

UNIVERSIDAD PÚBLICA DE EL ALTO

CARRERA INGENIERÍA DE SISTEMAS



PROYECTO DE GRADO

SISTEMA WEB PARA LA VENTA DE LADRILLOS Y
CONTROL DE PERSONAL

CASO: FÁBRICA DE CERÁMICA “EMANUEL”

Para optar al título de Licenciatura en Ingeniería de Sistemas

Mención: INFORMÁTICA Y COMUNICACIONES

Postulante : Univ. Armin Rodrigo Forra Layme
Tutor Metodológico : Ing. Marisol Arguedas Balladares
Tutor Especialista : Ing. Fernando Chambi Guachalla
Tutor revisor : Ing. Gabriel Reynaldo Sirpa Huayhua

EL ALTO – BOLIVIA
2020

DEDICATORIA

Dedicarlo este trabajo con mucho cariño:

A Dios que me ha dado la vida y la fortaleza para este proyecto de grado.

De la misma manera a mi estimado padre, madre y hermanas por el gran esfuerzo, amor y confianza que depositaron en mí, por sus sabias enseñanzas y consejos en largo de mi carrera universitaria.

AGRADECIMIENTO

Agradecer a todas aquellas personas quienes han tenido que ver en el desarrollo y conclusión de este proyecto.

Agradecer con mucho afecto a mis distinguidos tutores; a mi tutor metodológico Ing. Arguedas Balladares Marisol por su conocimiento, apoyo, confianza, tiempo, persistencia, paciencia y motivación que me brindo hacia mi persona.

A mi tutor especialista Ing. Chambi Guachalla Fernando por compartir su conocimiento, brindarme su orientación, sugerencias con paciencia y motivación durante el desarrollo del software del presente proyecto.

Por último, a mi tutor revisor Ing. Sirpa Huayhua Gabriel Reynaldo por su disponibilidad de tiempo, orientación y las observaciones brindadas por su amplio conocimiento en la realización de la documentación del proyecto.

RESUMEN

La ciencia y la tecnología van creciendo de manera acelerada y junto a ellas avanzan las ciencias informáticas y computacionales, las cuales se encuentran inmensas en nuestras vidas.

Se denomina aplicación web a aquellas herramientas que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador. En otras palabras, es una aplicación software que se codifica en un lenguaje soportado por los navegadores web en la que se confía la ejecución al navegador.

Se sabe que en la actualidad la sociedad está completamente informatizada, es un hecho que millones de personas están conectadas por medio de computadoras que están conectadas a la red. Todos los equipos no solo nos sirven para la búsqueda de información o chat en internet si no también se las usan en entidades públicas o privadas para el manejo de información, control de personal, administración de actividades y toma de decisiones.

Una vez realizado el análisis profundo en la fábrica de ladrillos “EMANUEL” se ha identificado claramente las deficiencias en el manejo de información acerca del registro y seguimiento de ingreso de materia prima, reportes diarios, semanales y mensuales de cantidad de cerámica vendida y controlar el historial de cada uno de los trabajadores.

Por tal motivo surge una gran necesidad de automatizar las tareas que se desempeñan en la fábrica de cerámica “EMANUEL”, utilizando nuevas tendencias metodológicas de análisis y diseño del sistema y herramientas de desarrollo javascript, Ajax, css3, bootstrap, html5, jquery, servidor Apache, base de datos en MySQL y el lenguaje de Programación PHP para así satisfacer los requerimientos de los encargados de la fábrica.

También se ha realizado el control de calidad d con la ayuda del estándar ISO/IEC 9126 así como la seguridad de la información con el estándar ISO-27002 y finalmente la estimación de costo del software con la aplicación de COCOMO.

ÍNDICE

1	MARCO PRELIMINAR	1
1.1	INTRODUCCIÓN	1
1.2	ANTECEDENTES	2
1.2.1	ANTECEDENTES DE LA FÁBRICA.....	2
1.2.2	NACIONAL.....	3
1.3	PLANTEAMIENTO DEL PROBLEMA.....	3
1.3.1	PROBLEMA PRINCIPAL.....	4
1.3.2	PROBLEMAS SECUNDARIOS	4
1.4	OBJETIVOS.....	5
1.4.1	OBJETIVO GENERAL	5
1.4.2	OBJETIVOS ESPECÍFICOS	5
1.5	JUSTIFICACIÓN	5
1.5.1	TÉCNICA	5
1.5.2	ECONÓMICA	6
1.5.3	SOCIAL.....	6
1.6	METODOLOGÍA	6
1.6.1	METODOLOGÍA DE INGENIERÍA	6
1.6.2	MÉTRICA DE CALIDAD DE SOFTWARE.....	7
1.6.3	MÉTODO DE ESTIMACIÓN DE COSTO DE SOFTWARE	7
1.7	HERRAMIENTAS.....	8
1.7.1.	LENGUAJE DE DESARROLLO	8
1.7.2	GESTOR DE BASE DE DATOS.....	9
1.7.3	HERRAMIENTAS DE INTERFAZ DE DISEÑO FRONTEND.....	10
1.8	LÍMITES Y ALCANCES.....	11
1.8.1	LÍMITES	11
1.8.2	ALCANCES.....	11
1.9	APORTES.....	12
2	MARCO TEÓRICO.....	13
2.1	SISTEMA	13
2.5	SOFTWARE.....	13
2.2.1	TIPOS PRINCIPALES DE SOFTWARE	14

2.5	SISTEMA DE INFORMACIÓN	14
2.5	DEFINICIÓN DE VENTA.....	18
2.4.1	CONTROL DE PERSONAL	18
2.5	METODOLOGÍA DE DESARROLLO.....	18
2.5.1	METODOLOGÍA DE UWE	19
2.5.2	ACTIVIDADES DE MODELADO DE UWE	19
2.5.3	CARACTERÍSTICA DE LA METODOLOGÍA UWE	20
2.5.4	MODELO DE LA METODOLOGÍA UWE.....	20
2.5.4.1	MODELO DE CASO DE USO.....	20
2.5.4.2	MODELO DE CONTENIDO	24
2.5.4.3	MODELO DE NAVEGACIÓN.....	25
2.5.4.4	MODELO DE ESTRUCTURA DEL PROCESO.....	26
2.5.4.5	MODELO DE PRESENTACIÓN.....	27
2.5.5	FASE DE LA METODOLOGÍA UWE.....	28
2.5.5.1	CAPTURA, ANÁLISIS Y ESPECIFICACIÓN DE REQUISITOS	28
2.5.5.2	DISEÑO DEL SISTEMA.....	28
2.5.5.3	CODIFICACIÓN DEL SOFTWARE	29
2.5.5.4	PRUEBAS.....	29
2.5.5.5	LA INSTALACIÓN O FASE DE IMPLEMENTACIÓN	29
2.5.5.6	MANTENIMIENTO.....	29
2.6	HERRAMIENTAS DE DESARROLLO.....	29
2.6.1	SERVIDOR APACHE.....	29
2.6.2	BASE DE DATOS	31
2.6.2.1	GESTOR DE BASE DE DATOS MARIADB.....	33
2.6.3	LENGUAJE DE PROGRAMACIÓN PHP.....	34
2.6.3.1	FRAMEWORK CODEIGNITER	35
2.6.4	HERRAMIENTA DE DISEÑO.....	36
2.6.4.1	HTML5.....	36
2.6.4.2	CSS3	38
2.6.4.3	JQUERY	39
2.6.4.4	JAVASCRIPT.....	39
2.6.4.5	AJAX.....	41
2.6.4.6	FRAMEWORK BOOTSTRAP	42

2.7	MÉTRICA DE CALIDAD DE SOFTWARE.....	43
2.7.1	ISO 9000.....	44
2.7.2	ESTÁNDAR ISO/IEC 9126.....	45
2.7.3	MÉTRICAS DE CALIDAD DEL MODELO DE ISO-9126	45
2.7.3.1	EVALUACIÓN INTERNA, EXTERNA Y CALIDAD DE USO ISO/IEC 9126.....	47
2.7.3.1.1	FUNCIONALIDAD.....	47
2.7.3.1.2	CONFIABILIDAD.....	48
2.7.3.1.3	USABILIDAD.....	48
2.7.3.1.4	EFICIENCIA.....	49
2.7.3.1.5	CAPACIDAD DE MANTENIMIENTO.....	50
2.7.3.1.6	PORTABILIDAD.....	50
2.7.4	MÉTRICAS BASADAS EN LA FUNCIÓN.....	52
2.8	MÉTODO DE ESTIMACIÓN DE COSTO DE SOFTWARE	55
2.8.1	MODELO COCOMO	55
2.8.1.1	MODELOS DE ESTIMACIÓN DE COSTO CON COCOMO.....	56
2.9	SEGURIDAD DE APLICACIONES WEB.....	60
2.9.1	CAJA NEGRA Y BLANCA.....	62
2.9.1.1	PRUEBAS (DISEÑO DE CASOS DE PRUEBA)	62
2.9.1.2	PRUEBAS DE CAJA NEGRA	63
2.9.1.3	PRUEBAS DE CAJA BLANCA.....	63
3	MARCO APLICATIVO.....	64
3.1	DESCRIPCIÓN GENERAL DE LA INSTITUCIÓN.....	64
3.2	OBTENCIÓN DE REQUISITOS	64
3.2.1	DEFINICIÓN DE ACTORES	65
3.2.2	LISTA DE REQUERIMIENTO DEL SISTEMA.....	67
3.2.2.1	REQUERIMIENTOS FUNCIONALES	67
3.2.2.2	REQUERIMIENTOS NO FUNCIONALES	68
3.3	APLICACIÓN DEL MODELO UWE	69
3.3.1	MODELO DE CASO DE USO	69
3.3.1.1	ANÁLISIS DE REQUERIMIENTOS.....	69
3.3.1.1.1	DIAGRAMA DE CASO DE USO	69
3.3.1.1.2	MODELO DE DIAGRAMA DE CASO DE USO: GENERAL DEL SISTEMA	70
3.3.1.1.3	DIAGRAMA DE CASO DE USO: ADMINISTRADOR/A PRINCIPAL.....	70

3.3.1.1.3.1	DIAGRAMA DE ADMINISTRACIÓN DE PERSONAL	71
3.3.1.1.3.2	DIAGRAMA DE ADMINISTRACIÓN DE USUARIOS	72
3.3.1.1.3.3	DIAGRAMA DE ADMINISTRACIÓN SERVICIOS BÁSICOS.....	73
3.3.1.1.3.4	DIAGRAMA DE ADMINISTRACIÓN DE PRODUCTOS	74
3.3.1.1.3.5	DIAGRAMA DE ADMINISTRACIÓN VENTA DE PRODUCTO.....	75
3.3.1.1.3.6	DIAGRAMA DE ADMINISTRACIÓN DE PÁGINA WEB	76
3.3.1.1.3.7	DIAGRAMA DE PLANILLA	77
3.3.1.1.4	DIAGRAMA DE CASO DE USO: ENCARGADO/A	78
3.3.1.1.4.1	DIAGRAMA DE CASO DE USO: MÓDULOS ASIGNADOS.....	79
3.3.1.1.5	DIAGRAMA DE CASO DE USO: SECRETARIA.....	80
3.3.1.1.5.1	DIAGRAMA DE CASO DE USO: MÓDULOS COMPRA MATERIAL.....	81
3.3.1.1.6	DIAGRAMA DE CASO DE USO: CLIENTE.....	82
3.3.2	MODELO DE CONTENIDO.	83
3.3.3	MODELO DE NAVEGACIÓN.....	85
3.3.3.1	MODELO DE NAVEGACIÓN: ADMINISTRADOR/A PRINCIPAL.	85
3.3.3.2	MODELO DE NAVEGACIÓN: ENCARGADO/A.....	86
3.3.3.3	MODELO DE NAVEGACIÓN: SECRETARIA.	86
3.3.3.4	MODELO DE NAVEGACIÓN: CLIENTE.	87
3.3.4	MODELO DE ESTRUCTURA DE PROCESO.....	87
3.3.5	MODELO DE PRESENTACIÓN.....	88
3.3.5.1	MODELO DE PRESENTACIÓN: ADMINISTRADOR/A PRINCIPAL.	89
3.3.5.2	MODELO DE PRESENTACIÓN: ENCARGADO/A.....	90
3.3.5.3	MODELO DE PRESENTACIÓN: SECRETARIA.	91
3.3.5.4	MODELO DE PRESENTACIÓN: CLIENTE.....	92
3.4	FASE DE LA METODOLOGÍA UWE.....	94
3.4.1	CAPTURA, ANÁLISIS Y ESPECIFICACIÓN DE REQUISITOS.	94
3.4.2	DISEÑO DEL SISTEMA.....	94
3.4.3	CODIFICACIÓN DEL SOFTWARE.	95
3.4.4	PRUEBAS.....	96
3.4.5	LA INSTALACIÓN O FASE DE IMPLEMENTACIÓN.	100
3.4.6	MANTENIMIENTO:	100
3.5	MÉTRICAS DE CALIDAD DE SOFTWARE.....	101
3.5.1	FUNCIONALIDAD	101

3.5.2	CONFIABILIDAD.....	108
3.5.3	MANTENIBILIDAD.....	109
3.5.4	USABILIDAD.....	110
3.5.5	PORTABILIDAD.....	112
3.5.6	RESULTADOS.....	113
3.6	MÉTODO DE ESTIMACIÓN DE COSTO DE SOFTWARE.....	114
3.6.1	MÉTODO DE ESTIMACIÓN DE COCOMO.....	114
3.7	SEGURIDAD DEL SISTEMA.....	117
3.7.1	ISO-27002.....	117
3.7.1.1	SEGURIDAD LÓGICA.....	118
3.7.1.2	SEGURIDAD FÍSICA.....	118
3.7.1.3	SEGURIDAD ORGANIZATIVA.....	118
3.7.2	PRUEBAS DE CAJA NEGRA Y BLANCA.....	118
3.7.2.1	PRUEBAS DE CAJA NEGRA.....	118
3.7.2.2	PRUEBA DE CAJA BLANCA.....	120
4	CONCLUSIONES Y RECOMENDACIONES.....	123
4.1	CONCLUSIONES.....	123
4.2	RECOMENDACIONES.....	124
	BIBLIOGRAFÍA.....	125

ÍNDICE DE FIGURAS

CAPITULO I MARCO PRELIMINAR

FIGURA Nº 1. 1 HERRAMIENTAS DE DESARROLLO.....	8
--	---

CAPITULO II MARCO TEORICO

FIGURA Nº 2. 1 CICLO DE VIDA CLÁSICO DE DESARROLLO DE SOFTWARE	17
FIGURA Nº 2. 2 ESTEREOTIPOS DE DIAGRAMA DE CASO DE USO	21
FIGURA Nº 2. 3 ACTOR.....	21
FIGURA Nº 2. 4 CASO DE USO	22
FIGURA Nº 2. 5 ASOCIACIÓN.....	22
FIGURA Nº 2. 6 DEPENDENCIA O INSTANCIACIÓN.....	23
FIGURA Nº 2. 7 GENERALIZACIÓN	23
FIGURA Nº 2. 8 DIAGRAMA DE CASO DE USO EN LA METODOLOGÍA UWE	24
FIGURA Nº 2. 9 DIAGRAMA DE CONTENIDO EN LA METODOLOGÍA UWE	25
FIGURA Nº 2. 10 ESTEREOTIPOS DE DIAGRAMA DE NAVEGACIÓN.....	25
FIGURA Nº 2. 11 DIAGRAMA DE NAVEGACIÓN EN LA METODOLOGÍA UWE.....	26
FIGURA Nº 2. 12 ESTEREOTIPOS DE DIAGRAMA DE PRESENTACIÓN	27
FIGURA Nº 2. 13 FASE DE LA METODOLOGÍA UWE	28
FIGURA Nº 2. 14 SERVIDOR WEB.....	31
FIGURA Nº 2. 15 SERVIDOR APACHE.....	31
FIGURA Nº 2. 16 CONCEPTOS BÁSICOS DE BASES DE DATOS.....	32
FIGURA Nº 2. 17 FUNCIONAMIENTO DE CSS3	39
FIGURA Nº 2. 18 DEFINICIÓN JAVASCRIPT	40
FIGURA Nº 2. 19 DETALLES DE COEFICIENTES DE COCOMO II	60

CAPITULO III MARCO APLICATIVO

FIGURA Nº 3. 1 MODELO DE DIAGRAMA DE CASO DE USO: GENERAL DEL SISTEMA ..	70
FIGURA Nº 3. 2 DIAGRAMA DE ADMINISTRACIÓN DE PERSONAL.....	71
FIGURA Nº 3. 3 DIAGRAMA DE ADMINISTRACIÓN DE USUARIOS	72
FIGURA Nº 3. 4 DIAGRAMA DE ADMINISTRACIÓN COMPRA MATERIALES.	73
FIGURA Nº 3. 5 DIAGRAMA DE ADMINISTRACIÓN DE PRODUCTOS	74
FIGURA Nº 3. 6 DIAGRAMA DE ADMINISTRACIÓN VENTA DE PRODUCTO.	75

FIGURA Nº 3. 7 DIAGRAMA DE ADMINISTRACIÓN DE PÁGINA WEB.....	76
FIGURA Nº 3. 8 DIAGRAMA DE CASO DE USO: MÓDULOS ASIGNADOS.	79
FIGURA Nº 3. 9 DIAGRAMA DE CASO DE USO: MÓDULOS ASIGNADOS.	81
FIGURA Nº 3. 10 DIAGRAMA DE CASO DE USO: CLIENTE.	82
FIGURA Nº 3. 11 MODELO DE CONTENIDO	84
FIGURA Nº 3. 12 MODELO DE NAVEGACIÓN: ADMINISTRADOR/A PRINCIPAL.....	85
FIGURA Nº 3. 13 MODELO DE NAVEGACIÓN: ENCARGADO/A.	86
FIGURA Nº 3. 14 MODELO DE NAVEGACIÓN: SECRETARIA.....	86
FIGURA Nº 3. 15 MODELO DE NAVEGACIÓN: CLIENTE.	87
FIGURA Nº 3. 16 MODELO DE ESTRUCTURA DE PROCESO	88
FIGURA Nº 3. 17 MODELO DE PRESENTACIÓN: ADMINISTRADOR/A PRINCIPAL LOGIN..	89
FIGURA Nº 3. 18 MODELO DE PRESENTACIÓN: ADMINISTRADOR/A PRINCIPAL.....	90
FIGURA Nº 3. 19 MODELO DE PRESENTACIÓN: ENCARGADO/A LOGIN	90
FIGURA Nº 3. 20 MODELO DE PRESENTACIÓN: ENCARGADO/A.....	91
FIGURA Nº 3. 21 MODELO DE PRESENTACIÓN: SECRETARIA LOGIN.....	91
FIGURA Nº 3. 22 MODELO DE PRESENTACIÓN: SECRETARIA.	92
FIGURA Nº 3. 23 MODELO DE PRESENTACIÓN: CLIENTE WEB	92
FIGURA Nº 3. 24 MODELO DE PRESENTACIÓN: CLIENTE FORM REGISTRO.....	93
FIGURA Nº 3. 25 MODELO DE PRESENTACIÓN: CLIENTE LOGIN	93
FIGURA Nº 3. 26 MODELO DE PRESENTACIÓN: CLIENTE RESERVA PRODUCTO	94
FIGURA Nº 3. 27 DISEÑO DEL SISTEMA ADMINISTRADOR.....	95
FIGURA Nº 3. 28 CODIFICACIÓN DEL SOFTWARE	96
FIGURA Nº 3. 29 PRUEBAS MENÚ INICIO.....	97
FIGURA Nº 3. 30 PRUEBAS LOGIN USUARIO.....	97
FIGURA Nº 3. 31 PRUEBAS MENÚ ADMINISTRADOR	98
FIGURA Nº 3. 32 PRUEBAS DE ADMINISTRACIÓN DE PRODUCTO.....	98
FIGURA Nº 3. 33 PRUEBAS REALIZAR VENTA A CLIENTE	99
FIGURA Nº 3. 34 PRUEBAS DE COMPROBANTE	99
FIGURA Nº 3. 35 INSTALACIÓN O FASE DE IMPLEMENTACIÓN	100

ÍNDICE DE TABLAS

CAPITULO II MARCO TEÓRICO

TABLA Nº 2. 1 TABLA DE FACTOR DE PONDERACIÓN	53
TABLA Nº 2. 2 FACTOR DE ESCALA	54
TABLA Nº 2. 3 VALORACIÓN MÉTRICA PUNTO FUNCIÓN.....	54
TABLA Nº 2. 4 PUNTOS OBJETO.....	57

CAPITULO III MARCO APLICATIVO

TABLA Nº 3. 1 OBTENCIÓN DE REQUISITOS	65
TABLA Nº 3. 2 DEFINICIÓN DE ACTORES	66
TABLA Nº 3. 3 LISTA DE REQUERIMIENTO DEL SISTEMA	67
TABLA Nº 3. 4 REQUERIMIENTOS FUNCIONALES	68
TABLA Nº 3. 5 REQUERIMIENTOS NO FUNCIONALES	68
TABLA Nº 3. 6 CASO DE USO: ADMINISTRACIÓN DE PERSONAL	71
TABLA Nº 3. 7 CASO DE USO: ADMINISTRACIÓN DE USUARIOS	72
TABLA Nº 3. 8 CASO DE USO: ADMINISTRACIÓN SERVICIOS BÁSICOS	73
TABLA Nº 3. 9 CASO DE USO: ADMINISTRACIÓN DE PRODUCTOS	74
TABLA Nº 3. 10 CASO DE USO: ADMINISTRACIÓN VENTA DE PRODUCTO.....	75
TABLA Nº 3. 11 CASO DE USO: ADMINISTRACIÓN DE PÁGINA WEB	76
TABLA Nº 3. 12 CASO DE USO: ADMINISTRACIÓN DE PÁGINA WEB	78
TABLA Nº 3. 13 CASO DE USO: ENCARGADO/A	80
TABLA Nº 3. 14 CASO DE USO: SECRETARIA.....	81
TABLA Nº 3. 15 CASO DE USO: CLIENTE.....	82
TABLA Nº 3. 16 CARACTERÍSTICA DE LA FUNCIONALIDAD	101
TABLA Nº 3. 17 NÚMERO DE ENTRADAS DE USUARIO	102
TABLA Nº 3. 18 NÚMERO DE SALIDAS USUARIO	103
TABLA Nº 3. 19 NÚMERO DE PETICIONES DEL USUARIO	103
TABLA Nº 3. 20 NÚMERO DE ARCHIVOS.....	104
TABLA Nº 3. 21 NÚMERO DE INTERFACES EXTERNAS	104
TABLA Nº 3. 22 FACTORES DE PONDERACIÓN	105
TABLA Nº 3. 23 VALORES DE AJUSTE DE COMPLEJIDAD	106
TABLA Nº 3. 24 AJUSTE DE PREGUNTAS	110
TABLA Nº 3. 25 AJUSTE DE PREGUNTAS	111

TABLA Nº 3. 26 PORTABILIDAD.....	112
TABLA Nº 3. 27 RESULTADOS.....	113
TABLA Nº 3. 28 APLICACIÓN DEL MODELO INTERMEDIO	114
TABLA Nº 3. 29 ECUACIONES DEL MODELO COCOMO II.....	115
TABLA Nº 3. 30 CÁLCULO DE ATRIBUTOS FAE	115

CAPÍTULO I

MARCO PRELIMINAR

1 MARCO PRELIMINAR

1.1 Introducción

En la actualidad puede apreciarse, que un sistema web es indispensable para el funcionamiento de cualquier organización, ya que nos ayudan de gran manera en la toma de decisiones oportunas, confiables y efectivas en cuanto a la planificación, programación y administración con el fin de garantizar el éxito deseado, limitar los riesgos y reducir los costos.

Se sabe que en la actualidad la sociedad está completamente informatizada, es un hecho que millones de personas están conectadas por medio de computadoras que están conectadas a la red. Todos los equipos no solo nos sirven para la búsqueda de información o chat en internet si no también se las usan en entidades públicas o privadas para el manejo de información, control de personal, venta y administración de actividades y toma de decisiones.

Debido a esta razón nace la idea de automatizar las actividades cotidianas en las diferentes organizaciones; cabe destacar el gran avance de las telecomunicaciones y el proceso que han experimentado las ciencias informáticas que obliga a estar en tono y entrar al moderno mundo de la tecnología, ser competitivos y no quedarse relegados en las tareas que proporcionan grandes beneficios a proyectos futuros.

Una vez realizado el análisis profundo en la fábrica de ladrillos “EMANUEL” se ha identificado claramente las deficiencias en el manejo de información en la venta de ladrillos, ya que no cuentan con un registro automatizado y por esta razón es muy difícil realizar un reporte de cantidad de ventas de cerámica mensual o semanalmente, por otro lado realizar reservas en línea y también poder ofrecer nuestros productos en la web y controlarlo desde un ordenador, tanto al stock con el que cuenta la fábrica, como a todo el personal en cuanto a la asistencia, pago de salarios, tiempo de permanencia y de contrato con la fábrica y poder realizar todo lo mencionado de manera rápida y efectiva sin causar descuentos por errores de índole manual que recurrentemente sucedía.

Por tal motivo surge una gran necesidad de automatizar las tareas que se desempeñan en la fábrica de cerámica “EMANUEL”, utilizando nuevas tendencias metodológicas de análisis y diseño con la metodología UWE y herramientas como UML, servidor Apache, base de datos en MySQL y el lenguaje de Programación PHP para así satisfacer los requerimientos de los encargados de la fábrica.

1.2 ANTECEDENTES

1.2.1 Antecedentes de la fábrica

La fábrica dio sus primeros pasos en el año 1998 con el nombre de cerámica “LA CONQUISTA”, como derecho de ingreso se dio 5000 ladrillos y así poder ser parte de la asociación “CERÁMICA ROJA” la cual fue fundada por 20 socios. En la actualidad la asociación cuenta con 40 fábricas legalmente inscritas.

La fábrica se ubica en la avenida final Buenos Aires, número 3060, zona Alpacoma Bajo perteneciente al Distrito 6 de Achocalla a la cual pertenece la licencia de funcionamiento, la superficie del terreno es de 5000 m².

En los inicios la preparación de la arcilla se realizaba manualmente y no se contaba con personal, solo era un trabajo en familia. Se tuvo muchos tropiezos en el afán de mejorar la calidad de los ladrillos ya que existía mucha competencia en el rubro.

Debido a dificultades en los papeleos en el Ministerio de Trabajo en el año 2012 se tuvo que cambiar el nombre de la fábrica que actualmente lleva el nombre de Cerámica “EMANUEL” que cuenta con Registro Patronal N° 01-331-080 y N° NIT: 2542156010.

La maquinaria con la que cuenta actualmente la fábrica es semi industrial en la cual elaboran aproximadamente una cantidad de 35000 ladrillos semanalmente.

Antecedentes internacionales

- ❖ Según (Maluff, 2013) “Implementación de sistemas de información de gestión comercial para mejorar los procesos de comercialización del grupo Autonort” Verificar si la implementación de Sistemas de Información de Gestión Comercial mejorará los procesos de comercialización del Grupo Autonort en el año 2013, pasando de un crecimiento de sus ventas de 3.1% en agosto a uno de 4% a septiembre., Universidad Privada Antenor Orrego; Trujillo Perú.
- ❖ Según (Godas, 2013) “Sistema para control de inventario, venta y generación de datos comerciales de restaurante” Desarrollar un sistema para control de inventario, venta y generación de datos comerciales para restaurante OK Corral. Universidad Austral de Chile; Puerto Montt Chile.

1.2.2 Nacional

- ❖ Según (Mamani, 2008) “Sistema de información vía web para el control de almacenes fábrica de fideos Santa Rosa” Tiene como actividad de elaborar, envasar y la distribución de fideos. Para el desarrollo del sistema se empleó la metodología SCRUM como herramienta, el uso de Lenguaje Unificado de Modelado (UML) y para realizar la técnica de inventario de bienes y modelo primero en entrar primero en Salir (PEPS). UMSA; La Paz.
- ❖ Según (H, 2001) “Sistema de control de ventas e inventarios COIMPE LTDA” Donde priorizan la estimación del tiempo de elaboración de los diferentes productos (piedras preciosas). Con la metodología OMT, el modelo de inventarios que se planteó fue la determinística. UMSS; Cochabamba.

1.3 PLANTEAMIENTO DEL PROBLEMA

Debido al proceso manual de la administración de la información, el procesamiento de los reportes diarios, semanales y mensuales de todas las ventas realizadas y el control del personal se obtienen los siguientes problemas.

1.3.1 Problema principal

La fábrica de Cerámica “EMANUEL” zona Alpacoma Bajo de Achocalla. En la actualidad las ventas de la fábrica y control de personal, son manuales lo cual genera una información desactualizada y no permiten una gestión adecuada, lo cual no cuenta con una información centralizada en donde se registre de manera detallada toda la información y salidas de las ventas, esto provoca inestabilidad en la fábrica EMANUEL.

1.3.2 Problemas secundarios

- ❖ No cuenta con información completa en cuanto a las ventas por que se realiza de forma manual en un libro, lo que ocasiona no poder realizar resúmenes de los movimientos que hay en la fábrica en tiempo real y en determinados periodos de tiempo, porque el procesamiento es manual.
- ❖ En las ventas es necesario un control del stock disponible y mínimo de forma oportuna.
- ❖ No existe información actualizada respecto a los proveedores de materia prima lo que dificulta su adquisición.
- ❖ No existe información actualizada de materia prima lo que ocasiona que sobre o falté estos insumos y paralicé la producción.
- ❖ Hace falta realizar pre reservas desde cualquier lugar, para poder realizar una compra de manera segura y confiable.
- ❖ No se cuenta con la información centralizada del personal que nos permita administrar su asistencia y los salarios.
- ❖ No existe información de estadísticas en cuanto a las ventas/compras y planillas del personal, por lo tanto, se describe la siguiente pregunta.

¿De qué manera el desarrollo de un sistema web coadyuvará en la venta de ladrillos y control de personal, realizándolo ágil y oportunamente, según los requerimientos de la fábrica para la correcta toma de decisiones?

1.4 OBJETIVOS

1.4.1 Objetivo general

Desarrollar un Sistema Web para la Venta de ladrillos y control del personal, el mismo que genere información confiable y oportuna coadyuve en la toma de decisiones y el crecimiento de la fábrica de cerámica “EMANUEL” de la localidad de Achocalla.

1.4.2 Objetivos específicos

- ❖ Automatizar la información completa respecto a la venta de ladrillos para la obtención de resúmenes de los movimientos en tiempo real y de manera oportuna.
- ❖ Sistematizar y diseñar la base de datos en cuanto a ventas para poder controlar el stock disponible y mínimo de forma oportuna.
- ❖ Sistematizar la información de los proveedores para disponer de sus direcciones y teléfonos y facilitar la adquisición.
- ❖ Sistematizar la información actualizada de materia prima para abastecer oportunamente los insumos y no paralizar la producción
- ❖ Automatizar las pre reservas para que los clientes puedan acceder desde cualquier lugar de manera segura y confiable.
- ❖ Desarrollar una base de datos en donde se administre la información sobre asistencias y salarios de todo el personal.
- ❖ Automatizar la información de estadísticas sobre las ventas/compras y la administración de las planillas del personal.

1.5 JUSTIFICACIÓN

1.5.1 Técnica

Actualmente la Cerámica Emanuel no cuenta con una unidad de alojamiento de Hosting, lo cual deberá realizar una adquisición alojamiento Hosting básico, para el equipamiento necesario para el funcionamiento del sistema.

El sistema desarrollado es multiplataforma puede funcionar tanto en GNU/Linux o Windows; escalable con capacidad de crecer en el tiempo.

1.5.2 Económica

Con el desarrollo del sistema web para la venta de ladrillos y control de personal, se generará información actualizada y oportuna lo que minimizara las pérdidas económicas para la fábrica de cerámica “EMANUEL” .

1.5.3 Social

El sistema web coadyuvará el control de las ventas de ladrillos, control del personal realizando un seguimiento sistemático adecuado facilitando al personal de ventas.

Prestará una administración de reportes más rápida y segura para la fábrica y así también de manera más confiable para la clientela de la empresa.

Beneficiará de gran manera al personal de la fábrica, ya que podrán tener un historial de manera automatizada durante su labor.

1.6 METODOLOGÍA

1.6.1 Metodología de ingeniería

UWE UML (UML-Based Web Engineering) Es una herramienta para modelar aplicaciones web, utilizada en la ingeniería web, prestando especial atención en sistematización y personalización (sistemas adaptativos). UWE es una propuesta basada en el proceso unificado y UML pero adaptados a la web. UWE está especializada en la especificación de aplicaciones adaptativas, y por tanto hace especial hincapié en características de personalización, como es la definición de un modelo de usuario o una etapa de definición de características adaptativas de la navegación en función de las preferencias, conocimiento o tareas de usuario. UWE es un proceso del desarrollo para aplicaciones Web enfocado sobre el diseño sistemático, la personalización y la generación semiautomática de escenarios que guíen el proceso de desarrollo de una aplicación Web. UWE describe una metodología de diseño sistemática, basada en las técnicas de UML, la notación de UML y los mecanismos de extensión de UML. Según (thewolf, 2015)

1.6.2 Métrica de calidad de software

La norma ISO/IEC 9126 permite especificar y evaluar la calidad del software desde diferentes criterios asociados con adquisición, requerimientos, desarrollo, uso, evaluación, soporte, mantenimiento, aseguramiento de la calidad y auditoria de software.

Los modelos de calidad para el software se describen así. Calidad Interna: Es la totalidad de las características del producto de software desde una perspectiva interna.

La calidad interna es medida y evaluada con base a los requerimientos internos de calidad. Calidad externa: Es la totalidad de las características del producto de software desde una perspectiva externa. El cual es típicamente medida y evaluada en un ambiente simulado, con datos simulados y usando métricas externas.

Calidad de uso: Es la perspectiva del usuario de la calidad del producto de software cuando este es usado en un ambiente específico y en un contexto de uso específico. Según (Villamizar, 2017)

1.6.3 Método de estimación de costo de software

Modelo COCOMO II, modelo de estimación que se encuentra en la jerarquía de modelos de estimación de software con el nombre de COCOMO, por Constructive Cost Model (Modelo Constructivo de Coste).

El modelo COCOMO original se ha convertido en uno de los modelos de estimación de coste del software más utilizados y estudiados en la industria. Según (Adriana Gómez, 2016).

COCOMO es una herramienta basada en la línea de código la cual le hace muy poderoso para la estimación de costos y no como otros que solamente miden el esfuerzo en base al tamaño. Hoy en día es necesario para un administrador de proyectos Poseer una herramienta de estimación de costos; y esta herramienta puede ser COCOMO. Representa el más extenso modelo empírico para la estimación de software publicado hasta la fecha. Con la realización de este trabajo ampliamos nuestro conocimiento acerca de la estimación de costos que es fundamental para un analista, administrador de proyecto.

1.7 HERRAMIENTAS

Las herramientas que se utilizarán para el desarrollo del presente proyecto son:

FIGURA N° 1. 1 HERRAMIENTAS DE DESARROLLO

SERVIDOR	SERVIDOR DE BASE DE DATOS	LENGUAJE DE PROGRAMACIÓN	HERRAMIENTA DE DISEÑO	FRAMEWORK
Apache	✓ MySQL ✓ MariaDB	PHP	✓ HTML ✓ JavaScript ✓ Css3 ✓ Ajax ✓ JQuery	✓ Codeigniter ✓ Bootstrap

Fuente: (Elaboración Propia)

1.7.1. Lenguaje de desarrollo

PHP: Es un lenguaje para programar scripts del lado del servidor, que se incrustan dentro del código HTML. Este lenguaje es gratuito y multiplataforma.

PHP es el acrónimo de Hipertext Preprocesor. Es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación. Según (Alvarez, Qué es PHP, 2001).

PHP se utiliza para generar páginas web dinámicas a aquellas cuyo contenido no es el mismo siempre. Se envían los datos de la solicitud al servidor que los procesa, reúne los datos (por eso decimos que es un 10 proceso dinámico) y el servidor lo que devuelve es una página HTML como si fuera estática.

FRAMEWORKS PHP: Hay una amplia gama de frameworks para aplicaciones web disponibles que utilizan PHP, entre los más conocidos destacan:

- ❖ Cakephp
- ❖ Zend framework
- ❖ Symfony
- ❖ Yii
- ❖ Phalcon
- ❖ Codeigniter
- ❖ Laravel

Framework De CodeIgniter: es un framework para el desarrollo de aplicaciones en PHP que utiliza el MVC. Permite a los programadores Web mejorar la forma de trabajar y hacerlo a mayor velocidad. Al igual que cualquier framework está pensado para gente que tiene un dominio, al menos medio, del lenguaje de programación PHP. Siempre hay que controlar PHP “a pelo” para empezar a trabajar de forma eficiente con este framework (o cualquier otro). Según (Colectiva, 2015).

JAVASCRIPT: JavaScript es un lenguaje de scripting multiplataforma y orientado a objetos. Es un lenguaje pequeño y liviano. JavaScript contiene una librería estándar de objetos, tales como Array, Date, y Math, y un conjunto central de elementos del lenguaje, tales como operadores, estructuras de control, y sentencias.

JavaScript es un lenguaje con muchas posibilidades, utilizado para crear pequeños programas que luego son insertados en una página web y en programas más grandes, orientados a objetos mucho más complejos. Con JavaScript podemos crear diferentes efectos e interactuar con nuestros usuarios. Según (Caballero, 2018)

1.7.2 Gestor de base de datos

MYSQL: Es el servidor de bases de datos relacionales más popular, desarrollado y MySQL es un sistema de administración de bases de datos con una colección estructurada de datos. MySQL es un sistema de gestión de bases de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. Una de las características más interesantes de MySQL es que permite recurrir a bases de datos multiusuario a través de la web y en diferentes lenguajes de programación que se adaptan a diferentes necesidades y requerimientos. Por otro lado, MySQL es conocida por desarrollar alta velocidad en la búsqueda de datos e información, a diferencia de sistemas anteriores. Las plataformas que utiliza son de variado tipo y entre ellas podemos mencionar LAMP, MAMP, SAMP, BAMP y WAMP (aplicables a Mac, Windows, Linux, BSD, Open Solaris, Perl y Python entre otras). Según (checa, 2018)

1.7.3 Herramientas de interfaz de diseño frontend

HTML5: es un lenguaje de marcado, las siglas de HTML significan Hyper Text Markup Language usado para estructurar y presentar el contenido para la web. Es uno de los aspectos fundamentales para el funcionamiento de los sitios, pero no es el primero. HTML5 es la última versión de HTML. El término representa dos conceptos diferentes:

- ❖ Se trata de una nueva versión de HTML, con nuevos elementos, atributos y comportamientos.
- ❖ Contiene un conjunto más amplio de tecnologías que permite a los sitios Web y a las aplicaciones ser más diversas y de gran alcance. A este conjunto se le llama HTML5 y amigos, a menudo reducido a HTML5

BOOTSTRAP: es un framework o conjunto de herramientas de Código abierto para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como, extensiones. 12 de JavaScript opcionales adicionales y se puede descargar de dos maneras, compilado o mediante el código fuente original. Según (Fontela, 2015)

AJAX: es una tecnología asíncrona, en el sentido de que los datos adicionales se requieren al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página. Ajax es una técnica válida para múltiples plataformas y utilizable en muchos sistemas operativos y navegadores, dado que está basado en estándares abiertos como JavaScript y Document Object Model (DOM). (instituto de nuevas tecnologías, 2017) El principal objetivo del AJAX, es intercambiar información entre el servidor y el cliente (navegadores) sin la necesidad de recargar la página. De esta forma, ganamos en usabilidad, experiencia y productividad del usuario final.

JQUERY: es una librería de componentes para el framework jQuery que provee un conjunto de plugins, efectos visuales. JQuery es una librería de JavaScript (JavaScript es un lenguaje de programación muy usado en desarrollo web). Esta librería de código abierto simplifica la tarea de programar en JavaScript y permite agregar interactividad a un sitio web sin tener conocimientos del lenguaje. Según (Angel, 2009).

1.8 LÍMITES Y ALCANCES

1.8.1 Límites

El presente proyecto se delimita al movimiento de venta de ladrillos y control del personal de la fábrica de cerámica “EMANUEL”, actualmente se realiza manualmente y eso provoca inseguridad en la cantidad exacta de salida en todas las ventas realizadas mensual y semanalmente.

El sistema web pretende facilitar estas desventajas y convertirlos en una actividad muy sencilla para el administrador y pueda realizar sus reportes diarios, semanales y mensuales de manera rápida y óptima.

El sistema web no emitirá facturas, a pedido del propietario solo emitirá comprobantes.

1.8.2 Alcances

El presente perfil de proyecto de grado de desarrollar un sistema de información a la fábrica de cerámica “EMANUEL”, estará aplicado al control de ingreso de materia prima y salida de ladrillos, principalmente al control de historial, actividades y salarios del personal.

El sistema deberá permitir a la administración de la fábrica, realizar de una forma más rápida y adecuada la gestión y administración de estas de forma que los tiempos entre los procesos actuales se vean afectados de forma positiva y la labor de distintas personas que intervienen en estos se beneficie de manera directa.

El proyecto contemplará los siguientes aspectos:

- ❖ Control de ventas: Se actualizarán los reportes en base a los métodos y modelos diarios, semanales o mensuales.
- ❖ Control de pedidos: Se encarga de registrar y controlar los pedidos de acuerdo a las reservas obtenidas.
- ❖ Módulo de control de asistencia.
- ❖ Control de planillas del personal.
- ❖ Venta de cerámica al cliente.
- ❖ Reportes y estadísticas: Servirá para la toma de decisiones a futuro de parte del dueño en cuanto a la venta de ladrillos y control del pago de personal y servicios de la fábrica.

1.9 APORTES

Los aportes que ofrecerá este proyecto será automatizar sus procesos de seguimientos de las actividades dentro la fábrica, reportes y control de ventas, de ingresos y salidas de mercadería lo cual permitirá minimizar y optimizar tiempos de ejecución generando información que coadyuve a la fácil y correcta toma de decisiones con respecto al manejo económico.

El principal aporte en el presente proyecto son los diferentes conocimientos adquiridos durante la formación en la Universidad Pública de El Alto en la carrera de Ingeniería de Sistemas.

Se aporta con un sistema web que contará con una herramienta de automatización a la medida de sus requerimientos, para un óptimo control de la información venta de ladrillos y control de personal.

El administrador contará con reportes los cuales le ayudará a dar una información rápida y oportuna a los compradores mayoristas.

CAPÍTULO II

MARCO TEÓRICO

2 MARCO TEÓRICO

En el siguiente capítulo se expone los fundamentos teóricos de la metodología y herramientas a usar durante el desarrollo de sistema, en el presente proyecto.

2.1 SISTEMA

Se entiende por un sistema a un conjunto ordenado de componentes relacionados entre sí, ya se trate de elementos materiales o conceptuales, dotado de una estructura, una composición y un entorno particulares. Se trata de un término que aplica a diversas áreas del saber, como la física, la biología y la informática o computación.

El mundo puede abordarse desde una perspectiva sistemática o sistematicista, en la que todos los objetos forman parte de algún tipo de sistema, desde las partículas de un átomo hasta la corteza cerebral, la democracia representativa o los números enteros. Visto así, un sistema no es otra cosa que un segmento de la realidad que puede estudiarse de manera independiente del resto, pero en el cual sus componentes se hallan interconectados.

Los sistemas son objeto de estudio de la Teoría de Sistemas o Teoría General de Sistemas, una disciplina que los aborda sean cuales sean desde una perspectiva múltiple, interdisciplinaria. Según ella cualquier sistema es reconocible dados sus límites y partes interrelacionadas e interdependientes (sus llamados subsistemas), a punto tal que la modificación de un elemento modifica necesariamente el funcionamiento del resto del sistema. Según (Raffino, 2019)

2.5 SOFTWARE

El término software es un vocablo inglés, que ha sido tomado por otros idiomas como el español para hacer referencia a determinados aplicativos en la informática. Este término designa al equipo lógico de una computadora, opuesto a los aspectos físicos de la misma.

El software está compuesto por un conjunto de programas que son diseñados para cumplir una determinada función dentro de un sistema, ya sean estos realizados por parte de los usuarios o por las mismas corporaciones dedicadas a la informática.

El concepto de software, como bien dijimos anteriormente, compone la parte lógica de un sistema de computación, permitiéndole el funcionamiento. Esto quiere decir entonces que no solo los programas son y forman un software, sino que la información del usuario y los datos procesados integran el software, ya que forma parte de él todo componente intangible y no físico.

2.2.1 Tipos principales de software

Un software de aplicación está diseñado para la realización de una o más tareas a la vez.

Software de sistema: Este grupo clasifica a los programas que dan al usuario la capacidad de relacionarse con el sistema, para entonces ejercer control por sobre el hardware. El software de sistema también se ofrece como soporte para otros programas. Ejemplos: sistemas operativos, servidores, etcétera.

Software de programación: Programas directamente diseñados como herramientas que le permiten a un programador el desarrollo de programas informáticos. Influyen en su utilización diferentes técnicas utilizadas y lenguaje de programación específico. Ejemplos: compiladores, editores multimedia, etcétera.

Software de aplicación: Programas diseñados para la realización de una o más tareas específicas a la vez, pudiendo ser automáticos o asistidos. Ejemplos: videojuegos, aplicaciones ofimáticas, etcétera. Según (Estela, software, 2019)

2.5 SISTEMA DE INFORMACIÓN

En informática, los sistemas de información ayudan a administrar, recolectar, recuperar, procesar, almacenar y distribuir información relevante para los procesos fundamentales y las particularidades de cada organización.

La importancia de un sistema de información radica en la eficiencia en la correlación de una gran cantidad de datos ingresados a través de procesos diseñados para cada área con el objetivo de producir información válida para la posterior toma de decisiones.

Características de un sistema de información

Un sistema de información se caracteriza principalmente por la eficiencia que procesa los datos en relación al área de acción. Los sistemas de información se alimentan de los procesos y herramientas de estadística, probabilidad, inteligencia de negocio, producción, marketing, entre otros para llegar a la mejor solución.

Un sistema de información se destaca por su diseño, facilidad de uso, flexibilidad, mantenimiento automático de los registros, apoyo en toma de decisiones críticas y mantener el anonimato en informaciones no relevantes. Según (Chen, 2008)

Componentes de un sistema de información

Los componentes que forman un sistema de comunicación son:

- ❖ la entrada: por donde se alimentan los datos,
- ❖ el proceso: uso de las herramientas de las áreas contempladas para relacionar, resumir o concluir,
- ❖ la salida: refleja la producción de la información, y
- ❖ la retroalimentación: los resultados obtenidos son ingresados y procesados nuevamente.

Actividad de sistema de información

Como sabemos todo Sistema de información es un conjunto de partes que se relacionan entre sí para lograr un objetivo que contribuirá a la toma de decisiones y esto es posible gracias a las diversas actividades que se llevan a cabo, lo cual estimula que la organización desarrolle sus procesos de una mejor manera, obteniendo así una eficiente labor. Los Sistemas de Información que realizan las diversas actividades principales haciendo uso de la computadora, lo cual permiten un mejor desempeño de los sistemas de información en las organizaciones públicas y/o privadas.

Ciclo de vida clásico de desarrollo de software

Ciclo de vida clásico o en cascada: El modelo de ciclo de vida en cascada es el modelo más simple en desarrollo de software. En él las etapas se llevan a cabo una detrás de otra de forma lineal, así sólo cuando la primera fase se termina se puede empezar con la segunda, y así progresivamente.

Este modelo asume que todo se lleva a cabo y tiene lugar tal y como se había planeado en la fase anterior, y no es necesario pensar en asuntos pasados que podrían surgir en la siguiente fase. Este modelo no funcionará correctamente si se dejan asuntos de lado en la fase previa. La naturaleza secuencial del modelo no permite volver atrás y deshacer o volver a hacer acciones.

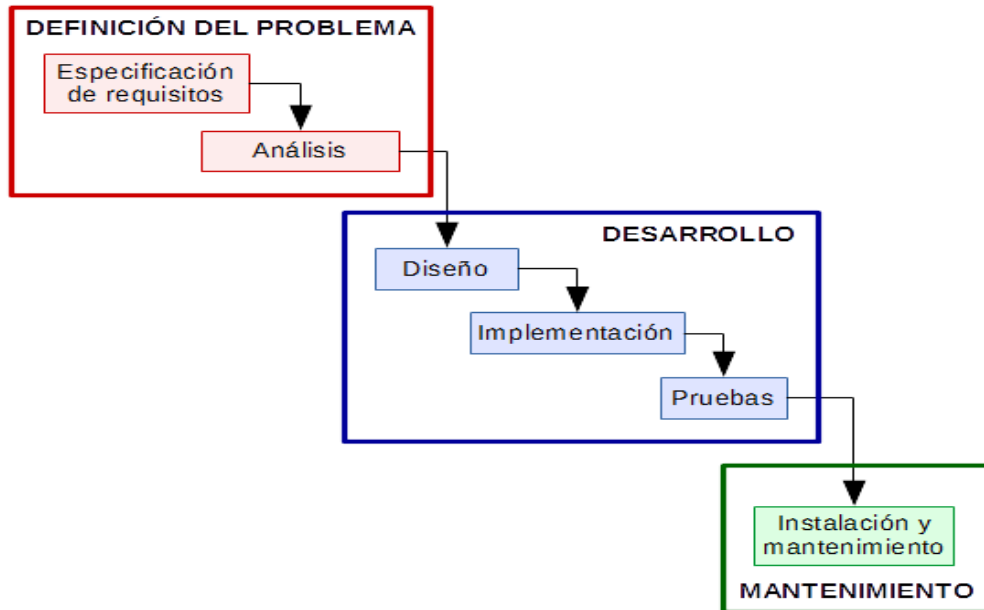
Este modelo es recomendable cuando el desarrollador ya ha diseñado y desarrollado aplicaciones similares con anterioridad, es decir, tiene la experiencia suficiente para terminar con una etapa y comenzar la siguiente. Según (Resan, 2016).

Son tres las fases en que se agrupan las etapas de este tipo de ciclo de vida:

- ❖ Definición del problema, que incluye tanto la especificación de requisitos como el análisis del sistema.
- ❖ Desarrollo, que abarca el diseño, implementación y pruebas del sistema.
- ❖ Mantenimiento, es decir, la instalación y el mantenimiento del sistema.

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

FIGURA Nº 2. 1 Ciclo de vida clásico de desarrollo de software



Fuente: (Resan, 2016).

WEB (WORLD WIDE WEB)

Una web es aquella que consiste en un documento electrónico que contiene información, cuyo formato se adapta para estar insertado en la World Wide Web, de manera que los usuarios a nivel mundial puedan entrar a la misma por medio del uso de un navegador, visualizándola con un dispositivo móvil como un smartphone o un monitor de computadora.

El formato de esta información puede ser HTML y desde una web se puede entrar a otra con enlaces de hipertexto. Los recursos para la presentación de la información de una web pueden ser scripts, imágenes digitales, hojas en forma de cascada, entre otros. Para que una web pueda estar en línea y ser vista por cualquier persona, requiere de un hospedaje, cosa que puede hacerse en un servidor remoto o local. La composición de una página web consiste básicamente en información que puede presentarse con módulos multimedia y texto, lo cual se complementa con hiperenlaces para conducir a otras secciones de dicha web o a otras webs. Según (Merino, 2013).

2.5 Definición de Venta

Del latín vendĭta, venta es la acción y efecto de vender (traspasar la propiedad de algo a otra persona tras el pago de un precio convenido). El término se usa tanto para nombrar a la operación en sí misma como a la cantidad de cosas que se venden. Por ejemplo: “La venta de tortas fue un éxito: hemos recaudado más de doscientos pesos”, “Mi padre me informó que la venta de la fábrica fue suspendida ante ciertas trabas legales”, “Juan Antonio se dedica a la venta de inmuebles”.

La venta también es el contrato a través del cual se transfiere una cosa propia a dominio ajeno por el precio pactado. La venta puede ser algo potencial (un producto que está a la venta pero que aún no ha sido comprado) o una operación ya concretada (en este caso, implica necesariamente la compra). Según (Porto, Definición de venta, 2013).

2.4.1 Control de Personal

Un adecuado sistema de control de personal es esencial para la buena administración en una empresa o un negocio, se trata de tener el control de entrada y salida de los empleados para mejorar la productividad, con seguimiento de horarios, grupos de acceso, zonas permitidas/restringidas y la certeza de un incremento en la productividad de la empresa.

El módulo de control de asistencia del personal le proporciona la capacidad configurar y administrar el sistema desde cualquier navegador web estándar, desde la red corporativa o a través de Internet. La información se genera en tiempo real y sin ningún hardware adicional o instalaciones de software. Por otra parte, todos los datos de comunicación entre el sistema de control de personal y la red son encriptados para proporcionar seguridad adicional a los usuarios. Según (Felipe, 2015)

2.5 METODOLOGÍA DE DESARROLLO

Metodologías de desarrollo de Software. Lograr la construcción de un sistema informático eficiente, que cumpla con los requerimientos planteados, es una tarea realmente intensa y sobre todo difícil de cumplir.

Las metodologías para el desarrollo del software imponen un proceso disciplinado sobre el desarrollo de software con el fin de hacerlo más predecible y eficiente. Una metodología de desarrollo de software tiene como principal objetivo aumentar la calidad del software que se produce en todas y cada una de sus fases de desarrollo. No existe una metodología de software universal, ya que toda metodología debe ser adaptada a las características de cada proyecto (equipo de desarrollo, recursos, etc.) exigiéndose así que el proceso sea configurable. Las metodologías de desarrollo se pueden dividir en dos grupos de acuerdo con sus características y los objetivos que persiguen: ágiles y robustas.

2.5.1 Metodología de UWE

Es una herramienta que nos permitirá modelar aplicaciones web, utilizada en la ingeniería web, prestando especial atención en sistematización y personalización (sistemas adaptativos). UWE es una propuesta basada en el proceso unificado y UML pero adaptados a la web. En requisitos separa las fases de captura, definición y validación. Hace además una clasificación y un tratamiento especial dependiendo del carácter de cada requisito. Según (Maximilians, 2016).

Es una herramienta que nos permitirá modelar aplicaciones web, utilizada en la ingeniería web, prestando especial atención en sistematización y personalización (sistemas adaptativos). UWE es una propuesta basada en el proceso unificado y UML pero adaptados a la web. En requisitos separa las fases de captura, definición y validación. Hace además una clasificación y un tratamiento especial dependiendo del carácter de cada requisito.

2.5.2 Actividades de modelado de UWE

Las actividades base de modelado de UWE son el análisis de requerimientos, el modelo conceptual, el modelo navegaciones y el modelo de presentación. A estos modelos se pueden sumar otros modelos como lo son el modelo de interacción y la visualización de Escenarios Web.

2.5.3 Característica de la metodología UWE

UML en una técnica de modelamiento propietaria, es la aceptación de UML en el proceso de desarrollo de software, la flexibilidad para la definición de un lenguaje de modelamiento específico en el dominio WEB, también llamado perfil UML.

UWE hace uso de notación UML pura y los tipos de diagramas UML en donde sea posible para el análisis y diseño de aplicaciones WEB.

2.5.4 Modelo de la metodología UWE

El modelo que propone UWE está compuesto por sub-modelos.

- ❖ **MODELO DE CASO DE USO:** Modelo para capturar los requisitos del sistema
- ❖ **MODELO DE CONTENIDO:** Es un modelo conceptual para el desarrollo del contenido.
- ❖ **MODELO DE NAVEGACIÓN:** En el cual se encuentra la presentación del sistema y el modelo de flujo.
- ❖ **MODELO DE ESTRUCTURA DEL PROCESO:** Incluye el modelo de la interfaz de usuario y el modelo de ciclo de vida del objeto.
- ❖ **MODELO DE PRESENTACIÓN:** En cuanto a los requisitos, UWE los clasifica dependiendo del carácter de cada uno. Además, distinguen entre las fases de captura, definición y validación de requisitos.

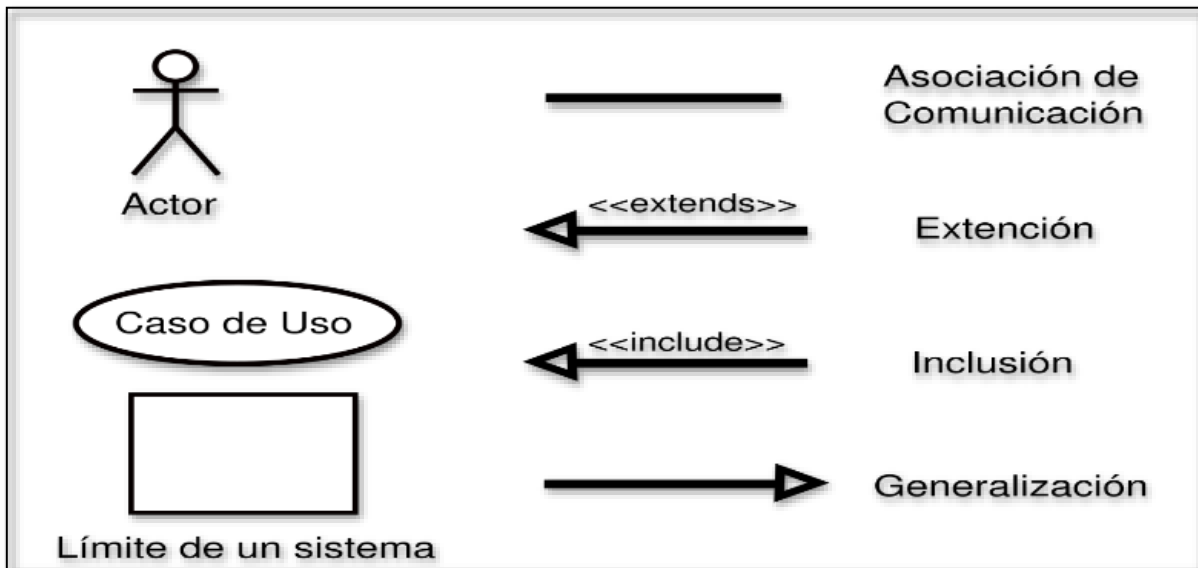
2.5.4.1 Modelo De Caso De Uso

Dentro de este ciclo se realiza un análisis del sistema construyendo para ello diagramas de casos de uso en una descripción de los pasos o las actividades que deberían realizarse para llevar a cabo algún proceso.

Los personajes o entidades que participaran en un caso de uso se denominan actores UWE provee diferentes estereotipos. Un caso de uso especifica el comportamiento de un sistema o una parte del mismo, y es una descripción de un conjunto de secuenciales de acciones, donde cada secuencia representa la interacción de los elementos externos del sistema (actores) con el propio sistema.

Un diagrama de casos de uso consta de los siguientes elementos:

FIGURA Nº 2. 2 Estereotipos de Diagrama de Caso de Uso

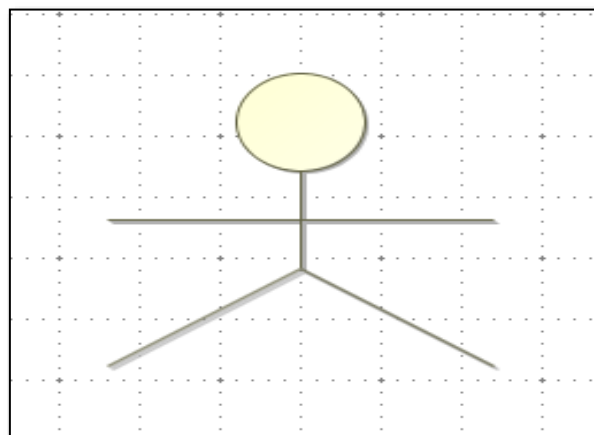


Fuente: (Elaboración Propia)

- ❖ Actor.
- ❖ Casos de Uso.
- ❖ Relaciones de Uso, Herencia y Comunicación.

ACTOR:

FIGURA Nº 2. 3 Actor

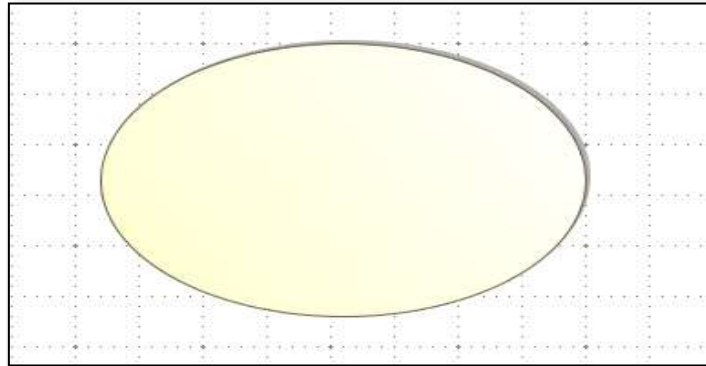


Fuente: (Soto, 2008)

Un Actor es un rol que un usuario juega con respecto al sistema. Es importante destacar el uso de la palabra rol, pues con esto se especifica que un Actor no necesariamente representa a una persona en particular, sino más bien la labor que realiza frente al sistema.

CASO DE USO:

FIGURA Nº 2. 4 Caso De Uso

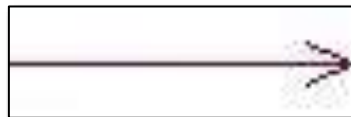


Fuente: (Soto, 2008)

Es una operación/tarea específica que se realiza tras una orden de algún agente externo, sea desde una petición de un actor o bien desde la invocación desde otro caso de uso.

ASOCIACIÓN

FIGURA Nº 2. 5 Asociación

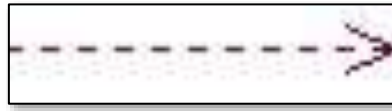


Fuente: (Soto, 2008)

Es el tipo de relación más básica que indica la invocación desde un actor o caso de uso a otra operación (caso de uso). Dicha relación se denota con una flecha simple.

DEPENDENCIA

FIGURA N° 2. 6 Dependencia o Instanciación

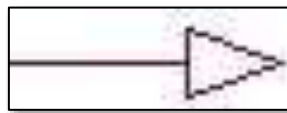


Fuente: (Soto, 2008)

Es una forma muy particular de relación entre clases, en la cual una clase depende de otra, es decir, se instancia (se crea). Dicha relación se denota con una flecha punteada.

GENERALIZACIÓN

FIGURA N° 2. 7 Generalización



Fuente: (Soto, 2008)

Este tipo de relación es uno de los más utilizados, cumple una doble función dependiendo de su estereotipo, que puede ser de Uso (<<uses>>) o de Herencia (<<extends>>).

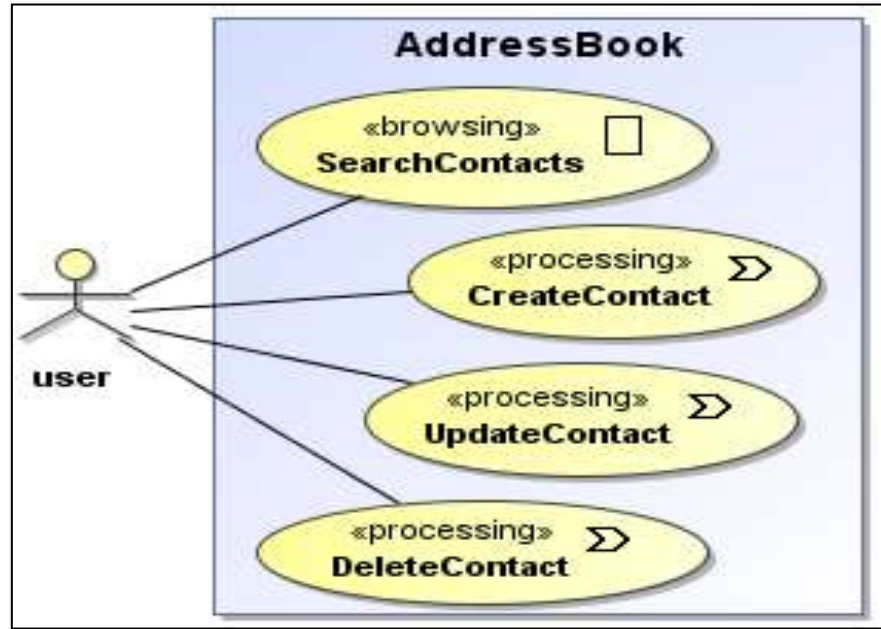
Este tipo de relación está orientado exclusivamente para casos de uso (y no para actores).

EXTENDS: Se recomienda utilizar cuando un caso de uso es similar a otro (características).

USES: Se recomienda utilizar cuando se tiene un conjunto de características que son similares en más de un caso de uso y no se desea mantener copiada la descripción de la característica.

De lo anterior cabe mencionar que tiene el mismo paradigma en diseño y modelamiento de clases, en donde está la duda clásica de usar o heredar.

FIGURA N° 2. 8 Diagrama de Caso de Uso en la Metodología UWE



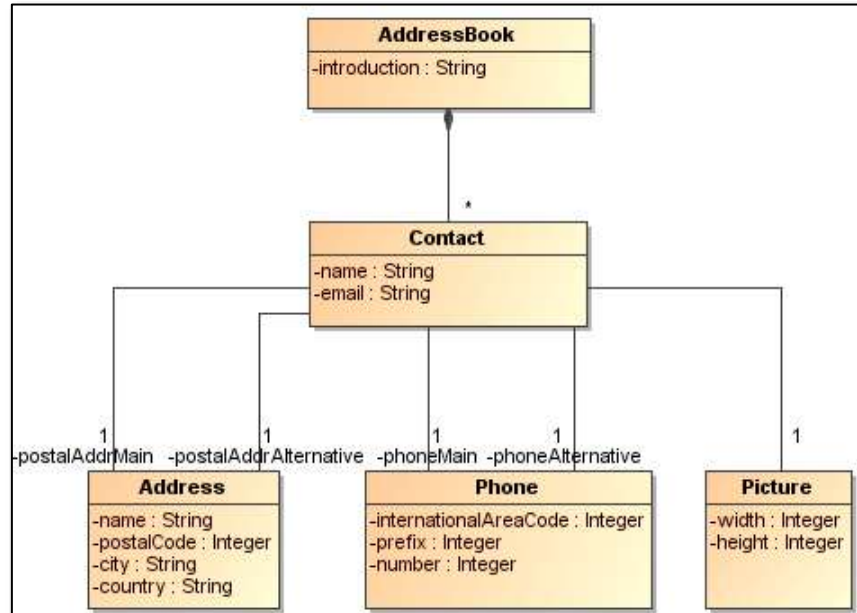
Fuente: (Soto, 2008).

2.5.4.2 Modelo de contenido

Dentro de este ciclo se modela el universo de la aplicación, creando para ello el modelo de dominio.

Diagrama de contenido. Este es un diagrama UML normal de clases, por ello debemos pensar en las clases que son necesarias. Los diagramas de clases se describen las estructuras estáticas de un sistema.

FIGURA N° 2. 9 Diagrama de Contenido en la Metodología UWE

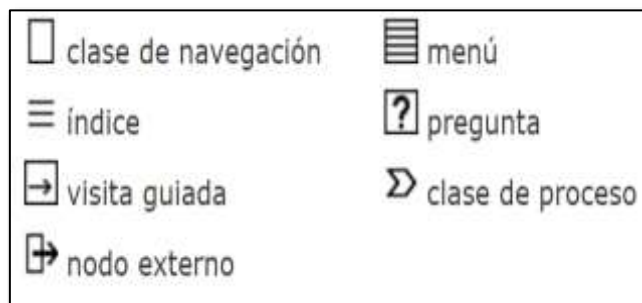


Fuente: (Soto, 2008).

2.5.4.3 Modelo de navegación

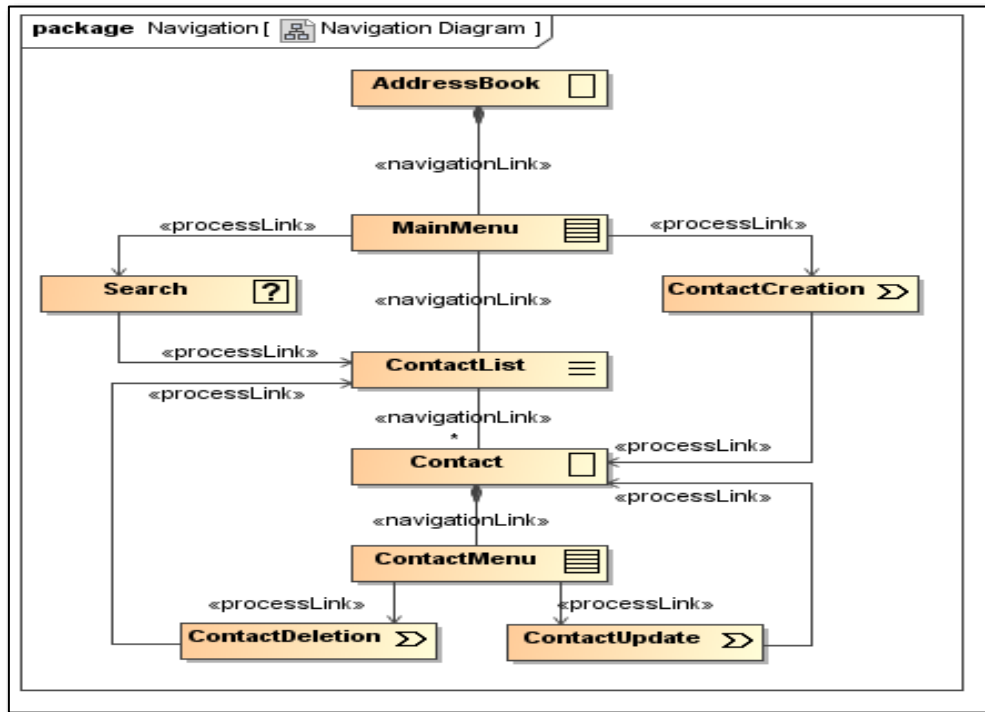
Dentro de este ciclo se define la navegación entre los distintos objetos del dominio. Para ello se construyen los modelos de Espacio de navegación y Estructura de navegación. Ello significa que necesitamos un diagrama conteniendo nodos y enlaces (links). ¿Pero que es un nodo? Nodos son unidades de navegación y están conectados por medio de enlaces. Nodos pueden ser presentados en diferentes páginas o en una misma página.

FIGURA N° 2. 10 Estereotipos de Diagrama de Navegación



Fuente: (Soto, 2008).

FIGURA N° 2. 11 Diagrama de Navegación en la Metodología UWE



Fuente: (Soto, 2008).

2.5.4.4 Modelo de estructura del proceso

En este modelo se define las acciones que realizan las clases de proceso especificando en el modelo de navegación, como se explicó anteriormente, el modelo de Proceso se divide en dos partes.

MODELO DE ESTRUCTURA DEL PROCESO:

En el cual se incluye las relaciones entre clases de proceso, se crea un diagrama de clases donde cada una se presente con un estereotipo de clases de proceso. Se conjuntan estas clases y se asocian a una superclase que representa el proceso, se agrega algunas clases si es necesario para denotar algunas interacciones u operaciones en común.

MODELO DE FLUJO DEL PROCESO:

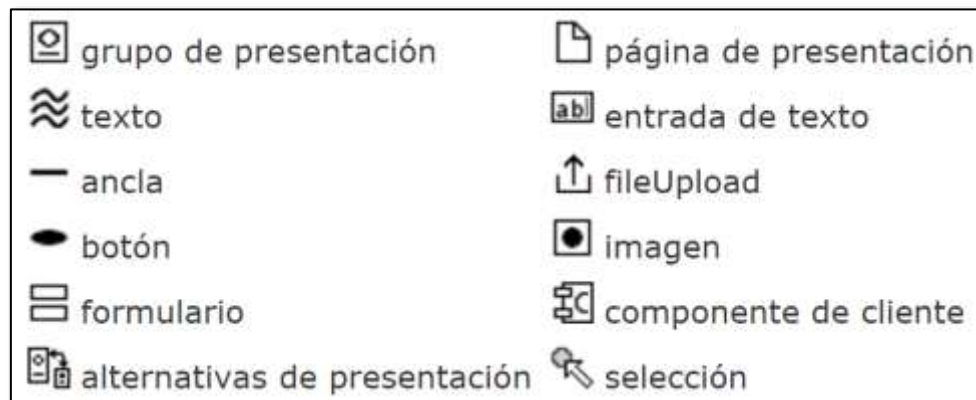
La conducta de un proceso es representando mediante un diagrama de actividades UML, describiendo el flujo de una clase de proceso, lo que sucede cuando un usuario navegación hacia una clase de proceso.

2.5.4.5 Modelo de presentación

La presentación se describe en función de distintos modelos estándares UML. El Modelo de Navegación no indica cuáles son las clases de navegación y de proceso que pertenecen a una página web. Podemos usar un Diagrama de Presentación con el fin de proveer esta información la metodología UWE presenta diferentes estereotipos.

Las propiedades pueden anidarse, por ejemplo, cada contacto cubre diferentes textos y botones. Ello significa, que para cada contacto la correspondiente dirección de correo y los correspondientes campos de teléfonos y direcciones serán visualizados en la página.

FIGURA N° 2. 12 Estereotipos de Diagrama de Presentación



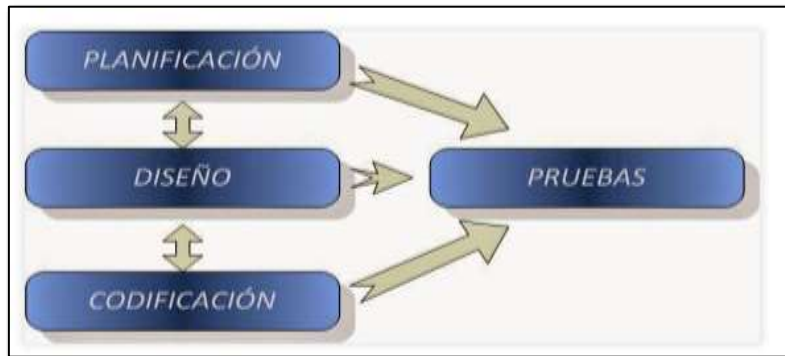
Fuente: (Soto, 2008).

En los siguientes diagramas, los estereotipos son solamente representados por sus iconos. En MagicDraw se puede configurar la visualización de ambos: nombres e iconos de los estereotipos. Por ello, los estereotipos presentan alternativas y es usado nuevamente para evitar los múltiples modelos de todo el contenido de ambos formularios de ingreso de datos.

2.5.5 Fase de la metodología UWE

UWE cubre todo el ciclo de vida de este tipo de aplicaciones centrandose además su atención en aplicaciones personalizadas o adaptativas. Las fases o etapas a utilizar son:

FIGURA Nº 2. 13 FASE DE LA METODOLOGÍA UWE



Fuente: (Quiroga, 2015)

2.5.5.1 Captura, análisis y especificación de requisitos

En simple palabras y básicamente, durante esta fase, se adquieren, reúnen y especifican las características funcionales y no funcionales que deberá cumplir la aplicación web.

Trata de diferente forma las necesidades de información, las necesidades de navegación, las necesidades de adaptación y las de interfaz de usuario, así como algunos requisitos adicionales. Centra el trabajo en el estudio de los casos de uso, la generación de los glosarios y el prototipado de la interfaz de usuario.

2.5.5.2 Diseño del sistema

Se basa en la especificación de requisitos producido por el análisis de los requerimientos (fase de análisis), el diseño define cómo estos requisitos se cumplirán, la estructura que debe darse a la aplicación web.

2.5.5.3 Codificación del software

Durante esta etapa se realizan las tareas que comúnmente se conocen como programación; que consiste, esencialmente, en llevar a código fuente, en el lenguaje de programación elegido, todo lo diseñado en la fase anterior.

2.5.5.4 Pruebas

Las pruebas se utilizan para asegurar el correcto funcionamiento de secciones de código del sistema.

2.5.5.5 La Instalación o fase de Implementación

Proceso por el cual los programas desarrollados son transferidos apropiadamente al computador destino, inicializados, y, eventualmente, configurados; todo ello con el propósito de ser ya utilizados por el usuario final.

Esto incluye la implementación de la arquitectura, de la estructura del hiperespacio, del modelo de usuario, de la interfaz de usuario, de los mecanismos adaptativos y las tareas referentes a la integración de todas estas implementaciones.

2.5.5.6 Mantenimiento

Es el proceso de control, mejora y optimización del software ya desarrollado e instalado, que también incluye depuración de errores y defectos que puedan haberse filtrado de la fase de pruebas de control.

2.6 HERRAMIENTAS DE DESARROLLO

2.6.1 Servidor apache

Apache es un software especializado en ofrecer servicios de servidor web. Es versátil, ligero y muy útil, además de ser completamente gratuito y de código abierto. Su popularidad es tal que, actualmente, cerca del 50% de las páginas web de todo el mundo se ejecutan en un servidor de este tipo.

Aunque se le conoce así, su nombre completo es Apache HTTP Server, y sus responsables tienen también un nombre similar: Apache Software Foundation. Esta es la firma responsable de todo el código que da forma a este software para servidores que cualquiera puede utilizar sin necesidad de pagar, como también modificar a su total antojo al ser completamente abierto.

En este artículo compartimos y explicamos de forma detallada cómo instalar en una computadora de escritorio o Laptop el servidor Apache.

Los parámetros que son necesario modificar para que funcione correctamente en su forma elemental y posteriormente agregarle módulos y otras funcionalidades, de acuerdo a las necesidades y propósitos de quien lo vaya a utilizar. Como hacer funcionar Apache de forma similar a un servidor de internet y cargar las páginas en nuestro navegador offline, como si estuviéramos conectados a la red.

¿Qué es un servidor web?

Un servidor web como su nombre lo indica, es un software instalado en el equipo con todas las condiciones necesarias para servir o entregar páginas web que le sean solicitadas por un navegador, asegurando que se muestren y representen todos los elementos necesarios para su correcto funcionamiento y visualización. Existen varios tipos de servidores web, Apache es un software de código abierto, libre de uso y totalmente configurable, es en este momento el más utilizado en la red, ya sea en plataformas Linux o Windows.

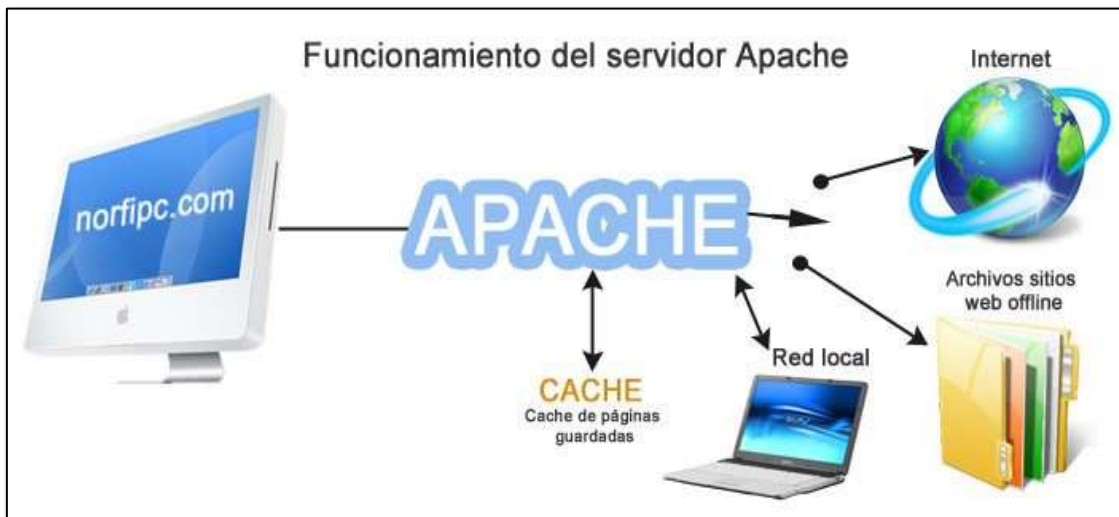
Al instalarlo en nuestra PC dispondremos de un servidor completo, con todos los requisitos para ejecutarlo de forma local. Según (Bernal., 2019).

FIGURA N° 2. 14 Servidor web



Fuente: (Bernal., 2019)

FIGURA N° 2. 15 Servidor apache



Fuente: (Bernal., 2019)

2.6.2 Base de datos

El concepto de base, procedente del latín basis, tiene múltiples usos. El término puede utilizarse con referencia al sostén o fundamento de algo. Un dato, por otra parte, es una información concreta, un testimonio, una prueba o una documentación.

Con estas definiciones ya podemos detallar los alcances de la noción de base de datos, que se emplea con mucha frecuencia en el ámbito de la informática. Se conoce como base de datos al conjunto de informaciones que está organizado y estructurado de un modo específico para que su contenido pueda ser tratado y analizado de manera rápida y sencilla. Según (Estela, Base de datos, 2019)

Las bases de datos, por lo tanto, presentan datos estructurados de acuerdo a diferentes parámetros. Al disponer la información de una cierta forma, el usuario puede encontrar aquello que busca con facilidad, a diferencia de lo que le sucedería si todos los datos estuvieran mezclados y sin ningún tipo de orden.

El nombre de un campo generalmente identifica la información almacenada en el campo. Por ejemplo, los campos pueden llamarse Nombre, Dirección o Número telefónico. Cada campo tiene un tipo que identifica la clase de información que puede almacenar: números, fechas, caracteres alfanuméricos y otros. Como cada campo contiene un tipo específico de datos, usted puede realizar cálculos y otras operaciones con la información guardada en ellos. Por ejemplo, puede sumar los números de dos campos. Puede comparar la fecha de un campo con la de otro. Puede mostrar el nombre de una persona (almacenado en un campo) después de su apellido (almacenado en otro campo) para construir la primera línea de una etiqueta de correo.

El conjunto de registros que utilizan los mismos campos conforma una tabla. Una base de datos puede contener muchas tablas. La siguiente imagen muestra cómo se relacionan estos conceptos.

FIGURA N° 2. 16 Conceptos básicos de bases de datos



Fuente: (Bembibre, 2009)

2.6.2.1 Gestor de base de datos Mariadb

MariaDB es un sistema de base de datos que proviene de MySQL, pero con licencia GPL, desarrollado por Michael Widenius, fundador de MySQL y la comunidad de desarrolladores de software libre.

Como hemos comentado anteriormente, MariaDB se trata de una “sustituto” de MySQL que corre bajo una licencia GPL. Por lo tanto, incorpora todas las mejoras de la versión de MySQL en la que se encuentre y además incluye diferentes mejoras:

Lo primero que tiene que quedar totalmente claro es que al ser MariaDB compatible con MySQL, la migración a MariaDB es simple y directa, no hay que adaptar el código ni nada.

Ventajas de MariaDB.

Nuevos motores de almacenamiento, para la mayoría de usuarios lo interesante es, que viene a reemplazar a MyISAM y también tenemos XtraDB que reemplaza a InnoDB. Los nuevos motores de almacenamiento son:

- ❖ Aria: Un motor de almacenamiento a prueba de fallos basado en MyISAM.
- ❖ XtraDB: El reemplazo del motor InnoDB basado en el plug-in de InnoDB.
- ❖ PBXT: Un motor de almacenamiento transaccional con una gran cantidad de nuevas y bonitas características.
- ❖ FederatedX: El reemplazo del motor Federated.

Mejoras de velocidad sobre todo en consultas complejas cuando se usa el motor de almacenamiento Aria, ya que Aria cachea los datos de tablas temporales en memoria, lo que supone un rendimiento frente al uso del disco duro (que es lo que emplea MyISAM).

Se añaden nuevas tablas de sistema (INFORMATION_SCHEMA) para almacenar estadísticas que nos pueden ayudar a optimizar las bases de datos.

El sistema para manejar las conexiones se ha mejorado, ya que implementa el sistema pool-of-threads de MySQL 6.0 con el que podemos tener más de 200.000 conexiones a MariaDB.

En general se han hecho muchas modificaciones para mejorar el rendimiento, velocidad e incluso implementar características nuevas.

Desventajas de MariaDB.

La única desventaja es la propia de tener que desinstalar MySQL e instalar MariaDB, pero la migración es muy sencilla. Según (Ramos, 2018)

2.6.3 Lenguaje de programación php

PHP son las siglas en inglés de “Hypertext Pre-Processor” que al traducirlo al español pierde un poco el sentido, mejor lo analizamos y encontramos que significa “Lenguaje de Programación Interpretado”. Este lenguaje es el que le debemos la visualización de contenido dinámico en las páginas web. Todo el código PHP es invisible para el usuario, porque todas las interacciones que se desarrollan en este lenguaje son por completo transformadas para que se puedan ver **imágenes**, variedad de multimedia y los formatos con los que somos capaces de interactuar añadiendo o descargando información de ellos.

Un lenguaje del lado del servidor es aquel que se ejecuta en el servidor web, justo antes de que se envíe la página a través de Internet al cliente. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente.

El cliente solamente recibe una página con el código HTML resultante de la ejecución de la PHP. Según (Alvarez, Qué es PHP, 2001).

VENTAJAS Y DESVENTAJAS.

Ventajas:

- ❖ Es un lenguaje multiplataforma.
- ❖ Completamente orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una Base de Datos.
- ❖ El código fuente escrito en PHP es invisible al navegador y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable.
- ❖ Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con varios gestores de base de datos.
- ❖ Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones).

Desventaja:

Como es un lenguaje que se interpreta en ejecución para ciertos usos puede resultar un inconveniente que el código fuente no pueda ser ocultado. La ofuscación es una técnica que puede dificultar la lectura del código, pero no la impide y, en ciertos casos, representa un costo en tiempos de ejecución. Según (MARTINEZ, 2011)

2.6.3.1 Framework CODEIGNITER

CodeIgniter es un framework para el desarrollo de aplicaciones en PHP, que utiliza el MVC. Esto permite a los programadores o desarrolladores Web mejorar su forma de trabajar, además de dar una mayor velocidad a la hora de crear páginas Webs.

¿Qué es MVC?

El MVC o Modelo Vista Controlador es un patrón de arquitectura de software que separa la lógica de control, la interfaz del usuario y los datos del sistema. Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir por un lado define los componentes para la representación de la información y por otro lado la interacción del usuario. Según (Fontán, 2012).

- ❖ **Modelo:** se trata de la capa que trabaja con los datos, por lo que tiene los mecanismos para acceder a la información y también actualizar su estado. Estos datos estarán habitualmente en una base de datos.
- ❖ **Vista:** esta capa contiene el código de la aplicación que va a producir la visualización de las interfaces de usuario, es decir, el código que permitirá renderizar los estados de nuestra aplicación en HTML.
- ❖ **Controlador:** contiene el código necesario para responder a las acciones que solicita la aplicación, como por ejemplo realizar una compra o visualizar un elemento. Esta capa sirve de enlace entre la vista y el modelo.

Algunas ventajas de usar el framework CodeIgniter:

- ❖ Las páginas se procesan más rápido, el núcleo de CodeIgniter es bastante ligero.
- ❖ Es sencillo de instalar, basta con subir los archivos al ftp y tocar un archivo de configuración para definir el acceso a la base de datos.
- ❖ Existe abundante documentación en la red.
- ❖ Facilidad de edición del código ya creado.
- ❖ Facilidad para crear nuevos módulos, páginas o funcionalidades.
- ❖ Estandarización del código
- ❖ Separación de la lógica y arquitectura de la web, el MVC.
- ❖ Cualquier servidor que soporte PHP + MySQL sirve para CodeIgniter.
- ❖ CodeIgniter se encuentra bajo una licencia open source, es código libre.

2.6.4 Herramienta de diseño

2.6.4.1 Html5

HTML5 (HyperText Markup Language) es la quinta revisión del lenguaje de marcado estándar que se emplea para la web. Es uno de los lenguajes de marcado más usados en todo el mundo y la razón es bastante obvia: gracias a HTML5 podemos crear la estructura de una página web. Texto, imágenes y material multimedia pueden mostrarse correctamente gracias a HTML5.

HTML5 es la última versión de HTML. El término representa dos conceptos diferentes:

- ❖ Se trata de una nueva versión de HTML, con nuevos elementos, atributos y comportamientos.
- ❖ Contiene un conjunto más amplio de tecnologías que permite a los sitios Web y a las aplicaciones ser más diversas y de gran alcance. A este conjunto se le llama HTML5 y amigos, a menudo reducido a HTML5.

Diseñado para ser utilizable por todos los desarrolladores de Open Web, esta página referencia numerosos recursos sobre las tecnologías de HTML5, clasificados en varios grupos según su función.

- ❖ Semántica: Permite describir con mayor precisión cuál es su contenido.
- ❖ Conectividad: Permite comunicarse con el servidor de formas nuevas e innovadoras.
- ❖ Sin conexión y almacenamiento: Permite a las páginas web almacenar datos localmente en el lado del cliente y operar sin conexión de manera más eficiente.
- ❖ Multimedia: Nos otorga un excelente soporte para utilizar contenido multimedia como lo son audio y video nativamente.
- ❖ Gráficos y efectos 2D/3D: Proporciona una amplia gama de nuevas características que se ocupan de los gráficos en la web como lo son canvas 2D, WebGL, SVG, etc.
- ❖ Rendimiento e Integración: Proporciona una mayor optimización de la velocidad y un mejor uso del hardware.
- ❖ Acceso al dispositivo: Proporciona APIs para el uso de varios componentes internos de entrada y salida de nuestro dispositivo.
- ❖ CSS3: Nos ofrece una nueva gran variedad de opciones para hacer diseños más sofisticados. Según (Martin, 2019)

2.6.4.2 Ccss3

CSS es un lenguaje de diseño gráfico que permite definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado. Es muy usado para establecer el diseño visual de los documentos web e interfaces de usuario escritas en HTML.

CSS3 es una tecnología que ha tenido una evolución en el tiempo, que actualmente se encuentra en su versión 3, como su propio nombre indica.

Sus siglas corresponden a “Cascading Style Sheets”, que tiene el siguiente significado:

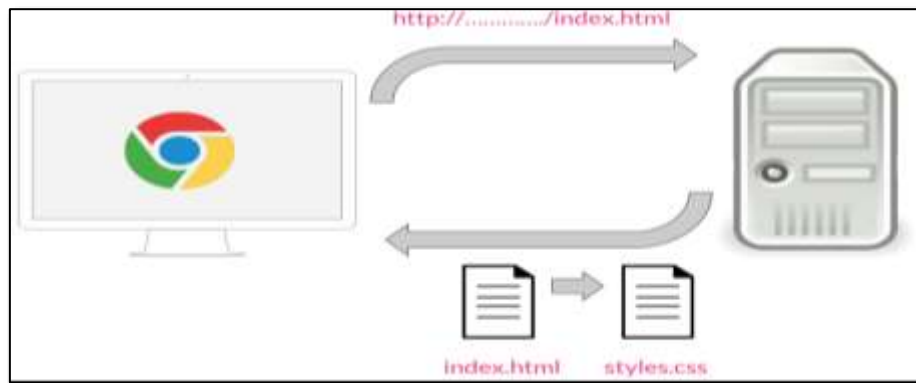
- ❖ Cascading, que significa que los estilos que aplicamos a los elementos de una página web se propagan a los elementos que contiene, se propagan en cascada.
- ❖ Style, porque mediante CSS lo que hacemos es aplicar estilos visuales a los distintos elementos de nuestra página web.
- ❖ Sheets, que significa hojas, porque los estilos de una página web se añaden en ficheros aparte, en ficheros con la extensión .css de manera general.

CSS ha ido evolucionando desde hace 25 años a la versión actual, que es CSS3, pero su función es la misma, sirve para dar estilos visuales a las páginas web. Según (Jiménez, 2019)

Funcionamiento

Vamos a ver en un pequeño esquema qué es lo que sucede desde que se solicita una página web hasta que se le aplica el estilo. El proceso es el siguiente:

FIGURA N° 2. 17 Funcionamiento de css3



Fuente: (Jiménez, 2019)

2.6.4.3 JQuery

Se denomina jQuery a una librería o biblioteca de JavaScript que facilita la programación en este lenguaje. Por lo general se emplea para añadir elementos interactivos a una página web sin necesidad de tener que programar demasiado.

En el caso de jQuery, la biblioteca es de código abierto (todos pueden contribuir al desarrollo ya que el acceso al código fuente es libre) y está basada en el lenguaje JavaScript. Lo que hace es ofrecer diversas funcionalidades que, de otro modo, exigirían desarrollar más código, por lo tanto ayudan a ahorrar tiempo. Según (Porto, 2018)

2.6.4.4 Javascript

JavaScript consiste en un lenguaje de programación interpretado, que habitualmente se utiliza en sitios web para ejecutar acciones en el lado del cliente, estando embebido en el código fuente de la página web

Técnicamente, constituye un dialecto del estándar ECMAScript, propuesto por la entidad internacional de estándares de información y comunicación ECMA International y diseñado inicialmente por Netscape y, posteriormente, por la Fundación Mozilla. También constituye un estándar ISO.

Si bien recibe en su sintaxis y forma de trabajo una mayor herencia del lenguaje C, también adopta nombres y convenciones que encontramos en Java, aunque se quedan en eso, en parecidos “cosméticos”.

Debido a su propósito y uso general, todos los navegadores web modernos interpretan correctamente JavaScript, siendo un lenguaje universal y multiplataforma. Y, habitualmente, su uso por parte de los desarrolladores se relaciona con la interfaz de usuario presentada por el sitio web.

JavaScript permite, en una página web, crear elementos como cuadros de diálogo, recoger información entrada por el usuario y mandarla al servidor para ser procesada.

Con el tiempo, JavaScript ha evolucionado, incluyendo funcionalidades de otros lenguajes más avanzados, como clases para programación orientada a objetos, o módulos para la reutilización de código fuente. Según (González, 2016).

FIGURA N° 2. 18 Definición Javascript



Fuente: (González, 2016)

Pero pese a estos cambios y actualizaciones, JavaScript mantiene la compatibilidad hacia atrás:

- ❖ Es un lenguaje estructurado e imperativo, tomando como modelo la estructura de programación del lenguaje C
- ❖ Tipos dinámicos, de forma que cuando definimos el valor de la variable, se define el tipo de datos
- ❖ Orientado a objetos, con funciones que son consideradas como objetos
- ❖ Los objetos y métodos mediante los que el lenguaje puede interactuar con el usuario se encuentran limitados por el entorno de ejecución como, por ejemplo, el browser
- ❖ De igual forma que Perl, es compatible con expresiones regulares

2.6.4.5 Ajax

AJAX son las siglas de Asynchronous JavaScript And XML, (Javascript asíncrono y XML). No es en sí un lenguaje de programación, sino una nueva técnica que combina varios lenguajes de programación.

La ventaja de ajax respecto a otros lenguajes de programación web es la asincronía. Esto consiste en que cuando queremos intercambiar datos con el servidor (por ejemplo enviar o comprobar un formulario, consultar una base de datos, etc), la página no se queda parada esperando la respuesta, sino que se pueden seguir ejecutando acciones mientras tanto.

Con ajax podemos crear páginas interactivas. En éstas solicitamos datos al servidor, los cuales podemos tener guardados en otras páginas o en bases de datos. El servidor devuelve los datos, los cuales se cargan en la misma página y en segundo plano. Lo de "segundo plano" significa que mientras esperamos que se reciban los datos la página no se queda parada, y el usuario o la programación de la página puede seguir haciendo otras cosas.

Para poder entender este manual debemos tener conocimientos de HTML, CSS, y Javascript. Es conveniente también tener nociones de XML y acceso al DOM. Para las últimas páginas necesitamos también tener conocimientos de PHP, ya que es el lenguaje que se emplea para procesar los datos en el servidor. Según (Alsina, 2018)

2.6.4.6 Framework Bootstrap

Bootstrap es conjunto conceptos, prácticas y criterios (framework) desarrollado por Mark Otto y Jacob Thornton dentro de Twitter con la intención de estandarizar el conjunto de herramientas que utilizaban todos los involucrados en el desarrollo del front-end.

Bootstrap, es un framework originalmente creado por Twitter, que permite crear interfaces web con CSS y JavaScript, cuya particularidad es la de adaptar la interfaz del sitio web al tamaño del dispositivo en que se visualice. Es decir, el sitio web se adapta automáticamente al tamaño de una PC, una Tablet u otro dispositivo. Esta técnica de diseño y desarrollo se conoce como “responsive design” o diseño adaptativo.

De esta manera crearon un conjunto de librerías JavaScript y CSS que toda la compañía debía usar evitando que las partes desarrolladas por un equipo no pudiesen ser mantenidas por otros.

Bootstrap nos ayuda a maquetar un sitio web con rapidez y, sobretodo, ayudándonos a que el diseño sea correcto y usable tanto en dispositivos convencionales como en los táctiles (responsive web design). Para hacerlo, nos ofrece una serie de estilos CSS y librerías JavaScript que nos ayudarán de una manera rápida a desarrollar nuestro sitio web y sobretodo es recomendable para el desarrollo de prototipos y tener un tiempo de respuesta realmente bueno.

La base de Bootstrap es un diseño de rejilla dividido en 12 columnas con un ancho total de 940 píxeles. De esta manera, la maquetación se simplifica haciendo que solamente se deban encajar correctamente nuestras secciones dentro de esta parrilla y Bootstrap se encargará de que la visualización sea correcta en todos los dispositivos. Nos ahorramos, todo el trabajo de revisar con varios navegadores, el desarrollo de CSS específicos para móviles, etc. Según (Ruiz, 2014)

2.7 MÉTRICA DE CALIDAD DE SOFTWARE

- ❖ **Calidad de software:** "La calidad del software es el grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario". (IEEE, Std. 610-1990). "Concordancia del software producido con los requerimientos explícitamente establecidos, con los estándares de desarrollo prefijados y con los requerimientos implícitos no establecidos formalmente, que desea el usuario". Según (Pressman, 1998).
- ❖ **Calidad:** La calidad es un término que actualmente se encuentra en multitud de contextos y con el que se busca despertar en quien lo escucha una sensación positiva, transmitiendo la idea de que algo es mejor. La palabra calidad tiene muchos significados. La calidad de un producto o servicio es la percepción que el cliente tiene del mismo, es una fijación mental del consumidor que asume conformidad con un producto o servicio determinado, que solo permanece hasta el punto de necesitar nuevas especificaciones. La calidad es la capacidad de un producto o servicio para satisfacer las necesidades del cliente o usuario.

MÉTRICA: Históricamente se habló de métrica en referencia a los sistemas que existían para escribir versos diferenciados en base al número de sílabas que contenía cada verso, así como en referencia al estudio y "medición" de la cantidad de sílabas y estrofas que contenían los versos.

En informática, el término métrica hace referencia a la medición del software en base a parámetros predeterminados, como puede ser el número de líneas de código de que consta o el volumen de documentación asociada. A veces en vez de hablar de métrica se usa el término "Indicadores" del software. Algunos ingenieros lo usan como sinónimos mientras que otros les atribuyen significados distintos.

Algunas métricas o indicadores pueden ser:

a) Índice de productividad = tamaño / esfuerzo = líneas de código generado / horas trabajadas.

b) Tasa de defectos = defectos / tamaño = número de errores / líneas de código generadas.

❖ **Métricas de calidad del software:** Los sistemas de métricas de calidad del software tradicionales se han centrado fundamentalmente en las métricas de procesos, de productos y de recursos. Los sistemas de métricas hoy en día son los usados en los perfiles o aplicaciones para probar las aplicaciones. Este tipo de aplicaciones usan sistemas de métricas en tiempo de ejecución para medir tiempos, buscar cuellos de botella en las aplicaciones, medir capacidades máximas, etcétera. Así, las métricas tratan de servir de medio para entender, monitorizar, controlar, predecir y probar el desarrollo software y los proyectos de mantenimiento.

Los tres objetivos fundamentales de la medición son:

1. Entender qué ocurre durante el desarrollo y el mantenimiento.
2. Controlar qué es lo que ocurre en nuestros proyectos.
3. Mejorar nuestros procesos y nuestros productos.

2.7.1 ISO 9000

La serie de Normas ISO 9000 son un conjunto de enunciados, los cuales especifican que elementos deben integrar el Sistema de Gestión de la Calidad de una Organización y como deben funcionar en conjunto estos elementos para asegurar la calidad de los bienes y servicios que produce la Organización. Al hablar de Organización nos estamos refiriendo a una Empresa, Compañía o cualquier Estructura Organizada que genere o comercialice productos o servicios de algún tipo.

2.7.2 Estándar ISO/IEC 9126

Esta norma Internacional fue publicada en 1992, la cual es usada para la evaluación de la calidad de software, llamado “Information technology-Software product evaluation-Quality characteristics and guidelines for their use”; o también conocido como ISO 9126 (o ISO/IEC 9126).

Este estándar describe 6 características generales: Funcionalidad, Confiabilidad, Usabilidad, Eficiencia, Mantenibilidad, y Portabilidad.

La norma ISO/IEC 9126 permite especificar y evaluar la calidad del software desde diferentes criterios asociados con adquisición, requerimientos, desarrollo, uso, evaluación, soporte, mantenimiento, aseguramiento de la calidad y auditoria de software.

2.7.3 Métricas de Calidad del Modelo de ISO-9126

El estándar ISO/IEC 9126 ha sido desarrollado en un intento de identificar los atributos clave de calidad para un producto de software (Pressman, 2002).

Este estándar es una simplificación del Modelo de McCall (Losavio, 2003), e identifica seis características básicas de calidad que pueden estar presentes en cualquier producto de software. El estándar provee una descomposición de las características en sub características, que se muestran en la Tabla 2.1

Características de Calidad – Modelo ISO/IEC 9126

Características	Sub Características	
Funcionalidad	<ul style="list-style-type: none"> ❖ Adecuacion ❖ Exactitud ❖ Interoperatvidad 	<ul style="list-style-type: none"> ❖ Seguridad ❖ Conformidad
Fiabilidad	<ul style="list-style-type: none"> ❖ Madurez ❖ Tolerancia a fallas 	<ul style="list-style-type: none"> ❖ Recuperabilidad ❖ Conformidad
Usabilidad	<ul style="list-style-type: none"> ❖ Comprensibilidad ❖ Facilidad de Aprendizaje ❖ Operabilidad 	<ul style="list-style-type: none"> ❖ Atracción ❖ Conformidad
Eficiencia	<ul style="list-style-type: none"> ❖ Comportamiento Temporal ❖ Utilización de recursos 	<ul style="list-style-type: none"> ❖ Conformidad
Mantenibilidad	<ul style="list-style-type: none"> ❖ Analizabilidad ❖ Cambiabilidad ❖ Estabilidad 	<ul style="list-style-type: none"> ❖ Capacidad de prueba ❖ Conformidad
Portabilidad	<ul style="list-style-type: none"> ❖ Adaptabilidad ❖ Facilidad de Instalación ❖ Coexistencia 	<ul style="list-style-type: none"> ❖ Remplazabilidad ❖ Conformidad

Fuente: (Pressman, 2002)

Al unir la calidad interna y externa con la calidad en uso se define un modelo de evaluación más completo, se puede pensar que la usabilidad del modelo de calidad externa e interna pueda ser igual al modelo de calidad en uso, pero no, la usabilidad es la forma como los profesionales interpretan o asimilan la funcionabilidad del software y la calidad en uso se puede asumir como la forma que lo asimila o maneja el usuario final. Si se unen los dos modelos, se puede definir que los seis indicadores del primer modelo tienen sus atributos y el modelo de calidad en uso sus 4 indicadores pasarían hacer sus atributos, mirándolo gráficamente quedaría así:

Norma de Evaluación ISO/IEC 9126

Se establecen categorías para las cualidades de la calidad externa e interna y calidad en uso del software, teniendo en cuenta estos 7 indicadores (funcionalidad, confiabilidad, utilidad, eficiencia, capacidad de mantenimiento, portabilidad y calidad en uso), que se subdividen a su vez en varios indicadores; estas se pueden medir por métrica interna o externa.

2.7.3.1 Evaluación Interna, externa y Calidad de Uso ISO/IEC 9126

Las definiciones se dan para cada característica y sub-característica de calidad del software que influye en la calidad. Para cada característica y sub-característica, la capacidad del software es determinada por un conjunto de atributos internos que pueden ser medidos. Las características y sub-características se pueden medir externamente por la capacidad del sistema que contiene el software.

2.7.3.1.1 Funcionalidad

Funcionalidad es la capacidad del software de cumplir y proveer las funciones para satisfacer las necesidades explícitas e implícitas cuando es utilizado en condiciones específicas. A continuación, se muestra la característica de Funcionalidad y las sub-características que cubre:

Característica de funcionalidad.

La funcionalidad se divide en 5 criterios:

Adecuación: La capacidad del software para proveer un adecuado conjunto de funciones que cumplan las tareas y objetivos especificados por el usuario.

Exactitud: La capacidad del software para hacer procesos y entregar los resultados solicitados con precisión o de forma esperada.

Interoperabilidad: La capacidad del software de interactuar con uno o más sistemas específicos.

Seguridad: La capacidad del software para proteger la información y los datos de manera que los usuarios o los sistemas no autorizados no puedan acceder a ellos para realizar operaciones, y la capacidad de aceptar el acceso a los datos de los usuarios o sistemas autorizados

Conformidad de la funcionalidad: La capacidad del software de cumplir los estándares referentes a la funcionalidad.

2.7.3.1.2 Confiabilidad.

La confiabilidad es la capacidad del software para asegurar un nivel de funcionamiento adecuado cuando es utilizado en condiciones específicas. En este caso a la confiabilidad se amplía sostener un nivel especificado de funcionamiento y no una función requerida.

Característica de Confiabilidad

La confiabilidad se divide en 4 criterios:

- ❖ **Madurez:** La capacidad que tiene el software para evitar fallas cuando encuentra errores. Ejemplo, la forma como el software advierte al usuario cuando realiza operaciones en la unidad de diskett vacía, o cuando no encuentra espacio suficiente el disco duro donde esta almacenando los datos.
- ❖ **Tolerancia a errores:** La capacidad que tiene el software para mantener un nivel de funcionamiento en caso de errores.
- ❖ **Recuperabilidad:** La capacidad que tiene el software para restablecer su funcionamiento adecuado y recuperar los datos afectados en el caso de una falla.
- ❖ **Conformidad de la fiabilidad:** La capacidad del software de cumplir a los estándares o normas relacionadas a la fiabilidad.

2.7.3.1.3 Usabilidad

La usabilidad es la capacidad del software de ser entendido, aprendido, y usado en forma fácil y atractiva. Algunos criterios de funcionalidad, fiabilidad y eficiencia afectan la usabilidad, pero para los propósitos de la ISO/IEC 9126 ellos no clasifican como usabilidad. La usabilidad está determinada por los usuarios finales y los usuarios indirectos del software, dirigidos a todos los ambientes, a la preparación del uso y el resultado obtenido.

Característica de Usabilidad

La usabilidad se divide en 5 criterios.

- ❖ **Entendimiento:** La capacidad que tiene el software para permitir al usuario entender si es adecuado, y de una manera fácil como ser utilizado para las tareas y las condiciones particulares de la aplicación. En este criterio se debe tener en cuenta la documentación y de las ayudas que el software entrega.
- ❖ **Aprendizaje:** La forma como el software permite al usuario aprender su uso. También es importante considerar la documentación.
- ❖ **Operabilidad:** La manera como el software permite al usuario operarlo y controlarlo.
- ❖ **Atracción:** La presentación del software debe ser atractiva al usuario. Esto se refiere a las cualidades del software para hacer más agradable al usuario, ejemplo, el diseño gráfico.
- ❖ **Conformidad de uso:** La capacidad del software de cumplir los estándares o normas relacionadas a su usabilidad.

2.7.3.1.4 Eficiencia

La eficiencia del software es la forma del desempeño adecuado, de acuerdo a al número recursos utilizados según las condiciones planteadas. Se debe tener en cuenta otros aspectos como la configuración de hardware, el sistema operativo, entre otros.

Característica de Eficiencia

La eficiencia se divide en 3 criterios:

- ❖ **Comportamiento de tiempos:** Los tiempos adecuados de respuesta y procesamiento, el rendimiento cuando realiza su función en condiciones específicas. Ejemplo, ejecutar el procedimiento más complejo del software y esperar su tiempo de respuesta, realizar la misma función, pero con más cantidad de registros.

- ❖ **Utilización de recursos:** La capacidad del software para utilizar cantidades y tipos adecuados de recursos cuando este funciona bajo requerimientos o condiciones establecidas. Ejemplo, los recursos humanos, el hardware, dispositivos externos.
- ❖ **Conformidad de eficiencia:** La capacidad que tiene el software para cumplir con los estándares o convenciones relacionados a la eficiencia.

2.7.3.1.5 Capacidad de mantenimiento

La capacidad de mantenimiento es la cualidad que tiene el software para ser modificado. Incluyendo correcciones o mejoras del software, a cambios en el entorno, y especificaciones de requerimientos funcionales.

Característica de Mantenimiento

El mantenimiento se divide en 5 criterios:

- ❖ **Capacidad de ser analizado:** La forma como el software permite diagnósticos de deficiencias o causas de fallas, o la identificación de partes modificadas.
- ❖ **Cambiabilidad:** La capacidad del software para que la implementación de una modificación se pueda realizar, incluye también codificación, diseño y documentación de cambios.
- ❖ **Estabilidad:** La forma como el software evita efectos inesperados para modificaciones del mismo.
- ❖ **Facilidad de prueba:** La forma como el software permite realizar pruebas a las modificaciones sin poner el riesgo los datos.
- ❖ **Conformidad de facilidad de mantenimiento:** La capacidad que tiene el software para cumplir con los estándares de facilidad de mantenimiento.

2.7.3.1.6 Portabilidad

La capacidad que tiene el software para ser trasladado de un entorno a otro.

Característica de portabilidad

La usabilidad se divide en 5 criterios:

- ❖ **Adaptabilidad:** Es como el software se adapta a diferentes entornos especificados (hardware o sistemas operativos) sin que implique reacciones negativas ante el cambio. Incluye la escalabilidad de capacidad interna (Ejemplo: Campos en pantalla, tablas, volúmenes de transacciones, formatos de reporte, etc.).
- ❖ **Facilidad de instalación:** La facilidad del software para ser instalado en un entorno específico o por el usuario final.
- ❖ **Coexistencia:** La capacidad que tiene el software para coexistir con otro o varios softwares, la forma de compartir recursos comunes con otro software o dispositivo.
- ❖ **Reemplazabilidad:** La capacidad que tiene el software para ser reemplazado por otro software del mismo tipo, y para el mismo objetivo.
- ❖ Ejemplo, la Reemplazabilidad de una nueva versión es importante para el usuario, la propiedad de poder migrar los datos a otro software de diferente proveedor.
- ❖ **Conformidad de portabilidad:** La capacidad que tiene el software para cumplir con los estándares relacionados a la portabilidad.

Calidad en uso

Calidad en uso es la calidad del software que el usuario final refleja, la forma como el usuario final logra realizar los procesos con satisfacción, eficiencia y exactitud. La calidad en uso debe asegurar la prueba o revisión de todas las opciones que el usuario trabaja diariamente y los procesos que realiza esporádicamente relacionados con el mismo software.

Característica Calidad de uso

La calidad de uso se divide en 4 criterios:

- ❖ **Eficacia:** La capacidad del software para permitir a los usuarios finales realizar los procesos con exactitud e integridad.
- ❖ **Productividad:** La forma como el software permite a los usuarios emplear cantidades apropiadas de recursos, en relación a la eficacia lograda en un contexto específico de uso. Para una empresa es muy importante que el software no afecte a la productividad del empleado
- ❖ **Seguridad:** Se refiere al que el Software no tenga niveles de riesgo para causar daño a las personas, instituciones, software, propiedad intelectual o entorno. Los riesgos son normalmente el resultado de deficiencias en la funcionalidad (Incluyendo seguridad), fiabilidad, usabilidad o facilidad de mantenimiento.
- ❖ **Satisfacción:** La satisfacción es la respuesta del usuario a la interacción con el software, e incluye las actitudes hacia el uso del mismo. Según (Dianelys, 2014).

2.7.4 Métricas Basadas en la Función

La métrica punto función (PF) se usa de manera efectiva como medio para medir la funcionalidad que entrega un sistema. PF se deriva empleando una relación empírica basada en medidas contables del dominio de la información del software y las evaluaciones de la complejidad de este.

TABLA N° 2. 1 Tabla de Factor de Ponderación

Parámetros de medición	Cuenta	Factores de Ponderación			Total
		Simple	Medio	Complejo	
1 Nro. de Entradas de Usuario	X	3	4	6	x
2 Nro. de Salidas de Usuario	X	4	5	7	x
3 Nro. de Peticiones de Usuario	X	3	4	6	x
4 Nro. de Archivos	X	7	10	15	x
5 Nro. de Interfaces Externas	X	5	7	10	x
Cuenta Total					x

Fuente: (PRESSMAN, 2002)

Número de entradas de usuario: Se cuenta cada entrada del usuario que proporcione al software diferentes datos aplicados a la aplicación. Las entradas deben ser distinguidas de las peticiones que se contabilizan por separado.

Número de Salidas del usuario: Se encuentra cada salida que proporciona al usuario información orientada a la aplicación. En este contexto se refieren a informes, pantallas, mensajes de error.

Número de peticiones al usuario: Una petición está definida como una entrada interactiva que resulta de la generación de algún tipo de respuesta en forma de salida interactiva se cuenta cada petición por separado.

Numero de archivos: Se cuenta cada archivo maestro lógico es decir una agrupación lógica de datos que puede ser una parte en gran base de datos o un archivo independiente.

Numero de interfaces externas: Se cuentan todas las interfaces legibles por la máquina, por ejemplo: archivo de datos, en cintas o discos que son utilizados para transmitir información a otro sistema.

A cada conteo se le asocia un valor de complejidad, no obstante, la determinación de la complejidad es un poco subjetiva.

Para Calcular los puntos función se usa la siguiente ecuación.

$$PF = \text{Cuenta Total} * (0.65 + 0.01 * \sum Fi)$$

Cuenta Total es la suma de todas las entradas de PF obtenidas de la tabla anterior Fi donde i puede ser de uno hasta 14 los valores de ajuste de complejidad basados en las respuestas a las cuestiones señaladas de la tabla

TABLA N° 2. 2 Factor de Escala

0	1	2	3	4	5
Sin Influencia	Incidental	Moderado	Medio	Significativo	Esencial

Fuente: (PRESSMAN, 2002)

TABLA N° 2. 3 Valoración Métrica Punto Función

Nro.	Preguntas	Fi
1	¿Requiere el sistema copia de seguridad y recuperación?	
2	¿Requiere comunicación de datos?	
3	¿Existen funciones de procesos distribuidos?	
4	¿El rendimiento es crítico?	
5	¿Sera ejecutado el sistema en entorno existente y fuertemente utilizado?	
6	¿Entrada de datos EN LINEA?	
7	¿Requiere la entrada de datos interactiva que las transiciones de entrada se llevan a cabo sobre múltiples pantallas o variadas opciones?	
8	¿Se actualizan los archivos maestros de forma interactiva?	
9	¿Son complejas de las entradas de salidas de archivos?	
10	¿Lógica del proceso Interno Compleja?	
11	¿Se diseña el código para ser reutilizable?	

12	¿Están incluidas en el diseño conversiones de instalación?	
13	¿Instalaciones Múltiples?	
14	¿Facilidad de Cambios?	

Fuente: (PRESSMAN, 2002)

Los valores constantes de la ecuación anterior y los factores de peso aplicados en las encuestas de los ámbitos de información han sido determinados empíricamente.

2.8 MÉTODO DE ESTIMACIÓN DE COSTO DE SOFTWARE

2.8.1 MODELO COCOMO

Modelo de estimación que se encuentra en la jerarquía de modelos de estimación de software con el nombre de COCOMO, por Constructive Cost Model (Modelo Constructivo de Coste). El modelo original se ha convertido en uno de los modelos de estimación de coste del software más utilizados y estudiados en la industria.

CARACTERÍSTICAS

- ❖ Es una herramienta basada en las líneas de código la cual la hace muy poderosa para la estimación de costos y no como otros que solamente miden el esfuerzo en base al tamaño.
- ❖ Representa el más extenso modelo empírico para la estimación de software.
- ❖ Existen herramientas automáticas que estiman costos basados en COCOMO.

Objetivos para la construcción de COCOMO II

- ❖ Desarrollar un modelo de estimación de costo y cronograma de proyectos de software que se adaptara tanto a las prácticas de desarrollo de la década del 90 como a las futuras.
- ❖ Construir una base de datos de proyectos de software que permitiera la calibración continua del modelo, y así incrementar la precisión en la estimación.
- ❖ Implementar una herramienta de software que soportara el modelo.

- ❖ Proveer un marco analítico cuantitativo y un conjunto de herramientas y técnicas que evaluarán el impacto de las mejoras tecnológicas de software sobre los costos y tiempos en las diferentes etapas del ciclo de vida de desarrollo.

Modelos de COCOMO II

Los tres modelos de COCOMO II se adaptan tanto a las necesidades de los diferentes sectores, como al tipo y cantidad de información disponible en cada etapa del ciclo de vida de desarrollo, lo que se conoce por granularidad de la información. Estos tres modelos son:

1. **Modelo de composición de aplicación.** Utilizado durante las primeras etapas de la Ingeniería del software, donde el prototipado de las interfaces de usuario, la interacción del sistema y del software, la evaluación del rendimiento, y la evaluación de la madurez de la tecnología son de suma importancia.
2. **Modelo de fase de diseño previo.** Utilizado una vez que se han estabilizado los requisitos y que se ha establecido la arquitectura básica del software.
3. **Modelo de fase posterior a la arquitectura.** Utilizado durante la construcción del software.

Según (Mercado, 2018)

2.8.1.1 Modelos de Estimación de costo con COCOMO

En la estimación del tamaño de software COCOMO utiliza tres técnicas:

- ❖ Puntos Objeto,
- ❖ Puntos Función No Ajustados y
- ❖ Líneas de Código Fuente.

Además, se emplean otros parámetros relativos al tamaño que contemplan aspectos tales como: reusó, reingeniería, conversión y mantenimiento.

Puntos objeto: El procedimiento para determinar Puntos Objeto en un proyecto software se resume en:

1. Determinar Cantidad de Objetos: Estimar la cantidad de pantallas, reportes, componentes que contendrá la aplicación.
2. Clasificar cada instancia de un objeto según sus niveles de complejidad (simple, media o difícil)
3. Dar el peso a cada objeto según el nivel de complejidad. Los pesos reflejan el esfuerzo relativo requerido para implementar una instancia de ese nivel de complejidad.
4. Determinar la cantidad de Puntos Objeto, sumando todos los pesos de las instancias de los tipos de objetos especificados.

TABLA Nº 2. 4 Puntos Objeto

Para Pantallas			
Cantidad de Vistas Contenidas	Cantidad y fuente de las tablas de datos		
	Total < 4 (< 2 servidor < 3 cliente)	Total < 8 (< 2 - 3 servidor < 3 - 5 cliente)	Total 8 + (> 3 servidor < 5 cliente)
< 3	Simple	Simple	Media
3 - 7	Simple	Media	Difícil
> 8	Media	Difícil	Difícil
Para Reportes			
Cantidad de Vistas Contenidas	Cantidad y fuente de las tablas de datos		
	Total < 4 (< 2 servidor < 3 cliente)	Total < 8 (< 2- 3 servidor < 3-5 cliente)	Total 8 + (> 3 servidor < 5 cliente)
0 o 1	Simple	Simple	Media
2 o 3	Simple	Media	Difícil
4 +	Media	Difícil	Difícil

Fuente: (Modelos de estimación, 2012).

Puntos Función:

El modelo COCOMO II usa Puntos Función y/o Líneas de Código Fuente (SLOC) como base para medir tamaño en los modelos de estimación de Diseño Temprano y Post-Arquitectura.

Los puntos función están basados en información disponible en las etapas tempranas del ciclo de vida del desarrollo de software.

COCOMO II considera solamente UFP (Puntos Función no ajustados).

$$\mathbf{FP = UFP \times TCF}$$

Donde UFP: Puntos Función no Ajustados

TCF: Factor de Complejidad Técnica

Para calcular los UFP, se deben identificar los siguientes elementos:

- ❖ **Entradas Externas (Inputs):** Entrada de datos del usuario o de control que ingresan desde el exterior del sistema para agregar y/o cambiar datos a un archivo lógico interno.
- ❖ **Salidas Externas (Outputs):** Salida de datos de usuario o de control que deja el límite del sistema de software.
- ❖ **Archivo Lógicos Internos (Archivos):** Incluye cada archivo lógico, es decir cada grupo lógico de datos que es generado, usado, o mantenido por el sistema de software.
- ❖ **Archivos Externos de Interface (Interfaces):** Archivos transferidos o compartidos entre sistemas de software.
- ❖ **Solicitudes Externas (Queries):** Combinación única de entrada-salida, donde una entrada causa y genera una salida inmediata, como un tipo de solicitud externa.

Una vez identificados los elementos se clasifican de acuerdo al grado de complejidad en: bajo, promedio o alto. Se asigna un peso a cada ítem según el tipo y el grado de complejidad correspondiente. Finalmente, los UFP son calculados sumando los pesos de todos los ítems identificados.

Líneas de Código Fuente (SLOC):

El objetivo es medir la cantidad de trabajo intelectual puesto en el desarrollo de un programa. Definir una línea de código es difícil debido a que existen diferencias conceptuales cuando se cuentan sentencias ejecutables y de declaraciones de datos en lenguajes diferentes.

A los efectos de COCOMO II, se eliminan las categorías de software que consumen poco esfuerzo. Así no están incluidas librerías de soporte, sistemas operativos, librerías comerciales, etc., ni tampoco el código generado con generadores de código fuente.

Conversión de Puntos Función a Líneas de Código Fuente (SLOC):

Para determinar el esfuerzo nominal en el modelo COCOMO II los puntos función no ajustados tienen que ser convertidos a líneas de código fuente considerando el lenguaje de implementación.

Por otro lado, el modelo COCOMO (COnstructive COst MOdel) Calcula esfuerzo y coste en función del tamaño del programa (LDC). COCOMO está definido para tres tipos de proyectos de Software:

1. Modo orgánico: Proyectos pequeños y sencillos, con equipos de experiencia en la aplicación y requisitos poco rígidos.
2. Modo semi-acoplado: Proyectos intermedios (más complejos), con equipos que poseen variados niveles de experiencia y requisitos más rígidos.
3. Modo empotrado: Proyectos que deben ser desarrollados en un conjunto de Hardware, Software y restricciones muy grandes.

Según (Modelos de estimación, 2012).

FIGURA Nº 2. 19 Detalles de coeficientes de COCOMO II

PARA PUNTOS DE FUNCIÓN:			COCOMO BASICO																
DOMINIO	COMPLEJIDAD	PESO	Modo (Tipo de Proyecto)	a	b	c	d												
Salidas	Alta	7	Orgánico	2.4	1.05	2.5	0.38												
	Media	5																	
	Baja	4																	
Entradas	Alta	6	Semiacoplado	3.0	1.12	2.5	0.35												
	Media	4																	
	Baja	3																	
Consultas	Alta	7	Empotrado	3.6	1.20	2.5	0.32												
	Media	5																	
	Baja	4																	
Archivo Interno	Alta	15	$E = a * (KLDC)^b$ $D = c * (E)^d$																
	Media	10																	
	Baja	7																	
Archivo Externo / Interfaces	Alta	10	COCOMO INTERMEDIO <table border="1"> <thead> <tr> <th>Modo (Tipo de Proyecto)</th> <th>a</th> <th>b</th> </tr> </thead> <tbody> <tr> <td>Orgánico</td> <td>3.2</td> <td>1.05</td> </tr> <tr> <td>Semiacoplado</td> <td>3.0</td> <td>1.12</td> </tr> <tr> <td>Empotrado</td> <td>2.8</td> <td>1.20</td> </tr> </tbody> </table>					Modo (Tipo de Proyecto)	a	b	Orgánico	3.2	1.05	Semiacoplado	3.0	1.12	Empotrado	2.8	1.20
	Modo (Tipo de Proyecto)	a						b											
	Orgánico	3.2	1.05																
Semiacoplado	3.0	1.12																	
Empotrado	2.8	1.20																	
Media	7																		
Baja	5																		

Fuente: (Modelos de estimación, 2012).

2.9 SEGURIDAD DE APLICACIONES WEB

La seguridad de las aplicaciones Web es un tema crítico y complejo para los desarrolladores Web. Ya que requiere realizar estudios y tener entendimiento sobre los puntos de vulnerabilidad en cada aplicación Web.

Se puede clasificar los puntos más importantes:

Seguridad en el servidor

- ❖ Servidor Web
- ❖ Servidor de Base de Datos
- ❖ Lenguaje de programación

Seguridad en la Aplicación

- ❖ Control de Acceso
- ❖ Validación de datos de entrada
- ❖ Pruebas de código

Seguridad en la Comunicación

- ❖ Protocolos de seguridad
- ❖ Certificados de Comunicación

Copias de Seguridad

- ❖ Backup de la Base de Datos
- ❖ Backup del control de Acceso

Las pruebas de seguridad están diseñadas para probar la vulnerabilidad en el ambiente del lado del cliente, las comunicaciones de red que ocurren mientras los datos pasan del cliente al servidor y de vuelta y el ambiente del lado del servidor. Cada uno de estos dominios puede recibir ataques y es labor de quien prueba la seguridad descubrir las debilidades que pueden explotar quienes tengan la intención de hacerlo (Pressman, 2005)

A continuación, se muestra las vulnerabilidades al lado del cliente

- ❖ Vulnerabilidad en los navegadores
- ❖ Vulnerabilidad en los correos electrónicos
- ❖ Acceso no autorizado a cookies
- ❖ Simulación en la comunicación con el servidor

Por lo tanto, se puede aplicar las siguientes medidas

Cortafuegos: Mecanismo de filtrado combinación de hardware y software para examinando cada paquete de información entrante para garantizar que no sea un dato sospechoso

Uso de Autenticación: Una contraseña o clave es una forma de autenticación que utiliza información secreta para controlar el acceso hacia algún recurso. A aquellos que desean acceder a la información se les solicita una clave; si conocen o no conocen la contraseña, se concede o se niega el acceso a la información según sea el caso.

Modelo de Criptografía: La encriptación está asociada con la transformación de un mensaje inteligente a una forma no inteligente con la ayuda de una clave secreta antes de que sea colocada en un medio inseguro.

Algoritmo básico del MD5: El mensaje se divide en bloques de 512 bits, añadiendo bits si es necesario al último bloque (de 1 a 448, por una parte, más 64 fijos de un valor $k \bmod 264$). Con cada bloque de 512 se realiza 64 iteraciones. En cada iteración la función MD5 se alimenta con el bloque correspondiente y con el resultado de la iteración anterior, un vector de cuatro elementos de 32 bits cada uno (con valores predeterminados en la primera iteración). La salida de cada iteración se convierte en el nuevo vector (128 bits).

2.9.1 Caja negra y blanca

Es poner en producción el producto en servidores que cumplan con los requerimientos especificados, además de realizar las pruebas necesarias sobre la nueva infraestructura para el acceso de los usuarios. Según (Sommerville, 2005).

2.9.1.1 Pruebas (Diseño de Casos de Prueba)

Para la realización de estas es necesario la creación de casos de prueba especificando la forma de probar el sistema como un todo. Según (Sommerville, 2005). esto incluye:

- ❖ Realizar pruebas de instalación en la plataforma de producción definida.
- ❖ Pruebas de configuración
- ❖ Pruebas negativas, encontrar debilidades del sistema
- ❖ Pruebas de tensión o de estrés al no existir recursos suficientes
- ❖ Prueba de integración del sistema

2.9.1.2 Pruebas de Caja Negra

También denominadas pruebas de comportamientos, se centran en los requisitos funcionales del sistema. Esta prueba permite obtener conjunto de condiciones de entrada que ejerciten los requisitos funcionales del sistema.

Este tipo de pruebas intentan encontrar errores de tipo:

- ❖ Funciones incorrectas o ausentes.
- ❖ Errores de interfaz.
- ❖ Errores de estructuras de datos.
- ❖ Errores de rendimiento.
- ❖ Errores de inicio y fin.

2.9.1.3 Pruebas de Caja Blanca

Denominada también prueba de caja de cristal, es un método de diseño de casos de prueba que usa la estructura de control de diseño para obtenerlos. Según (Sommerville, 2005).

Con estas pruebas se pretenden:

- ❖ Garantizar que se ejecuta al menos una vez todos los caminos independientes de cada módulo.
- ❖ Ejerciten todas las decisiones lógicas en su vertiente verdadera y falsa.
- ❖ Ejecuten todos los bucles en sus límites.

CAPÍTULO III

MARCO APLICATIVO

3 MARCO APLICATIVO

En esta etapa de desarrollo del sistema de información para el control de ingreso de materia prima, salida de cerámica y control del personal, se aplican la metodología, normas y las técnicas mencionadas en el marco teórico.

3.1 DESCRIPCIÓN GENERAL DE LA INSTITUCIÓN.

La fábrica se ubica en la avenida final Buenos Aires, número 3060, zona Alpacoma Bajo perteneciente al Distrito 6 de Achocalla a la cual pertenece la licencia de funcionamiento, la superficie del terreno es de 5000 m².

En los inicios la preparación de la arcilla se realizaba manualmente y no se contaba con personal, solo era un trabajo en familia. Se tuvo muchos tropiezos en el afán de mejorar la calidad de los ladrillos ya que existía mucha competencia en el rubro.

Debido a dificultades en los papeleos en el Ministerio de Trabajo en el año 2012 se tuvo que cambiar el nombre de la fábrica que actualmente lleva el nombre de Cerámica “EMANUEL” que cuenta con Registro Patronal N° 01-331-080 y N° NIT: 2542156010.

La maquinaria con la que cuenta actualmente la fábrica es semi industrial en la cual elaboran aproximadamente una cantidad de 35000 ladrillos semanalmente.

En la actualidad la Cerámica Emanuel no cuenta con un sistema de ventas y control de personal, lo cual los controles se realizan en hojas, en el cual dificulta para la toma de decisiones de la fábrica.

3.2 OBTENCIÓN DE REQUISITOS

La tarea de ingeniería de requerimiento es muy fundamental para que un sistema tenga éxito, en este sentido para el presente proyecto se realizaron las actividades correspondientes, que presentaremos en la siguiente tabla:

TABLA N° 3. 1 OBTENCIÓN DE REQUISITOS

TAREAS REALIZADAS	CARACTERÍSTICAS
Entrevista	Se realizaron las entrevistas con las siguientes personas: <ul style="list-style-type: none"> - Responsable de la fábrica “EMANUEL”. - Encargado de presupuestos.
Observaciones	Se ha podido observar las presentes dificultades en la fábrica “Emanuel”. Se genera información desactualizada y no permiten la correcta toma de decisiones, no cuenta con una base de datos en donde se registre de manera detallada los ingresos y salidas de caja; esto provoca inestabilidad especialmente en los gastos como ser en salarios, impuestos, compra de materia prima y pago de gastos recurrentes como ser agua, luz, gas entre otros.
Documentación	Se ha podido recabar, con toda la documentación necesaria para el proceso de desarrollo del sistema.

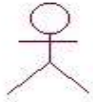
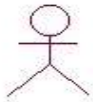
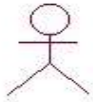
Fuente: (Elaboración propia).

3.2.1 Definición de actores

Las identificaciones de los actores, nos permite conocer a los personajes involucrados en el proceso del sistema web para la venta de ladrillos y control de personal “EMANUEL”.

A objeto de modelar los casos de uso, en la siguiente tabla presentaremos las listas de los actores tanto la descripción de sus actividades relacionadas.

TABLA N° 3. 2 Definición de actores

ACTORES	DESCRIPCIONES
 Administrador/a principal	El administrador/a contempla con los siguientes accesos. <ul style="list-style-type: none"> ❖ Módulo de usuario ❖ Módulo de administración de personal. ❖ Módulo de pagos a servicios básicos. ❖ Módulo de Compra de materiales. ❖ Módulo de administración de producto. ❖ Módulo de administración de ventas. ❖ Módulo de administración de reservas. ❖ Módulo de administración de página web. ❖ Módulo de control de asistencia. ❖ Módulo de planilla.
 Encargado/a	<ul style="list-style-type: none"> ❖ Módulo de pagos a servicios básicos. ❖ Módulo de Compra de materiales. ❖ Módulo de administración de producto. ❖ Módulo de administración de ventas. ❖ Módulo de administración de reservas. ❖ Módulo de administración de página web. ❖ Módulo de control de asistencia.
 Secretaria Recurso humano	<ul style="list-style-type: none"> ❖ Módulo de pagos a servicios básicos. ❖ Módulo de Compra de materiales. ❖ Módulo de administración de página web. ❖ Módulo de control de asistencia.

Fuente: (Elaboración propia).

3.2.2 Lista de requerimiento del sistema

La obtención correcta del requerimiento que se puede llegar a describir con claridad, en forma consistente, el comportamiento del sistema. Las funciones que deben realizarse se clasifican en tres categorías como se detallan en la siguiente tabla.

TABLA N° 3. 3 Lista de requerimiento del sistema

FUNCIONES	SIGNIFICADO
Evidente	Alerta de confirmación, el usuario debe saber que acción se ha realizado.
Oculto	Debe realizarse, aunque no es visible para los usuarios. Esto se aplica en varios módulos donde la información es guardar en un mecanismo persistente de almacenamiento oculto.
Superflua	Esta función es opcional en el cual no repercuten la forma significativa en el costo ni en otras funciones.

Fuente: (Elaboración propia).

3.2.2.1 Requerimientos funcionales

Los requisitos funcionales en la siguiente tabla muestran las características que necesita el sistema a partir de la información obtenida como parte de la tarea de obtención de requisitos, se detallan a continuación en la siguiente tabla.

TABLA N° 3. 4 Requerimientos funcionales

Ref.	Función	Categoría
R.1	Control de acceso seguro y diferenciado por tipo de usuarios o rol.	Evidente
R.2	Gestión de Usuario del Sistema.	Evidente
R.3	Desplegar vistas y menús de acuerdo al rango de cada usuario.	Oculto
R.4	Registro de personal administrativa	Evidente
R.5	Registros de productos y sus características.	Evidente
R.6	Determinar el estado de cada usuario activado.	Evidente
R.7	Realizar un control de stock producto.	Evidente
R.8	Realizar un control productos reservados.	Evidente
R.9	Gestionar reporte en formato PDF.	Evidente
R.10	Búsqueda en el tiempo instante en los datos requeridos.	Evidente

Fuente: (elaboración propia).

3.2.2.2 Requerimientos no Funcionales

El requerimiento no funcional muestra su comportamiento en la siguiente tabla.

TABLA N° 3. 5 Requerimientos no Funcionales

Ref.	Función	Categoría
R.1	El sistema debe visualizarse y funcionar correctamente en cualquier Navegador, Internet Explorer, Mozilla Firefox, Chrome.	Evidente
R.2	El sistema no debe tardar más de diez segundos en mostrar los resultados de una búsqueda.	Evidente
R.3	El sistema debe tener seguridad en el acceso a la información del sistema	Evidente

Fuente: (Elaboración Propia).

3.3 APLICACIÓN DEL MODELO UWE

3.3.1 Modelo de caso de uso

Dentro de este ciclo se realiza un análisis del sistema construyendo para ello diagramas de casos de uso en una descripción de los pasos o las actividades que realiza al cabo de los procesos del sistema.

3.3.1.1 Análisis de requerimientos

En este punto se plasma el análisis de requerimiento del sistema mediante el diseño de diagrama de caso de uso comercial el cual describe el comportamiento del sistema. El diagrama de caso de uso permite que describa el comportamiento del sistema frente a las acciones de los diferentes actores, así como las funcionalidades del sistema.

3.3.1.1.1 Diagrama de caso de uso

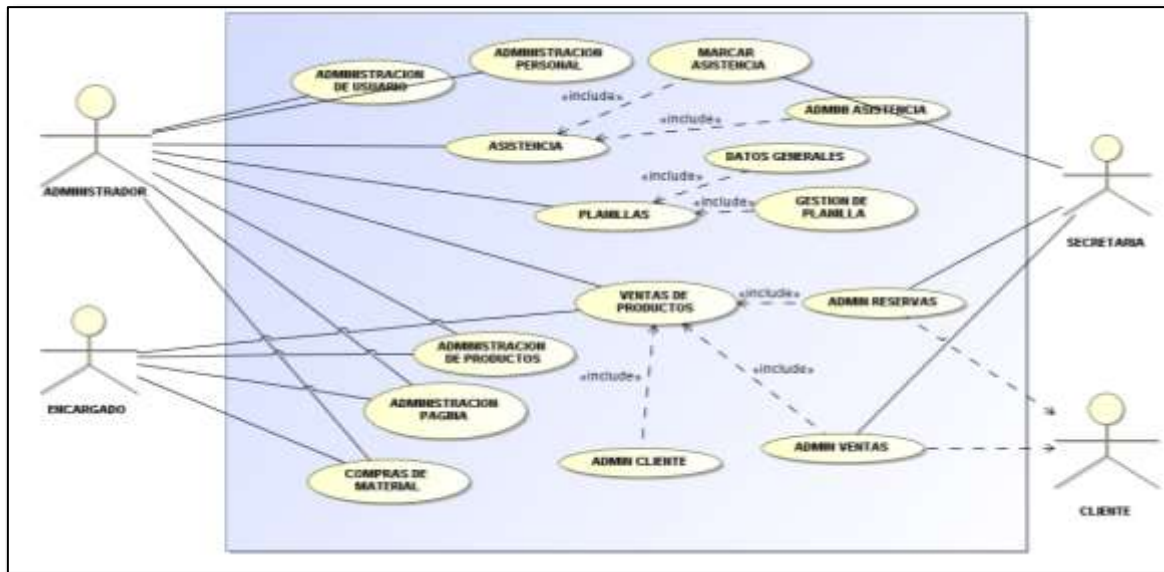
A continuación, se hace el modelamiento donde se puede apreciar cómo interactúan los actores sobre los casos de uso.

Realizaremos una presentación previa de los diagramas de casos de uso a continuación.

- ❖ Modelo de diagrama de caso de uso: General del Sistema
- ❖ Diagrama de caso de uso: Administrador/a principal.
- ❖ Diagrama de caso de uso: Encargado/a.
- ❖ Diagrama de caso de uso: Secretaria.
- ❖ Diagrama de caso de uso: Cliente.

3.3.1.1.2 Modelo de diagrama de caso de uso: General del Sistema

FIGURA N° 3. 1 Modelo de diagrama de caso de uso: General del Sistema



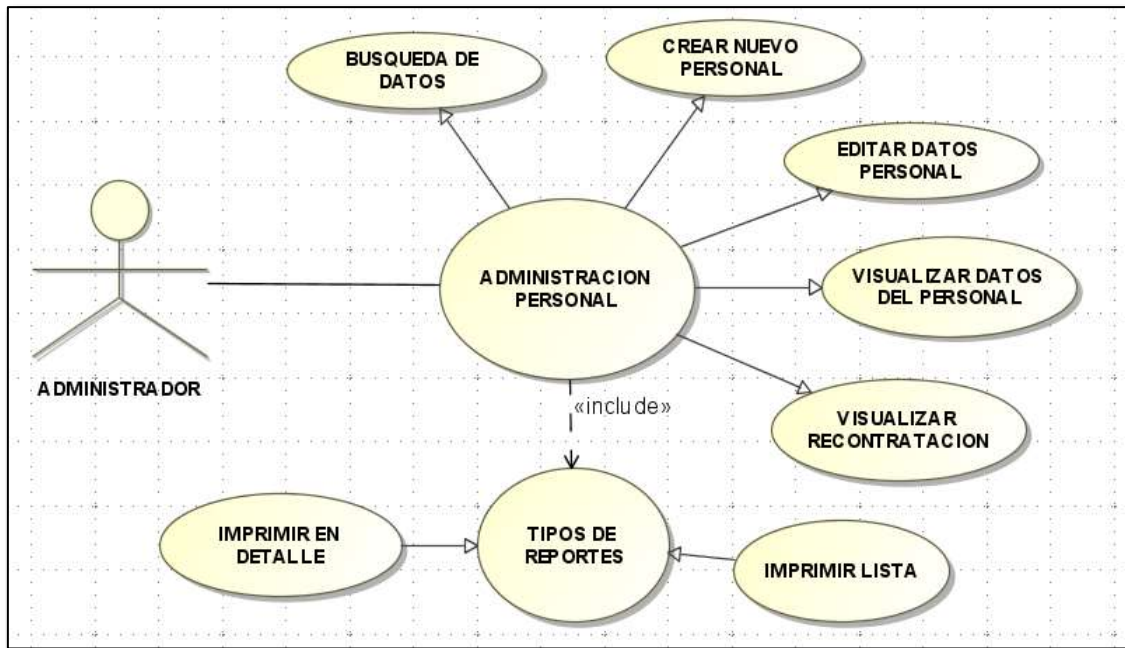
Fuente: (Elaboración propia).

3.3.1.1.3 Diagrama de caso de uso: Administrador/a principal

A continuación, se presentará el modelamiento de diagrama de caso de uso en detalle del administrador que interactúa con el sistema cada módulo.

3.3.1.1.3.1 Diagrama de administración de personal

FIGURA Nº 3. 2 Diagrama de administración de personal



Fuente: (Elaboración propia).

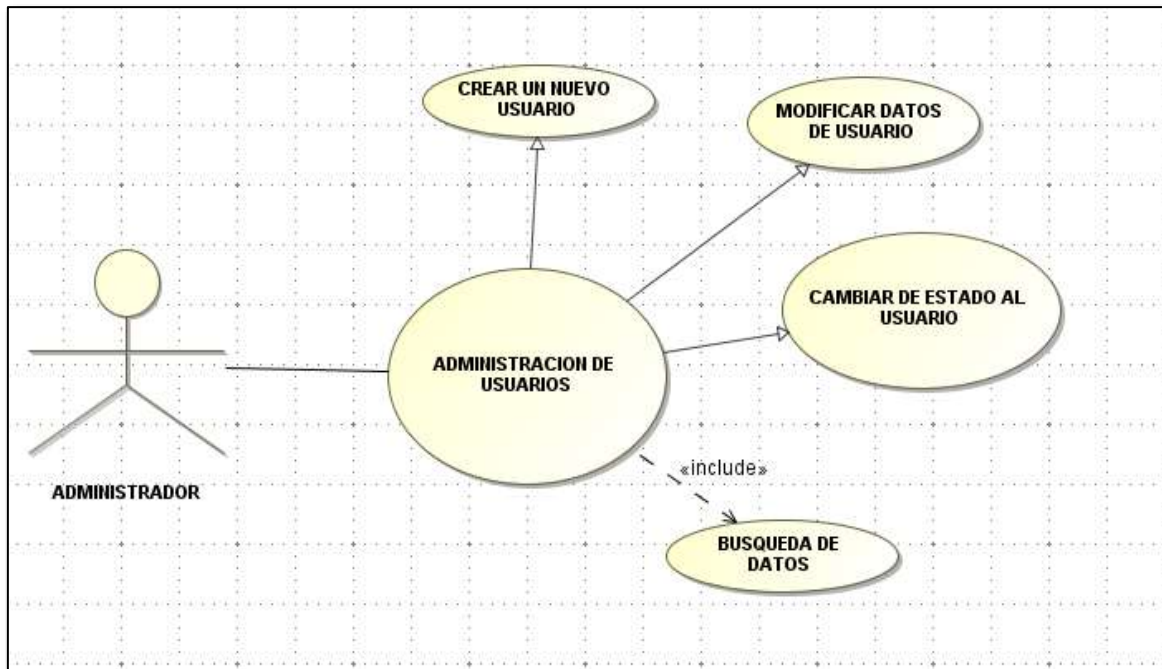
TABLA Nº 3. 6 Caso de uso: Administración de personal

Caso de uso: administración de personal	
Actores	Administrador principal.
Tipo	Primario esencial.
Descripción	El administrador principal tiene las siguientes opciones, visualizar el contenido de registro del personal, crear nuevo registro, editar los datos, visualizar datos personales, visualizar datos de contratación, realizar búsqueda de datos en tiempo real, imprimir lista de personal no detallada y por último imprimir lista del personal en detalle en formato PDF.

Fuente: (Elaboración propia).

3.3.1.1.3.2 Diagrama de administración de usuarios

FIGURA N° 3. 3 Diagrama de administración de usuarios



Fuente: (Elaboración propia).

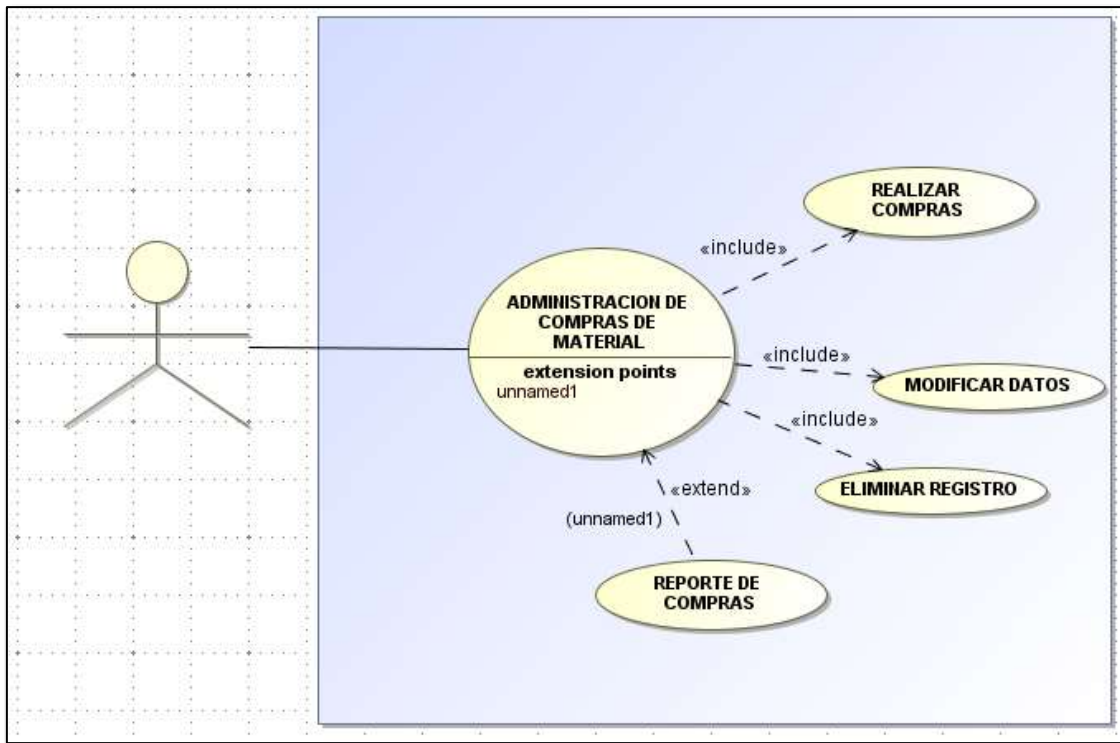
TABLA N° 3. 7 Caso de uso: Administración de usuarios

Caso de uso: administración de usuarios	
Actores	Administrador principal.
Tipo	Primario esencial.
Descripción	El administrador principal tiene las siguientes opciones en este módulo de administración de usuario, visualizar el contenido de registro de usuarios, crear nuevo usuario, editar datos del usuario, cambiar estado del usuario y por último buscar datos en tiempo real.

Fuente: (Elaboración propia).

3.3.1.1.3.3 Diagrama de administración servicios básicos

FIGURA Nº 3. 4 Diagrama de administración compra materiales.



Fuente: (Elaboración propia).

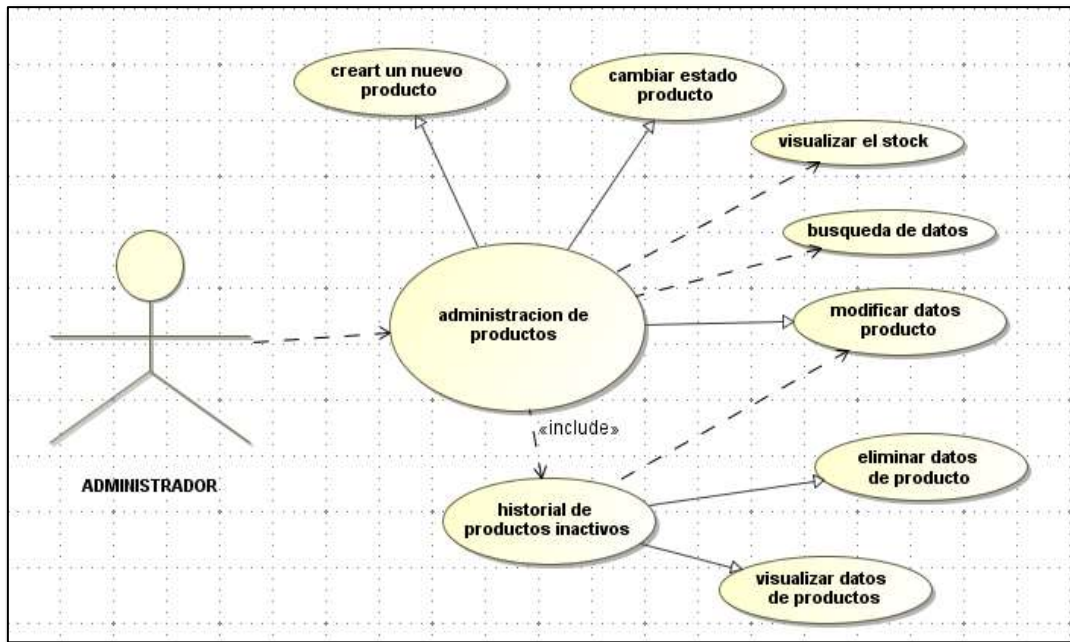
TABLA Nº 3. 8 Caso de uso: Administración servicios básicos

Caso de uso: administración servicios básicos	
Actores	Administrador principal.
Tipo	Primario esencial.
Descripción	El administrador principal tiene las siguientes opciones en el siguiente módulo de compra de materia, visualizaremos el contenido de registro de compra de material, crear un nuevo registro, editar datos, eliminar registros y por último buscar datos en tiempo real.

Fuente: (Elaboración propia).

3.3.1.1.3.4 Diagrama de administración de productos

FIGURA Nº 3. 5 Diagrama de administración de productos



Fuente: (Elaboración propia).

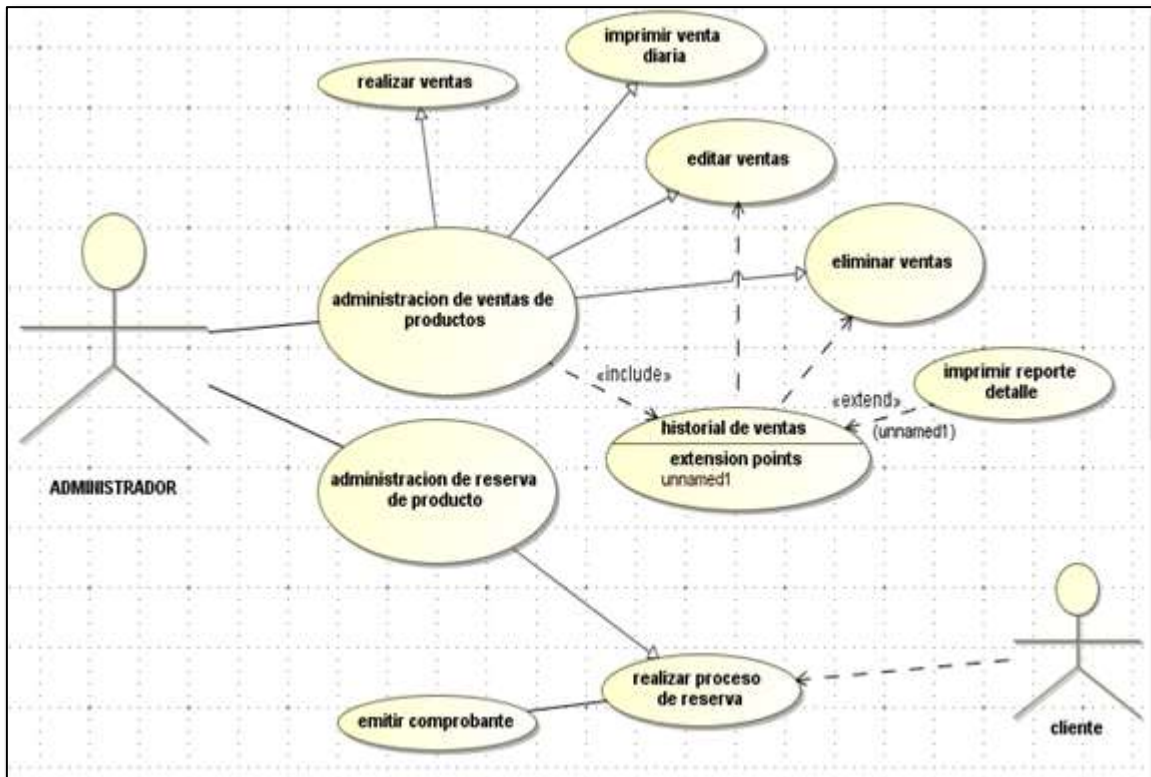
TABLA Nº 3. 9 Caso de uso: Administración de productos

Caso de uso: administración de productos	
Actores	Administrador principal.
Tipo	Primario esencial.
Descripción	El administrador principal tiene las siguientes opciones en este módulo, visualizar el contenido de registro de productos, crear un nuevo registro de producto, editar datos de producto, eliminar registros y por último buscar datos en tiempo real.

Fuente: (Elaboración propia).

3.3.1.1.3.5 Diagrama de administración venta de producto

FIGURA N° 3. 6 Diagrama de administración venta de producto.



Fuente: (Elaboración propia).

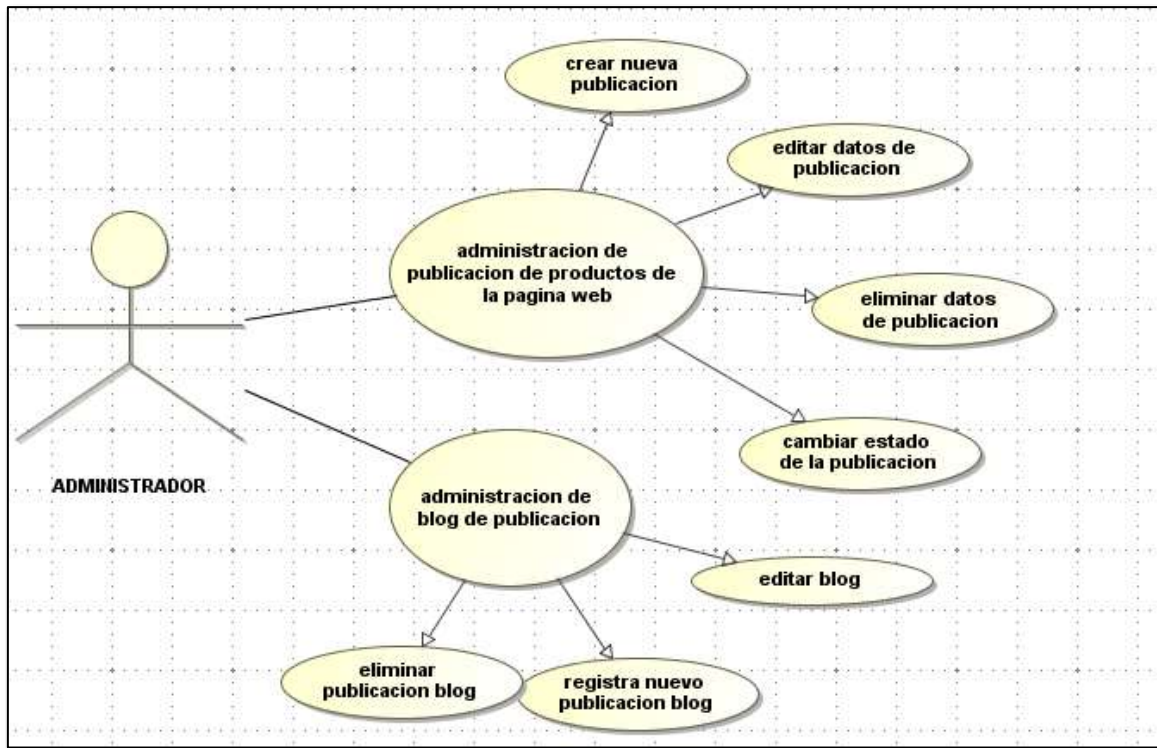
TABLA N° 3. 10 Caso de uso: Administración venta de producto

Caso de uso: administración venta de producto	
Actores	Administrador principal.
Tipo	Primario esencial.
Descripción	El administrador principal tiene las siguientes opciones en estos dos módulos, visualizar el contenido de registro de la venta diaria, realizar la venta de producto, editar venta de producto, eliminar venta y por último emitir comprobante de producto al cliente.

Fuente: (Elaboración propia).

3.3.1.1.3.6 Diagrama de administración de página web

FIGURA Nº 3. 7 Diagrama de administración de página web.



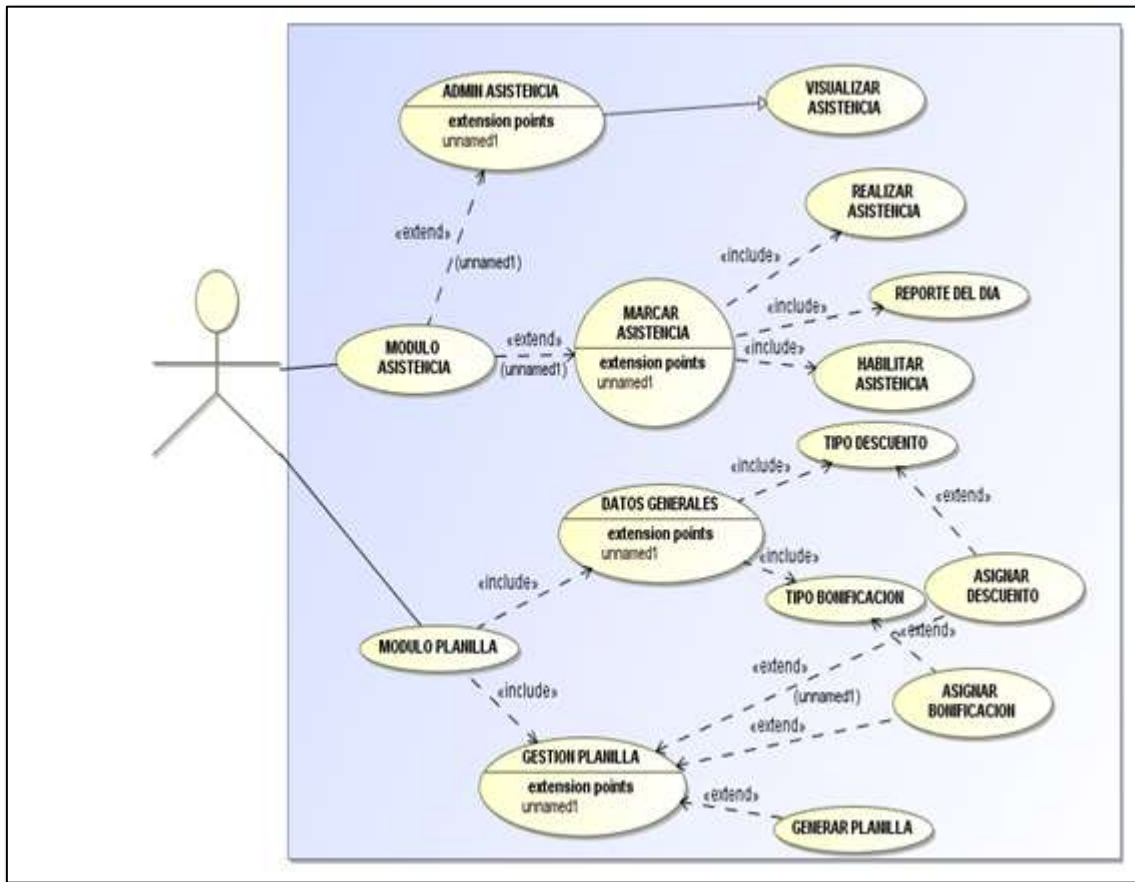
Fuente (Elaboración propia).

TABLA Nº 3. 11 Caso de uso: Administración de página web

Caso de uso: administración de página web	
Actores	Administrador principal.
Tipo	Primario esencial.
Descripción	El administrador principal tiene las siguientes opciones en este módulo, visualizar el contenido de registro de publicación de producto, realizar un nuevo registro, editar registro, eliminar registró y por último cambiar de estado de publicación.

Fuente: (Elaboración propia).

3.3.1.1.3.7 Diagrama de planilla



Fuente (Elaboración propia).

TABLA N° 3. 12 Caso de uso: Administración de página web

Caso de uso: administración de página web	
Actores	Administrador principal.
Tipo	Primario esencial.
Descripción	<p>El administrador principal tiene las siguientes opciones en este módulo de asistencia y el módulo de planilla, en el módulo de asistencia presentamos dos sub-módulos los cuales son:</p> <p>Administrar la asistencia, el cual lista a todos los funcionarios y poder visualizar su información sobre la asistencia del día.</p> <p>Marcar asistencia, en este módulo el administrador contempla con el control de asistencia diaria de poder marcar entrada y salida, de la misma manera presentar un reporte del día, en formato pdf.</p> <p>En el módulo de planilla el administrador puede asignar descuentos al funcionario, de la misma manera el administrador puede asignar bonificaciones y por último genera una planilla de pagos realizando el cierre de mes de la planilla.</p>

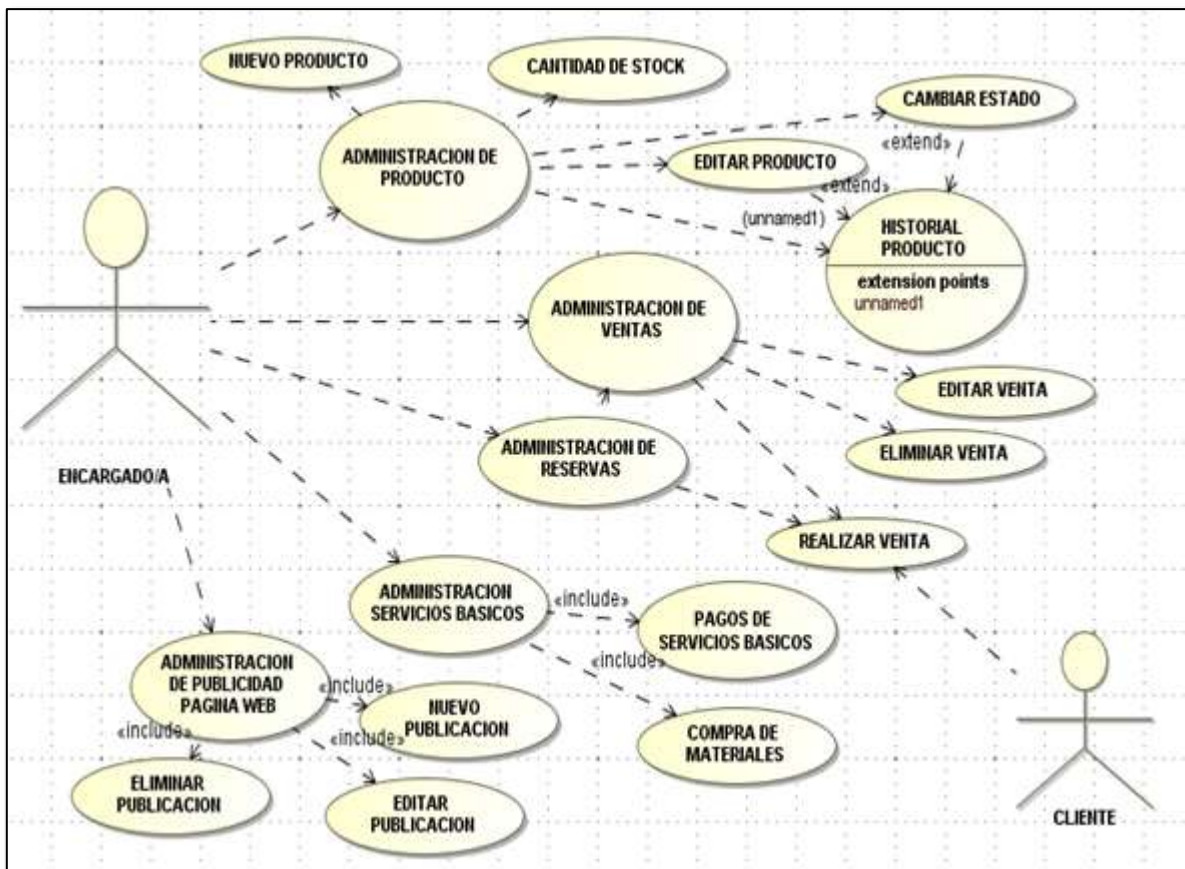
Fuente: (Elaboración propia).

3.3.1.1.4 Diagrama de caso de uso: Encargado/a

A continuación, se presentará el modelamiento general de diagrama de caso de uso del encargado/a que interactúa con el sistema cada módulo.

3.3.1.1.4.1 Diagrama de caso de uso: módulos asignados

FIGURA N° 3. 8 Diagrama de caso de uso: módulos asignados.



Fuente: (Elaboración propia).

TABLA Nº 3. 13 Caso de uso: Encargado/a

Caso de uso: Encargado/a	
Actores	Encargado/a
Tipo	Primario esencial.
Descripción	<p>El encargado del sistema, debe contemplar con ciertos privilegios del sistema lo cual le permite la administración de los siguientes módulos que mencionaremos a continuación.</p> <p>Módulo de producto: en este módulo nos permite administrar, modificar, insertar, listar el stock, eliminar y búsqueda en tiempo real.</p> <p>Administrar la venta y reserva: en este módulo de la misma manera el encargado puede realizar las ventas, modificar la venta, eliminar la venta y de la misma manera puede realizar venta de producto reservado.</p> <p>Administrar los servicios básicos: en este módulo el usuario puede realizar los pagos de todos los servicios básicos y de la misma manera podrá realizar las compras de materia prima para la cerámica.</p> <p>Administrar la página web: en este módulo el encargado tiene el control total de la publicación de poder inserta nueva publicación, editar la publicación, eliminar la publicación y cambiar de estado la publicación.</p>

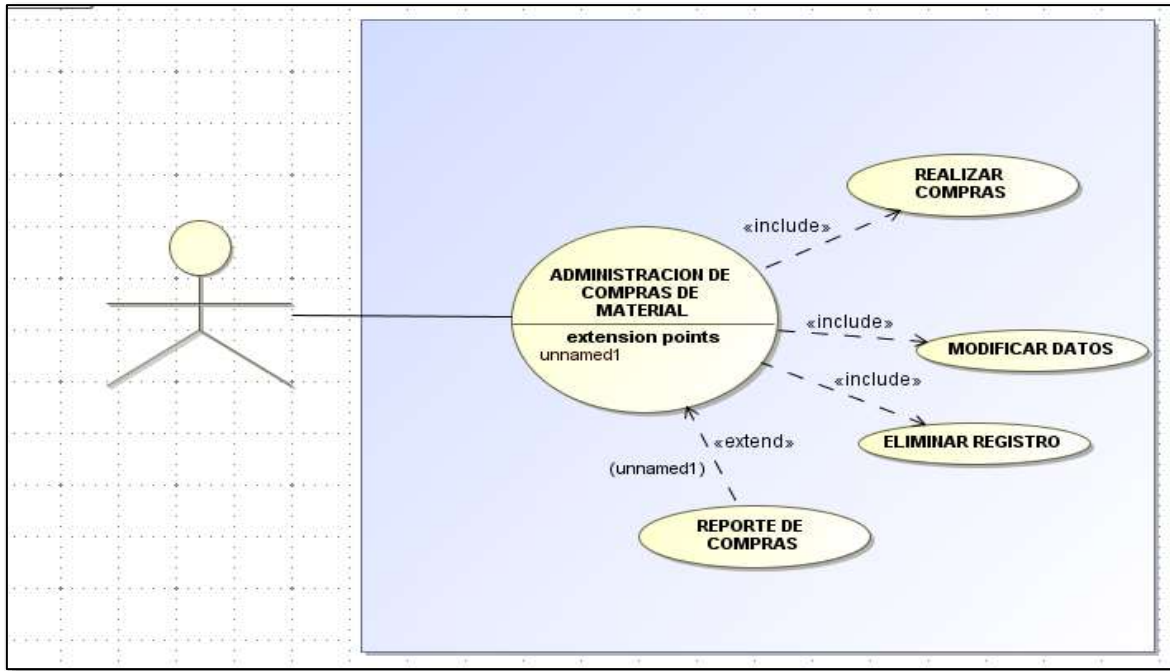
Fuente: (Elaboración propia).

3.3.1.1.5 Diagrama de caso de uso: Secretaria

A continuación, se presentará el modelamiento general de diagrama de caso de uso de secretaria que interactúa con el sistema cada módulo.

3.3.1.1.5.1 Diagrama de caso de uso: módulos compra material.

FIGURA N° 3. 9 Diagrama de caso de uso: módulos asignados.



Fuente (Elaboración propia).

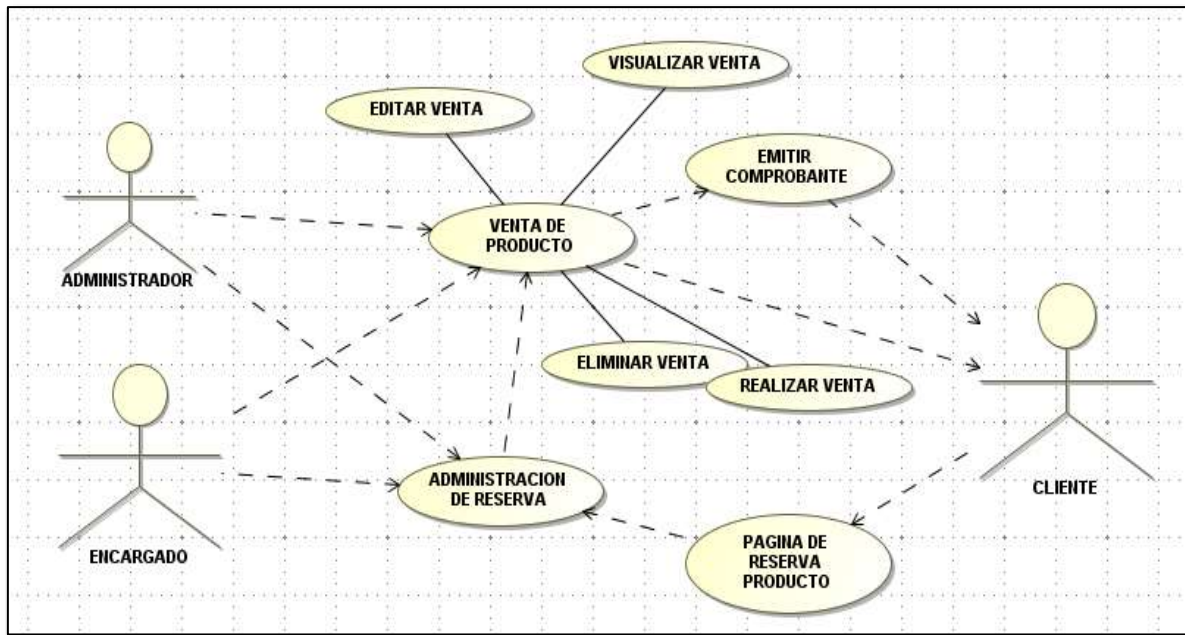
TABLA N° 3. 14 Caso de uso: Secretaria.

Caso de uso: Secretaria.	
Actores	Secretaria.
Tipo	Primario esencial.
Descripción	La secretaria o el secretario, debe contemplar con los siguientes privilegios del sistema, a los cuales se le asigna los siguientes módulos. Administrar de compra de material, el usuario realiza compras, editar compras y generar reportes en formato pdf, para las tomas de decisiones.

Fuente: (Elaboración propia).

3.3.1.1.6 Diagrama de caso de uso: cliente.

FIGURA N° 3. 10 Diagrama de caso de uso: cliente.



Fuente: (Elaboración propia).

TABLA N° 3. 15 Caso de uso: cliente

Caso de uso: cliente	
Actores	Cliente
Tipo	Secundario
Descripción	<p>El cliente debe realizar compra directa o reserva de producto mediante la página el cual interactuar con el encargado o el administrador, para realizar la venta del material.</p> <p>De la misma manera el administrador o el encardo debe contemplar con los privilegios de poder editar, eliminar, visualizar y generar el reporte adecuado para el cliente.</p>

Fuente: (Elaboración propia).

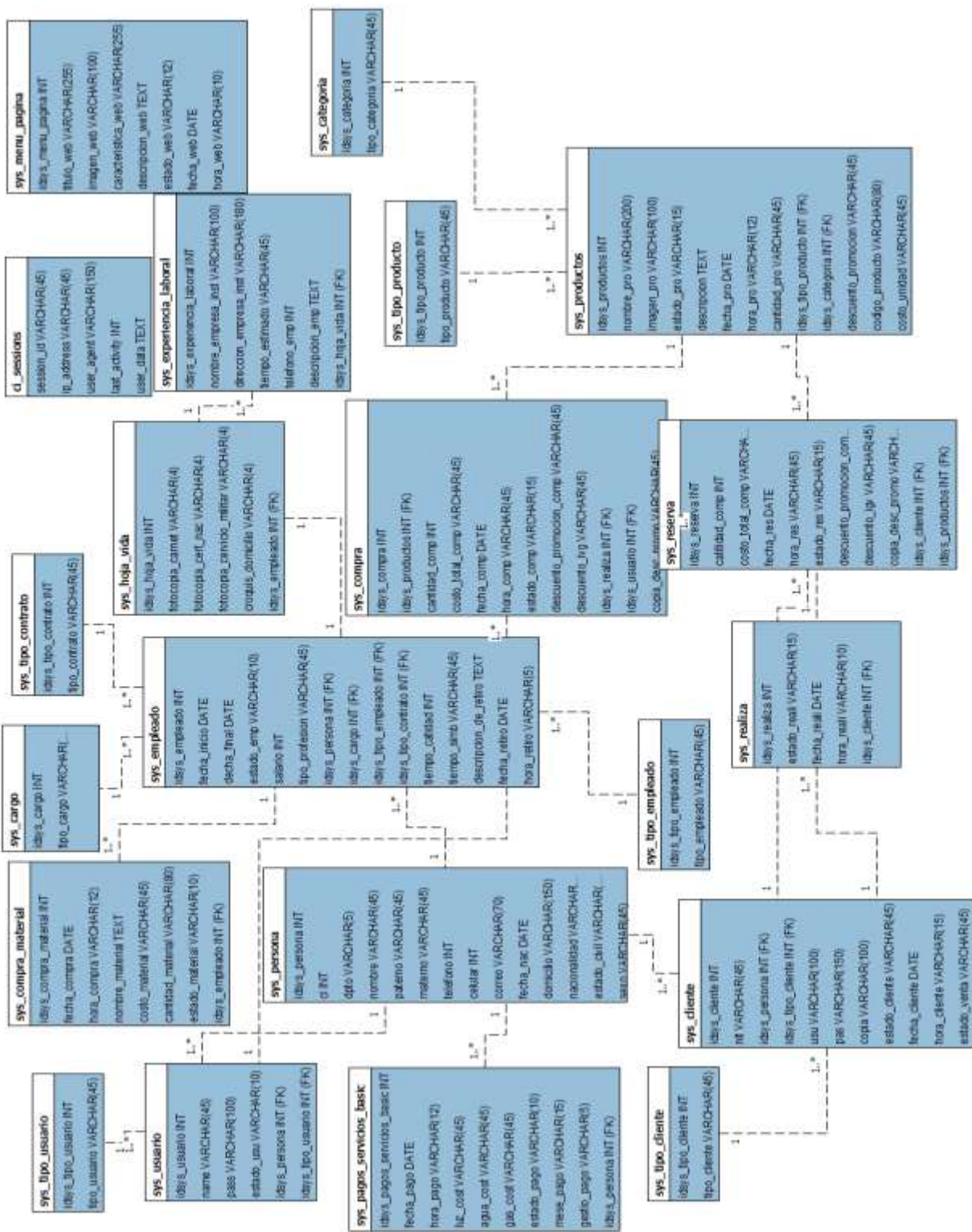
3.3.2 Modelo de contenido.

El modelo de contenido para aplicaciones Web en UWE, utilizan modelos estándares UML para modelar estructuras de clases, asociaciones y paquetes. El diagrama de contenido tiene por propósito mostrar las relaciones entre las entidades y la estructura de los datos que se encuentran alojados en el sistema el modelo de contenido contiene la información relevante almacenada en el sistema como se estructura como se relaciona.

El diseño conceptual se basa en el modelo de análisis e incluye los objetos involucrados en las actividades típicas que los usuarios realizan con la aplicación.

El propósito del modelo de contenido es proporcionar una especificación visual de la información relevante para el dominio del sistema web, que comprende principalmente el contenido de la aplicación Web.

FIGURA N° 3. 11 Modelo de contenido



Fuente: (Elaboración propia).

El modelo muestra la relación entre las entidades del sistema, pudiendo destacar la importancia de la entidad de usuario en el sistema.

3.3.3 Modelo De Navegación.

A continuación, se realiza el modelamiento de modelo de navegación, donde se puede apreciar cómo interactúan los usuarios en el modelo de navegación en cada clase.

- ❖ Modelo De Navegación: Administrador/a principal.
- ❖ Modelo De Navegación: Encargado/a.
- ❖ Modelo De Navegación: Secretaria.
- ❖ Modelo De Navegación: cliente.

Estos tienen por objetivo de ilustrar los vínculos lógicos y de navegación entre clases. El modelo de clase navegación y las asociaciones de navegación general del proyecto, que expresa la navegación directa que se muestra en los siguientes modelos de navegación.

3.3.3.1 Modelo De Navegación: Administrador/a principal.

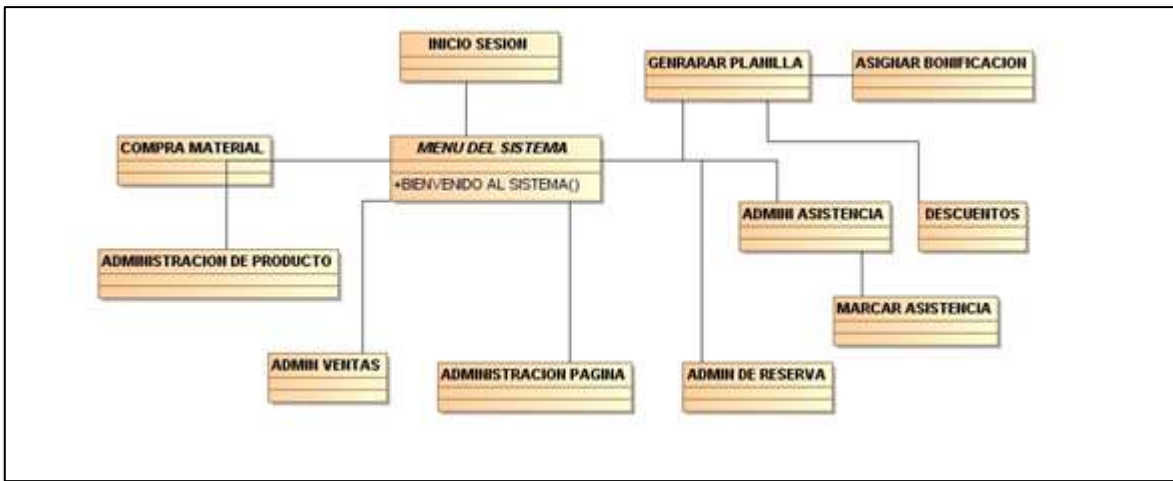
FIGURA N° 3. 12 Modelo De Navegación: Administrador/a principal



Fuente: (Elaboración propia).

3.3.3.2 Modelo De Navegación: Encargado/a.

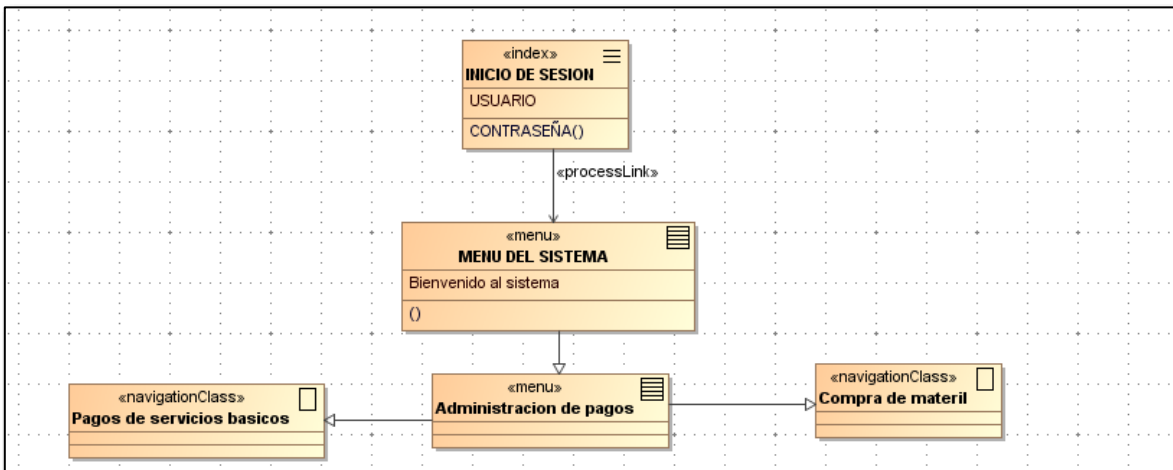
FIGURA Nº 3. 13 Modelo De Navegación: Encargado/a.



Fuente: (Elaboración propia).

3.3.3.3 Modelo De Navegación: Secretaria.

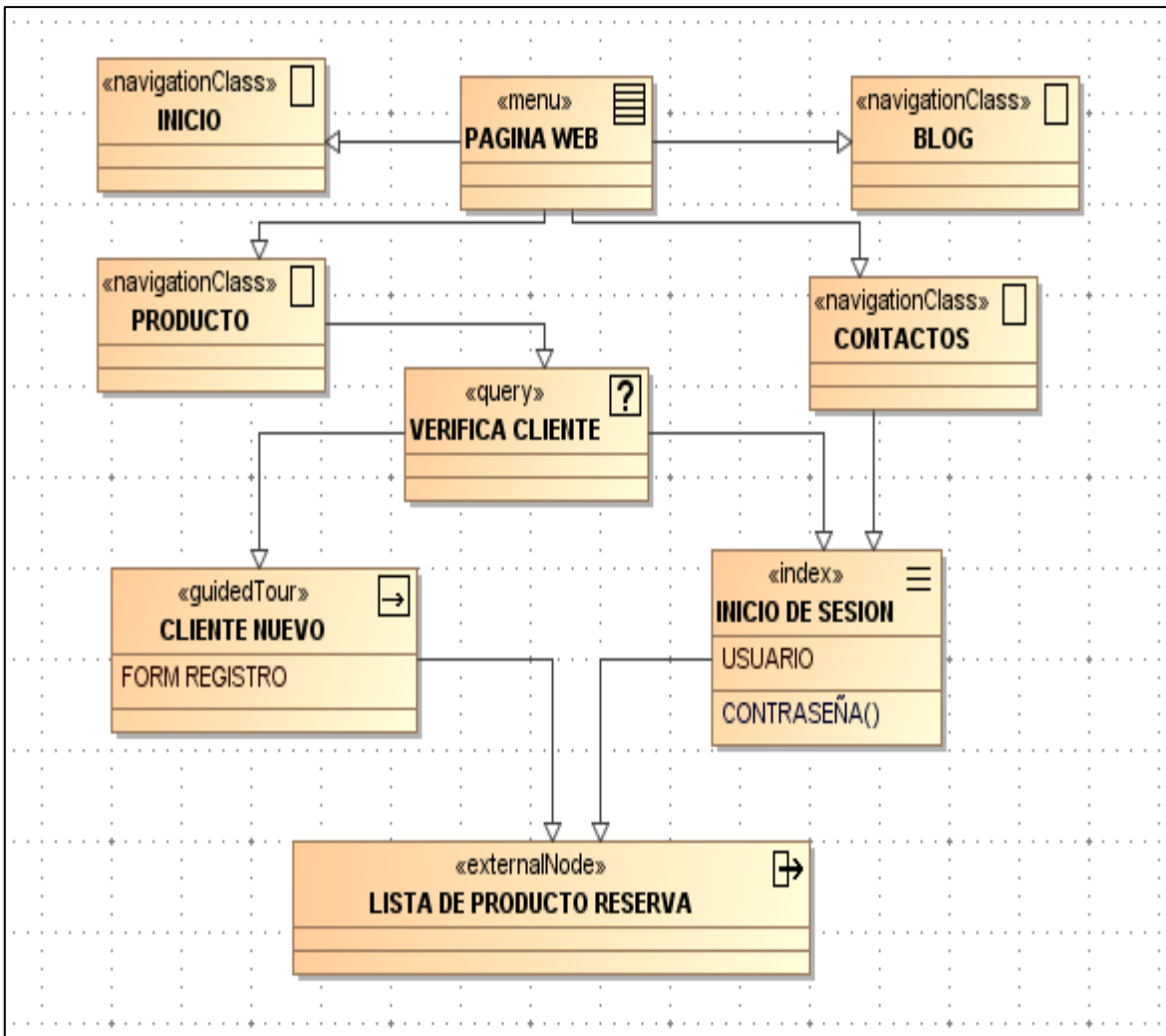
FIGURA Nº 3. 14 Modelo De Navegación: Secretaria.



Fuente: (Elaboración propia).

3.3.3.4 Modelo De Navegación: cliente.

FIGURA N° 3. 15 Modelo De Navegación: cliente.

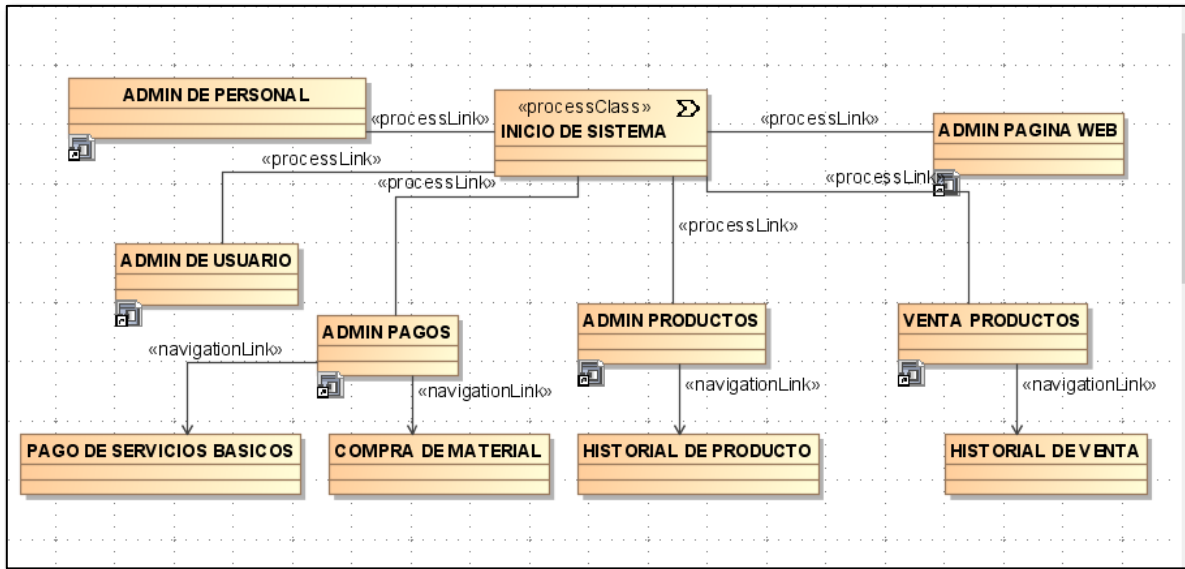


Fuente: (Elaboración propia).

3.3.4 Modelo de Estructura de proceso.

Este modelo de estructura de proceso permite ilustrar los vínculos lógicos y de navegación entre clases, menús, índices y clases de proceso para tener una mejor comprensión.

FIGURA N° 3. 16 Modelo de Estructura de proceso



Fuente: (Elaboración propia).

3.3.5 Modelo de Presentación

A continuación, se muestran los modelos de presentación del sistema web para la venta de ladrillos y control de personal, según la metodología UWE que propone para la construcción de las páginas en forma de bosquejos derivadas donde los usuarios podrán acceder al sistema mostrado los menús correspondientes según el tipo de usuario.

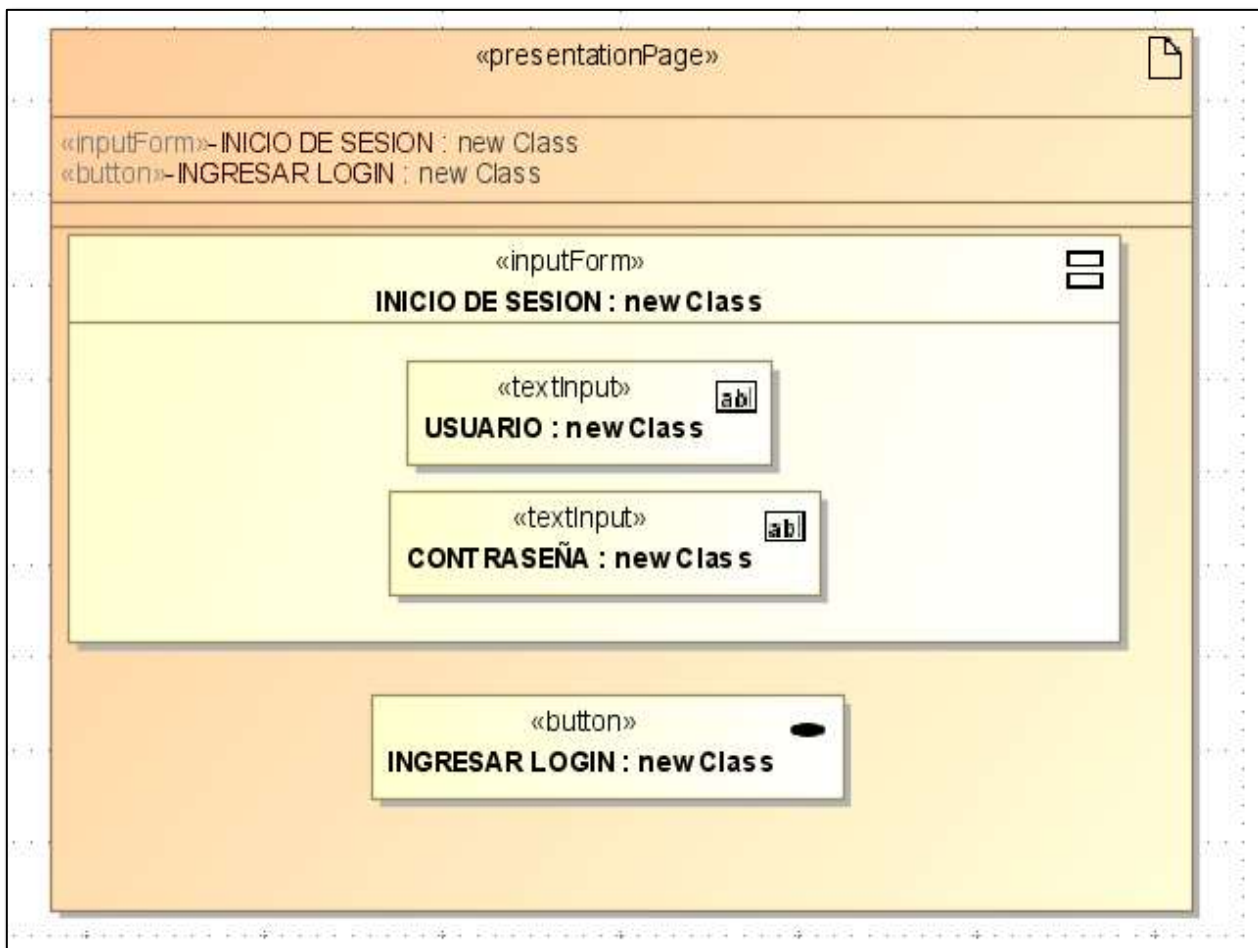
- ❖ Modelo de Presentación: Administrador/a principal.
- ❖ Modelo de Presentación: Encargado/a.
- ❖ Modelo de Presentación: Secretaria.
- ❖ Modelo de Presentación: cliente.

El objetivo es ilustrar los vínculos lógicos de modelo de presentación entre clases. Las propiedades pueden anidarse, por ejemplo, cada contacto cubre diferentes textos y botones. Ello significa, acciones que realizan el sistema que para cada contacto la correspondiente dirección de correo y los correspondientes campos de teléfonos y direcciones serán visualizados.

3.3.5.1 Modelo de Presentación: Administrador/a principal.

En este modelo de presentación se puede visualizar el interfaz de inicio de sesión del sistema, para el usuario principal.

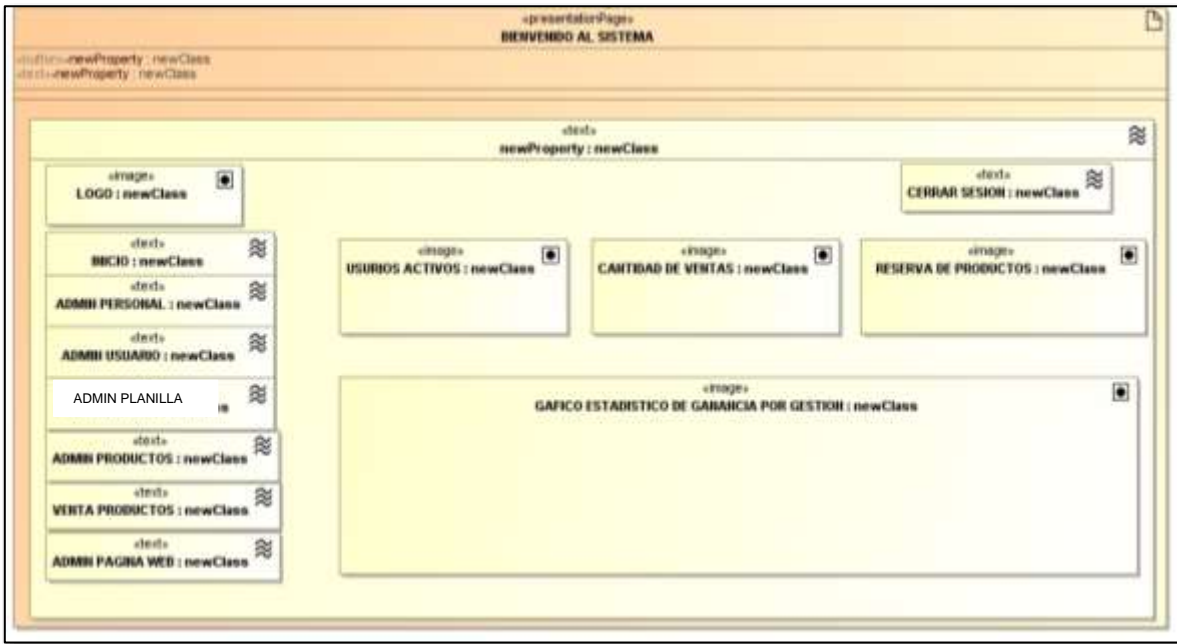
FIGURA N° 3. 17 Modelo de Presentación: Administrador/a principal login



Fuente: (Elaboración propia).

Una vez ingresado al sistema como usuario principal nos presenta un interfaz de bienvenida al sistema con sus respectivos menús, dependiendo del tipo de rol que contempla el usuario ingresado, como se puede visualizar en el gráfico.

FIGURA Nº 3. 18 Modelo de Presentación: Administrador/a principal

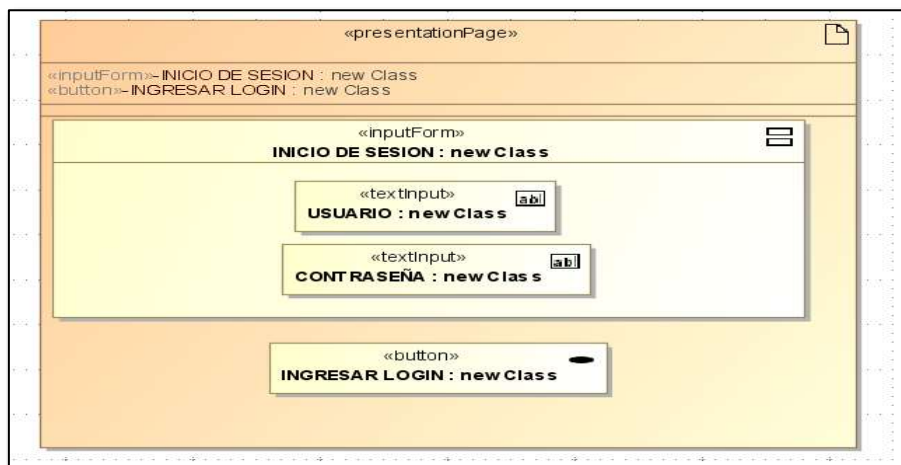


Fuente: (Elaboración propia).

3.3.5.2 Modelo de Presentación: Encargado/a.

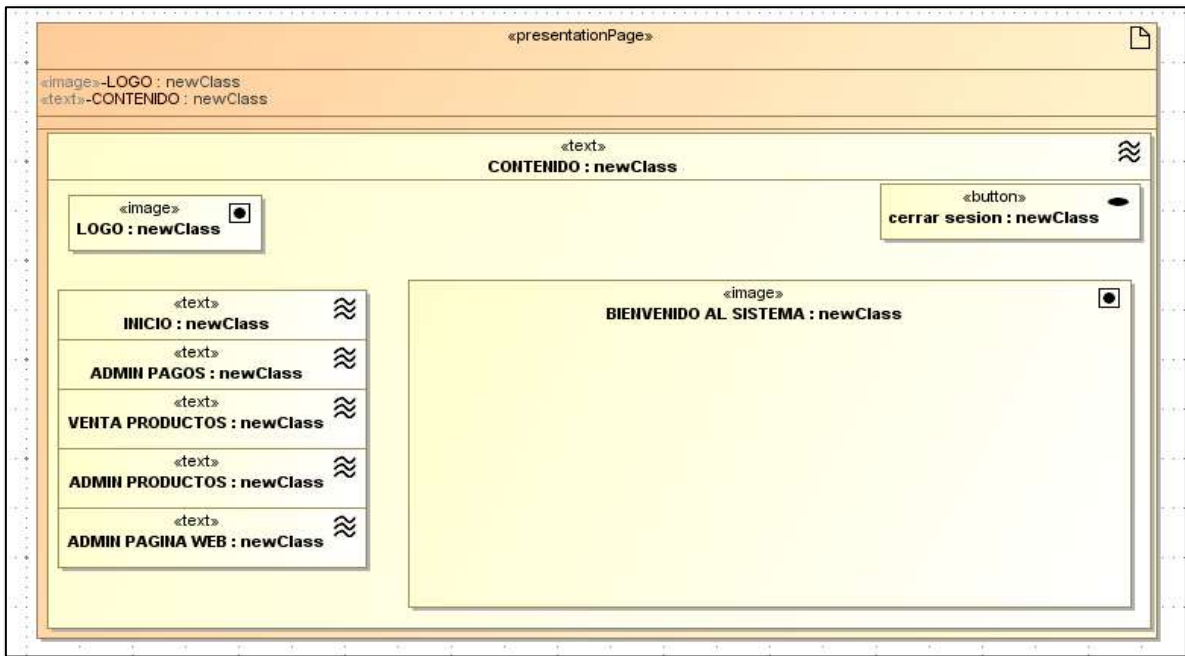
En este modelo de presentación se puede visualizar el interfaz de inicio de sesión del sistema, para el usuario rol encargado del sistema.

FIGURA Nº 3. 19 Modelo de Presentación: Encargado/a login



Fuente: (Elaboración propia).

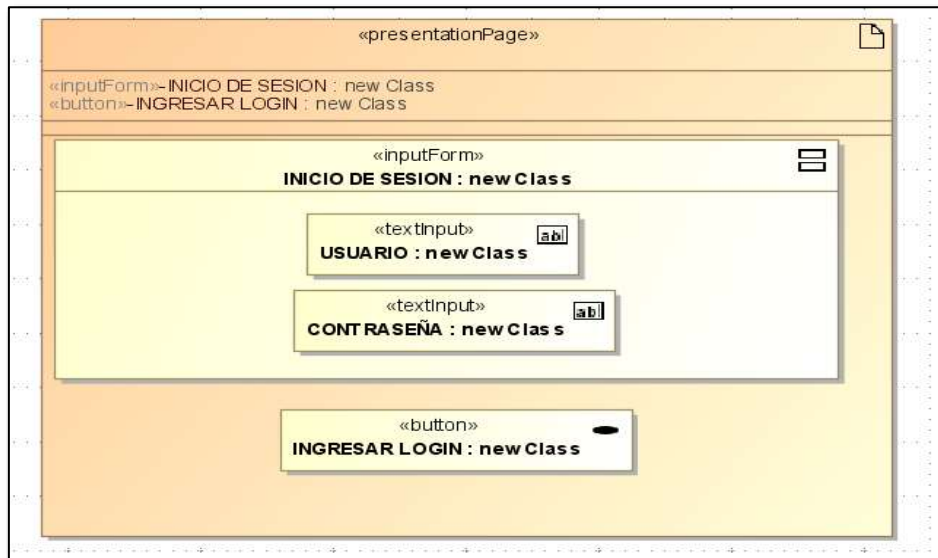
FIGURA N° 3. 20 Modelo de Presentación: Encargado/a



Fuente: (Elaboración propia).

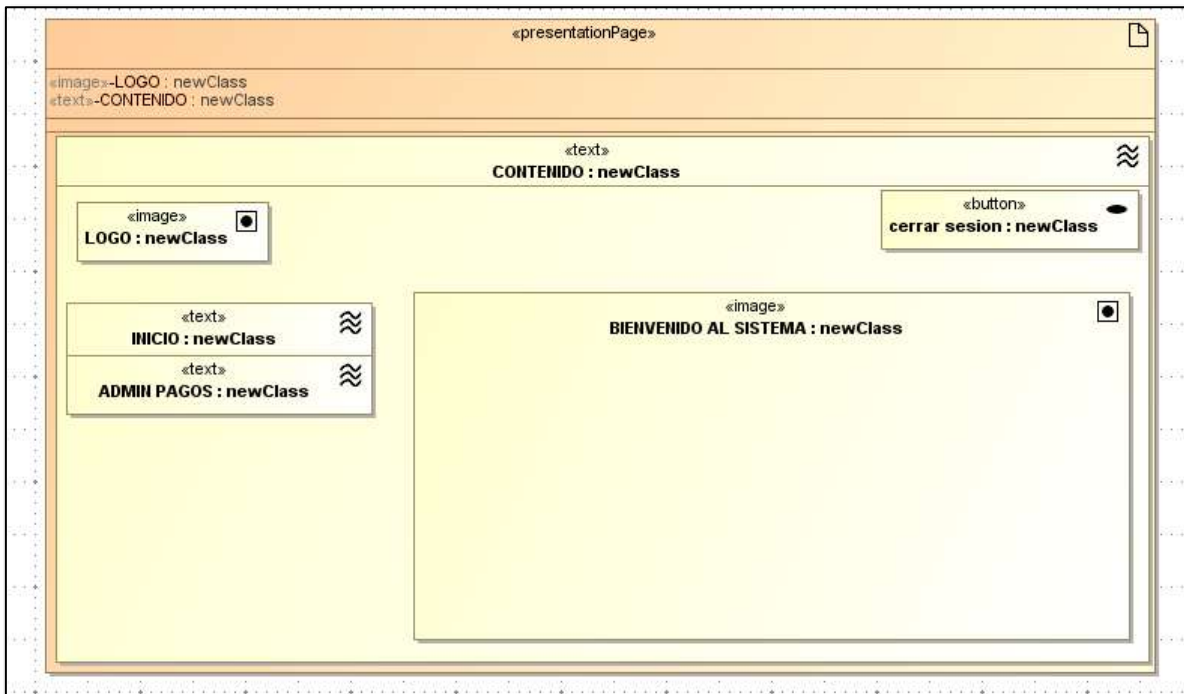
3.3.5.3 Modelo de Presentación: Secretaria.

FIGURA N° 3. 21 Modelo de Presentación: Secretaria login



Fuente: (Elaboración propia).

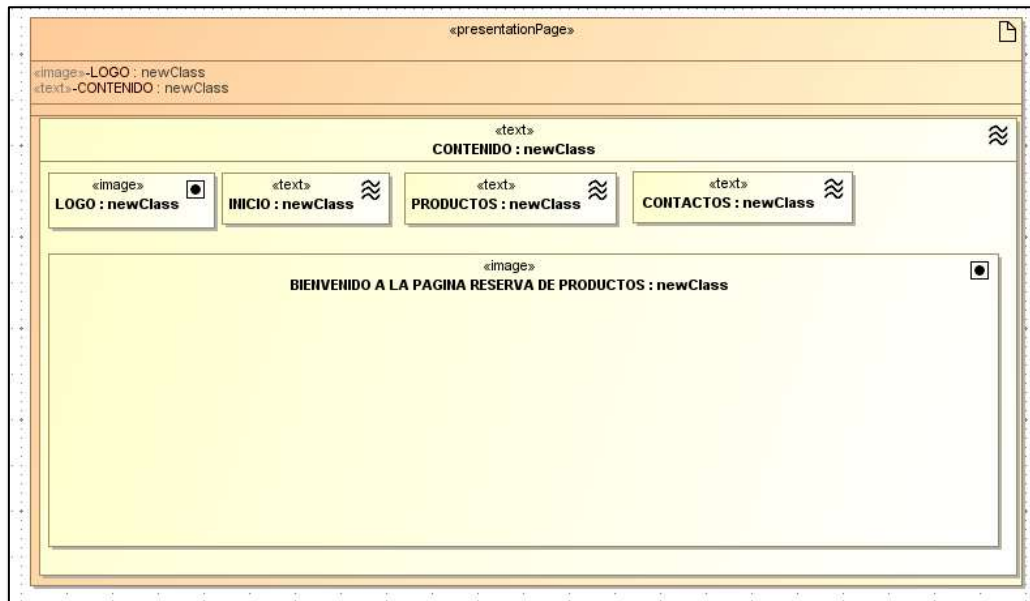
FIGURA Nº 3. 22 Modelo de Presentación: Secretaria.



Fuente: (Elaboración propia).

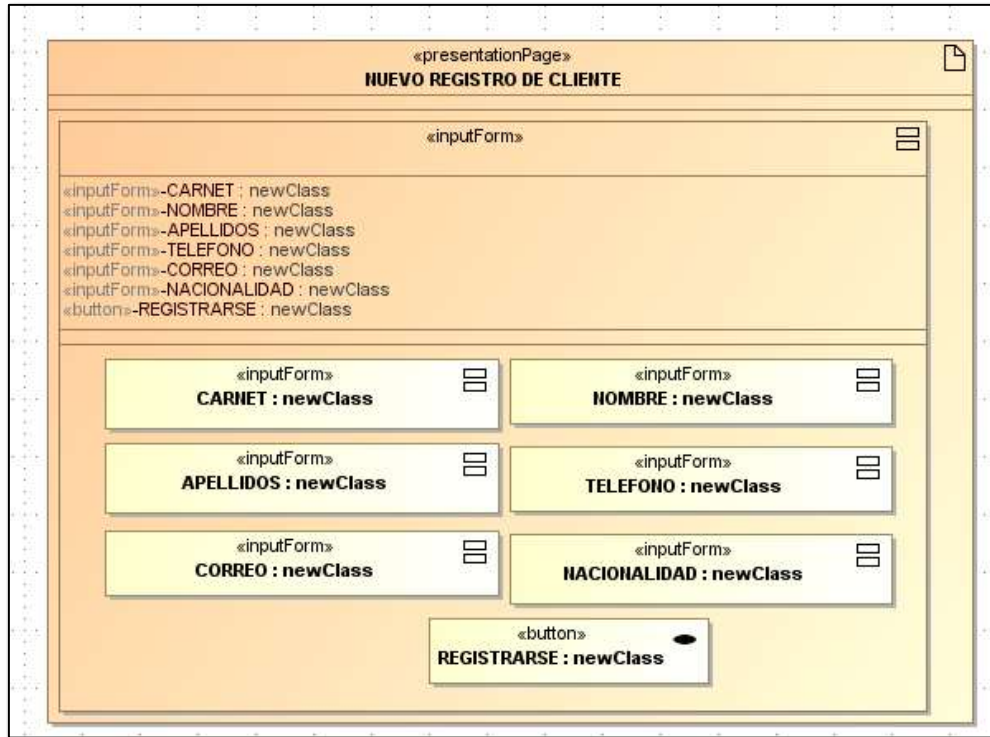
3.3.5.4 Modelo de Presentación: cliente.

FIGURA Nº 3. 23 Modelo de Presentación: cliente web



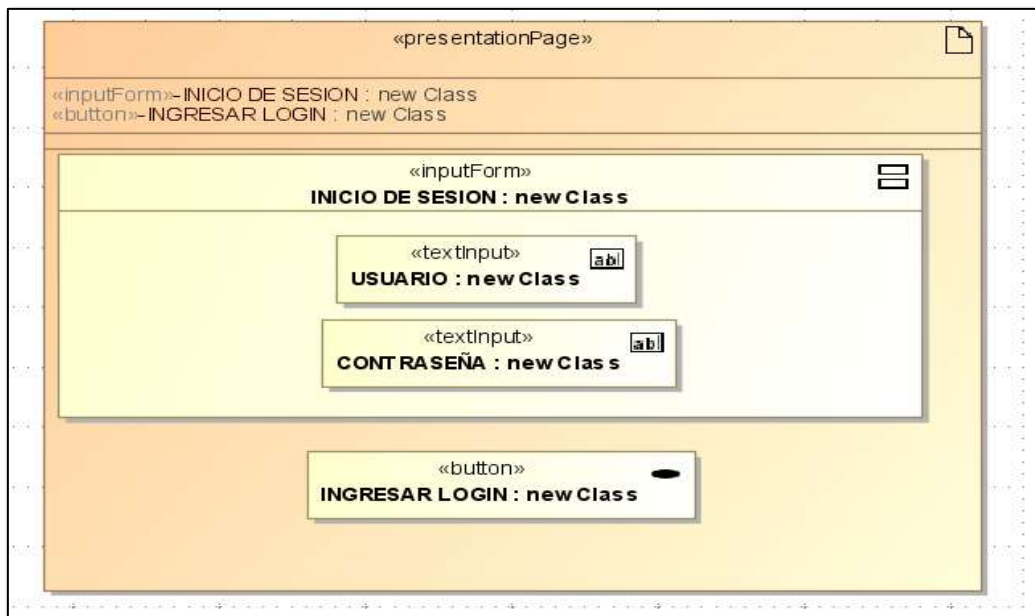
Fuente: (Elaboración propia).

FIGURA N° 3. 24 Modelo de Presentación: cliente form registro



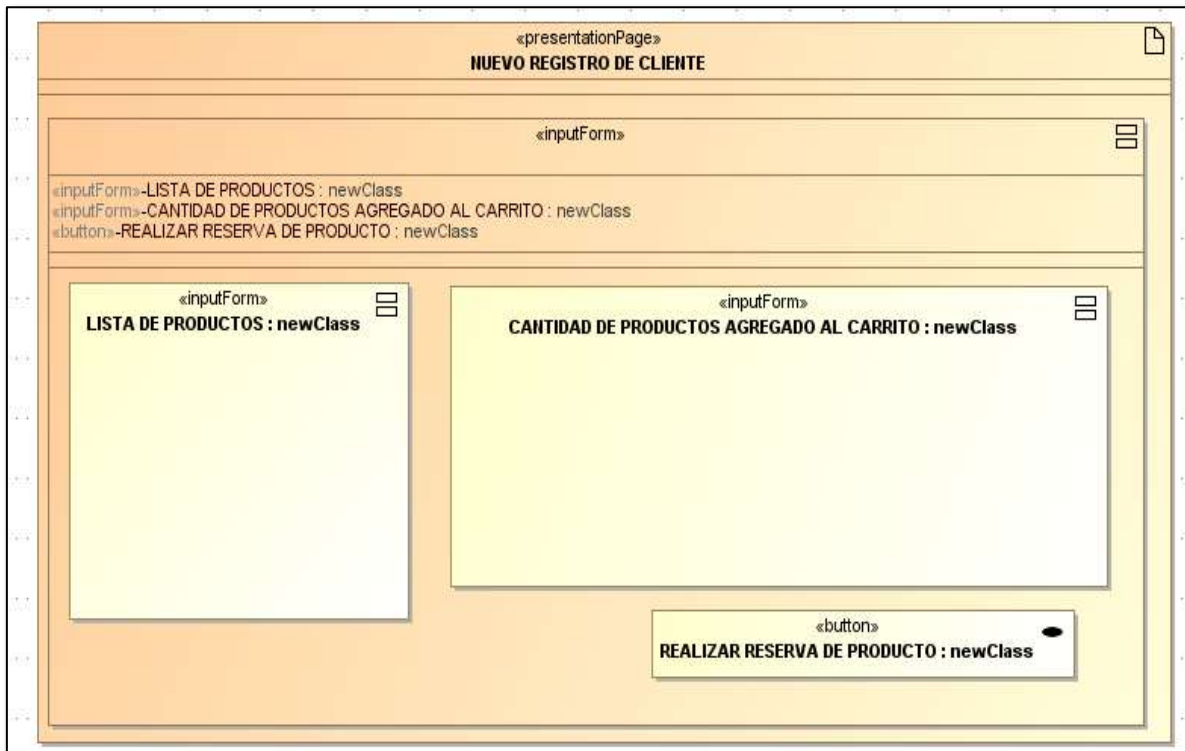
Fuente: (Elaboración propia).

FIGURA N° 3. 25 Modelo de Presentación: cliente login



Fuente: (Elaboración propia).

FIGURA N° 3. 26 Modelo de Presentación: cliente reserva producto



Fuente: (Elaboración propia).

3.4 FASE DE LA METODOLOGÍA UWE.

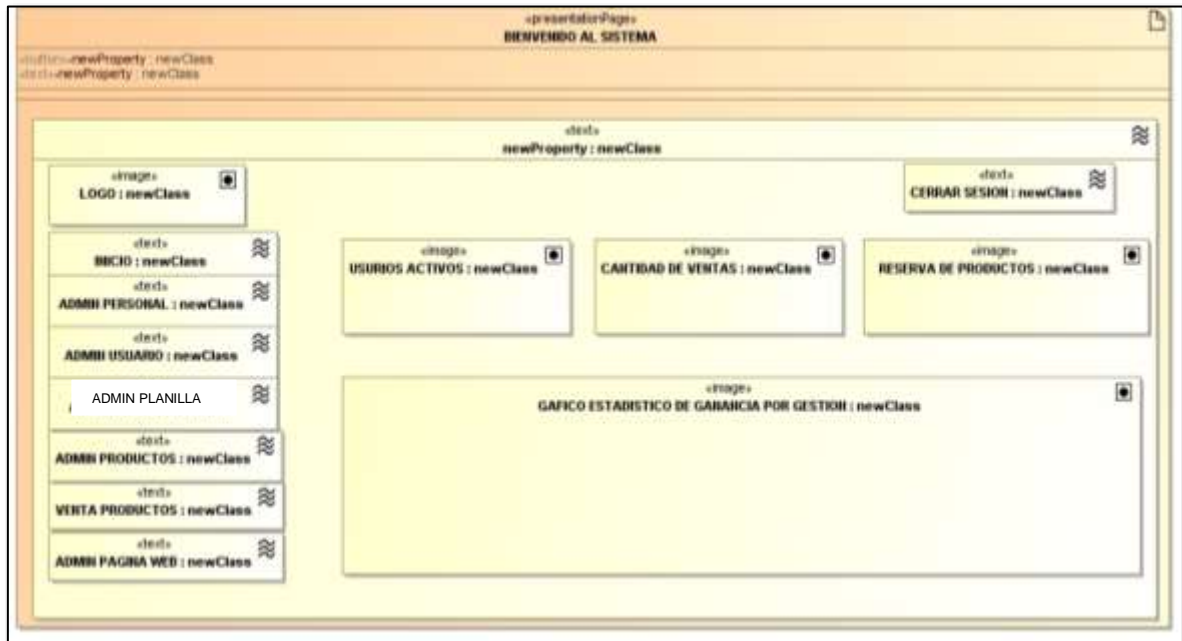
3.4.1 Captura, análisis y especificación de requisitos.

El requerimiento que se ha realizado en la institución es de desarrollar un sistema de información para el control de ingreso de materia prima, salida de cerámica y control del personal, el mismo que coadyuve en la toma de decisiones en la fábrica de cerámica “EMANUEL” de la localidad de Achocalla. Donde detallamos con más claridad en el índice 3.2 obtención de requisitos.

3.4.2 Diseño del sistema.

Se basa en la especificación de requisitos producido por el análisis de los requerimientos (fase de análisis), el diseño define cómo estos requisitos, la estructura que debe darse a la aplicación web.

FIGURA N° 3. 27 Diseño del sistema administrador



Fuente: (Elaboración propia).

3.4.3 Codificación del software.

Durante esta etapa se realizan las tareas que comúnmente se conocen como programación; que consiste, esencialmente, en llevar a código fuente, en el lenguaje de programación PHP.

FIGURA N° 3. 28 Codificación del software

```
224 <li>a href='<php echo base_url()>#controller_usuario/ver_producto/ver_producto'>ADMIN DE VISITAS</li> class="badge badge-emerald pull=
225
226 <li>a href='<php echo base_url()>#controller_usuario/ver_producto/reserva_producto'>ADMIN DE RESERVAS</li> class="badge badge-emerald
227
228 </ul>
229 </li>
230
231 <li class="sub-menu" >a href='<php echo base_url()>#controller_usuario/ver_producto/ver_producto'> class="btn btn-outline-light btn-sm" >
232 </li>
233
234 <li class="sub-menu" >a href='<php echo base_url()>#controller_usuario/ver_producto/ver_producto'> class="btn btn-outline-light btn-sm" >
235 </li>
236
237 <li class="sub-menu" >a href='<php echo base_url()>#controller_usuario/ver_producto/ver_producto'> class="btn btn-outline-light btn-sm" >
238 </li>
239
240 <li class="sub-menu" >a href='<php echo base_url()>#controller_usuario/ver_producto/ver_producto'> class="btn btn-outline-light btn-sm" >
241 </li>
242
243 <li class="sub-menu" >a href='<php echo base_url()>#controller_usuario/ver_producto/ver_producto'> class="btn btn-outline-light btn-sm" >
244 </li>
245
246 <li class="sub-menu" >a href='<php echo base_url()>#controller_usuario/ver_producto/ver_producto'> class="btn btn-outline-light btn-sm" >
247 </li>
248
249 </li>
250
251 <li class="sub-menu" >a href='<php echo base_url()>#controller_usuario/ver_producto/ver_producto'> class="btn btn-outline-light btn-sm" >
252 </li>
253
254 <li class="sub-menu" >a href='<php echo base_url()>#controller_usuario/ver_producto/ver_producto'> class="btn btn-outline-light btn-sm" >
255 </li>
256
257 </li>
258
259 <li class="sub-menu" >a href='<php echo base_url()>#controller_usuario/ver_producto/ver_producto'> class="btn btn-outline-light btn-sm" >
260 </li>
261
262 <li class="sub-menu" >a href='<php echo base_url()>#controller_usuario/ver_producto/ver_producto'> class="btn btn-outline-light btn-sm" >
263 </li>
264
265 </li>
266
267 </li>
268
269 </li>
270
271 </li>
272
273 </li>
274
275 </li>
276
277 </li>
278
279 </li>
280
281 </li>
282
283 </li>
284
285 </li>
286
287 </li>
288
289 </li>
290
291 </li>
292
293 </li>
294
295 </li>
296
297 </li>
298
299 </li>
300
301 </li>
302
303 </li>
304
305 </li>
306
307 </li>
308
309 </li>
310
311 </li>
312
313 </li>
314
315 </li>
316
317 </li>
318
319 </li>
320
321 </li>
322
323 </li>
324
325 </li>
326
327 </li>
328
329 </li>
330
331 </li>
332
333 </li>
334
335 </li>
336
337 </li>
338
339 </li>
340
341 </li>
342
343 </li>
344
345 </li>
346
347 </li>
348
349 </li>
350
351 </li>
352
353 </li>
354
355 </li>
356
357 </li>
358
359 </li>
360
361 </li>
362
363 </li>
364
365 </li>
366
367 </li>
368
369 </li>
370
371 </li>
372
373 </li>
374
375 </li>
376
377 </li>
378
379 </li>
380
381 </li>
382
383 </li>
384
385 </li>
386
387 </li>
388
389 </li>
390
391 </li>
392
393 </li>
394
395 </li>
396
397 </li>
398
399 </li>
400
401 </li>
402
403 </li>
404
405 </li>
406
407 </li>
408
409 </li>
410
411 </li>
412
413 </li>
414
415 </li>
416
417 </li>
418
419 </li>
420
421 </li>
422
423 </li>
424
425 </li>
426
427 </li>
428
429 </li>
430
431 </li>
432
433 </li>
434
435 </li>
436
437 </li>
438
439 </li>
440
441 </li>
442
443 </li>
444
445 </li>
446
447 </li>
448
449 </li>
450
451 </li>
452
453 </li>
454
455 </li>
456
457 </li>
458
459 </li>
460
461 </li>
462
463 </li>
464
465 </li>
466
467 </li>
468
469 </li>
470
471 </li>
472
473 </li>
474
475 </li>
476
477 </li>
478
479 </li>
480
481 </li>
482
483 </li>
484
485 </li>
486
487 </li>
488
489 </li>
490
491 </li>
492
493 </li>
494
495 </li>
496
497 </li>
498
499 </li>
500
501 </li>
502
503 </li>
504
505 </li>
506
507 </li>
508
509 </li>
510
511 </li>
512
513 </li>
514
515 </li>
516
517 </li>
518
519 </li>
520
521 </li>
522
523 </li>
524
525 </li>
526
527 </li>
528
529 </li>
530
531 </li>
532
533 </li>
534
535 </li>
536
537 </li>
538
539 </li>
540
541 </li>
542
543 </li>
544
545 </li>
546
547 </li>
548
549 </li>
550
551 </li>
552
553 </li>
554
555 </li>
556
557 </li>
558
559 </li>
560
561 </li>
562
563 </li>
564
565 </li>
566
567 </li>
568
569 </li>
570
571 </li>
572
573 </li>
574
575 </li>
576
577 </li>
578
579 </li>
580
581 </li>
582
583 </li>
584
585 </li>
586
587 </li>
588
589 </li>
590
591 </li>
592
593 </li>
594
595 </li>
596
597 </li>
598
599 </li>
600
601 </li>
602
603 </li>
604
605 </li>
606
607 </li>
608
609 </li>
610
611 </li>
612
613 </li>
614
615 </li>
616
617 </li>
618
619 </li>
620
621 </li>
622
623 </li>
624
625 </li>
626
627 </li>
628
629 </li>
630
631 </li>
632
633 </li>
634
635 </li>
636
637 </li>
638
639 </li>
640
641 </li>
642
643 </li>
644
645 </li>
646
647 </li>
648
649 </li>
650
651 </li>
652
653 </li>
654
655 </li>
656
657 </li>
658
659 </li>
660
661 </li>
662
663 </li>
664
665 </li>
666
667 </li>
668
669 </li>
670
671 </li>
672
673 </li>
674
675 </li>
676
677 </li>
678
679 </li>
680
681 </li>
682
683 </li>
684
685 </li>
686
687 </li>
688
689 </li>
690
691 </li>
692
693 </li>
694
695 </li>
696
697 </li>
698
699 </li>
700
701 </li>
702
703 </li>
704
705 </li>
706
707 </li>
708
709 </li>
710
711 </li>
712
713 </li>
714
715 </li>
716
717 </li>
718
719 </li>
720
721 </li>
722
723 </li>
724
725 </li>
726
727 </li>
728
729 </li>
730
731 </li>
732
733 </li>
734
735 </li>
736
737 </li>
738
739 </li>
740
741 </li>
742
743 </li>
744
745 </li>
746
747 </li>
748
749 </li>
750
751 </li>
752
753 </li>
754
755 </li>
756
757 </li>
758
759 </li>
760
761 </li>
762
763 </li>
764
765 </li>
766
767 </li>
768
769 </li>
770
771 </li>
772
773 </li>
774
775 </li>
776
777 </li>
778
779 </li>
780
781 </li>
782
783 </li>
784
785 </li>
786
787 </li>
788
789 </li>
790
791 </li>
792
793 </li>
794
795 </li>
796
797 </li>
798
799 </li>
800
801 </li>
802
803 </li>
804
805 </li>
806
807 </li>
808
809 </li>
810
811 </li>
812
813 </li>
814
815 </li>
816
817 </li>
818
819 </li>
820
821 </li>
822
823 </li>
824
825 </li>
826
827 </li>
828
829 </li>
830
831 </li>
832
833 </li>
834
835 </li>
836
837 </li>
838
839 </li>
840
841 </li>
842
843 </li>
844
845 </li>
846
847 </li>
848
849 </li>
850
851 </li>
852
853 </li>
854
855 </li>
856
857 </li>
858
859 </li>
860
861 </li>
862
863 </li>
864
865 </li>
866
867 </li>
868
869 </li>
870
871 </li>
872
873 </li>
874
875 </li>
876
877 </li>
878
879 </li>
880
881 </li>
882
883 </li>
884
885 </li>
886
887 </li>
888
889 </li>
890
891 </li>
892
893 </li>
894
895 </li>
896
897 </li>
898
899 </li>
900
901 </li>
902
903 </li>
904
905 </li>
906
907 </li>
908
909 </li>
910
911 </li>
912
913 </li>
914
915 </li>
916
917 </li>
918
919 </li>
920
921 </li>
922
923 </li>
924
925 </li>
926
927 </li>
928
929 </li>
929 </pre>
```

Fuente: (Elaboración propia).

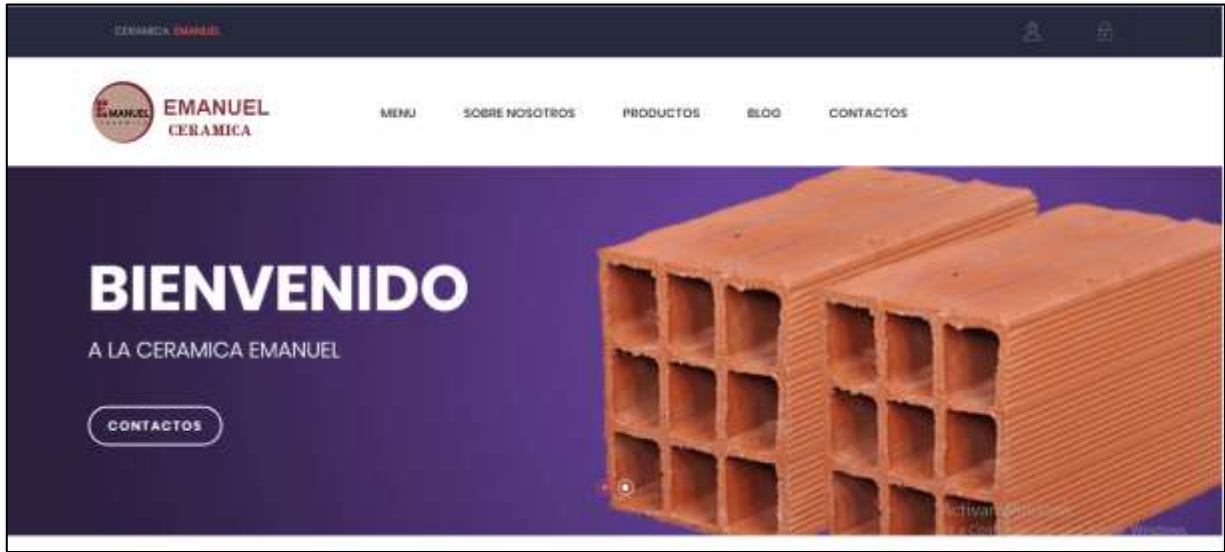
3.4.4 Pruebas.

Las pruebas se utilizan para asegurar el correcto funcionamiento de los distintos módulos, los formularios y secciones de código del sistema.

Para más detalle sobre contenidos de los módulos podemos ingresar a anexos manual de usuario, donde presentaremos en detalle los interfaces o panel de distintos módulos que contempla el sistema.

En la siguiente figura se presenta la interfaz de la página web, donde se puede visualizar los productos que ofrece la fábrica de cerámica Emanuel y de la misma manera los clientes podrán realizar las respectivas reservas del producto.

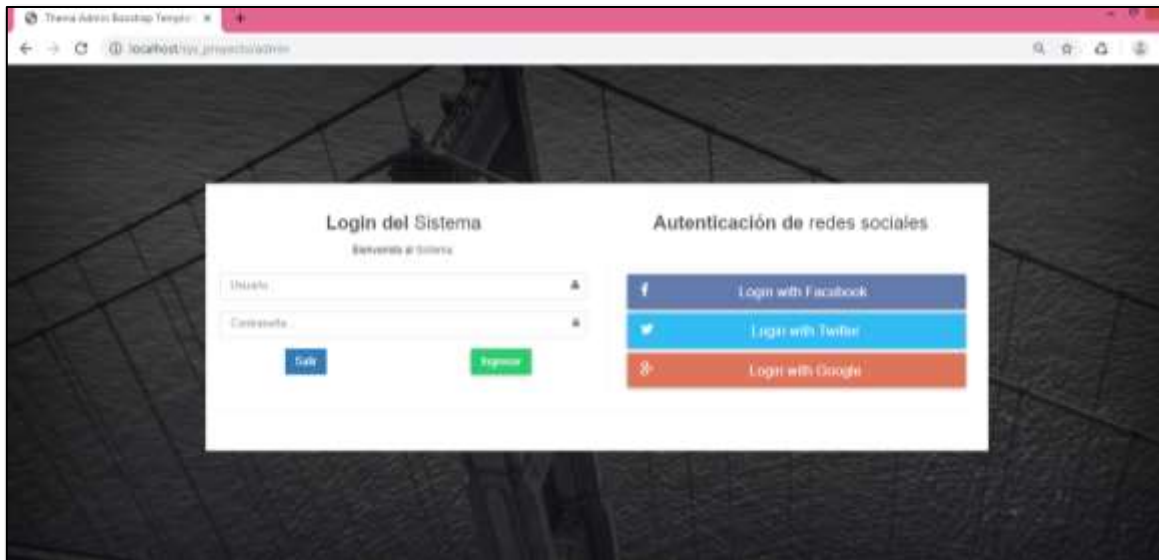
FIGURA N° 3. 29 Pruebas menú inicio



Fuente: (Elaboración propia).

En la siguiente figura se puede visualizar un inicio de sesión para el ingreso al sistema de administración, donde nos solicita un usuario y una contraseña.

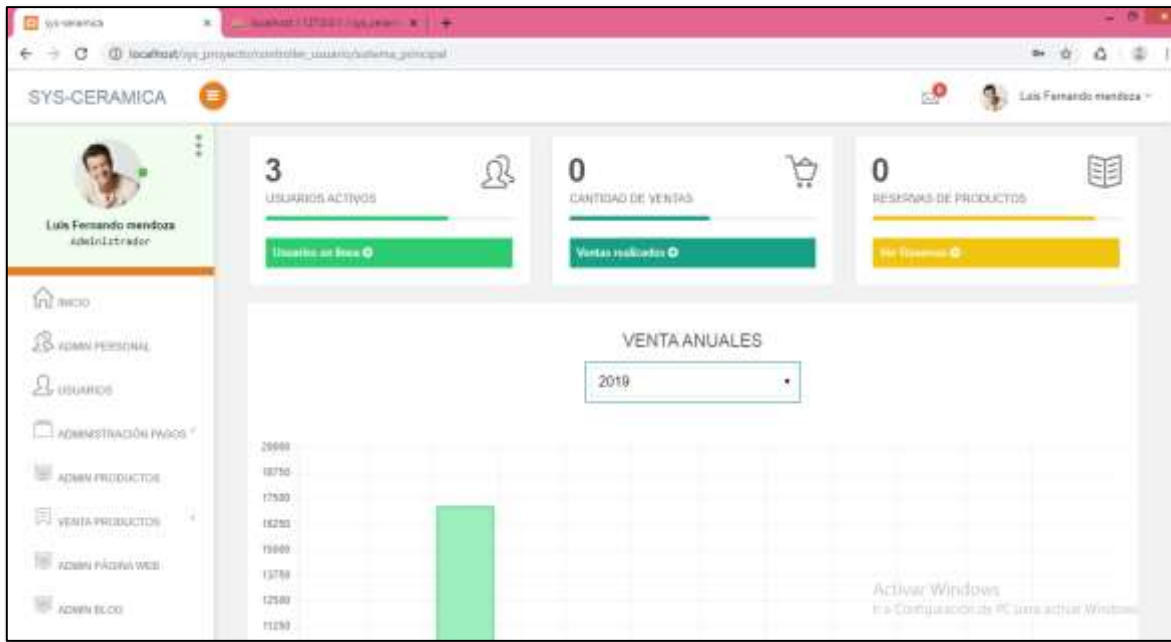
FIGURA N° 3. 30 Pruebas login usuario



Fuente: (Elaboración propia).

Una vez ingresado al sistema con un respectivo rol designado por el administrador del sistema nos permite visualizar los respectivos menús, donde pueda realizar sus respectivas funciones.

FIGURA N° 3. 31 Pruebas menú administrador



Fuente: (Elaboración propia).

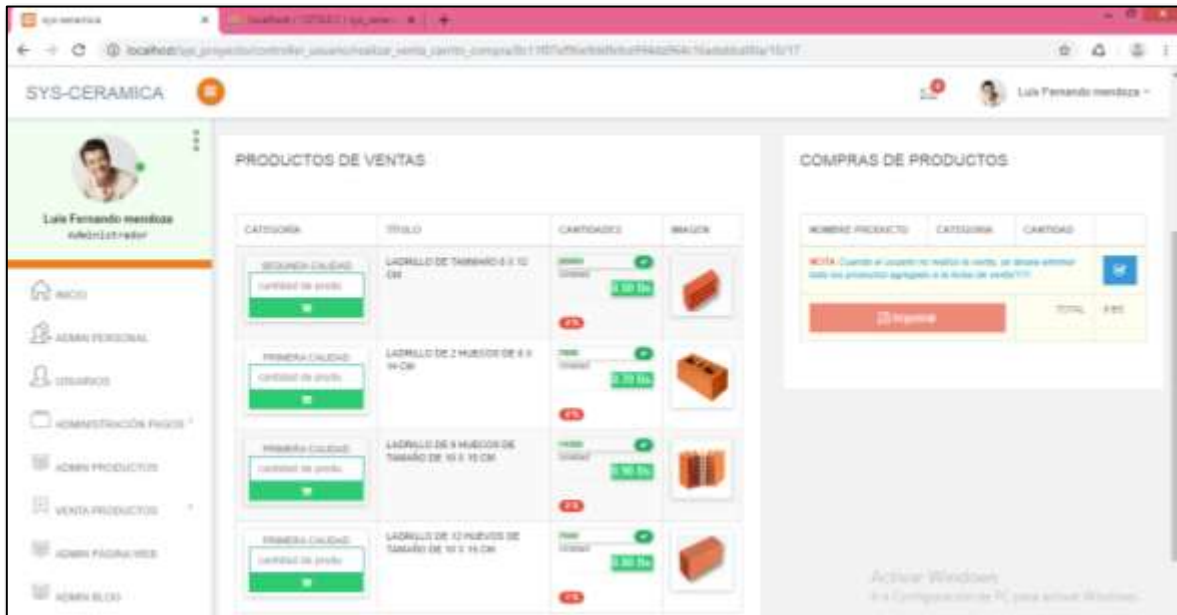
FIGURA N° 3. 32 Pruebas de administración de producto

The screenshot shows the product administration interface. It features a table with columns for 'TIPO PRODUCTO', 'FINALE', 'CATEGORIA', 'CANTIDADES STOCK', 'PRECIO', 'IMAGEN', and 'DESCRIPCIÓN'. The table lists four types of ceramic tiles with their respective stock levels, prices, and descriptions.

TIPO PRODUCTO	FINALE	CATEGORIA	CANTIDADES STOCK	PRECIO	IMAGEN	DESCRIPCIÓN
TIPO PARED	LADRILLO DE TIRABANDA A X 45 CM	PRESEDA CAJEDOS	11980 Disponible	1.50 Bs		Detalle técnico solo en subproducto principal. Precios por m ² . 02/2020 a 12/2020
TIPO PARED	LADRILLO DE 8 HUECOS DE 30 X 45 CM	PRESEDA CAJEDOS	11980 Disponible	1.70 Bs		Detalle técnico solo en subproducto principal. Precios por m ² . 02/2020 a 12/2020
TIPO EN PISO	LADRILLO DE 8 HUECOS DE TIRABANDA DE 30 X 45 CM	PRESEDA CAJEDOS	11980 Disponible	1.50 Bs		Detalle técnico solo en subproducto principal. Precios por m ² . 02/2020 a 12/2020
TIPO PARED	LADRILLO DE 12 HUECOS DE TIRABANDA DE 30 X 45 CM	PRESEDA CAJEDOS	11980 Disponible	1.50 Bs		Detalle técnico solo en subproducto principal. Precios por m ² . 02/2020 a 12/2020

Fuente: (Elaboración propia).

FIGURA N° 3. 33 Pruebas realizar venta a cliente



Fuente: (Elaboración propia).

FIGURA N° 3. 34 Pruebas de comprobante

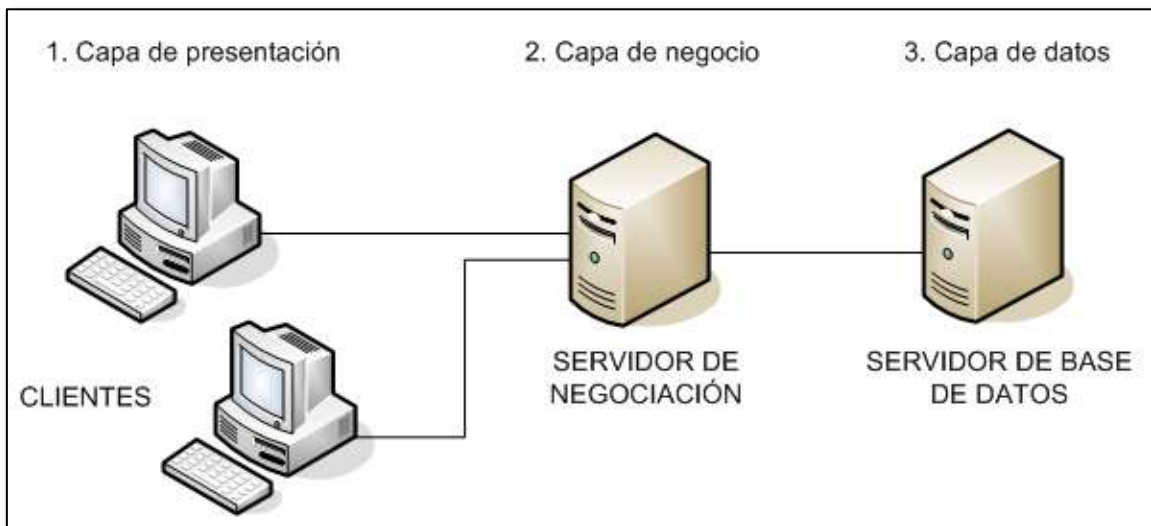


Fuente: (Elaboración propia).

3.4.5 La Instalación o fase de Implementación.

Proceso por el cual el sistema desarrollado son transferidos apropiadamente a un servidor donde se alojarán el sistema por el cual donde se solicitará un dominio para el acceso al sistema, todo ellos con el propósito de ser ya utilizados por el usuario final. Esto incluye la implementación de la arquitectura, al criterio del usuario, de los mecanismos adaptativos y las tareas referentes a la integración de todas estas implementaciones.

FIGURA N° 3. 35 Instalación o fase de Implementación



Fuente: (Arenas Paredes , 2011)

3.4.6 Mantenimiento:

Es el proceso de control, mejora y optimización del software ya desarrollado e instalado, que también incluye depuración de errores y defectos que puedan haberse filtrado de la fase de pruebas de control.

El mantenimiento costa la implementación de algunos módulos o mejora de ciertos interfaces grafica del sistema al criterio del usuario.

3.5 MÉTRICAS DE CALIDAD DE SOFTWARE

En este índice se verá el desarrollo de la medición de la calidad del software mediante la métrica ISO-9126 lo cual es un estándar internacional para la evaluación de software, que establece que cualquier componente de la calidad pueda ser descrito por las características de Funcionalidad, Confiabilidad, Mantenibilidad, Usabilidad y Portabilidad.

3.5.1 Funcionalidad

La funcionalidad no se puede medir directamente por esta razón corresponde derivar medidas directas como es el punto función que cuantifica el tamaño y la complejidad del sistema en términos de las funciones del usuario. Para la funcionalidad o medición del sistema, se debe determinar las siguientes características:

TABLA Nº 3. 16 Característica de la funcionalidad

CARACTERÍSTICAS	DESCRIPCIÓN
Número de Entradas de Usuario	Se origina al usuario, cuando realiza el ingreso de datos orientado a la aplicación
Número de Salidas de Usuario	Se cuenta cada salida que proporciona información orientada a la aplicación del usuario.
Numero de petición de Usuario	En una entrada en línea que lleva a la generación de algunas respuestas inmediatas por parte del software.
Numero de archivos	Se cuenta cada archivo lógico maestro, cada archivo lógico interno que es un agrupamiento lógico de datos como ser parte de una base de datos o archivo independientes
Numero de Interfaces externas	Se cuentan todas las interfaces legibles por la máquina.

Fuente: (Elaboración Propia)

Número de Entradas de usuario, se cuenta cada entrada de usuario que proporciona diferentes datos orientados a la aplicación. Las entradas se deberían diferenciar de las peticiones, las cuales se cuentan de forma separada.

TABLA N° 3. 17 Número de Entradas de Usuario

Entradas de Usuario		
1	Módulo de administración de personal	6
2	Módulo de administración de usuario	3
3	Módulo de administración pagos	13
4	Módulo de administración productos	6
5	Módulo de ventas productos	15
6	Módulo administración página web	7
Total		50

Fuente: (Elaboración Propia)

Número de Salidas Usuario, se cuenta cada salida que proporciona al usuario información orientada a la aplicación. En este contexto la salida se refiere a informes, pantallas, mensajes de error, etc. Los elementos de datos particulares dentro de un informe no se cuentan de forma separada.

TABLA N° 3. 18 Número de Salidas Usuario

Salidas de Usuario		
1	Módulo de administración de personal	7
2	Módulo de administración de usuario	4
3	Módulo de administración pagos	15
4	Módulo de administración productos	9
5	Módulo de ventas productos	16
6	Módulo administración página web	8
Total		59

Fuente: (Elaboración Propia)

Número de Peticiones de usuario, una petición se define como una entrada interactiva que produce la generación de alguna respuesta del software inmediata en forma de salida interactiva. Se cuenta cada petición por separada.

TABLA N° 3. 19 Número de Peticiones del Usuario

Peticiones del Usuario		
1	Módulo de administración de personal	5
2	Módulo de administración de usuario	3
3	Módulo de administración pagos	5
4	Módulo de administración productos	7
5	Módulo de ventas productos	10
6	Módulo administración página web	4
Total		34

Fuente: (Elaboración Propia).

Número de Archivos, se cuenta archivo maestro Lógico

TABLA N° 3. 20 Número de Archivos

Número de Archivos		
1	Módulo de administración de personal	5
2	Módulo de administración de usuario	6
3	Módulo de administración pagos	5
4	Módulo de administración productos	11
5	Módulo de ventas productos	8
6	Módulo administración página web	12
Total		47

Fuente: (Elaboración Propia)

Número de interfaces Externas, se cuenta todas las interfaces legibles por la máquina.

TABLA N° 3. 21 Número de Interfaces Externas

Interfaces Externas		
1	Internet	5
Total		5

Fuente: (Elaboración Propia)

Para realizar el cálculo de la cuenta total con factores de ponderación se debe realizar con la siguiente tabla.

TABLA N° 3. 22 Factores de Ponderación

Parámetros de medida		Cuenta	Factor	Total
1	Nro. de Entradas de Usuario	50	4	200
2	Nro. de Salidas de Usuario	59	5	295
3	Nro. de Peticiones de Usuario	34	4	136
4	Nro. de Archivos	47	10	470
5	Nro. de Interfaces Externas	5	7	35
Cuenta total				1136

Fuente: (Elaboración Propia)

Valores de ajuste de complejidad según las respuestas a las siguientes preguntas que se muestra en la siguiente tabla.

TABLA N° 3. 23 Valores de Ajuste de Complejidad

Nro	Factor de Complejidad	Sin Influencia	Incidental	Moderada	Medio	Significativa	Esencial	Fi
		0	1	2	3	4	5	
1	¿Requiere el sistema copia de seguridad y recuperación?						X	5
2	¿Requiere comunicación de datos?						X	5
3	¿Existen funciones de procesos distribuidos?				X			3
4	¿El rendimiento es crítico				X			3
5	¿Sera ejecutado el sistema en entorno existente y fuertemente utilizado?						X	5
6	¿Entrada de datos EN LINEA?						X	5
7	¿Requiere la entrada de datos interactiva que las transiciones de entrada se llevan a cabo sobre múltiples pantallas o variadas opciones?						X	5
8	¿Se actualizan los archivos maestros de forma interactiva?						X	5
9	¿Son complejas de las entradas de salidas de archivos?			X				2
10	¿Lógica del proceso Interno Compleja?					X		3
11	¿Se diseña el código para ser reutilizable?						X	5
12	¿Están incluidas en el diseño conversiones de instalación?			X				2
13	¿instalaciones Múltiples?			X				2
14	¿Facilidad de Cambios?						X	5
Factor de Complejidad Total (FCT)								56

Fuente: (Elaboración Propia)

Para calcular los puntos función (PF), utilizaremos la relación siguiente

$$\mathbf{PF = Cuenta Total * (0.65 + 0.01 * \sum Fi)}$$

Dónde:

Cuenta Total: Nivel de complejidad del sistema con respecto al usuario

(0.65+0.01*∑Fi): Ajuste de complejidad según el dominio de la información.

0.01: Factor de conversión, es decir un error de 1%

0.65: Valor mínimo de ajuste

Calculando el punto función de según la ecuación:

$$\mathbf{PF = 1136 * [0.65 + 0.01 * 56]}$$

$$\mathbf{PF = 1374.56}$$

Si calculamos al 100% el nivel de confianza consideramos que la sumatoria de $F_i=70$ como el máximo valor de ajuste de complejidad se tiene:

$$\mathbf{PF \max = Cuenta Total * [1 + 0.01 * \sum Fi]}$$

$$\mathbf{PF \max = 1136 * [0.65 + 0.01 * 70]}$$

$$\mathbf{PF = 1533.6}$$

Con los máximos valores de ajuste de complejidad se tiene que la funcionalidad real es:

$$\mathbf{Funcionalidad = \frac{1374.56}{1533.6} = 0,90}$$

$$\mathbf{Funcionalidad = 0,90 * 100 = 90\%}$$

Entonces la funcionalidad del sistema es un 90% esto quiere decir que el sistema tiene un 90% que funcione sin riesgos de fallo y operatividad constante y 10% de colapso de sistema.

3.5.2 Confiabilidad

La confiabilidad del sistema se define como la probabilidad de operación libre de fallos de un programa de computadora en un entorno determinado y durante un tiempo específico.

Para calcular confiabilidad del sistema se toma en cuenta el periodo de tiempo en el cual se ejecuta y se obtiene muestras

$$F(t) = f * e^{(-\mu * t)}$$

En el inicio de ejecución $t_0=0$ lo que significa el tiempo inicial en el cual dará inicio el funcionamiento del sistema.

$$F(0) = f * e^{(-\mu * t_0)}$$

Se observa el trabajo del sistema hasta que produce una falla en el instante T, el cual se aproxima a una variable aleatoria continua.

Como se aproxima a variables aleatorias continuas, la confiabilidad a ser obtenida en términos probabilísticos.

Entonces el término en el cual el sistema trabaja sin falla está dado por la ecuación (2) y tiempo en el cual no falla el sistema está dado por ecuación (3).

$$P(T \leq t) = F(t) \quad (2) \text{ Probabilidad de fallas}$$

$$P(T \leq t) = 1 - F(t) \quad (3) \text{ Probabilidad de trabajo sin falla}$$

En un periodo de 20 días como tiempo de prueba se define de cada 10 ejecuciones 1 falla.

Conociendo la funcionalidad del 90% del sistema calculamos para el periodo establecido.

$$P(T \leq t) = 1 - F(t)$$

$$F(t) = 1 - 90 + e^{(-\frac{1}{10} * 20)}$$

$$F(t) = 1 - 0,122$$

$$F(t) = 0,88$$

La confiabilidad del sistema es del 88% en un periodo de 20 días como tiempo de prueba.

3.5.3 Mantenibilidad

El mantenimiento se desarrolla para mejorar el sistema en respuesta a los nuevos requerimientos y los reglamentos que está regida por la por la institución.

El estándar IEE94 sugiere un índice de madurez del software (IMS) que proporciona un indicador en la estabilidad de un producto, se lo determina con la siguiente formula.

$$IMS = \frac{[Mt - (Fa + Fc + Fd)]}{Mt}$$

Mt = Número de Módulos la versión actual

Fa = Número de Módulos en la versión actual que se han añadido

Fc: Número de Módulos en la versión actual que se han Cambiado.

Fd: Número de Módulos en la versión anterior que se han borrado en la versión actual.

Calculado el IMS

$$IMS = [6 - (1 + 0 + 0)] / 6$$

$$IMS = 0,83$$

Con lo que podemos decir que el nuevo sistema tiene una estabilidad de 83% que es la facilidad de mantenimiento, el 17% restante es el margen de error correspondiente a los cambios y modificaciones efectuados desde el prototipo de la versión actual.

Puesto que es un sistema diseñado con los requerimientos actuales con el tiempo surgirán nuevos requerimientos los cuales cambiará el valor del índice de madurez del software.

Mantenimiento Correctivo, el sistema presenta diseño modular y es por eso que tolera variaciones en su corrección.

Mantenimiento Adaptativo. Se realiza cuando en la organización se produce algún cambio haciendo que el sistema sufra modificaciones.

El sistema por su programación modular permitirá fácilmente hacer modificaciones en sus módulos o integrar nuevos al sistema.

3.5.4 Usabilidad

La usabilidad es lo mismo decir facilidad de uso, esta métrica nos muestra el costo de aprender a manejar el producto, lo cual se calcula con la siguiente fórmula:

$$FU = \left[\frac{\text{Sum}(xi)}{n} \right] * 100$$

Donde:

Xi: es la sumatoria de valores

n: es el número de preguntas

Para responder a las preguntas se debe considerar la siguiente tabla:

TABLA N° 3. 24 Ajuste de Preguntas

ESCALA	VALOR
Muy bueno	5
Bueno	4
Regular	3
Malo	2
Pésimo	1

Fuente: (Elaboración Propia)

TABLA N° 3. 25 Ajuste de Preguntas

PREGUNTAS	Respuestas		Ponderación %
	SI	NO	
¿Puede Utilizar con facilidad el sistema?	4	1	80%
¿Puede Controlar operaciones que el sistema solicita?	4	1	80%
¿Las Respuestas del sistema son complicadas?	1	4	80%
¿El Sistema permitió la retroalimentación de información?	5	0	100%
¿El sistema cuenta con interface agradable a la vista?	5	0	100%
¿La respuesta del sistema es satisfactoria?	4	1	80%
¿Le parece complicada las funciones del sistema?	1	4	80%
¿Se hace difícil o dificultoso aprender a manejar el sistema?	1	4	80%
¿Los resultados que proporciona el sistema facilitan el trabajo?	5	0	100%
¿Durante el uso del sistema se produjo errores?	1	4	80%
USABILIDAD			86%

Fuente: (Elaboración Propia)

Existe un 86% de comprensión o entendimiento de los usuarios con respecto a la capacidad del sistema.

3.5.5 Portabilidad

Para la portabilidad del sistema se tomará en cuenta dos aspectos como ser a nivel aplicación, nivel hardware la portabilidad la dividimos en dos secciones portabilidad del lado del servidor y portabilidad del lado del cliente.

- *Adaptabilidad.* Capacidad del producto que le permite ser adaptado de forma efectiva y eficiente a diferentes entornos determinados de hardware, software, operacionales o de uso.
- *Capacidad para ser ejecutado.* Facilidad con la que el producto se puede ejecutar de forma exitosa en un determinado sistema operativo.
- *Capacidad para ser reemplazado.* Capacidad del producto para ser utilizado en lugar de otro producto software determinado con el mismo propósito y en el mismo entorno.

Para el cálculo de la portabilidad se tomó en cuenta la siguiente tabla que contiene las características que se mencionaron anteriormente.

TABLA N° 3. 26 Portabilidad

FACTOR DE PORTABILIDAD	VALOR %
Puede ser abierto en cualquier sistema operativo	96
Se puede adaptar a los ambientes con facilidad (empresas similares)	100
Es fácil de usar	100
Es capaz de reemplazar a un sistema similar	96
TOTAL	98

Fuente: (Elaboración Propia)

La interpretación a este resultado significa que nuestro sistema tiene la capacidad de ejecutarse en cualquier sistema operativo.

Portabilidad = 98%

Portabilidad lado servidor

A nivel sistema de software. El sistema porta bajo los siguientes sistemas operativos de la familia Microsoft, Windows 7, Windows 8, Windows 8.1, Windows 10 y todo el sistema operativo libre de Linux.

A nivel de base de datos se utiliza base de datos creada en MySql, la portabilidad se muestra que la base de datos que la base de datos puede ser migrada SQL server, Oracle y Postgres.

A nivel Hardware el sistema porta bajo las siguientes características mínimas de hardware microprocesador Pentium IV RAM de 1GB Como mínimo espacio en el disco duro 50 Gb como mínimo, monitor SVGA.

3.5.6 Resultados

De acuerdo a los resultados obtenidos se puede establecer la calidad total del sistema en base a los parámetros medidos anteriormente. La calidad está directamente relacionada con el grado de satisfacción con el usuario que ingresa al sistema.

TABLA N° 3. 27 Resultados

CARACTERÍSTICAS	RESULTADO
Funcionalidad	90
Confiabilidad	88
Mantenibilidad	83
Usabilidad	86
Portabilidad	98
TOTAL	89

Fuente: (Elaboración Propia)

Evaluación de calidad total = 89%

3.6 MÉTODO DE ESTIMACIÓN DE COSTO DE SOFTWARE

Existe distintos métodos para la estimación de costo de desarrollo de software, estos métodos no son otra cosa que establece una relación matemática entre el esfuerzo y el tiempo de desarrollo.

3.6.1 Método de estimación de COCOMO

La estimación de costos del sistema ha sido desarrollada bajo la KLDC (kilo-líneas de código) como se detalla a continuación.

Para calcular el esfuerzo, necesitaremos hallar la variable KLDC (kilo-líneas de código)

Este proyecto se implementa 6700 Líneas de Código en el lenguaje PHP.

Aplicando Conversiones se tiene

$$KLCD = (LCD)/1000$$

$$KLCD = 6700 / 1000$$

$$KLDC = 6.7 KLDC$$

Es un modelo Intermedio y sistema orgánico la evaluación del sistema ha sido considerada bajo las 6700 KLDC los coeficientes que se usarán serán los valores que se detallan en la siguiente tabla.

TABLA N° 3. 28 Aplicación del Modelo Intermedio

PROYECTO SOFTWARE	a	b	c	d
Orgánico	3,2	1,05	2,5	0,38
Semi acoplado	3,0	1,12	2,5	0,35
Empotrado	2,8	1,20	2,5	0,32

Fuente:(S. Pressman, 2010)

A continuación, presentaremos las ecuaciones que nos permitirán el costo total de software:

TABLA Nº 3. 29 Ecuaciones del modelo COCOMO II

Variable	Ecuación	Tipo/Unidad
Esfuerzo requerido por el proyecto	$E = a \times (KLDC)^b \times FAE$	Personas/mes
Tiempo requerido por el proyecto	$T = c \times (E)^n$	mese
Número de personas requeridas para el proyecto	$NP = \frac{E}{T}$	personas
Costo total	$CT = \text{Sueldo Mes} \times NP \times T$	\$us

Fuente:(S. Prentice-Hall, 1981)

Para hallar los valores de FAE, se utilizará la tabla de atributos multiplicadores

TABLA Nº 3. 30 Cálculo de Atributos FAE

Atributos	Valor					
	Muy bajo	Bajo	Nominal	Alto	Muy alto	Extra alto
Atributos de software						
Fiabilidad	0,75	0,88	1,00	1,15	1,40	
Tamaño de Base de datos		0,94	1,00	1,08	1,16	
Complejidad	0,70	0,85	1,00	1,15	1,30	1,65
Atributos de hardware						
Restricciones de tiempo de ejecución			1,00	1,11	1,30	1,66
Restricciones de memoria virtual			1,00	1,06	1,21	1,56
Volatilidad de la máquina virtual		0,87	1,00	1,15	1,30	
Tiempo de respuesta		0,87	1,00	1,07	1,15	

Atributos de personal						
Capacidad de análisis	1,46	1,19	1,00	0,86	0,71	
Experiencia en la aplicación	1,29	1,13	1,00	0,91	0,82	
Calidad de los programadores	1,42	1,17	1,00	0,86	0,70	
Experiencia en la máquina virtual	1,21	1,10	1,00	0,90		
Experiencia en el lenguaje	1,14	1,07	1,00	0,95		
Atributos del proyecto						
Técnicas actualizadas de programación	1,24	1,10	1,00	0,91	0,82	
Utilización de herramientas de software	1,24	1,10	1,00	0,91	0,83	
Restricciones de tiempo de desarrollo	1,22	1,08	1,00	1,04	1,10	
Total			0,72			

Fuente: (Elaboración Propia)

Por tanto, nuestro Factor de ajuste será:

$$FAE = 0,72$$

Aplicando y reemplazando valores a la fórmula de esfuerzo, se tiene:

$$E = a * KLCD^b * FAE \text{ (Personas/Mes)}$$

$$E = 3,2 * 6,7^{1,05} * 0,72$$

$$E = 17 \text{ (Personas Mes)}$$

Cálculo del Tiempo

$$T = c * Esfuerzo^d \text{ (Meses)}$$

$$T = 2,5 * 17^{0,38}$$

$$T = 7.33 \text{ (Meses)}$$

Cálculo de la Productividad

$$PR = \frac{LCD}{Esfuerzo}(\text{Meses})$$

$$PR = \frac{6700}{17}(\text{Meses})$$

$$PR = 394.1 \text{ (LCD/personas Mes)}$$

Cálculo del personal requerido

$$P = \frac{E}{T}(\text{Personas})$$

$$P = \frac{17}{7.33}(\text{Personas})$$

$$P = 2.31(\text{Personas}) \text{ equivalente a 2 personas}$$

Costo Total del Proyecto

$$(\text{Coste Mes}) = P * \text{Salario medio entre los programadores y analistas}$$

Costo Persona Mes 350\$

$$\text{Coste Mes} = 2 * 430 = 860\$us$$

$$\text{Costo Total} = 860 * 7 = 6020\$us$$

En resumen, se requiere 2 personas estimando un trabajo de 7 meses y con costo total de 6020\$us, 6.90 Equivalente en bolivianos a 41538 Bs.

3.7 SEGURIDAD DEL SISTEMA

3.7.1 ISO-27002

La ISO-27002 evalúa y rectifica la implementación mediante el cumplimiento de normas, así como la mejor forma de controles que permitan reducir el riesgo de sufrir incidentes de seguridad en el funcionamiento de la institución en la información, para lo cual se toma los siguientes tipos de seguridad.

3.7.1.1 Seguridad l3gica.

Gesti3n de comunicaci3n y operaciones.

- ❖ Los respaldos (Back-up) de la base de datos (BD).
- ❖ Los personales involucrados en el proceso del sistema deber3n cambiar el Password del sistema peri3dicamente una vez al mes.
- ❖ Recomendamos al personal involucrado en el sistema cada vez que ingrese al sistema en recomendable que cierre sesi3n.

3.7.1.2 Seguridad f3sica

Seguridad f3sica del entorno de alojamiento de sistema.

- ❖ Se recomienda los back-up 2 a 3 copias en distintos discos de almacenamiento.
- ❖ Las copias de la base de datos (BD) deber3 ser protegido en un 3rea donde solo el personal autorizado tenga el acceso.

3.7.1.3 Seguridad organizativa

En 3l seguridad del sistema se consideran las siguientes precauciones:

- ❖ Autenticaci3n de usuarios
- ❖ Manejo de tipos de usuario en la aplicaci3n
- ❖ Se aplica el paradigma de Modelo Vista Controlador para tener mayor seguridad en el c3digo
- ❖ Manejo de sesiones
- ❖ Encriptaci3n Sha1 en las contrase3as
- ❖ Manejo de roles y tipos usuario en el gestor de Base de Datos.

3.7.2 Pruebas de caja negra y blanca

3.7.2.1 Pruebas de Caja Negra

En la evaluaci3n fundamenta que el sistema se observa aspectos de funcionalidad, operativa en su proceso que realiza.

Se procede a realizar una evaluaci3n de acuerdo a los m3dulos para cotejar los resultados que entrega el sistema ya sea mediante reportes o interfaces de resultados.

Registro de Usuario

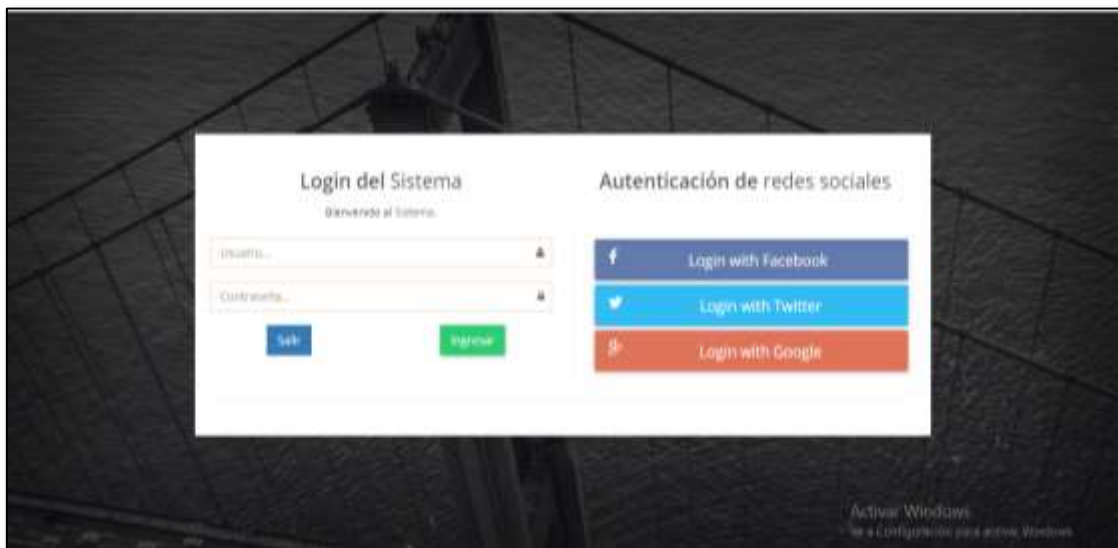
Datos de Ingreso: Se ingresa un usuario y una contraseña asignada por el administrador del sistema en momento otorgado asignación de cargo.

Prueba: Se espera que el sistema realice las respectivas validaciones para el ingreso de funcionarios.

Resultado: El sistema no permite el ingreso al personal no autorizado por el administrador del sistema.

Inicio de sesión al sistema

FIGURA NRO 3.5. 1 Inicio de sesión al sistema



Fuente (elaboración propia).

Datos de Ingreso: Es ingresa al módulo de inicio.

Prueba: Se espera que el sistema despliegue la información del sistema.

Resultado: El sistema despliega enlaces de todo el menú correspondiente a su privilegio o rol asignado.

FIGURA NRO 3.5. 2 Interfaz de inicio del sistema



Fuente (elaboración propia).

3.7.2.2 Prueba de Caja Blanca

Para las pruebas de caja blanca se proporcionará una medición cuantitativa de la complejidad lógica del sistema. La Complejidad ciclomática de un grado de flujo.

$V(G)$ establece el número de caminos independientes:

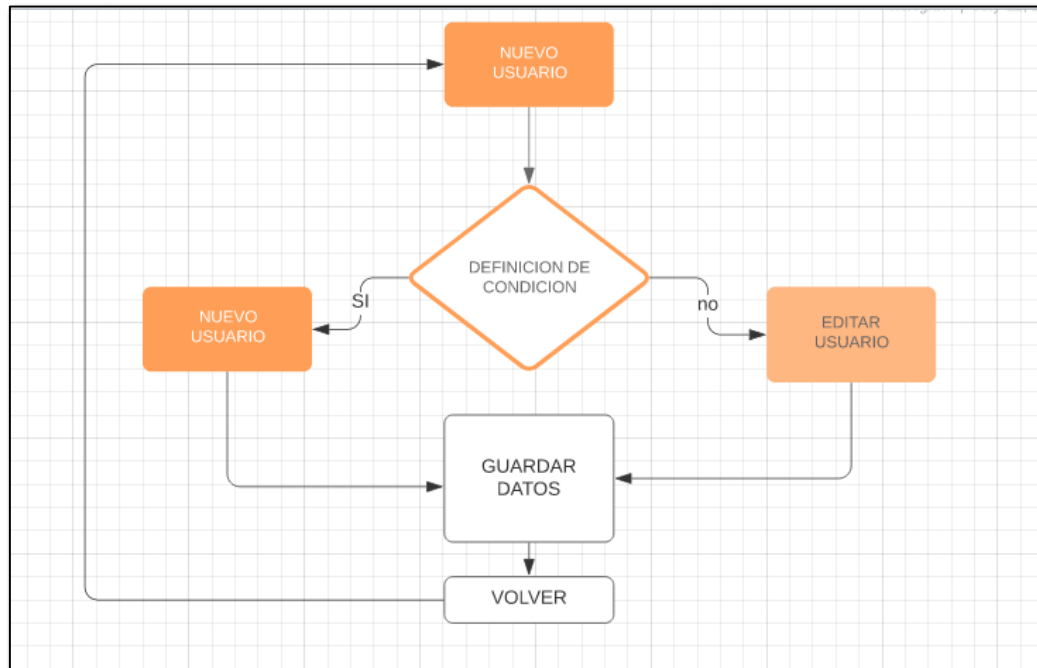
- ❖ El número de regiones del grafico de flujo.
- ❖ $V(G)=A-N+2$, donde A es el número de aristas y N es el número de nodos.
- ❖ $V(G)= P+1$, donde P es el número de nodos predicado.

Módulo de Usuario

Se vio necesario realizar pruebas de caja blanca para el módulo de administración de usuario del sistema debido al flujo grama que presentaba.

Pruebas de Caja blanca para el registro de usuario en el sistema.

FIGURA NRO 3.5. 3 Creación de nuevo usuario flujo drama.



Fuente (elaboración propia).

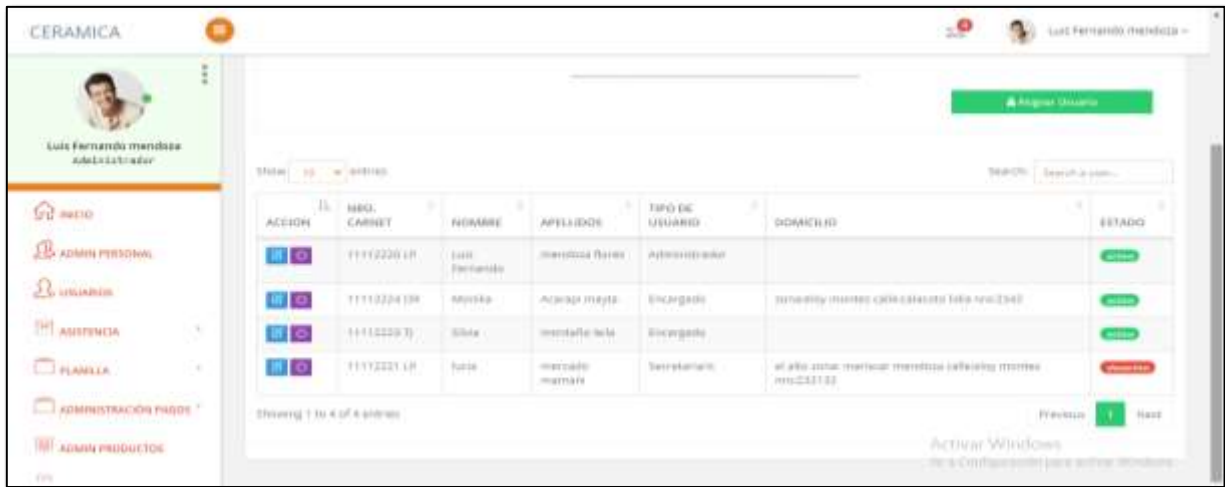
Para ilustrar el flujo grama se puede evidenciar el procedimiento en la figura anterior del módulo de administración de usuario.

FIGURA NRO 3.5. 4 formulario de creación de usuario

The screenshot shows a web application interface for 'CERAMICA'. At the top, there is a navigation bar with the company name and a user profile for 'Luis Fernando mendota'. A sidebar on the left contains menu items: 'Inicio', 'ADMIN PERSONAL', 'USUARIOS', 'ASISTENCIA', 'PUNTAJE', and 'ADMINISTRACION PUNTAJE'. The main content area displays a modal window titled 'CREAR NUEVO USUARIO'. The form contains the following fields: 'CARNET-IDENTIDAD' (11112222 LP), 'NOMBRE' (grover), 'APELLIDOS' (mayta mercado), 'TIPO DE EMPLEADO' (Administrativo), 'TIPO DE USUARIO' (dropdown), 'NAME' (GROVER_11112222), 'PASSWORD' (11112222), and 'REPETIR PASSWORD' (masked). At the bottom of the form are two buttons: 'Cancelar' and 'Guardar Usuario'.

Fuente (elaboración propia).

FIGURA NRO 3.5. 5 Listar usuario



Fuente (elaboración propia).

La figura 3.5.3 Muestra la complejidad ciclomática de un módulo de administración de usuario por lo que reemplazando el número de predicados en la segunda fórmula se obtiene:

$$V(G) = 1 + 1 = 2$$

Este valor determina dos caminos independientes, dos casos de prueba, para el módulo de administración de usuario. Por lo tanto, se realizará las pruebas necesarias para cada uno de los caminos como se muestra en la tabla.

TABLA NRO 3.5 1 Valor de determina dos casos de prueba

CAMINO 1	CAMINO 2
<ul style="list-style-type: none"> ❖ Ingresar al módulo de administración de usuario. ❖ Vuelve a ingresar al botón de nuevo usuario para el nuevo registro. 	<ul style="list-style-type: none"> ❖ Ingresar datos en el formulario de administración de usuario. ❖ Verifica y valida si existen usuario activo incorrecto en el formulario. ❖ Guarda información del usuario en el sistema. ❖ Listar todos los usuarios.

Fuente (elaboración propia).

CAPÍTULO IV

CONCLUSIONES Y RECOMENDACIONES

4 CONCLUSIONES Y RECOMENDACIONES

En este capítulo se dará las conclusiones y recomendaciones referentes al presente Proyecto de Grado.

4.1 CONCLUSIONES

Se concluyó con los objetivos planteados en el presente proyecto, desarrollar un sistema de información para el control de ingreso de materia prima, salida de cerámica elaborada y control del personal, el mismo que coadyuve en la toma de decisiones en la fábrica de cerámica “EMANUEL” de la localidad de Achocalla, que coadyuve con la administración y la optimización de la información, en la Desarrollar un sistema de información para el control de ingreso de materia prima, salida de cerámica elaborada y control del personal, el mismo que coadyuve en la toma de decisiones en la fábrica de cerámica “EMANUEL” de la localidad de Achocalla.

- ❖ El sistema de información para el control de ingreso de materia prima, salida de cerámica elaborada y control del personal, fue desarrollado de manera satisfactoria haciendo el uso de la metodología UWE con el cual se realizó, diseño y desarrollo para su implantación.
- ❖ Se automatizo la información respecto a la venta de ladrillos para obtener resúmenes de los movimientos de las ventas en tiempo real y de forma oportuna.
- ❖ Se sistematizó y diseñó la base de datos en cuanto a las ventas y se puede controlar el stock disponible y mínimo.
- ❖ Se sistematizó la información de los proveedores, de los cuales podemos obtener sus datos personales.
- ❖ Se sistematizó la información sobre la materia prima, la cual se actualiza oportunamente para abastecer en caso de que falte insumos.
- ❖ Se automatizo las pre reservas para los clientes, los cuales pueden acceder desde cualquier lugar de manera segura y confiable, las cuales tienen una vigencia de 24 horas.

- ❖ Se desarrolló una base de datos en donde se administra toda la información sobre las asistencias y salarios del personal.
- ❖ Se automatizó la información de las estadísticas sobre las ventas/compras, por otro lado, se cuenta con planillas del personal la cual nos permite administrarla de manera oportuna.

4.2 RECOMENDACIONES

En base a las políticas de seguridad propuesta y las observaciones realizadas durante las pruebas se elabora las siguientes recomendaciones.

- ❖ Es necesario realizar copias de seguridad de la base de datos.
- ❖ Asignar un administrador de desarrollo de software, que se encargue en el mantenimiento respectivo del sistema, que actualice a los nuevos requerimientos.
- ❖ Se debe tener mucho cuidado con respecto a las claves de usuario que se le proporciona a los administradores del sistema.
- ❖ Las actualizaciones deben ser coordinadas con la fábrica de cerámica “EMANUEL”.

BIBLIOGRAFÍA

- Alsina, G. G. (25 de noviembre de 2018). *Definición de AJAX*. Recuperado el 15 de Julio de 2019, de Definición de AJAX: <https://www.definicionabc.com/tecnologia/ajax.php>
- Alvarez, M. A. (09 de mayo de 2001). *Qué es PHP*. Recuperado el 21 de Junio de 2018, de <https://desarrolloweb.com/articulos/392.php>
- Alvarez, M. A. (09 de mayo de 2001). *Qué es PHP*. Recuperado el 15 de Julio de 2019, de Qué es PHP: <https://desarrolloweb.com/articulos/392.php>
- Angel, A. M. (05 de junio de 2009). *Uso de Ajax , html, jquery*. Recuperado el 08 de mayo de 2019, de <https://desarrolloweb.com/articulos/uso-ajax-jquery.html>
- Arenas Paredes , M. (10 de 8 de 2011). *ARQUITECTURA EN CAPAS*. Obtenido de <http://arquitecturaencapas.blogspot.com/2011/08/arquitectura-3-capas-programacion-por.html>
- Bembibre, V. (10 de Febrero de 2009). *Definición de Base de datos*. Recuperado el 10 de Julio de 2019, de <https://www.definicionabc.com/tecnologia/base-de-datos.php>
- Bernal., G. (16 de Junio de 2019). *Descripción completa del servidor web Apache*. Recuperado el 15 de Julio de 2019, de <https://www.hostinger.es/tutoriales/que-es-apache/>
- Caballero, J. G. (13 de junio de 2018). *¿Qué es JavaScript?* Recuperado el 12 de diciembre de 2018, de <https://devcode.la/blog/que-es-javascript/>
- Chavez, K. (29 de Noviembre de 2012). *Ingeniería Web*. Recuperado el 08 de Diciembre de 2013, de <http://kevinchavez93.blogspot.com/2012/11/ingenieria-web.html>
- checa, s. (30 de mayo de 2018). *Qué es MYSQL*. Recuperado el 01 de mayo de 2018, de <https://www.locurainformaticadigital.com/2018/05/30/sql-vs-mysql-que-son-y-cual-es-la-diferencia/>

- Chen, C. (12 de Julio de 2008). *Significado de Sistema de información*. Recuperado el 12 de Julio de 2019, de Qué es Sistema de información: <https://www.significados.com/sistema-de-informacion/>
- Colectiva, N. (9 de TUTORIALESABRIL de 2015). *CodeIgniter*. Obtenido de <http://blog.nubecolectiva.com/url-amigables-con-codeigniter/>
- Dianelys, V. R. (25 de noviembre de 2014). Recuperado el 12 de Julio de 2019, de <https://www.gestiopolis.com/estimacion-de-costos-de-desarrollo-de-software/>
- Estela, M. (10 de Enero de 2019). *Base de datos*. Recuperado el 10 de Junio de 2019, de Base de datos: <https://concepto.de/base-de-datos/>
- Estela, M. (13 de Febrero de 2019). *software*. Recuperado el 10 de Julio de 2019, de ¿Qué es software?: <https://concepto.de/software/>
- Felipe, S. (21 de marzo de 2015). *Control de Personal*. Recuperado el 03 de julio de 2020, de Control de Personal: <http://www.dointech.com.co/control-personal.html>
- Fontán, M. (25 de octubre de 2012). *CodeIgniter, un framework PHP para el desarrollo rápido de aplicaciones web*. Recuperado el 10 de Julio de 2019, de ¿Qué es MVC?: <http://www.adwe.es/codigo/codeigniter-framework-php-desarrollo-aplicaciones-web>
- Fontela, A. (16 de Julio de 2015). *¿Que es Bootstrap?* Recuperado el 23 de junio de 2018, de <https://raiolanetworks.es/blog/que-es-bootstrap/>
- Godas, P. A. (2013). *Sistema para control de inventario, venta y generacion de datos comerciales de restaurante*. Universidad Austral de Chile: Puerto Montt Chile. Recuperado el 14 de julio de 2018
- González, G. A. (28 de noviembre de 2016). *Definición de JavaScript*. Recuperado el 15 de Julio de 2019, de JavaScript: <https://www.definicionabc.com/tecnologia/javascript.php>

- H, J. C. (2001). *Sistema de control de ventas e inventarios COIMPE LTDA*. UMSS; Cochabamba. Recuperado el 14 de julio de 2018
- Jiménez, J. D. (20 de Enero de 2019). Recuperado el 15 de Julio de 2019, de <https://openwebinars.net/blog/que-es-css3/>
- Maluff, R. A. (2013). *Implementación de sistemas de informacion de gestion comercial para mejorar los procesos de*. Universidad Privada Antenor Orrego: Trujillo Perú. Recuperado el 14 de Julio de 2018
- Mamani, J. M. (2008). *Sistema de informacion via web para el control de almacenes fabrica de fideos santa rosa*. UMSA; La Paz. Recuperado el 14 de Julio de 2018
- Martin, A. (24 de Abril de 2019). *HTML5*. Recuperado el 15 de Julio de 2019, de HTML5: <https://developer.mozilla.org/es/docs/HTML/HTML5>
- MARTINEZ, K. (4 de abril de 2011). *TODO SOBRE PHP*. Recuperado el 12 de Julio de 2019, de VENTAJAS Y DESVENTAJAS: <http://klarimartinezbenjumea.blogspot.com/2011/04/ventajas-y-desventajas.html>
- Maximilians, L. (10 de 08 de 2016). *UWE – UML-based Web Engineering*. Obtenido de <http://uwe.pst.ifi.lmu.de/teachingTutorialPresentationSpanish.html>
- Mercado, J. (27 de Agosto de 2018). Recuperado el 11 de Junio de 2019, de <http://www.pmoinformatica.com/2018/08/tecnicas-estimacion-software.html>
- Merino, J. P. (21 de diciembre de 2013). *Definición de Web*. Recuperado el 12 de Junio de 2019, de <https://definicion.de/web/>
- Pandini, W. (30 de Diciembre de 2005). *seguridad de la información*. Recuperado el 10 de Junio de 2019, de ISO 27002: Buenas prácticas para gestión de la seguridad de la información.
- Pastor, R. (12 de enero de 2018). *MariaDB vs MySQL*. Recuperado el 17 de Jyulio de 2019, de MariaDB vs MySQL: <https://styde.net/mariadb-vs-mysql/>

- Porto, J. P. (23 de diciembre de 2013). *Definición de venta*. Recuperado el 03 de julio de 2020, de Definición de venta: <https://definicion.de/venta/>
- Porto, J. P. (30 de Diciembre de 2018). Recuperado el 15 de julio de 2019, de <https://definicion.de/jquery/>
- PRESSMAN, R. (2002). *Ingeniería de Software. Un Enfoque Práctico. Quinta Edición* (QUITA EDICIÓN ed.). MEXICO: MC GRAW HILL.
- Pressman, R. (2005). *Ingeniería del Software*.
- Quiroga, A. (23 de 03 de 2015). *Metodología UWE UML (UML-Based Web Engineering)*. Obtenido de <http://proyectogradoingenieriasistemas.blogspot.com/2015/03/metodologia-uwe-uml-uml-based-web.html>
- Raffino, M. E. (11 de enero de 2019). *Sistema*. Recuperado el 10 de Julio de 2019, de ¿Qué es un sistema?: <https://concepto.de/sistema/>
- Ramos, P. (12 de 01 de 2018). *MariaDB*. Recuperado el 10 de Junio de 2019, de MariaDB: <https://styde.net/mariadb-vs-mysql/>
- Resan, R. (05 de Junio de 2016). *Ciclo de vida clásico o en cascada*. Recuperado el 12 de Julio de 2019, de modelo de ciclo de vida en cascada: http://agrega.juntadeandalucia.es/repositorio/20022017/6b/es-an_2017022012_9122843/51_ciclo_de_vida_clsico_o_en_cascada.html
- ruiz, a. (20 de septiembre de 2014). *Bootstrap*. Recuperado el 15 de Julio de 2019, de ¿Qué es Bootstrap?: <https://www.mastermarketingdigital.com/everriculum/2014/09/20/que-es-bootstrap/>
- Soto, A. M. (2008). *Diagramas de Casos de Uso*. Obtenido de <https://es.scribd.com/doc/82603040/Casos-de-Uso-explicacion-y-ejemplos>
- thewolf, d. (25 de junio de 2015). *Metodología UWE*. Obtenido de <https://metodologiauwe.wordpress.com/2015/06/25/hello-world/>

Villamizar, H. (13 de febrero de 2017). *Norma ISO/IEC 9126 Y METRICAS DE CALIDAD*. Recuperado el 23 de 08 de 2018, de <https://prezi.com/yu357lmguc9n/norma-isoiec-9126-y-metricas-de-calidad/>

Wikipedia. (5 de noviembre de 2015). *Seguridad informática*. Recuperado el 19 de Junio de 2019, de https://es.wikipedia.org/wiki/Seguridad_inform%C3%A1tica

ANEXO

ANEXO A

El Alto, 29 de junio de 2020

AVAL DE CONFORMIDAD

Señor
Ing. David Carlos Mamani Quispe
DIRECTOR DE LA CARRERA DE INGENIERÍA DE SISTEMAS
HONORABLE CONSEJO DE CARRERA

Presente. -


REF. AVAL DE CONFORMIDAD PROYECTO DE GRADO

Distinguido Ingeniero:

En calidad de tutor especialista luego de haber realizado el proyecto de grado titulado "**SISTEMA WEB PARA LA VENTA DE LADRILLOS Y CONTROL DE PERSONAL**" que propone el postulante Univ. Armin Rodrigo Forra Layme con C.I. 6194913 L.P., doy mi conformidad para que se programe fecha y hora de defensa pública, evaluación correspondiente a la materia Taller de licenciatura II, de acuerdo a reglamento vigente de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular, reciba saludos cordiales.

Atentamente,


.....
Ing. Fernando Chambi Guachalla
C.I. 9136227 LP
TUTOR ESPECIALISTA

AVAL DE CONFORMIDAD

El Alto, 13 de julio de 2020

Señor
Ing. David Carlos Mamani Quispe
DIRECTOR DE LA CARRERA DE INGENIERÍA DE SISTEMAS
HONORABLE CONSEJO DE CARRERA

Presente.-

REF. AVAL DE CONFORMIDAD PROYECTO DE GRADO

Distinguido Ingeniero:

Mediante la presente tengo a bien comunicarle en calidad de tutor revisor mi conformidad del proyecto de grado titulado "**SISTEMA WEB PARA LA VENTA DE LADRILLOS Y CONTROL DE PERSONAL**" que propone el postulante Univ. Armin Rodrigo Forra Layme con C.I. 6194913 L.P. y después de haber subsanado las observaciones realizadas, se programe fecha y hora de defensa pública, evaluación correspondiente a la materia de Taller de licenciatura II, de acuerdo a reglamento vigente de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular, reciba saludos cordiales.

Atentamente,



.....
Lic. Gabriel Reynaldo Sirpa Huayhua
C.I. 6837773 L.P.
TUTOR REVISOR

El Alto, 10 de julio de 2020

AVAL DE CONFORMIDAD

Señor:

**Ing. David Carlos Mamani Quispe
DIRECTOR DE LA CARRERA DE INGENIERÍA DE SISTEMAS
HONORABLE CONSEJO DE CARRERA**

Presente. -

REF. AVAL DE CONFORMIDAD PROYECTO DE GRADO

Distinguido Ingeniero:

Mediante la presente tengo a bien comunicarle mi conformidad del proyecto de grado titulado "**SISTEMA WEB PARA LA VENTA DE LADRILLOS Y CONTROL DE PERSONAL**" que propone el postulante Univ. Armin Rodrigo Forra Layme con Cédula de Identidad 6194913 L.P. y Registro Universitario 6001596, para su defensa pública, evaluación correspondiente a la materia Taller de Licenciatura II, de acuerdo a reglamento vigente de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular, reciba saludos cordiales.

Atentamente,


.....
Ing. Mañsol Arguedas Balladares
TUTOR METODOLÓGICO



CERTIFICACIÓN

A SOLICITUD ESCRITA DEL INTERESADO, EL ADMINISTRADOR GENERAL Y PROPIETARIO DE LA FÁBRICA DE CERÁMICA "EMANUEL" ÁNGEL VÍCTOR PAREDES ALANOCA.

CERTIFICA QUE:

El señor, **ARMIN RODRIGO FORRA LAYME**, con cédula de identidad número **6194913** expedido en la ciudad de La Paz, desarrolló e implementó satisfactoriamente el "**SISTEMA WEB PARA LA VENTA DE LADRILLOS Y CONTROL DE PERSONAL**", para la fábrica de CERÁMICA "EMANUEL", dando conformidad del mismo, así de esta manera se dió el cumplimiento al proyecto presentado por el mencionado.

Es cuanto certifico en honor a la verdad para fines consiguientes a favor del interesado.

La Paz, 9 de julio de 2020



ÁNGEL VÍCTOR PAREDES ALANOCA
GERENTE GENERAL CERÁMICA EMANUEL