Acciones	<ol style="list-style-type: none"> 1. Seleccionar menú correspondiente. 2. Consultar estado requerimiento. 			

Tabla 25- Escenario consultar estado requerimiento.

Escenario	E24	Cerrar sesión.	Usuario	Cliente
Contexto	El cliente al finalizar todas las actividades tendrá la obligación de cerrar la sesión.			
Objetivo	Evitar que personas no autorizadas manipulen los datos.			
Acciones	<ol style="list-style-type: none"> 1. Seleccionar la opción correspondiente. 2. Salir cerrando la sesión. 			

Tabla 26- Escenario cerrar sesión del cliente.

ESPECIFICACIÓN DE CASOS DE USO

- **Casos de uso comunes para todos los actores del sistema.**

Caso de uso que representa actualizar datos para los usuarios administrador, empleado y cliente.

Caso de uso	Actualizar datos	
ID	CU-01	
Escenario	Actor	Administrador, Empleado, Cliente.
E2, E14, E20		
Breve descripción:		
Los usuarios necesitan actualizar sus datos, la cual deberán ingresar al menú actualizar datos de su perfil correspondiente.		
Pre-condiciones:		
Tener acceso al sistema.		

<p>Flujo de eventos:</p> <p>Flujo Básico:</p> <ol style="list-style-type: none"> 1. Usuario: Selecciona el menú “Mi usuario”. 2. Sistema: Despliega las opciones contenidas en el menú. 3. Usuario: Selecciona la opción “Actualizar datos”. 4. Sistema: Presenta una pantalla de información del usuario que posee el sistema. 5. Usuario: Ingresa la nueva información en los campos a actualizar y presiona en el botón guardar. 6. Sistema: Valida la información ingresada y guarda en el sistema. <p>Flujo Alternativo:</p> <ol style="list-style-type: none"> 1. Sistema: Presenta un mensaje indicando error en los campos ingresados.
<p>Post-condiciones:</p> <p>Datos guardados exitosamente.</p>

Tabla 27- CU-01 Actualizar datos.

Caso de uso que representa cambiar la contraseña de ingreso al sistema, para los usuarios administrador, empleado y cliente.

Caso de uso		Cambiar contraseña	
ID		CU-02	
Escenario		Actor	Administrador, Empleado, Cliente.
E3, E15, E21			
Breve descripción:			
Los usuarios necesitan cambiar las contraseñas en un tiempo periódico para lo cual deberán ingresar al menú cambiar contraseña del perfil que corresponda.			
Pre-condiciones:			
Tener acceso al sistema.			

<p>Flujo de eventos:</p> <p>Flujo Básico:</p> <ol style="list-style-type: none"> 1. Usuario: Selecciona el menú “Mi usuario”. 2. Sistema: Despliega las opciones contenidas en el menú. 3. Usuario: Selecciona el menú “Cambiar contraseña”. 4. Sistema: Presenta pantalla donde se ingresará: clave anterior, clave nueva y la confirmación de la clave nueva. 5. Usuario: Ingresa los datos solicitados y presiona en el botón guardar. 6. Sistema: Valida la información ingresada y guarda en el sistema. <p>Flujo Alternativo:</p> <ol style="list-style-type: none"> 1. Sistema: Presenta un mensaje indicando error en los campos ingresados o error en la confirmación de la clave nueva.
<p>Post-condiciones:</p> <p>Datos guardados exitosamente.</p>

Tabla 28- CU-02 Cambiar contraseña.

Caso de uso que representa cerrar sesión, para los usuarios administrador, empleado y cliente.

Caso de uso		Cerrar sesión	
ID		CU-03	
Escenario		Actor	Administrador, Empleado, Cliente.
E12, E18, E24			
Breve descripción:			
Los usuarios pueden salir del sistema cuando lo requieran.			
Pre-condiciones:			
Ingresar al sistema.			

Flujo de eventos:**Flujo Básico:**

1. Usuario: Selecciona el botón cerrar sesión.
2. Sistema: Cierra la sesión.
3. Usuario: Visualiza la pantalla principal del sistema.

Flujo Alternativo:

No aplica.

Post-condiciones:

Guarda los cambios realizados durante la sesión del usuario.

Tabla 29- CU-03 Cerrar sesión.

- **Casos de uso del actor administrador.**

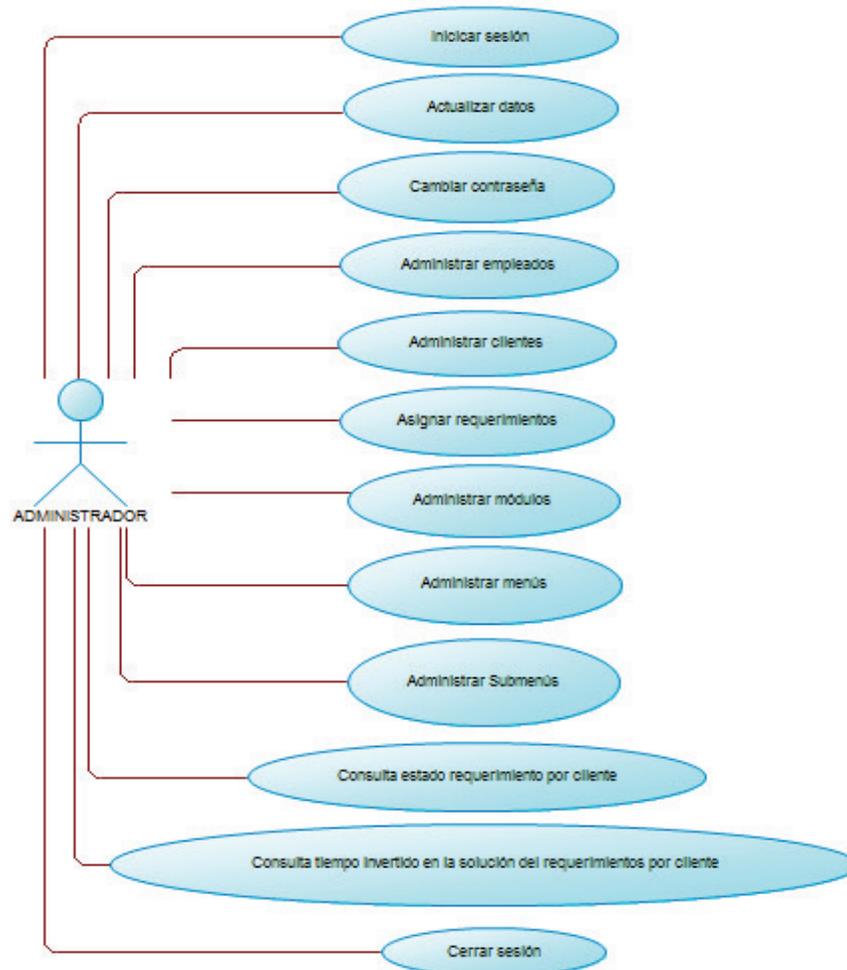


FIGURA 9- Diagrama del caso de uso del rol administrador.

Caso de uso		Iniciar sesión	
ID		CU-04	
Escenario	E1	Actor	Administrador
Breve descripción:			
El administrador necesita ingresar al sistema para lo cual deberá seleccionar el perfil administrador e ingresar su número de cedula y contraseña.			
Pre-condiciones:			
Tener una cuenta creada como administrador.			
Flujo de eventos:			
Flujo Básico:			
1. Usuario: Selecciona el perfil administrador e ingresa el número de cedula y contraseña, luego presionar el botón ingresar al sistema.			
2. Sistema: valida los datos ingresados.			
3. Sistema: Muestra la pantalla principal del perfil administrador.			
Flujo Alternativo:			
1. Sistema: Presenta un mensaje indicando que no existe un usuario con los datos ingresados.			
Post-condiciones:			
Almacena el id de sesión en el sistema en forma temporal.			

Tabla 30- CU-04 Iniciar sesión administrador.

Caso de uso		Administrar empleados	
ID		CU-05	
Escenario	E4	Actor	Administrador
Breve descripción:			
El administrador visualiza, edita y crea empleados en el sistema.			
Pre-condiciones:			
Tener acceso al sistema con perfil administrador.			

<p>Flujo de eventos:</p> <p>Flujo Básico:</p> <ol style="list-style-type: none"> 1. Usuario: Selecciona el menú “Operaciones”. 2. Sistema: Despliega las opciones contenidas en el menú. 3. Usuario: Selecciona el menú “Empleados”. 4. Sistema: Muestra las opciones de crear, editar y visualizar. 5. Usuario: Escoge la actividad que desea realizar. <p>Flujo Alternativo:</p> <p>No aplica</p>
<p>Post-condiciones:</p> <p>Guarda los cambios realizados según la opción seleccionada.</p>

Tabla 31- CU-05 Administrar empleados.

Caso de uso		Administrar Clientes	
ID		CU-06	
Escenario	E6	Actor	Administrador
<p>Breve descripción:</p> <p>El administrador visualiza, edita y crea clientes en el sistema.</p>			
<p>Pre-condiciones:</p> <p>Tener acceso al sistema con perfil administrador.</p>			
<p>Flujo de eventos:</p> <p>Flujo Básico:</p> <ol style="list-style-type: none"> 1. Usuario: Selecciona el menú “Operaciones”. 2. Sistema: Despliega las opciones contenidas en el menú. 3. Usuario: Selecciona el menú “Clientes”. 4. Sistema: Muestra las opciones de crear, editar y visualizar. 5. Usuario: Escoge la actividad que desea realizar. <p>Flujo Alternativo:</p> <p>No aplica</p>			
<p>Post-condiciones:</p> <p>Guarda los cambios realizados según la opción seleccionada.</p>			

Tabla 32- CU-06 Administrar clientes.

Caso de uso		Asignar requerimiento	
ID		CU-07	
Escenario	E6	Actor	Administrador
Breve descripción:			
El administrador asigna el requerimiento al empleado teniendo en cuenta la dificultad del requerimiento.			
Pre-condiciones:			
Tener acceso al sistema con perfil administrador.			
Flujo de eventos:			
Flujo Básico:			
<ol style="list-style-type: none"> 1. Usuario: Selecciona el menú “Asignación Tareas”. 2. Sistema: Despliega la opción contenida en el menú. 3. Usuario: Selecciona el menú “Requerimientos”. 4. Usuario: Selecciona el menú requerimientos. 5. Sistema: Muestra pantalla de requerimientos pendientes de asignar. 6. Usuario: Escoge el requerimiento y presiona botón asignar. 7. Sistema: Muestra una ventana en la cual consta los detalles del requerimiento; y los empleados a los que lo puede asignar. 8. Usuario: Selecciona el empleado y presiona el botón asignar. 9. Sistema: Muestra la pantalla de requerimientos pendientes en la cual ya no está el requerimiento que fue asignado. 			
Flujo Alternativo:			
No aplica			
Post-condiciones:			
Guarda los cambios realizados según la opción seleccionada.			

Tabla 33- CU-07 Asignar requerimientos.

Caso de uso		Administrar Módulos	
ID		CU-08	
Escenario	E7	Actor	Administrador
Breve descripción:			
El administrador puede visualizar, crear y activar módulos que pertenecen al otro sistema.			
Pre-condiciones:			
Tener acceso al sistema con perfil administrador.			
Flujo de eventos:			
Flujo Básico:			
1. Usuario: Selecciona el menú “Operaciones”.			
2. Sistema: Despliega las opciones contenidas en el menú.			
3. Usuario: Selecciona el menú “Módulos”.			
4. Sistema: Muestra los módulos creados en el sistema.			
5. Usuario: Puede activar o desactivar el módulo.			
6. Usuario: Presiona botón nuevo.			
7. Sistema: Muestra una pantalla para crear módulo.			
8. Usuario: Completa los datos mostrados y presiona en el botón guardar.			
9. Sistema: Valida y guarda el módulo.			
Flujo Alternativo:			
1. Sistema: Muestra errores en la creación del módulo en el sistema.			
Post-condiciones:			
Guarda el módulo y muestra en la pantalla de módulos del sistema.			

Tabla 34- CU-08 Administrar módulos.

Caso de uso		Administrar Menús	
ID		CU-09	
Escenario	E8	Actor	Administrador
Breve descripción:			
El administrador puede visualizar, crear y activar menús que pertenecen al otro sistema.			
Pre-condiciones:			
Tener acceso al sistema con perfil administrador.			
Flujo de eventos:			
Flujo Básico:			
<ol style="list-style-type: none"> 1. Usuario: Selecciona el menú “Operaciones”. 2. Sistema: Despliega las opciones contenidas en el menú. 3. Usuario: Selecciona el menú “Menús”. 4. Sistema: Muestra los menús creados en el sistema. 5. Usuario: Puede activar o desactivar el menú. 6. Usuario: Presiona botón nuevo. 7. Sistema: Muestra una pantalla para crear menús. 8. Usuario: Completa los datos mostrados y presiona en el botón guardar. 9. Sistema: Valida y guarda el menú. 			
Flujo Alternativo:			
<ol style="list-style-type: none"> 1. Sistema: Muestra errores en la creación del menú en el sistema. 			
Post-condiciones:			
Guarda el módulo y muestra en la pantalla de módulos del sistema.			

Tabla 35- CU-09 Administrar menús.

Caso de uso		Administrar Submenús	
ID		CU-10	
Escenario	E9	Actor	Administrador
Breve descripción:			
El administrador puede visualizar, crear y activar Submenús que pertenecen al otro sistema.			
Pre-condiciones:			
Tener acceso al sistema con perfil administrador.			
Flujo de eventos:			
Flujo Básico:			
<ol style="list-style-type: none"> 1. Usuario: Selecciona el menú “Operaciones”. 2. Sistema: Despliega las opciones contenidas en el menú. 3. Usuario: Selecciona el menú “Submenús”. 4. Sistema: Muestra los Submenús creados en el sistema. 5. Usuario: Puede activar o desactivar el menú. 6. Usuario: Presiona botón nuevo. 7. Sistema: Muestra una pantalla para crear Submenús. 8. Usuario: Completa los datos mostrados y presiona en el botón guardar. 9. Sistema: Valida y guarda el Submenú. 			
Flujo Alternativo:			
<ol style="list-style-type: none"> 1. Sistema: Muestra errores en la creación de submenú en el sistema. 			
Post-condiciones:			
Guarda el módulo y muestra en la pantalla de módulos del sistema.			

Tabla 36- CU-10 Administrar Submenús.

Caso de uso		Consulta estado requerimiento por cliente	
ID		CU-11	
Escenario	E10	Actor	Administrador
Breve descripción: El administrador puede realizar la consulta de requerimientos por clientes.			
Pre-condiciones: Tener acceso al sistema con perfil administrador.			
Flujo de eventos:			
Flujo Básico:			
<ol style="list-style-type: none"> 1. Usuario: Selecciona el menú “Consultas”. 2. Sistema: Despliega las opciones contenidas en el menú. 3. Usuario: Selecciona el menú “Requerimientos”. 4. Sistema: Muestra todos los requerimientos. 5. Usuario: Seleccionar requerimientos por cliente. 6. Sistema: Muestra requerimientos del cliente seleccionado. 			
Flujo Alternativo:			
<ol style="list-style-type: none"> 1. Sistema: Muestra que no existe requerimientos del cliente seleccionado. 			
Post-condiciones: No aplica			

Tabla 37- CU-11 Consulta estado de requerimiento por cliente.

Caso de uso		Consulta tiempo invertido en la solución de requerimientos por cliente	
ID		CU-12	
Escenario	E11	Actor	Administrador
Breve descripción:			
El administrador puede realizar la consulta de los tiempos que se invirtieron en la solución de requerimientos por clientes.			
Pre-condiciones:			
Tener acceso al sistema con perfil administrador.			
Flujo de eventos:			
Flujo Básico:			
<ol style="list-style-type: none"> 1. Usuario: Selecciona el menú “Consultas”. 2. Sistema: Despliega las opciones contenidas en el menú. 3. Usuario: Selecciona el menú “Tiempos”. 4. Sistema: Muestra todos los tiempos invertidos de los clientes, en la solución de requerimientos. 5. Usuario: Seleccionar requerimientos por cliente. 6. Sistema: Muestra requerimientos del cliente seleccionado con su respectivo tiempo invertido. 			
Flujo Alternativo:			
<ol style="list-style-type: none"> 1. Sistema: Muestra que no existe requerimientos del cliente seleccionado. 			
Post-condiciones:			
No aplica			

Tabla 38- CU-12 Consulta de tiempos invertidos en la solución de requerimiento por cliente.

- Caso de uso del actor empleado

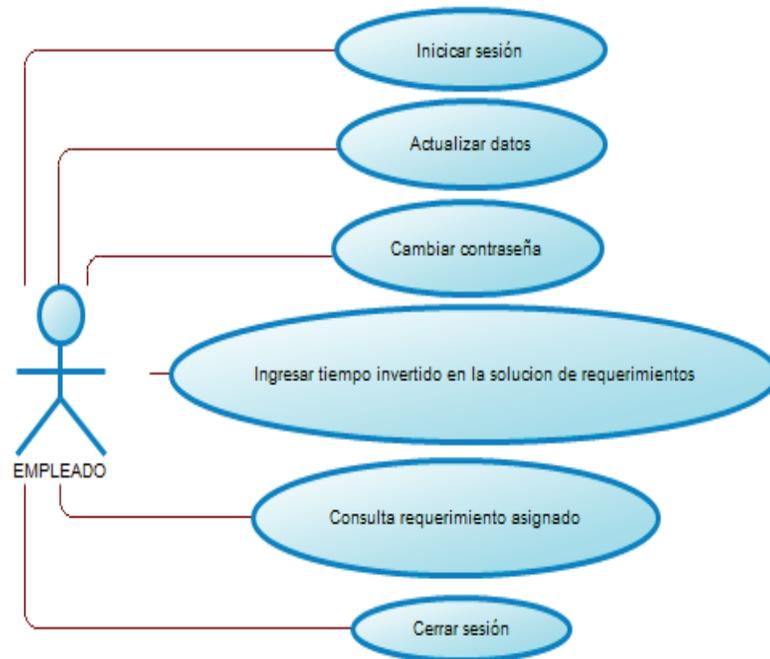


FIGURA 10- Diagrama de caso de uso del rol empleado.

Caso de uso		Iniciar sesión	
ID		CU-13	
Escenario	E13	Actor	Empleado
Breve descripción: El empleado necesita ingresar al sistema para lo cual deberá seleccionar el perfil empleado e ingresar su número de cedula y contraseña.			
Pre-condiciones: Tener una cuenta creada como empleado.			

<p>Flujo de eventos:</p> <p>Flujo Básico:</p> <ol style="list-style-type: none"> 1. Usuario: Selecciona el perfil empleado e ingresa el número de cedula y contraseña, luego presionar el botón ingresar al sistema. 2. Sistema: valida los datos ingresados. 3. Sistema: Muestra la pantalla principal del perfil empleado. <p>Flujo Alternativo:</p> <ol style="list-style-type: none"> 1. Sistema: Presenta un mensaje indicando que no existe un usuario con los datos ingresados.
<p>Post-condiciones:</p> <p>Almacena el id de sesión en el sistema en forma temporal.</p>

Tabla 39- CU-13 Iniciar sesión empleado.

Caso de uso		Ingresar tiempo invertido en la solución del requerimiento	
ID		CU-14	
Escenario	E16	Actor	Empleado
<p>Breve descripción:</p> <p>El empleado después de realizar el requerimiento tendrá que ingresar el tiempo que invirtió en la solución.</p>			
<p>Pre-condiciones:</p> <p>Tener acceso al sistema con perfil empleado.</p>			

Flujo de eventos:	
Flujo Básico:	
<ol style="list-style-type: none"> 1. Usuario: Selecciona el menú “Operaciones”. 2. Sistema: Despliega las opciones contenidas en el menú. 3. Usuario: Selecciona el menú “Ingreso Tiempos”. 4. Sistema: Muestra la pantalla con los clientes de los requerimientos que se le asignaron. 5. Usuario: Seleccionar el cliente. 6. Sistema: Muestra una pantalla con todos los requerimientos asignados pertenecientes al cliente seleccionado. 7. Usuario: Selecciona el requerimiento. 8. Sistema: Muestra una pantalla con campos que se requiere que sean ingresados. 9. Usuario: Completa los datos requeridos y presiona en el botón registrar tiempo. 10. Sistema: Guardar el tiempo con sus respectivos datos. 	
Flujo Alternativo:	
No aplica	
Post-condiciones:	
No aplica	

Tabla 40- CU-14 Ingreso de tiempo invertido en la solución del requerimiento por cliente.

Caso de uso		Consultar requerimiento asignado	
ID		CU-15	
Escenario	E17	Actor	Empleado
Breve descripción:			
El empleado tendrá la posibilidad de consultar los requerimientos que le fueron asignados.			
Pre-condiciones:			
Tener acceso al sistema con perfil empleado.			

<p>Flujo de eventos:</p> <p>Flujo Básico:</p> <ol style="list-style-type: none"> 1. Usuario: Selecciona el menú “Consultas”. 2. Sistema: Despliega las opciones contenidas en el menú. 3. Usuario: Selecciona el menú “Requerimientos”. 4. Sistema: Muestra la pantalla con todos los requerimientos asignados. <p>Flujo Alternativo:</p> <p>No aplica</p>
<p>Post-condiciones:</p> <p>No aplica</p>

Tabla 41- CU-15 Consulta requerimientos asignados.

- **Caso de uso del actor cliente.**

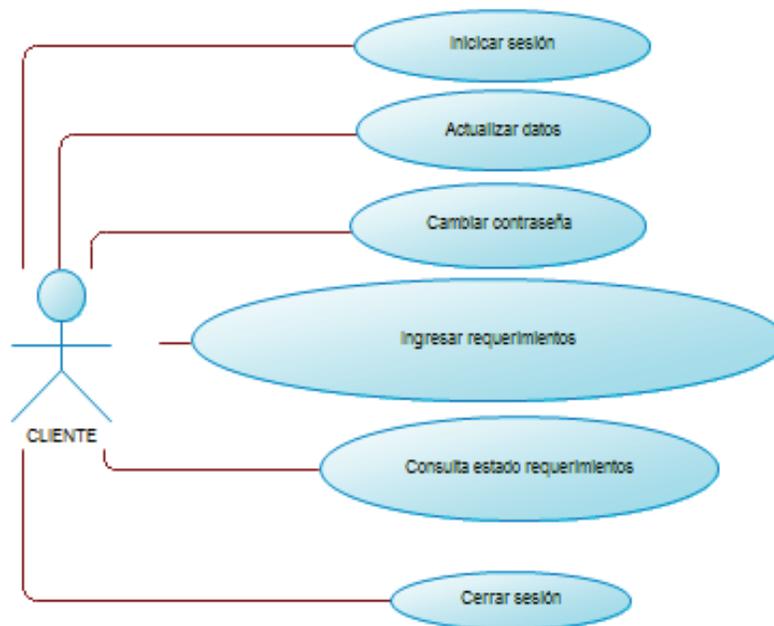


FIGURA 11- Diagrama del caso de uso del rol cliente.

Caso de uso		Iniciar sesión	
ID		CU-16	
Escenario	E19	Actor	Cliente
Breve descripción: El cliente necesita ingresar al sistema para lo cual deberá seleccionar el perfil cliente e ingresar el Ruc y su contraseña.			
Pre-condiciones: Tener una cuenta creada como cliente.			
Flujo de eventos:			
Flujo Básico:			
<ol style="list-style-type: none"> 1. Usuario: Selecciona el perfil cliente e ingresa el Ruc y contraseña, luego presionar en el botón ingresar al sistema. 2. Sistema: valida los datos ingresados. 3. Sistema: Muestra la pantalla principal del perfil cliente. 			
Flujo Alternativo:			
<ol style="list-style-type: none"> 1. Sistema: Presenta un mensaje indicando que no existe un usuario con los datos ingresados. 			
Post-condiciones: Almacena el id de sesión en el sistema en forma temporal.			

Tabla 42- CU-16 Iniciar sesión cliente.

Caso de uso		Ingresar requerimientos	
ID		CU-17	
Escenario	E22	Actor	Cliente
Breve descripción: El empleado deberá ingresar los requerimientos para su posterior solución.			
Pre-condiciones: Tener acceso al sistema con perfil cliente.			
Flujo de eventos:			
Flujo Básico:			
<ol style="list-style-type: none"> 1. Usuario: Selecciona el menú “Operación”. 2. Sistema: Despliega las opciones contenidas en el menú. 3. Usuario: Selecciona el menú “Ingreso Requerimiento”. 4. Sistema: Muestra la pantalla con todos los requerimientos pendientes. 5. Usuario: Selecciona el botón nuevo. 6. Sistema: Muestra una pantalla en las cuales debe ingresar el modulo, menú, submenú al que hace referencia el requerimiento y un detalle explicando la razón del requerimiento. 7. Usuario: Completa los campos requeridos y presiona el botón guardar. 8. Sistema: Guardar el requerimiento con su respectivos datos. 			
Flujo Alternativo:			
No aplica			
Post-condiciones: Muestra el requerimiento ingresado en la pantalla de requerimientos pendientes.			

Tabla 43- CU-17 Ingreso requerimientos.

Caso de uso		Consultar estado del requerimiento	
ID		CU-18	
Escenario	E23	Actor	Empleado
Breve descripción: El empleado tendrá la posibilidad de consultar los estados de los requerimientos.			
Pre-condiciones: Tener acceso al sistema con perfil empleado.			
Flujo de eventos:			
Flujo Básico:			
<ol style="list-style-type: none"> 1. Usuario: Selecciona el menú “Consultas”. 2. Sistema: Despliega las opciones contenidas en el menú. 3. Usuario: Selecciona el menú “Requerimientos”. 4. Sistema: Muestra la pantalla con todos los requerimientos ingresados con los respectivos estados. 			
Flujo Alternativo:			
No aplica			
Post-condiciones:			
No aplica			

Tabla 44- CU-18 Consulta estado de requerimientos.

3.3. DISEÑO DE BASE DE DATOS

En la FIGURA 12- Diseño de la base de datos., se muestra las diferentes tablas y relaciones que se han creado para realizar el desarrollo del sistema web.

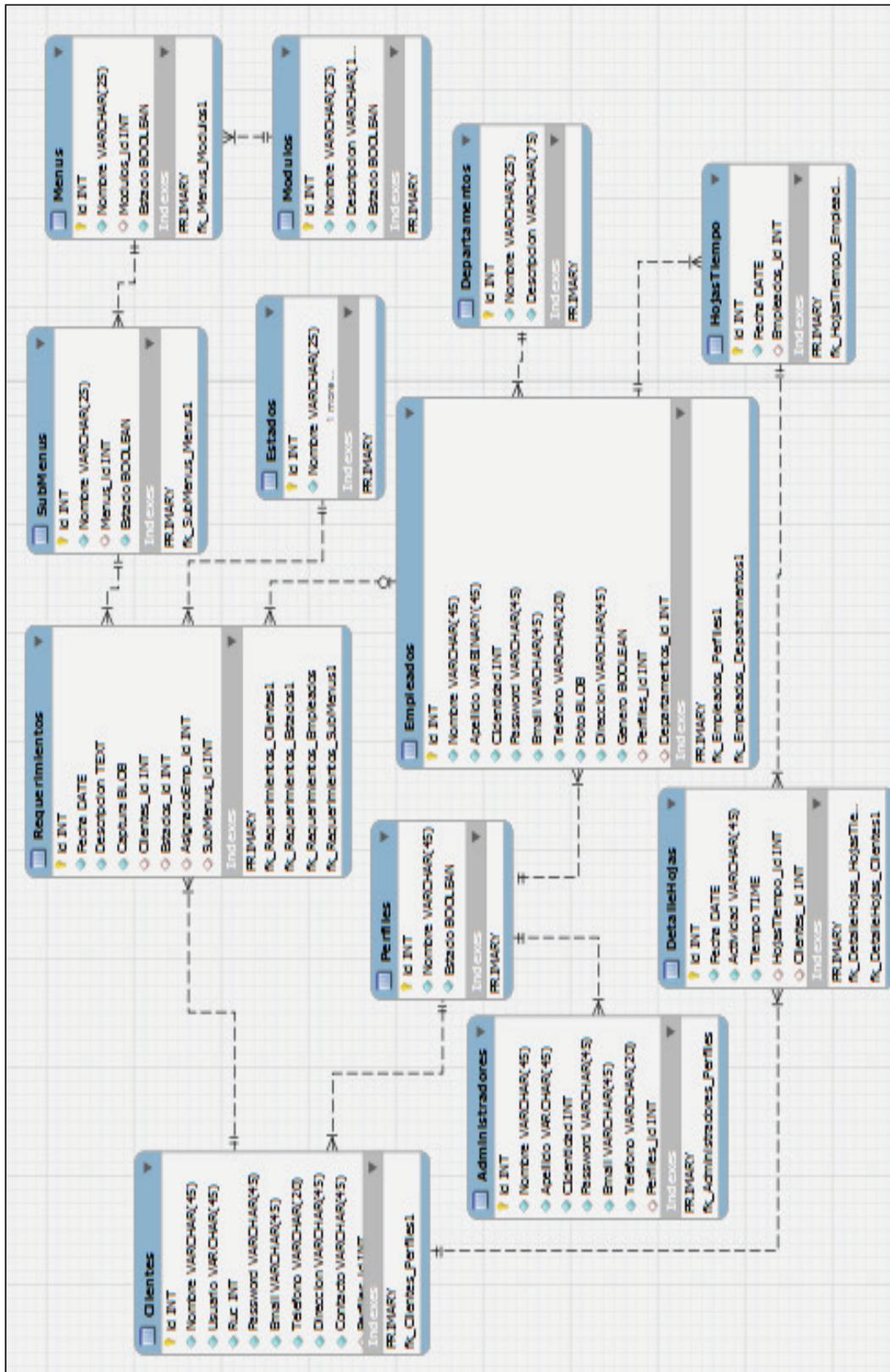


FIGURA 12- Diseño de la base de datos.

3.4. DISEÑO DE NAVEGACIONAL

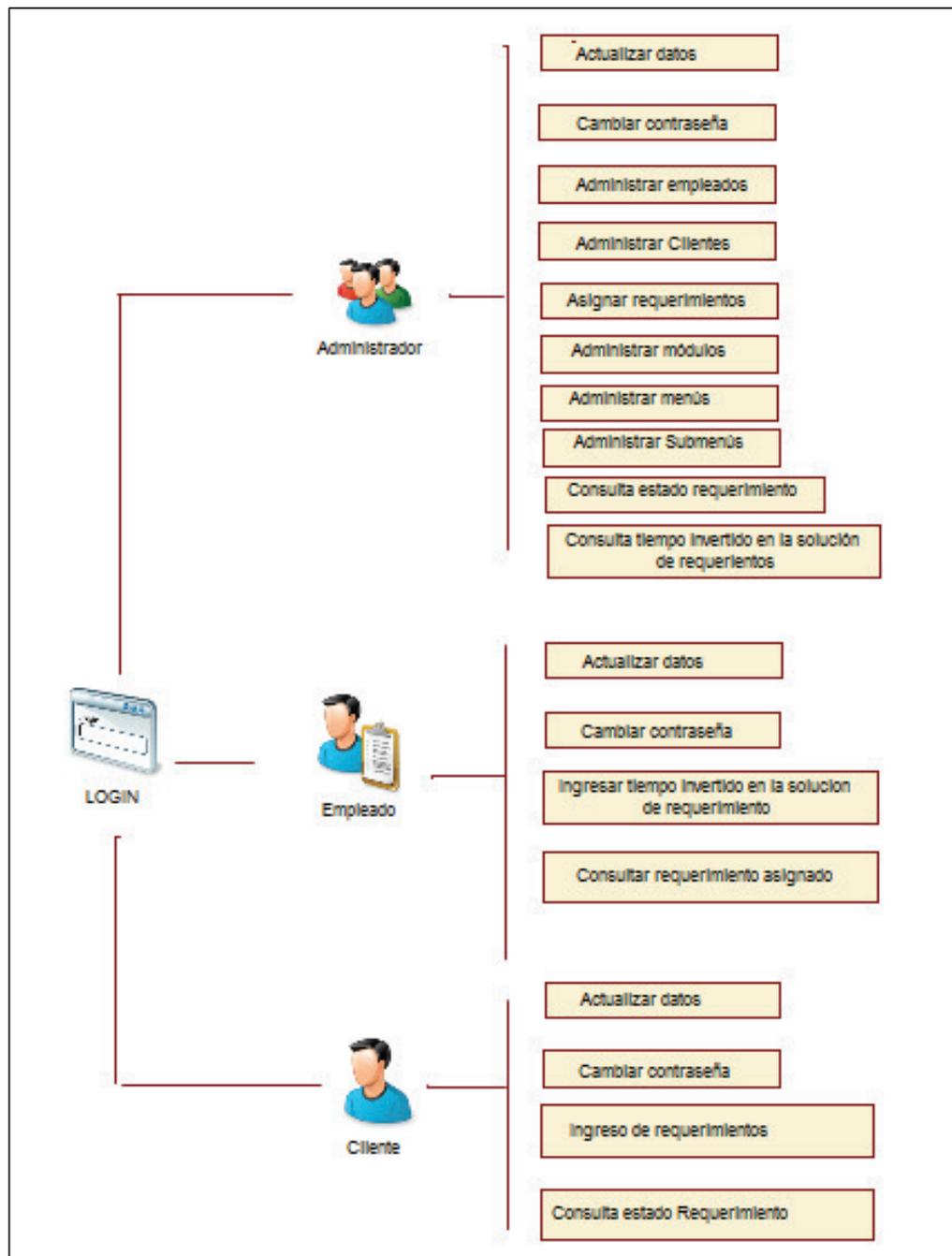


FIGURA 13- Diseño navegacional del sistema.

3.5. DISEÑO DE LA APLICACIÓN WEB

Para el diseño se ha utilizado una herramienta de maquetación para nuestro sitio web, como es NinjaMock. (<https://ninjamock.com/>).

3.5.1. DISEÑO DEL INICIO SESIÓN DEL SISTEMA

El modelo que se presenta en la FIGURA 14- Estructura inicio sesión, pertenece a la estructura que se implementará en la interfaz de inicio de sesión.

The image shows a web browser window with a login form. The browser's address bar contains 'http://'. The form is titled 'Inicio de Sesión' and features a dropdown menu with three options: 'ADMINISTRADOR', 'CLIENTE', and 'EMPLEADO'. Below the dropdown are two input fields labeled 'Cédula/Doc' and 'Password'. At the bottom of the form is a button labeled 'INGRESAR AL SISTEMA' with a checkmark icon. The footer of the page reads: 'Desarrollado por Alvaro Tiso G. 1 Gammasoft Todos los derechos reservados, Z91C'.

FIGURA 14- Estructura inicio sesión

3.5.2. DISEÑO PRINCIPAL DEL SISTEMA

Para el diseño principal se ha realizado una plantilla base que sirve como página principal para perfiles de los usuarios que existe en el sistema como son: Administrador, Empleado, Cliente. En la FIGURA 15- Diseño plantilla base se puede observar cómo se identificarán los diferentes bloques de la página web.

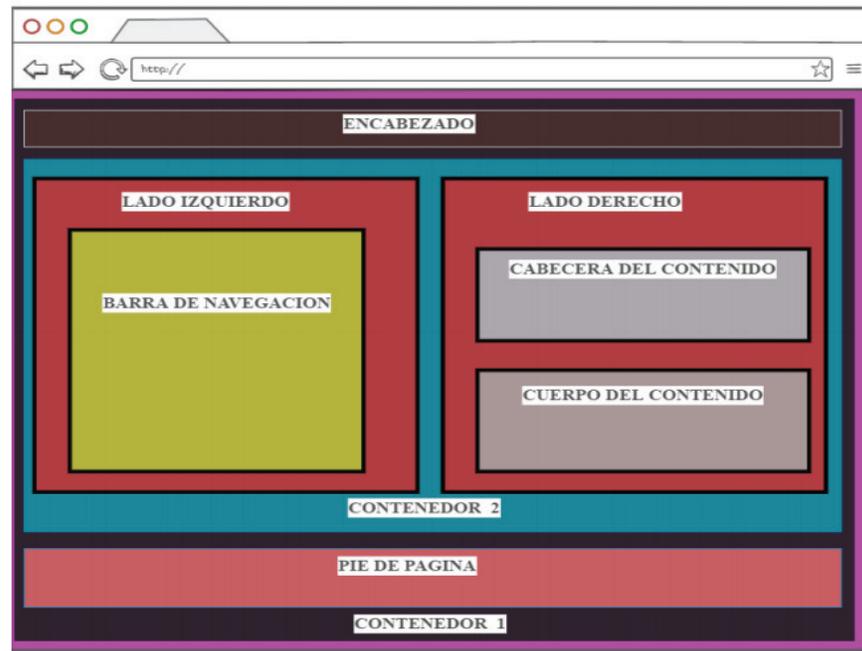


FIGURA 15- Diseño plantilla base

3.6. DESARROLLO DEL SISTEMA

Para el desarrollo del sistema web se ha utilizado el framework Symfony, ya que está diseñado para la optimización de tiempo y recursos.

Symfony está basada en una arquitectura MVC que se representa por capas como son el modelo, vista y controlador.

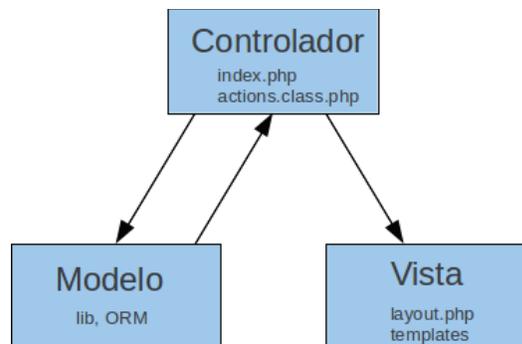


FIGURA 16- Relación que existe con elementos MVC.
(Rodrigues, 2016)

3.6.1. COMPONENTES QUE FORMAN EL PROYECTO

En el desarrollo de la aplicación se han empleado diferentes elementos que son de mucha importancia al momento del desarrollo del sistema como son:

- El Controlador.
- Doctrine.
- Twig.
- Sistema de enrutamiento.
- Bundle.

Los cuales fueron explicados en el marco teórico SYMFONY.

3.6.2. ESTRUCTURA DE DIRECTORIOS DEL PROYECTO

Para el desarrollo del sistema web se ha creado una estructura de directorios que ayudará a organizar el proyecto como se observa en la FIGURA 17- Estructura de directorios del proyecto.

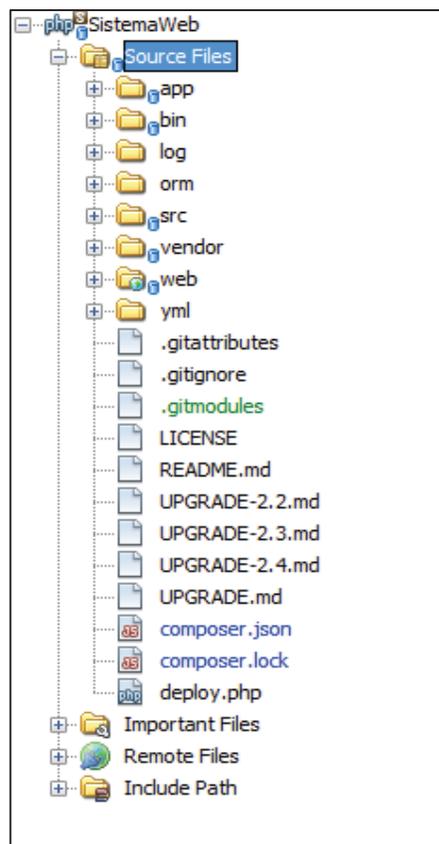


FIGURA 17- Estructura de directorios del proyecto.

En la siguiente sección se muestra los directorios de mayor relevancia que contiene nuestro sistema:

- **app:** Es la que contiene la configuración del sistema.
- **src:** Es la parte donde está el código PHP de nuestra aplicación.
- **vendor:** En esta carpeta se encuentran los archivos del framework y librerías de terceros.
- **web:** En esta carpeta están los archivos públicos del sistema como son los CSS, JavaScript e imágenes.

3.6.3. CONFIGURACIÓN DEL SISTEMA

La configuración del sistema se la realizó en el directorio app, modificando los siguientes archivos:

- **AppKernel.php:** Es un archivo PHP que es el punto de entrada de la aplicación, donde se registran los parámetros de configuración como son los bundles.
- **Parameters.yml:** Es un archivo de extensión yml, se ubica en la carpeta config del directorio app, sirve para la configuración de la base de datos.

3.6.4. CREACIÓN DE BUNDLE

En el desarrollo del sistema se ha creado un bundle de nombre firstBundle que está contenida en el directorio /src/Login de la estructura de la estructura de directorios.

Para la creación de un bundle se ha utilizado el siguiente comando:

```
php app/console generate:bundle Login/firstBundle
```

Este Bundle contendrá todos los elementos de nuestro Sistema web. Al momento de ejecutar el comando se crean diferentes directorios dentro de nuestro bundle como se puede observar en la FIGURA 18- Estructura de directorios del Bundle.

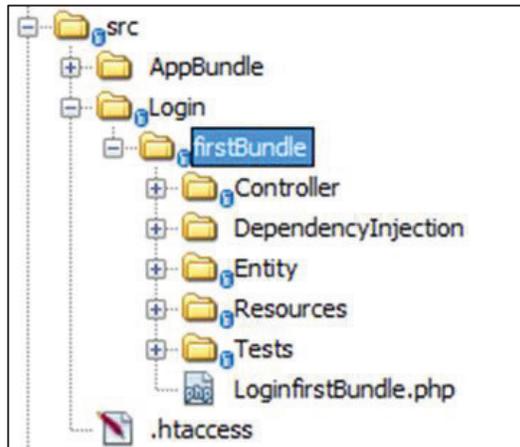


FIGURA 18- Estructura de directorios del Bundle.

3.6.5. CREACIÓN DE ENTIDADES

La creación de entidades es muy importante, ya que esto es el punto de partida para el desarrollo de los controladores, estas entidades son utilizadas para realizar diferentes acciones en la base de datos del sistema. Para generar las entidades a partir del modelo de base de datos del sistema, se puede utilizar doctrine a través de los siguientes comandos:

Primeramente, se importa los datos aplicando el siguiente comando.

```
php app/console doctrine:mapping:import
    LoginfirstBundle yml
```

Segundo se genera las entidades aplicando el comando

```
php app/console doctrine:Generate:entities
    LoginfirstBundle
```

En la FIGURA 19- Entidades de sistema. Se muestra como se ha generado las entidades de nuestro sistema.

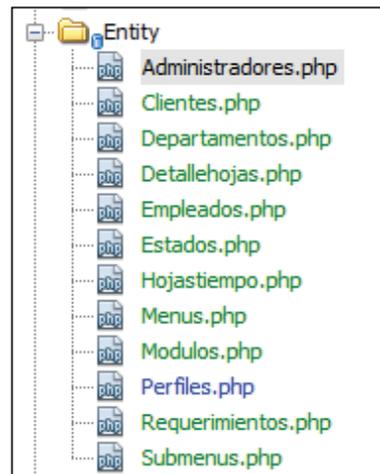


FIGURA 19- Entidades de sistema.

3.6.6. CREACIÓN DE CONTROLADORES

Para la creación de controladores se ha utilizado el comando:

```
php app/console generate:controller
LoginfirstBundle/ nombredel controlador
```

Para el desarrollo del sistema se ha creado los controladores que se muestran en FIGURA 20- Controladores del sistema.

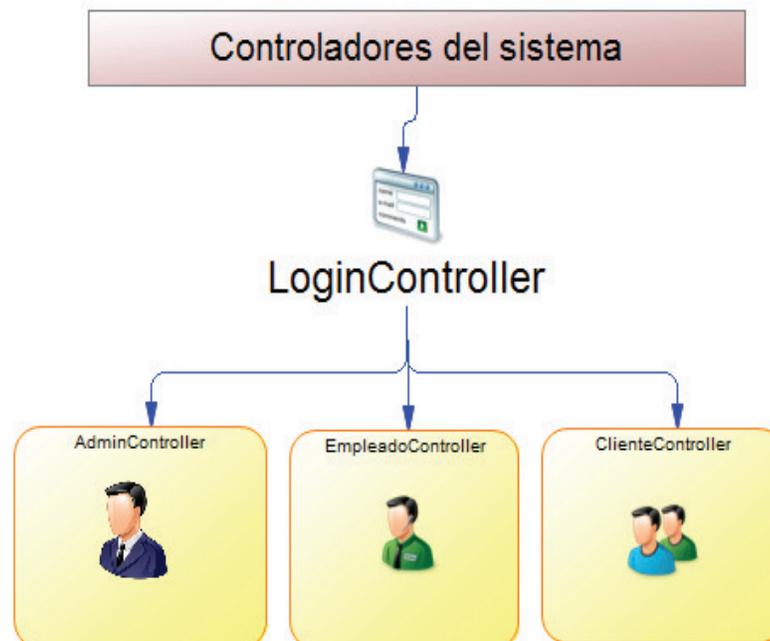


FIGURA 20- Controladores del sistema.

LoginController

Este controlador se ha definido como punto de seguridad del sistema, la cual consta de dos funcionalidades como son:

- **Inicio de sesión**

Es una función que permite el ingreso al sistema en sus diferentes perfiles de usuarios.

- **Cerrar sesión**

Es una función la cual permite cerrar la sesión del usuario ingresado al sistema.

AdminController

En este controlador se ha definido las diferentes acciones que puede realizar el administrador del sistema como son:

- Actualizar datos.
- Cambiar contraseña.
- Administrar empleados.
- Administrar clientes.
- Administrar módulos.
- Administrar menús.
- Administrar submenús.
- Asignar requerimientos.
- Consultar estado del requerimiento por cliente.
- Consultar tiempo invertido en la solución de requerimientos.

EmpleadoController

En este controlador se definen las acciones que puede realizar el empleado en el sistema como son:

- Actualizar datos.
- Cambiar contraseña.
- Ingresar tiempos invertidos en la solución de requerimientos.
- Consultar Requerimientos asignados.

ClienteController

En este controlador se definen las acciones que puede realizar el Cliente en el sistema como son:

- Actualizar datos.
- Cambiar contraseña.
- Ingresar requerimientos.
- Consultar Requerimientos.

3.6.7. CREACIÓN DE LA VISTA

Para la creación de la vista se ha utilizado el generador de plantillas Twig, para lo cual se ha generado una estructura de directorios que se muestra en la FIGURA 21- Estructura de directorios Twig.

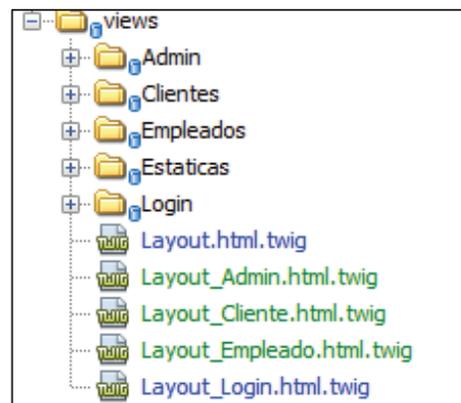


FIGURA 21- Estructura de directorios Twig.

La utilización de Twig permite la reutilización de código ya sea HTML o PHP ya que algunas páginas de nuestro sistema tienen el mismo aspecto no es necesario repetir código, permite tener código limpio.

Para esto se utiliza una plantilla base que se utilizará para todas las páginas de nuestro sistema. Esta plantilla está formada por bloques que permiten tener una herencia de plantillas.

- **BLOQUES EN TWIG**

Los bloques son elementos comunes que tiene la plantilla base y a su vez pueden ser reemplazadas por plantillas hijas.

Para el desarrollo del sistema se ha creado una plantilla base que permite la reutilización de código, estos trozos de código se definen bloques. La plantilla base Twig se ha codificado de acuerdo a la FIGURA 15- Diseño plantilla base con los bloques que corresponden esta representación se puede observar en la FIGURA 23- Código fuente plantilla base Twig.

- **PLANTILLA BASE TWIG**

La plantilla base se define en el directorio app/Resources como se puede ver en la FIGURA 22- Directorio plantilla base Twig.



FIGURA 22- Directorio plantilla base Twig.

```

{%block body %}
    {%block content1 %}
        {% block header %}
            Encabezado
        {% endblock %}
        {%block content2 %}|
            {% block asideLeft %}
                Lado Izquierda
                {%block section1 %}
                    Barra navegacion
                {%endblock%}
            {% endblock %}
            {% block asideRight %}
                Lado Derecho
                {%block section2 %}
                    content header
                {%endblock%}
                {%block section3 %}
                    content
                {%endblock%}
            {% endblock %}
        {%endblock%}
        {% block Footer %}
            pie de pagina
        {% endblock%}
    {% endblock %}
    {% block javascripts %}
    {% endblock %}
{% endblock %}

```

FIGURA 23- Código fuente plantilla base Twig.

La plantilla base creada se une con un Layout que no es más que un decorador de plantillas, para así formar el archivo Twig final. Esto se puede ver en la FIGURA 24- Integración de Layout con plantilla base.

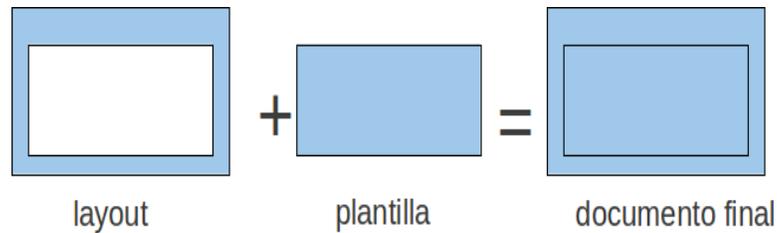


FIGURA 24- Integración de Layout con plantilla base.
(Rodrigues, 2016)

Para la vista se han creado los siguientes directorios y archivos que se pueden ver en Twig. Estos representan las interfaces de usuario por las clases Twig, archivos js, archivos CSS y formularios necesarios para la interacción con el cliente.

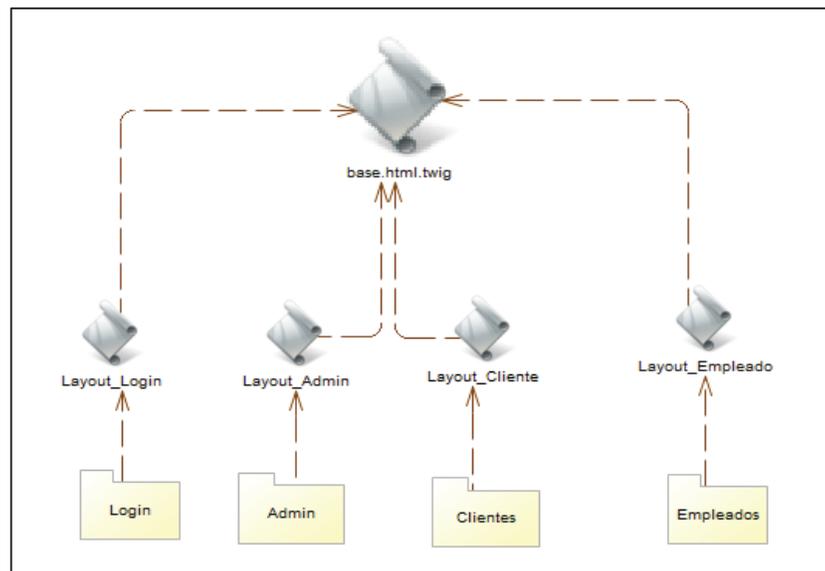


FIGURA 25- Archivos y directorios de la vista Twig.

Las carpetas que se muestran en la FIGURA 25- Archivos y directorios de la vista Twig., contienen los archivos finales Twig de nuestra aplicación.

También se ha creado directorios que contendrán imágenes, hojas de estilo, archivos js que ayudan a tener páginas dinámicas e interactivas

en nuestro sistema web. FIGURA 26- Directorio públicos de nuestro sistema.

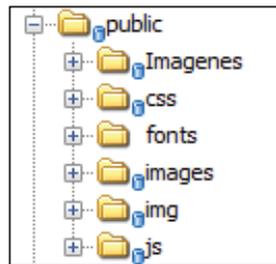


FIGURA 26- Directorio públicos de nuestro sistema.

• BOOTSTRAP EN EL PROYECTO

Para el desarrollo del sistema se utilizó Bootstrap ya que es una herramienta que contiene código CSS y JavaScript prediseñados que ayudan mucho al momento de realizar el diseño.

Para la utilización de Bootstrap se descargó los archivos necesarios como se muestra en la FIGURA 27- Archivos Bootstrap, estos en nuestro proyecto se encuentran ubicados en el directorio public de nuestro bundle.

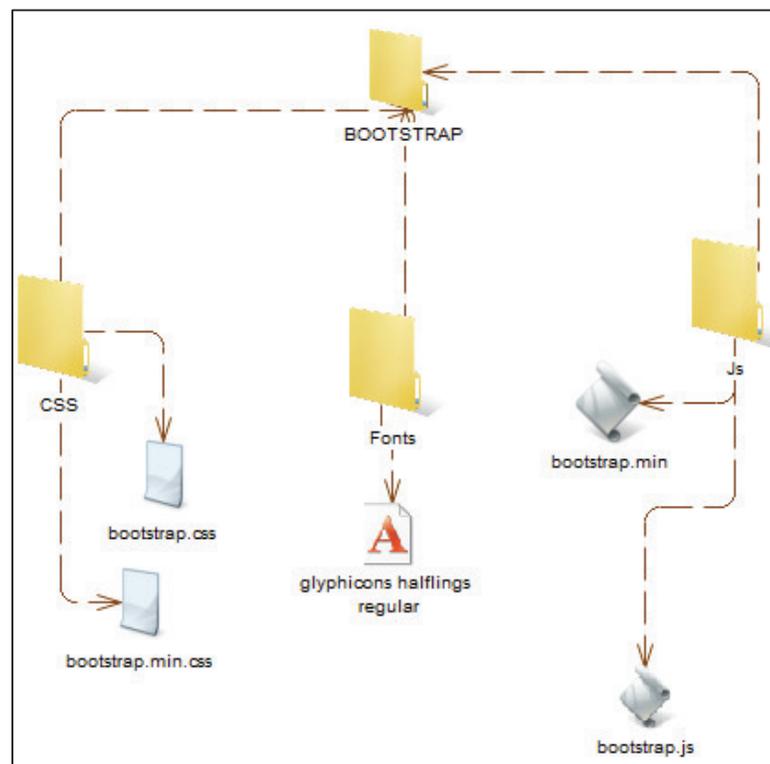


FIGURA 27- Archivos Bootstrap

3.7. IMPLEMENTACIÓN

Una vez terminadas las etapas del desarrollo de un sistema de información, llega el momento de la implementación:

Para la instalación es necesario disponer un entorno adecuado para que funcione el sistema tanto a nivel de hardware como software.

3.7.1. REQUISITOS PREVIOS PARA LA IMPLEMENTACIÓN

- **HARDWARE**

El servidor que alojará el sistema debe cumplir con las siguientes características para su funcionamiento como son:

HARDWARE
<ul style="list-style-type: none"> • Procesador Intel Core i5-3317U CPU 1.70GHZ. • Memoria RAM 4Gb. • Tarjeta de red. • Tarjeta de video. • Monitor de 19 pulgadas. • Accesorios como son: mouse, teclado.

Tabla 45- Requerimientos mínimos hardware para la implementación del sistema.

- **SOFTWARE**

SOFTWARE
<ul style="list-style-type: none"> • Sistema operativo Windows 7. • XAMPP. <ul style="list-style-type: none"> ○ Apache. ○ MYSQL. ○ PHP. • Composer.

Tabla 46- Requerimientos mínimos de software para la implementación del sistema.

Para que el sistema pueda ser accedido desde internet, el servidor debe contar con una IP pública la cual debe tener asociado un nombre de dominio.

- **INSTALACIÓN DE XAMPP**

Se realiza la instalación de XAMPP para posteriormente levantar los servicios de Apache y MySQL.

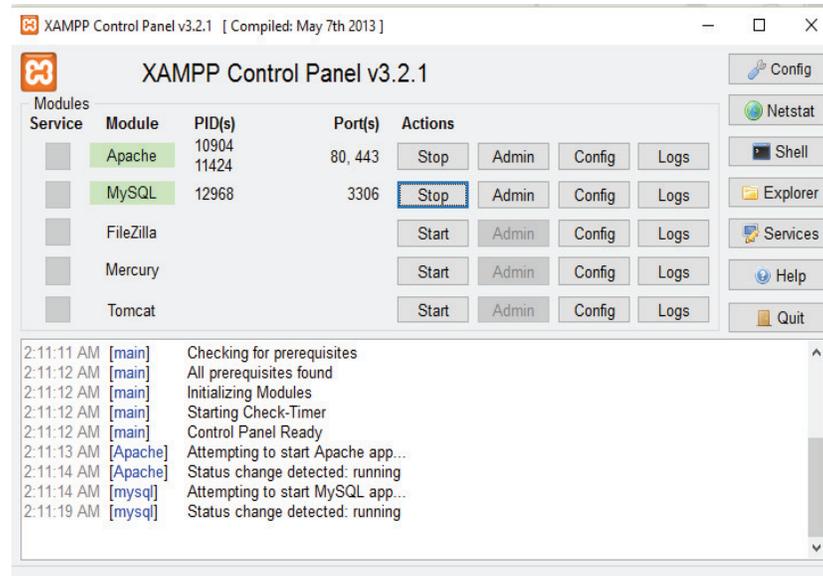


FIGURA 28- Instalación de XAMPP.

- **PRUEBA DEL SERVIDOR APACHE**

Se realiza la prueba de funcionamiento del servidor web Apache ejecutando <http://localhost/dashboard/>.



FIGURA 29- Prueba de funcionamiento servidor web Apache.

3.7.2. INTERFACES DEL SISTEMA

En base a los diseños descritos en el punto diseño del sistema, se han implementado las interfaces para el sistema, la información que se debe mostrar de acuerdo al perfil de usuario. Estas interfaces se muestran a continuación.

- **PÁGINA DE INGRESO AL SISTEMA**



FIGURA 30- Página de ingreso al sistema.

- **PÁGINA PRINCIPAL DEL USUARIO ADMINISTRADOR**



FIGURA 31- Página de principal Administrador.

- **PÁGINA PARA ACTUALIZAR DATOS DEL USUARIO ADMINISTRADOR**

Actualizar Información Administrador

Nombres: Alvaro David

Apellidos: Turo Guayta

Teléfono: 02699982

Correo Electrónico: alvaro@gmail.com.ec

Cédula Identidad/Usuario: 0510263903

Guardar

FIGURA 32- Página para actualizar datos del administrador.

- **PÁGINA DE ASIGNACIÓN DE REQUERIMIENTOS AL EMPLEADO**

Asignación Requerimiento

Id Requerimiento: Rq-3

Módulo: Contabilidad

Menu: Movimientos

Submenu: Ingreso Diarios

Descripción: Prueba de funcionamiento del SRT

Empleado: Seleccione Empleado

Cancelar Asignar

#	Fecha	Id Requerimiento	Estado	Módulo	Asignación
1	2016/04/20	Rq-3	Pendiente	Contabilidad	Asignación
2	2016/04/20	Rq-5	Pendiente	Comedor	Asignación

FIGURA 33- Página de asignación de requerimientos.

- **PÁGINA PRINCIPAL DEL PERFIL DE USUARIO EMPLEADO**



FIGURA 34- Página de principal Empleados.

- **PÁGINA INGRESO TIEMPOS INVERTIDOS EN LA SOLUCIÓN DE REQUERIMIENTOS**

The screenshot shows a web application interface for time entry. At the top left is the logo for 'GAMEDA Soft'. A dark red navigation bar contains a hamburger menu icon and the name 'Alvaro'. Below this is a dark sidebar with the text 'Bienvenido Alvaro' and a 'MENU PRINCIPAL' section containing links for 'Mi Usuario', 'Operaciones', and 'Consultas'. The main content area has a white background with the heading 'INICIO / Registro tiempos' and a blue header 'Registrar Tiempos'. Below this are two dropdown menus: 'Seleccione la Empresa:' and 'Seleccionar el requerimiento:'. Below these are two input fields: 'Ingrese el Tiempo:' with the value 'Tiempo' and 'Ingrese la actividad:'. A green 'Guardar' button is at the bottom.

FIGURA 35- Página ingreso tiempos invertidos en la solución de requerimientos.

- **PÁGINA PRINCIPAL DEL USUARIO CLIENTE**

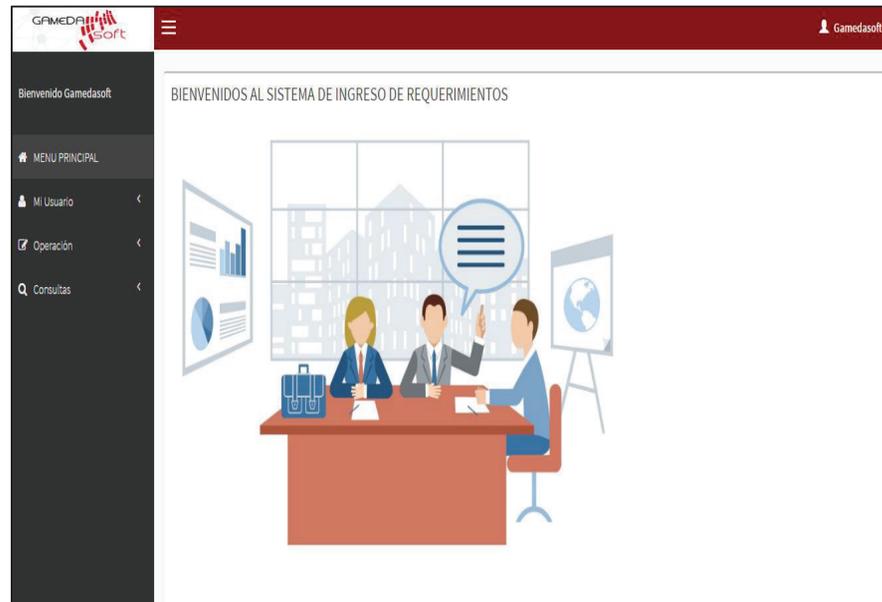


FIGURA 36- Página de principal cliente.

- **PÁGINA PARA EL INGRESO DE REQUERIMIENTOS**

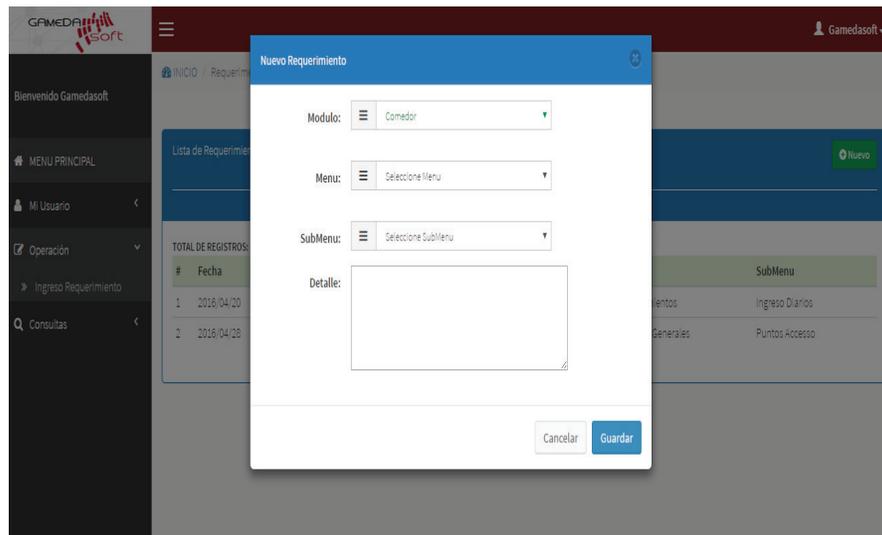


FIGURA 37- Página de ingreso de requerimientos.

3.8. PRUEBAS DE FUNCIONAMIENTO

Finalizada la etapa de implementación se procedió a realizar las pruebas del sistema para garantizar la calidad del producto. La FIGURA 38- Esquema de

pruebas del sistema. muestra un esquema donde se identifican los diferentes tipos de pruebas.

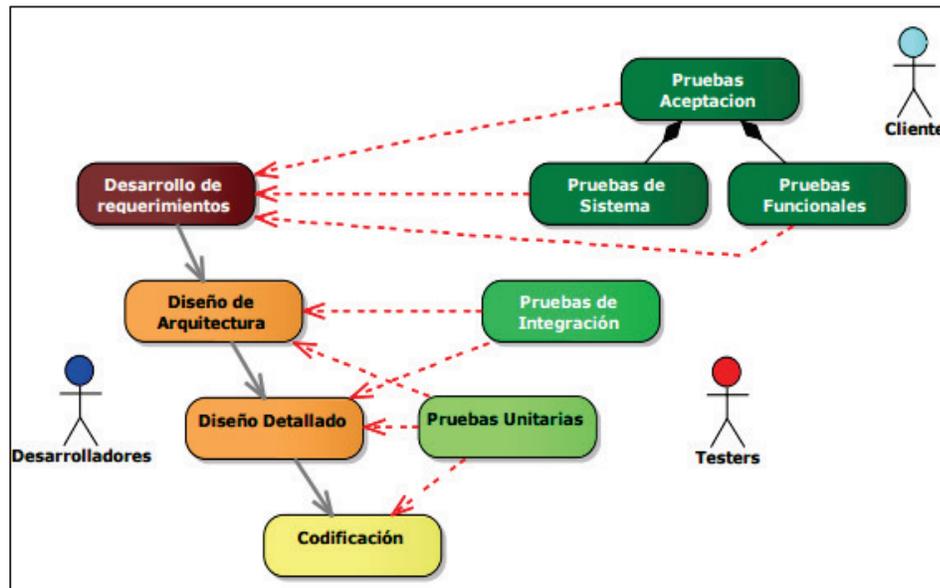


FIGURA 38- Esquema de pruebas del sistema.
(Myers, 2004)

3.8.1. PRUEBAS DE ACEPTACIÓN

Este tipo de pruebas se realiza a partir de los casos de uso definidos en la obtención de requerimientos para el desarrollo del sistema. Las pruebas de aceptación ayudan a descubrir errores funcionales, rendimiento y seguridad del sistema web. Para realizar las pruebas de aceptación se va seguir el flujo básico que realiza el sistema que se muestra en la FIGURA 39- Flujo principal del sistema.



FIGURA 39- Flujo principal del sistema.

TABLAS DE PRUEBAS DE ACEPTACIÓN DEL USUARIO CLIENTE.

- **PRUEBAS DE ACEPTACIÓN INICIO SESIÓN**

Prueba de inicio de sesión cliente	
Id	P.A.-01
Caso de uso	CU-16
Descripción: El cliente procede a seleccionar el perfil cliente e ingresa su número de Ruc y contraseña.	
Pasos de ejecución: 1. Seleccionar el perfil cliente. 2. Ingresar su número de Ruc. 3. Ingresar contraseña. 4. Presione el botón “Ingresar al sistema”.	
Resultado esperado: Que el sistema reconozca al usuario que ha iniciado la sesión.	Resultado obtenido: Se Ingresó al sistema con éxito y se visualiza el nombre del usuario.
Evaluación de la prueba: Se mostró el resultado esperado.	

Tabla 47- P.A.-01-Prueba de inicio de sesión cliente

Visualización en la interfaz del sistema

The screenshot displays a login form titled "Inicio de Sesión". It features a dropdown menu labeled "CLIENTE" with a downward arrow. Below it, the RUC number "1712345678001" is highlighted in yellow. Underneath is a password field with masked characters "*****". At the bottom, there is a red button with a white arrow icon and the text "INGRESAR AL SISTEMA".

FIGURA 40- Iniciar sesión cliente.

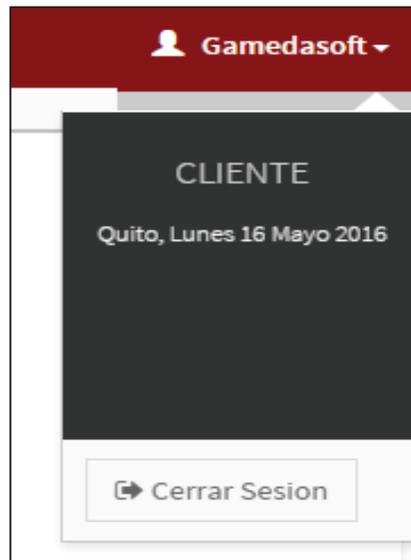


FIGURA 41- Prueba inicio sesión cliente.

• **PRUEBAS DE ACEPTACIÓN INGRESO DE REQUERIMIENTO**

Prueba de Ingreso requerimiento	
Id	P.A.-02
Caso de uso	CU-17
Descripción: El cliente procede a ingresar el requerimiento para su posterior solución.	
Pasos de ejecución: 1. Seleccionar la opción ingresar requerimiento en el menú “Operación”. 2. Presionar el botón nuevo. 3. Completar los campos requeridos. 4. Presione el botón “Guardar”.	
Resultado esperado: Que el sistema registre el requerimiento	Resultado obtenido: Se guardó el requerimiento.
Evaluación de la prueba:	

Se mostró el resultado esperado.

Tabla 48- P.A.-02-Prueba de Ingreso requerimiento.

Visualización en la interfaz del sistema

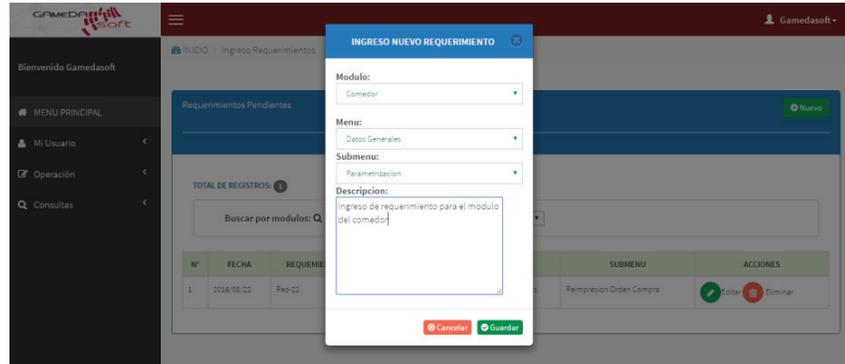


FIGURA 42- Ingreso de requerimiento.

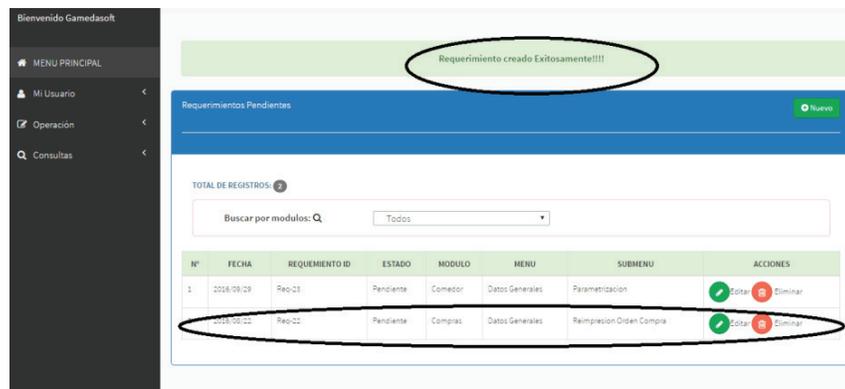


FIGURA 43- Registro de requerimiento.

TABLAS DE PRUEBAS DE ACEPTACIÓN DEL USUARIO ADMINISTRADOR.

• PRUEBAS DE ACEPTACIÓN INICIO SESIÓN

Prueba de inicio de sesión administrador	
Id	P.A.-03
Caso de uso	CU-04
Descripción:	

El administrador procede a seleccionar el perfil administrador e ingresar su número de cedula y contraseña.	
Pasos de ejecución:	
5. Seleccionar el perfil administrador.	
6. Ingresar su número de cédula.	
7. Ingresar contraseña.	
8. Presione el botón “Ingresar al sistema”.	
Resultado esperado:	Resultado obtenido:
Que el sistema reconozca al usuario que ha iniciado la sesión.	Se Ingresó al sistema con éxito y se visualiza el nombre del usuario.
Evaluación de la prueba:	
Se mostró el resultado esperado.	

Tabla 49- P.A.-04- Prueba de inicio de sesión Administrador.

Visualización en la interfaz del sistema

FIGURA 44- Inicio sesión Administrador.

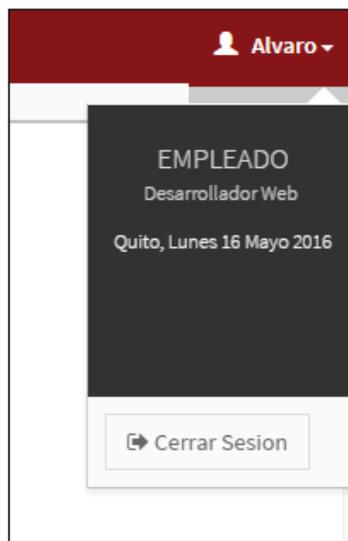


FIGURA 45- Prueba de inicio de sesión Administrador.

• **PRUEBAS DE ACEPTACIÓN ASIGNACIÓN DE REQUERIMIENTO**

Prueba de aceptación asignación de requerimiento	
Id	P.A.-04
Caso de uso	CU-07
Descripción: El administrador procede a asignar el requerimiento al empleado el requerimiento registrado por el cliente.	
Pasos de ejecución: 1. Seleccionar la opción requerimientos del menú “Asignación Tareas”. 2. Seleccionar el requerimiento a asignar. 3. Presionar el botón asignar. 4. Seleccionar el empleado. 5. Presionar el botón Asignar.	
Resultado esperado: Que el requerimiento se asigne a un empleado.	Resultado obtenido: Se asignó el requerimiento al empleado seleccionado.
Evaluación de la prueba: Se mostró el resultado esperado.	

Tabla 50- P.A.-04- Prueba de aceptación asignación de requerimiento.

Visualización en la interfaz del sistema

BIENVENIDO Alvaro David

INICIO / Requerimientos

Requerimientos pendientes por asignar

TOTAL DE REGISTROS: 2

N°	FECHA	REQUERIMIENTO ID	CLIENTE	ESTADO	MODULO	ASIGNACION
1	2016/09/28	Rq-03	Gamedasoft Sistemas Informaticos	Pendiente	Compras	Asignación
2	2016/09/02	Rq-02	Gamedasoft Sistemas Informaticos	Pendiente	Compras	Asignación

FIGURA 46-Selección de requerimiento a asignar.

BIENVENIDO Alvaro David

INICIO / Requerimientos

Requerimientos pendientes por asignar

TOTAL DE REGISTROS: 2

N°	FECHA	REQ.	CLIENTE	ESTADO	MODULO	ASIGNACION
1	2016/09/28	Rq-03	Gamedasoft Sistemas Informaticos	Pendiente	Compras	Asignación
2	2016/09/02	Rq-02	Gamedasoft Sistemas Informaticos	Pendiente	Compras	Asignación

ASIGNAR REQUERIMIENTO

Id Requerimiento: Rq-02
 Modulo: Compras
 Menu: Cursos Generales
 Submenú: Reimpresión Orden Compra
 Descripción: página 2 de requerimie update
 Empleado: Selección Empleado

Cancelar | Asignar

FIGURA 47- Seleccionamos el empleado a asignar.

Luego de la asignación el requerimiento pasa a los requerimientos asignados y está en proceso de solución.

BIENVENIDO Alvaro David

INICIO / Consulta Requerimientos

Filtros Requerimientos

Fecha Inicial:
 Fecha Final:
 Cliente: Todos
 Estado: Todos
 Modulo: Todos
 Empleado: Todos

Requerimientos

N°	FECHA	ID	CLIENTE	ESTADO	MODULO	EMPLEADO ASIGNADO
1	04/01/2016	Rq-1	Oleodavilla	Solucionado	Comedor	Allano Tuzo
2	04/02/2016	Rq-2	Gamedasoft Sistemas Informaticos	Solucionado	Contabilidad	Allano Tuzo
3	04/20/2016	Rq-3	Gamedasoft Sistemas Informaticos	Solucionado	Contabilidad	carlos Andrade
4	04/21/2016	Rq-4	Gamedasoft Sistemas Informaticos	En proceso	Comedor	David Villanco
5	04/28/2016	Rq-5	Gamedasoft Sistemas Informaticos	Solucionado	Comedor	Allano Tuzo
6	05/29/2016	Rq-11	Gamedasoft Sistemas Informaticos	Solucionado	Comedor	Fernando Caceres
7	05/29/2016	Rq-12	Gamedasoft Sistemas Informaticos	Solucionado	Recursos Humanos	Allano Tuzo
8	05/30/2016	Rq-13	Oleodavilla	Solucionado	Comedor	carlos Andrade
9	07/19/2016	Rq-19	Esfon Eon	Solucionado	Compras	Allano Tuzo
10	07/19/2016	Rq-20	Esfon Eon	Solucionado	Comedor	Allano Tuzo
11	08/02/2016	Rq-02	Gamedasoft Sistemas Informaticos	En proceso	Compras	Allano Tuzo

FIGURA 48- Requerimiento asignado.

TABLAS DE PRUEBAS DE ACEPTACIÓN DEL USUARIO EMPLEADO.

- PRUEBAS DE ACEPTACIÓN INICIAR SESIÓN

Prueba de inicio de sesión empleado	
Id	P.A.-05
Caso de uso	CU-13
Descripción: El empleado procede a seleccionar el perfil empleado e ingresar su número de cedula y contraseña.	
Pasos de ejecución: 1. Seleccionar el perfil administrador. 2. Ingresar su número de cédula. 3. Ingresar contraseña. 4. Presione el botón “Ingresar al sistema”.	
Resultado esperado: Que el sistema reconozca al usuario que ha iniciado la sesión.	Resultado obtenido: Se Ingresó al sistema con éxito y se visualiza el nombre del usuario.
Evaluación de la prueba: Se mostró el resultado esperado.	

Tabla 51- P.A.-05- Prueba de inicio de sesión empleado.

Visualización en la interfaz del sistema

FIGURA 49- Inicio sesión empleado.

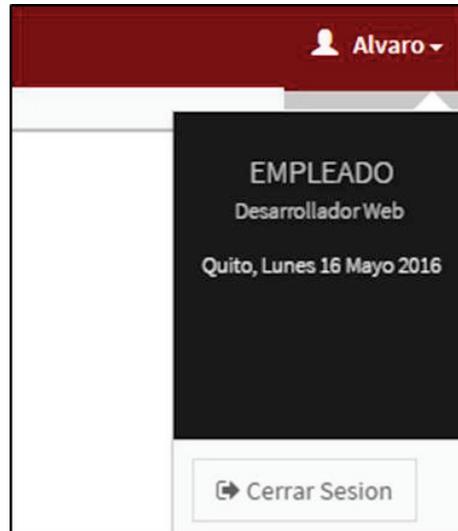


FIGURA 50- Prueba de inicio de sesión empleado.

- **PRUEBAS DE ACEPTACIÓN REGISTRO DEL TIEMPO INVERTIDO EN LA SOLUCIÓN DEL REQUERIMIENTO**

Prueba de aceptación registro de tiempos invertido en la solución del requerimiento	
Id	P.A.-06
Caso de uso	CU-14
Descripción: El administrador procede a registrar los tiempos que se invirtió en la solución del requerimiento.	
Pasos de ejecución: <ol style="list-style-type: none"> 1. Seleccionar la opción registró tiempos del menú “Operaciones”. 2. Seleccionar el cliente que desea ingresar. 3. Luego seleccionar el requerimiento que desea registrar el tiempo. 4. Ingresa el tiempo y la actividad que realizo para la solución. 5. Presionar el registrar. 	
Resultado esperado: Que se registre el tiempo y la actividad que se realizó para la solución del requerimiento.	Resultado obtenido: Se registró el tiempo del requerimiento.

Evaluación de la prueba:

Se mostró el resultado esperado.

Tabla 52- P.A.-04- Prueba de inicio de sesión Administrador.

Visualización en la interfaz del sistema

The screenshot shows the 'Registrar Tiempos' form in a web application. The left sidebar contains a menu with 'MENU PRINCIPAL', 'Mi Usuario', 'Operaciones', and 'Consultas'. The main content area has a header 'INICIO / Registro tiempos' and a sub-header 'Registrar Tiempos'. The form includes a dropdown for 'Selección de la Empresa' with a list of options: 'Selección Empresa', 'Gamedash Sistemas Informaticos', and 'Selección Requerimiento'. Below this are fields for 'Ingrese el Tiempo:' (containing 'Tiempo') and 'Ingrese la actividad:' (empty). A green 'Guardar' button is at the bottom right.

FIGURA 51- Seleccionar cliente ingreso tiempos.

The screenshot shows the 'Registrar Tiempos' form with the 'Seleccionar el requerimiento' dropdown menu open. The options are 'Selección Requerimiento' and 'Req 2Donwella 2 de requerime update'. The 'Ingreso el Tiempo:' field now contains 'Tiempo' and the 'Ingrese la actividad:' field is empty. The 'Guardar' button remains at the bottom right.

FIGURA 52- Seleccionar requerimiento.

The screenshot shows the 'Registrar Tiempos' form with the 'Ingreso el Tiempo:' field filled with '2:12' and the 'Ingrese la actividad:' field filled with 'prueba de funcionamiento'. The 'Guardar' button is at the bottom right.

FIGURA 53- Registro de tiempo del requerimiento.

CAPITULO IV

4. CONCLUSIONES Y RECOMENDACIONES

4.1. CONCLUSIONES

- Utilizar un framework permite a los desarrolladores de software seguir un estándar de desarrollo, la cual significa reducir costos y tiempo en el desarrollo y para su posterior mantenimiento. La utilización del framework Symfony en el desarrollo del sistema, facilitó el alcance de los objetivos propuestos en el proyecto y permitió aplicar estándares internacionales de programación en todas las fases de desarrollo del sistema.
- De acuerdo a los resultados obtenidos en las pruebas desarrolladas, se puede observar que el flujo para el control de tiempos de los desarrolladores es más eficiente y que se ha facilitado recepción de requerimientos de los clientes. El hecho de contar con el sistema en la empresa, permite al cliente verificar el estado de sus requerimientos de manera inmediata y asegura que la información ha sido recibida por parte de la empresa y que esta ha sido atendida. Adicionalmente, tener la información almacenada en una base de datos, simplifica el proceso de consulta de información y ayuda a que la información almacenada sea consistente reduce tiempos y ahorra dinero a las empresas.
- La calidad que brindan las herramientas de código abierto como PHP, Apache, MySQL, Symfony, Bootstrap, composer permite desarrollar sistemas con grandes funcionalidades y que cumplen con estándares internacionales tanto a nivel de código como a nivel de interfaces de usuario en periodos cortos de tiempo y a un costo mínimo.
- Finalmente, cabe destacar la importancia que tienen las fases de análisis y obtención de requerimientos, ya que estas permiten tener una visión general del sistema a desarrollar lo cual evita el desperdicio de tiempo en las demás fases, especialmente en el diseño y codificación.

4.2. RECOMENDACIONES

- Para nuevas versiones del sistema se podría agregar una nueva funcionalidad, que cuando el empleado va resolver un requerimiento exista un contador de tiempo desde su inicio hasta su finalización permitiendo tener un tiempo exacto en la solución de manera automática. Para la implementación de nuevas funcionalidades del sistema se recomienda el uso de la metodología OOHDM debido a los buenos resultados que se obtuvieron con esta metodología en el presente proyecto.
- Se recomienda el uso de herramientas de software libre para el desarrollo de software, ya que, a más del beneficio de los costos para el desarrollador, son herramienta que están en constante desarrollo para asegurar una alta calidad y facilidad de uso.
- A pesar de que el servidor donde fue implementado el sistema cumple con los requerimientos mínimos, si se desea mejorar el rendimiento del sistema, se recomienda implementarlo en un servidor web de altas prestaciones que sea capaz de manejar varias conexiones de usuario con un menor tiempo de respuesta.
- Debido a que el servidor se encuentra en la misma empresa, se recomienda que este cuente con una unidad de protección UPS con voltaje de 110/120 Vac en caso de cortes de energía eléctrica.

BIBLIOGRAFÍA

- documents.mx*. (2015). Obtenido de documents.mx:
<http://documents.mx/documents/oohdm-55b079a392811.html>
- Guerra Velez, G. (16 de Noviembre de 2011). *gestioninformacionyconocimiento*. Obtenido de Wikipedia: <http://gestioninformacionyconocimiento.blogspot.com/2011/11/oohdm-object-oriented-hypermedia-design.html>
- IEEE. (2006). *IEEE*. Obtenido de <https://prezi.com/ypibvtchc-0m/normas-ieee-sobre-software-e-ingenieria-de-software/>
- Myers, G. (2004). The Art of Software Testing, Second Edition. *The Art of Software Testing, Second Edition*.
- Potencier, F., & Zaninotto, F. (2008). *Symfony la guia definitiva*.
- Pressman, R. S. (2010). *Ingenieria del software un enfoque practico 7th.ed.* México, D. F.: MCGRAW-HILL INTERAMERICANA EDITORES, S.A. DE C.V.
- Rafale, D. (2008). *Rafael Dohms*. Obtenido de <http://blog.doh.ms/2008/08/18/ajax-e-php-aprendendo-a-base-1/?lang=pt-br>
- Rodrigues, J. (2016). *Juandarodriguez.es*. Obtenido de Juandarodriguez.es:
<http://juandarodriguez.es/cursosf14/unidad7.html>
- SitePoint*. (2015). Obtenido de SitePoint: <https://www.sitepoint.com/best-php-framework-2015-sitepoint-survey-results/>
- Sommerville, I. (2011). *INGENIRIA DEL SOFTWARE Novena edición*. México: PEARSON EDUCACION S.A.
- Symfony.com*. (2016). Obtenido de Symfony.com:
http://symfony.com/legacy/doc/jobeeet/1_4/es/04?orm=Doctrine
- Vélez Serrano, J. (2011). Diseñar y programar, todo es empezar. En J. Vélez Serrano, A. Peña Abril, & P. Gortazar Bellas, *Diseñar y programar, todo es empezar* (pág. 5). Madrid: DYKINSON.
- Yeison, F. (2011). *Sistemasy.blogspot.com*. Obtenido de Sistemasy.blogspot.com:
<http://sistemasy.blogspot.com/2011/03/enumere-los-tipos-de-diagrama-de-uml-y.html>