

UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA



PROYECTO DE GRADO
SISTEMA WEB DE GESTIÓN DE CLIENTES, REGISTRO
DE PEDIDOS Y ENVÍOS
CASO: FLORERÍA “DALIA”

Proyecto de Grado para obtener el Título de Licenciatura en Informática
Mención Ingeniería de Sistemas Informáticos

POR: ODON ORLANDO RIVERA CRUZ

TUTOR METODOLÓGICO: M.SC. ALDO RAMIRO VALDEZ
ALVARADO

ASESOR: LIC. VICTOR PABLO POZO DIAZ

LA PAZ – BOLIVIA

2021



**UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA**



LA CARRERA DE INFORMÁTICA DE LA FACULTAD DE CIENCIAS PURAS Y NATURALES PERTENECIENTE A LA UNIVERSIDAD MAYOR DE SAN ANDRÉS AUTORIZA EL USO DE LA INFORMACIÓN CONTENIDA EN ESTE DOCUMENTO SI LOS PROPÓSITOS SON ESTRICTAMENTE ACADÉMICOS.

LICENCIA DE USO

El usuario está autorizado a:

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

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

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

DEDICATORIA

Dedico este proyecto a:

A Dios, por permitirme llegar a uno más, de los tantos momentos de dicha y felicidad en mi vida.

A MI PAPA, ERNESTO y A MI MAMA, ALICIA por darme todo su amor, apoyo, motivación, paciencia, confianza y además depositar su fe, en mí.

A mi Compañera de vida REYNA, por alentarme y ayudarme en este último paso, además de acompañarme y quererme tal como soy.

A mis Hermanos Rivera Cruz; MERY, LOURDES, GONZALO Y CARLA por brindarme siempre su ayuda y cariño en estos años.

AGRADECIMIENTOS

Agradecer a Dios por haberme guiado y protegido durante estos años.

Agradecer a todos los integrantes de mi Familia por el apoyo incondicional que me dieron en estos años.

Agradecer a la Universidad Mayor de San Andrés y a la carrera de Informática por brindarme sus predios y así desarrollarme profesionalmente.

Agradecer a mi tutor Aldo Ramiro Valdez Alvarado, quien me guio en este último paso, con su experiencia y sabiduría es uno de los mejores docentes de la UMSA.

Agradecer a mi asesor Víctor Pablo Pozo Diaz, quien me enseñó que la perseverancia tiene resultados buenos, gracias a sus consejos y técnicas pude lograr llegar a esta aquí, es uno de los mejores docentes de la UMSA.

Agradecer a la Florería “Dalia” por abrirme las puertas a este emprendimiento y darme la oportunidad para realizar el proyecto.

Agradecer a todos mis docentes de la carrera de informática por las enseñanzas que me dieron en cada materia cursada.

Agradecer a todos mis amigos, compañeros de la carrera y amigos de mi barrio, por aminorarme y colaborar en diferentes oportunidades.

RESUMEN

El desarrollo de cualquier Sistema Web otorgó un lugar a un desprendimiento de muchas ideas al observar la tecnología seguir el cometer a las empresas, no es una labor simple a menos que sean trabajador o conocedor único de la entidad para la obtención y proceso de datos vemos en Bolivia que las múltiples organizaciones algunas mantienen un desarrollo de uso de tecnologías sin embargo cabe resaltar que van paso a paso por necesidad utilitaria optimizando tiempo con el uso necesario de las tecnologías.

El presente proyecto tiene como finalidad apoyar a la Florería “Dalia” por medio de la utilización del sistema que dejara mantener el control de Gestión de Clientes, Registro de Pedidos y Envíos, una vez concluido el trabajo en la Florería y poder ser expuesto funcionamiento.

La metodología SCRUM ofrece la posibilidad de adaptarse exitosamente a la Florería “Dalia”. De esta manera, se concibe la idea de una producción organizada de las tareas, además se desarrolla un plan y trabajo conjunto con el encargado de la Florería.

La participación y administración de SCRUM es sencilla y de fácil manejo para todas las etapas. De igual manera, se cuenta con un registro de labores realizadas y se le da seguimiento para lograrlo de forma eficiente.

El desarrollo del proyecto está enfocado bajo la metodología Scrum y la metodología de diseño Web UWE.

El programa obtenido es un producto de calidad según la metodología de evaluación de calidad de sistemas Web-Site Qem.

Al final se puede observar en las conclusiones que las metas planteadas han sido alcanzadas y que el producto desarrollado cumple con los requerimientos, funcionales de la Florería.

Palabras clave: Aplicación Web, Scrum, UWE, Gestión de Clientes.

ABSTRACT

The development of any Web System gave a place to a detachment of many ideas when observing the technology to follow the committing of the companies, it is not a simple task unless they are a worker or sole expert of the entity to obtain and process data. In Bolivia, the multiple organizations, some maintain a development in the use of technologies, however, it should be noted that they go step by step due to utilitarian necessity, optimizing time with the necessary use of technologies.

The purpose of this project is to support the "Dalia" Florist by using the system that will allow to maintain the control of Customer Management, Registration of Orders and Shipments, once the work in the Florist is finished and the operation can be exposed.

The SCRUM methodology offers the possibility of successfully adapting to the "Dalia" Florist. In this way, the idea of an organized production of the tasks is conceived, in addition a plan and joint work is developed with the person in charge of the Flower Shop.

The participation and administration of SCRUM is simple and easy to use for all stages. In the same way, there is a record of the work carried out and it is followed up to achieve it efficiently.

The development of the project is focused under the Scrum methodology and the UWE Web design methodology.

The program obtained is a quality product according to the Web-Site Qem systems quality evaluation methodology.

In the end, it can be seen in the conclusions that the goals set have been achieved and that the product developed meets the functional requirements of the flower shop.

Keywords: Web Application, Scrum, UWE, Customer Management.

INDICE DE CONTENIDO

CAPÍTULO I	1
MARCO REFERENCIAL	1
1.1. INTRODUCCIÓN	1
1.2. ANTECEDENTES	2
1.2.1. ANTECEDENTES INSTITUCIONALES	2
1.2.1.1. MISIÓN	2
1.2.2. ANTECEDENTES DE PROYECTOS SIMILARES	3
1.3. PLANTEAMIENTO DEL PROBLEMA	5
1.3.1. PROBLEMA CENTRAL	6
1.3.2. PROBLEMAS SECUNDARIOS	6
1.4. DEFINICIÓN DE OBJETIVOS	7
1.4.1. OBJETIVO GENERAL	7
1.4.2. OBJETIVOS ESPECÍFICOS	7
1.5. JUSTIFICACIÓN	7
1.5.1. JUSTIFICACIÓN ECONÓMICA	7
1.5.2. JUSTIFICACIÓN SOCIAL	8
1.5.3. JUSTIFICACIÓN TECNOLÓGICA	8
1.6. ALCANCES Y LÍMITES	8
1.6.1. ALCANCES	8
1.6.2. LÍMITES	9
1.7. APORTES	9
1.7.1. PRÁCTICO	9
1.7.2. TEÓRICO	9
1.8. METODOLOGÍA	10
CAPITULO II	12
MARCO TEÓRICO	12
2.1. MARCO TEÓRICO	12
2.2. INGENIERÍA DE SOFTWARE	12
2.3. METODOLOGÍAS DE DESARROLLO ÁGILES	13

2.4. METODOLOGÍA SCRUM	14
2.5. ARQUITECTURA DEL SISTEMA	26
CAPÍTULO III	27
MARCO APLICATIVO	27
3.1. INTRODUCCIÓN	27
3.3. PRE – GAME	28
3.3.1. ASIGNACION DE ROLES	28
3.3.2. RECOLECCIÓN DE REQUISITOS	28
3.3.3. PRODUCT BACKLOG	29
3.3.4. REQUERIMIENTO DE HARDWARE Y SOFTWARE	32
3.4. GAME	33
3.4.1. SPRINTS	33
3.4.1.1. PRIMER SPRINT DE ADMINISTRACIÓN DE USUARIOS Y GESTIÓN DE CLIENTES	34
3.4.1.2. PLANIFICACIÓN DEL SPRINT 1: ADMINISTRACIÓN DE USUARIOS Y GESTIÓN DE CLIENTES	34
3.4.1.3. ESPECIFICACIÓN DE CASO DE USO DEL SPRINT 1	36
3.4.1.4. DISEÑO NAVEGACIONAL ADMINISTRACIÓN DE USUARIOS Y GESTIÓN DE USUARIOS	39
3.4.1.5. DISEÑO DE PRESENTACIÓN	39
3.4.1.6. PANTALLAS DEL SPRINT 1	41
3.4.1.7. PRUEBA UNITARIA DEL SPRINT 1	43
3.4.2. DESARROLLO DEL SEGUNDO SPRINT, REGISTRO DE PEDIDOS Y ENVÍOS	43
3.4.2.1. PLANIFICACIÓN DEL SPRINT 2 REGISTRO DE PEDIDOS Y ENVÍOS	43
3.4.2.2. ESPECIFICACIÓN DE CASO DE USO DEL SPRINT 2	45
3.4.2.3. DISEÑO NAVEGACIONAL	47
3.4.2.4. DISEÑO DE PRESENTACIÓN	47
3.4.2.5. PANTALLA DEL SPRINT 2	49
3.4.2.6. PRUEBA UNITARIA DEL SPRINT 2	51

3.4.3. DESARROLLO DEL TERCER SPRINT, REPORTES	51
3.4.3.1. PLANIFICACIÓN DEL SPRINT 3 REPORTES	51
3.4.3.2. ESPECIFICACIÓN DE CASO DE USO DEL SPRINT 3	53
3.4.3.3. DISEÑO NAVEGACIONAL	54
3.4.3.4. DISEÑO DE PRESENTACIÓN	55
3.4.3.5. PANTALLA DEL SPRINT 3	56
3.5. DIAGRAMA ENTIDAD RELACIÓN	58
3.6. MODELO FÍSICO	59
CAPITULO IV.....	61
CALIDAD Y SEGURIDAD.....	61
4.1. INTRODUCCIÓN.....	61
4.2. CALIDAD DE SOFTWARE	61
4.3. NORMA ISO 9126.....	62
4.4. METODOLOGÍA WEB QEM.....	62
4.4. FASES DE WEB QEM	62
4.4.1.1. DEFINICIÓN DE LAS METAS DE EVALUACIÓN Y SELECCIÓN DEL PERFIL DE USUARIO	62
4.4.1.2. DEFINICIÓN DE LOS REQUERIMIENTOS DE CALIDAD Y/O COSTO	62
4.4.1.3. DEFINICIÓN DE CRITERIOS BÁSICOS Y PROCEDIMIENTOS DE MEDICIÓN.....	63
4.4.1.4. DEFINICIÓN DE ESTRUCTURAS DE AGREGACIÓN E IMPLEMENTACIÓN DE LA EVALUACIÓN GLOBAL	63
4.4.1.5. ANÁLISIS DE RESULTADOS Y RECOMENDACIONES.....	63
4.4.2. DESARROLLO DE CARACTERÍSTICAS DE WEB QEM	64
4.4.2.1. FUNCIONALIDAD.....	64
4.4.2.3. USABILIDAD	67
4.4.2.4. MANTENIBILIDAD.....	68
4.4.2.5. PORTABILIDAD	69
4.5. SEGURIDAD	70
4.5.1. ALGORITMO DE ENCRIPCIÓN SHA-2.....	70

4.5.2. VARIANTES DEL SHA-2.....	71
4.5.3. CAPA DE CONEXIÓN SEGURA SSL.....	71
4.5.3.1 IMPLEMENTACIÓN DE CERTIFICADO SSL CON LET'S ENCRYPT Y CERBOT	72
CAPITULO V	74
COSTO BENEFICIO.....	74
5.1. INTRODUCCIÓN	74
5.2. COCOMO II.....	74
5.3. COSTO DEL SISTEMA	76
5.5.1. VALOR ACTUAL NETO (VAN)	84
5.5.2. TASA INTERNA DE RETORNO (TIR)	85
5.6. COSTO BENEFICIO.....	86
CAPITULO VI.....	88
CONCLUSIONES Y RECOMENDACIONES	88
6.1. CONCLUSIONES	88
6.2. RECOMENDACIONES	88
BIBLIOGRAFIA	90
ANEXOS	92
ANEXO A – ARBOL DE PROBLEMAS.....	92
ANEXO B – ARBOL DE OBJETIVOS	93
ANEXO C – MARCO LÓGICO.....	94
ANEXO D – CRONOGRAMA DE ACTIVIDADES.....	95

ÍNDICE DE FIGURAS

Figura 2.1: Estructura de paquetes del metamodelo de UWE.....	23
Figura 2.2: Proceso UWE para transformaciones CIM2PIM y PIM2PIM	23
Figura 2.3: Ejemplos de Modelos de Contenido, Navegación y Presentación	24
Figura 2.4: Metamodelo de Integración de Procesos	25
Figura 2.5: Metamodelo para el modelo de proceso de datos.....	26
Figura 2.6: Diagrama de arquitectura del sistema.....	26
Figura 3.1: Proceso Scrum	27
Figura 3.2: Caso de uso para Administración de roles de Usuario	36
Figura 3.3: Caso de uso para Gestión de Clientes.....	37
Figura 3.4: Modelo Navegacional, Administración de Usuarios y Gestión de Clientes	39
Figura 3.5: Modelo de Presentación del Administrador de usuarios	40
Figura 3.6: Modelo de Presentación para listar y desplegar la información	40
Figura 3.7: Código php, para el ingreso de usuario	41
Figura 3.8: Pantalla de Inicio	41
Figura 3.9: Pantalla de Administración de Usuarios.....	42
Figura 3.10: Pantalla de Administración de Clientes.....	42
Figura 3.11: Caso de uso para el Registro de Pedidos y Envíos	45
Figura 3.12: Diseño de Navegacional Registro de Pedidos y Envíos	47
Figura 3.13: Diseño de Presentación Registro de Pedidos y Envíos.....	48
Figura 3.14: Código php, para registrar una nueva categoría	48
Figura 3.15: Pantalla Crear Venta	49
Figura 3.16: Pantalla Administración de Productos.....	49
Figura 3.17: Pantalla Categorías	50
Figura 3.18: Página Registro de Pedidos y Envíos	50
Figura 3.19: Caso de uso Reportes.....	53
Figura 3.20: Diseño Navegacional Reportes.....	55
Figura 3.21: Diseño de Presentación Reportes.....	56
Figura 3.22: Código php, para mostrar las ventas.....	56

Figura 3.23: Pantalla Reportes, gráfico de ventas	57
Figura 3.24: Pantalla Gráficos, productos mas vendidos y clientes potenciales.....	57
Figura 3.25: Diagrama Entidad Relación	59
Figura 3.26: Modelo Físico	60
Figura 4.1: Fases de WebQEM	64
Figura 4.2: Certificación <i>Let's Encrypt</i> activa.....	73
Figura 5.1: Líneas de código software COCOMO.....	80
Figura 5.2: Estimación de costo software COCOMO.....	81
Figura 5.3: Punto fusión software COCOMO.....	82
Figura 5.4: Estimación de costo software COCOMO.....	82
Figura 5.5: Factores de escala software COCOMO.....	83

ÍNDICE DE TABLAS

Tabla 1.1: Problema Causa-Efecto	6
Tabla 3.1: Asignación de Roles.....	28
Tabla 3.2: Lista de Requisitos.....	29
Tabla 3.3: Detalles de los Módulos	32
Tabla 3.4: Requerimiento de Hardware y Software.....	33
Tabla 3.5: Primer Sprint	35
Tabla 3.6: Administración de Usuarios y Gestión de Clientes	38
Tabla 3.7: Prueba unitario del Sprint 1.....	43
Tabla 3.8: Segundo Sprint.....	44
Tabla 3.9: Registro de Pedidos y Envíos	46
Tabla 3.10: Prueba unitario del Sprint 2.....	51
Tabla 3.11: Tercer Sprint	52
Tabla 3.12: Reportes	54
Tabla 3.13: Prueba Unitaria del Sprint 3.....	58
Tabla 4.1: Factor de ajuste de complejidad.....	65
Tabla 4.2: Preguntas para obtener el grado de usabilidad	68
Tabla 4.3: Evaluación de mantenibilidad.....	69
Tabla 4.4: Calidad global.....	70
Tabla 5.1: Tabla de modos de desarrollo.....	75
Tabla 5.2: Puntos fusión COCOMO II.....	77
Tabla 5.3: Tabla modo de conductores de costo.....	78
Tabla 5.4: Costo total desarrollo de software.....	83
Tabla 5.5: Flujo de caja por años.....	84

CAPÍTULO I

MARCO REFERENCIAL

1.1. INTRODUCCIÓN

La implementación de Tecnologías de la Información y Comunicación (TIC) desde finales del siglo pasado produjo, a escala mundial, importantes cambios en todo ámbito, sea privado o público, desde la forma de organización personal e institucional hasta el proceso de comunicación, convivencia, aprendizaje y, por ende, en el ámbito laboral.

Cada vez es más común que en cualquier tipo de entidad como, por ejemplo, una empresa, asociación, negocio, institución u otros, se necesite una solución informática ajustada a sus necesidades. Y cada vez es más habitual que se nombre una “Aplicación Web” como la respuesta a estas necesidades.

Bolivia ha comenzado a incursionar en esta industria del desarrollo de software hace aproximadamente diez años y hoy es uno de los países sudamericanos que oferta este producto a costos más reducidos que otros. Los primeros años fueron de prueba para muchas empresas bolivianas, y sobrevivieron aquellas que sobre todo se enfocaron para atender a grandes empresas del mercado local (Cueto, 2009).

Las regiones que destacan más emprendimientos en este rubro corresponden al eje troncal: La Paz, Cochabamba y Santa Cruz, acompañados en los últimos años de Potosí y Sucre como resultado de la demanda de producción de software (Suaznabar, 2017).

Con una trayectoria de más de 20 años de servicio a la población en la ciudad de La Paz, la Florería “Dalia” fue creada en el año 1998 como un negocio familiar, se encuentra ubicada en la Avenida 16 de Julio, Calle Colon No. 161, edificio Barrosqira esta Florería de gran prestigio tiene como meta, expandirse por el interior del País, para que muchas personas reconozcan sus trabajos florales que son obras maestras diseñadas por el excelente personal de trabajo.

1.2. ANTECEDENTES

1.2.1. ANTECEDENTES INSTITUCIONALES

La Florería “Dalia” dedicada a realizar servicios de envíos de arreglos florales a domicilio para todo tipo de acontecimiento como lo son; cumpleaños, casamientos, aniversarios, nacimientos, bautizos y condolencias, atiende a sus clientes los 7 días de la semana y preferentemente en días festivos y feriados, realizando entrega de rosas, chocolates, peluches y desayunos a domicilio. Esta Florería sufre la deficiencia de no contar con un sistema informático el cual ayude a optimizar, ordenar y priorizar las tareas de trabajos recibidos en temporadas de alta demanda. También de realizar una buena administración y almacenamiento de documentos de clientes y empleados.

Los empleados de la Florería registran su asistencia en una planilla, en el cual se anota días de trabajo, hora de llegada y de salida, también tienen un cuaderno personal de anotaciones, en la recepción se tiene un block de notas, el cual se utiliza para anotar los pedidos de los clientes que llegan a la Florería, estas notas son transcritas en un Excel con el fin de generar un informe del día. A la Florería van clientes aseverando ser fieles y exigen algún tipo de rebaja o promoción, pero no todos los empleados conocen a estos clientes fieles.

1.2.1.1. MISIÓN

Brindar el mejor servicio de decoración floral, especializando el servicio en base al diseño y armado de arreglos florales para cualquier evento social y también realizar bellos diseños con flores para transmitir todo tipo de sentimientos humanos.

1.2.1.2. VISIÓN

Lograr alcanzar servicios de calidad de tal manera de que la florería se distinga de las demás en el diseño de arreglos florales, decoración de eventos como; cumpleaños, bodas, aniversarios, nacimientos, bautizos y condolencias, siendo nuestro principal objetivo la satisfacción de nuestros clientes en especial.

1.2.2. ANTECEDENTES DE PROYECTOS SIMILARES

En la biblioteca de la carrera de Informática en la Universidad Mayor de San Andrés (UMSA), se tomó como referencia proyectos similares, a continuación, se describe mencionados proyectos:

PROYECTO: “APLICACIÓN ANDROID PARA LA GESTIÓN Y CONTROL DE LA ENTREGA Y RECOJO DE CORRESPONDENCIA CASO: TNT EXPRESS”

La aplicación móvil permite la gestión y control del servicio de recojo, entrega de envíos y correspondencia apoyada en un modelo de ordenamiento y priorización, además que permite actualizar y brindar información oportuna con pruebas de entregas digitales y en línea a los clientes. El presente proyecto tiene como objetivo desarrollar una herramienta en Android que permite ayudar a actualizar la información de las entregas de correspondencia y optimizar los tiempos de respuesta en los estatus de cada envío y tener la información de forma oportuna. (Hinojosa, 2016)

PROYECTO: “SISTEMA WEB DE COMERCIO ELECTRÓNICO CASO: EMPRESA DE DISTRIBUCIÓN DE EQUIPOS DE COMPUTACIÓN HSB”

El proyecto describe un sistema con dos interfaces, uno es el administrador, el cual administra todos los elementos de la tienda, permite la edición, creación, y eliminación de cada uno de los elementos de la tienda, así mismo permite vender dentro del administrador como un medio dentro del almacén. La interfaz de la tienda, muestra todos los productos registrados en el administrador, y cuenta con carrito, deseos de clientes, libretas de dirección, proceso de compra y el proceso de pago en línea, así mismo permite el contacto constante con el cliente mediante correo electrónico, además permite la colaboración constante de clientes que publican productos solicitados. El proyecto está dirigido a la venta en línea, por lo tanto, se realiza el pago mediante Tigo Money o PayPal, y de forma adicional está el depósito bancario. (Gutiérrez, 2016)

PROYECTO: “SISTEMA WEB PARA EL CONTROL Y SEGUIMIENTO DE VENTAS DE PRODUCTOS ARTESANALES CASO: BOLIVIA TECH HUB”

El proyecto se centra en la venta de productos artesanales, desde el registro de productos, hasta reportes de ventas, administración de ventas, creación de carrito de compras, utilización de la pasarela de pago de Paypal y autenticación para controlar acceso al sistema. En la parte introductoria se muestra los antecedentes y actividades que realiza la institución, también se muestra el análisis de los problemas y los objetivos propuestos. Para el desarrollo del proyecto se aplicó la metodología de desarrollo ágil Scrum apoyándose en la metodología Kanban y la metodología de desarrollo UWE para el modelado del diseño. El Sistema es un producto de calidad de acuerdo a la métrica de calidad Web-Site QEM. Para el funcionamiento pleno del proyecto también se toma como tarea final implementar la seguridad, además de poner en funcionamiento en el servidor designado. (Limachi, 2018)

PROYECTO: “SISTEMA DE CONTROL DE INVENTARIOS CON ALERTAS TEMPRANAS PARA PEDIDOS EN LÍNEA CASO: ASOCIACIÓN MIXTA DE ORFEBRES Y RELOJEROS A.M.O.R.”

El proyecto, muestra los productos de muchos artesanos para que estos sean apreciados y se pueda alcanzar otros mercados, para ello presentaron el proyecto como un todo además de ser integral, en donde se controlaran los productos, se llevaran un control de todos los productos, previendo la cantidad que se requerirá de parte del comprador, este control al no tener llevaba a demorar los envíos y pedidos de los clientes, quienes optaban por hacerlo a otros artesanos, perdiendo la Asociación contratos por la falta de la implementación de un sistema Web. (Duran, 2016)

PROYECTO: “SISTEMA WEB DE CONTROL DE VENTAS E INVENTARIOS CASO: MICHELLINE”

El presente proyecto fue desarrollado para el control de ventas e inventarios, así saber el ingreso y egreso de productos, de acuerdo a ello poder distribuir de manera eficiente a los puntos de ventas de la empresa Michelline logrando mejorar sus ingresos. El control de ventas e inventarios es el término utilizado para describir el conjunto de procesos de entrada y salida cuyo objeto final es el control de productos de la empresa. El sistema

brinda una serie de funcionalidades que fueron plasmadas en las historias de usuario, estas funcionalidades buscan satisfacer los requerimientos de la empresa. (Aduviri, 2016)

1.3. PLANTEAMIENTO DEL PROBLEMA

Para que una empresa crezca y llegue a ser exitosa, es necesario que esta, tenga que actualizarse y aprovechar las nuevas tecnologías que facilitan y mejoran de gran manera el trabajo en la misma, además de lograr un mayor alcance en lo que se refiere a ventas.

En el actual mercado empresarial de Bolivia, es imprescindible usar sistemas informáticos para realizar tareas de básica y compleja naturaleza. La Florería “Dalia”, sufre la deficiencia de no contar con un sistema informático el cual ayude a optimizar tareas como:

Registro y Administración de los datos de los usuarios: la Florería no cuenta con un debido y sofisticado sistema que registre debidamente todos los datos de los usuarios, el problema ocurre cuando se pierden datos del cliente como también de los trabajadores, con ser; su teléfono, su dirección domiciliaria, lo conduce a la pérdida de clientela para la Florería.

Demora de tiempo: Actualmente estos pedidos o envíos se registran mediante notas en papel para después pasarlos a un Excel, ocasionando pérdida de datos, inseguridad y molestia en los clientes y trabajadores de la Florería, además cuando se realiza un pedido o envió, se observó que estos llegan a con retraso y desordenadamente, incumpliendo los días de entrega.

Informes incompletos: problema por el cual se genera mucha desconfianza a momento de saldar cuentas con los clientes y los propietarios, ya que en ese momento no existe un registro seguro y confiable acerca de todo el detalle que posteriormente sirvan para ayudar a la contabilidad y centralizar toda la información acerca de servicios que sea hayan brindado, además del movimiento del dinero y pedidos de clientes que tuvieron gran

demanda en determinados periodos del año.

1.3.1. PROBLEMA CENTRAL

¿De qué manera realizar el registro y administración de datos del usuario, además de identificar a clientes potenciales para su contacto permanente, como también el de registrar pedidos y envíos de los diferentes servicios que ofrece la Florería “Dalia”?

1.3.2. PROBLEMAS SEGUNDARIOS

Los problemas encontrados, en el proceso de brindar un adecuado servicio al usuario en la Florería “Dalia”. Son los siguientes:

	PROBLEMA	CAUSA	EFEECTO
1) Administración de usuarios	No se tiene un registro de los usuarios que intervienen en el proceso del servicio.	Para realizar determinadas actividades que existen en los usuarios, los cuales ocupan roles distintos.	Desorganización y molestia en los empleados de la Florería.
2) Gestión de clientes	No existe un registro centralizado con datos importantes de los clientes.	Mal manejo de los datos referidos a clientes y la manera de contactarse con el mismo.	Genera pérdida de posibles clientes potenciales y por tanto pérdida de dinero.
3) Registro de pedidos y control de envíos	La información de los pedidos realizados no se guarda correctamente, además no se cuenta con un control de envíos.	Atrasos en los pedidos y envíos. Además de la falta de un control cuidadoso con los envíos.	Reclamos por parte del gerente, así también de los clientes, por la mala organización.
4) Reporte	No se tiene una solución que permita generar reportes de todos los servicios.	Floja administración de la información mensual en la Florería.	Dudas e incertidumbre de los resultados mensuales de la Florería.

Tabla 1.1.: Problema Causa-Efecto
Fuente: (Propia, 2020)

1.4. DEFINICIÓN DE OBJETIVOS

1.4.1. OBJETIVO GENERAL

Desarrollar un sistema web de gestión de clientes, registro de pedidos y envíos para la Florería “Dalia”.

1.4.2. OBJETIVOS ESPECÍFICOS

- Registrar la información de todos los usuarios, como ser sus datos personales y correos electrónicos.
- Mostrar detalladamente reportes mensuales detallando la información de todos los servicios que han sido realizados en el día y también de todo el mes.
- Registrar los pedidos y envíos solicitados por los clientes de manera ordenada y correcta.

1.5. JUSTIFICACIÓN

1.5.1. JUSTIFICACIÓN ECONÓMICA

El sistema web permite que la Florería “Dalia” reduzca sus pérdidas económicas que resultan de la falta de automatización en los procesos de control y administración. Actualmente la Florería cuenta con 3 personas responsables del control de trabajos y su correspondiente seguimiento de forma digital y repetitiva, el sistema es suficiente para que este trio, interactúen con el mismo ya que anteriormente existían 7 personas responsables, dando así una significativa reducción en los pagos de sueldos al personal.

El usuario del sistema tiene la información, documentación y la administración de los trabajos las 24 horas al día desde cualquier ubicación al instante, evitando los gastos de transporte, cuando se requiere información que se encuentran en otras sucursales de la Florería. De la misma manera reduce gastos en impresiones pues anteriormente todos los documentos generados por el servicio eran impresos para el archivo de la Florería, pero el sistema web almacena toda la documentación en documentos digitales.

1.5.2. JUSTIFICACIÓN SOCIAL

El sistema web tiene un gran impacto entre la Florería y los clientes pues es de gran aporte para que el cliente reciba un servicio óptimo y eficiente, como es común mientras más rápido sea atendido el cliente en cuanto al servicio, solicitado más satisfecho quedara, siendo el tiempo un factor determinante, esto sin descuidar la calidad del servicio.

De esta manera la Florería brinda servicios de manera eficaz, fácil, confiable y dinámica, donde tanto el cliente como los usuarios de la Florería facilitan sus actividades, las cuales estarán al alcance de un celular, computadora o algún dispositivo con acceso a internet.

Socialmente el proyecto se justifica ya que se automatiza tanto el cliente y empleado todo el proceso de solicitud de arreglos.

1.5.3. JUSTIFICACIÓN TECNOLÓGICA

La Florería cuenta con el equipamiento necesario para la implementación del Sistema además de tener su propio hosting con las características necesarias el cual solo aloja a un sistema web de publicación y promoción de servicios.

También cuenta con un cableado estructurado, servicio de internet de alta velocidad y equipos de última generación donde todos cuentan con navegadores correctamente actualizados y conectados a la red.

1.6. ALCANCES Y LÍMITES

1.6.1. ALCANCES

Según nuestros objetivos propuestos, se desarrolla lo siguiente:

- Módulo de Administración de Usuarios y Gestión de Clientes, donde se registra toda la información de los usuarios y clientes para brindarles un mejor servicio, además de tenerlos en permanente contacto, ganando su confianza y lealtad con la Florería.

- Módulo Registro de Pedidos y Control de Envíos, donde se registra los pedidos y envíos solicitados por los clientes de manera ordenada y correcta para que estos lleguen en perfecto estado y así mejorar el desempeño de la Florería.
- Módulo de Reportes, donde se muestra detalladamente reportes mensuales detallando la información de todos los servicios que han sido realizados en el día y el mes, centralizando los mismos y así se obtuvo datos acerca del rendimiento de la Florería.

1.6.2. LÍMITES

- El sistema web no emite ni genera facturas por los servicios.
- El sistema web no realiza servicios de contabilidad ni control de personal.
- El sistema web esta implementado únicamente, en la sucursal de la ciudad de La Paz.
- El sistema web no registra al personal de la Florería.

1.7. APORTES

1.7.1. PRÁCTICO

El sistema web, es de gran utilidad para la “Florería Dalia” porque brinda información segura a los usuarios y clientes, logrando tener un correcto control y seguimiento a todas las actividades realizadas en el proceso de brindar un óptimo servicio al cliente.

1.7.2. TEÓRICO

El proyecto claramente es una referencia para el desarrollo de proyectos orientados al desarrollo web, porque se aplicó la metodología ágil llamada Scrum, que es un marco de trabajo iterativo e incremental y se estructura en ciclos de trabajo llamados Sprints.

En cuanto al modelado y análisis del sistema se empleará UML basado en ingeniería web (UWE).

1.8. METODOLOGÍA

Se aplica el método científico, el cual es un método de estudio sistemático de la naturaleza que incluye las técnicas de observación, reglas para el razonamiento y la predicción, ideas sobre la experimentación planificada y los modos de comunicar los resultados experimentales y teóricos.

Los aspectos de la investigación:

- La investigación exploratoria: que es considerada como el primer acercamiento científico a un problema. Se utiliza cuando este aún no ha sido abordado o no ha sido suficientemente estudiado y las condiciones existentes no son aún determinantes; para luego pasar a la Investigación.
- Descriptiva: que se efectúa cuando se desea describir, en todos sus componentes principales, una realidad.

En el proceso de desarrollo del sistema se aplica la metodología Scrum, la cual pertenece a la categoría de metodologías ágiles, aquellas que buscan gestionar de manera generalizada cómo se van completando las tareas.

En el desarrollo de la metodología Scrum, según los expertos se dieron cuenta de que tenía que haber una cierta disciplina detrás de la ciencia de la gestión de proyectos.

Los proyectos se pueden lograr entregar a tiempo y dentro del presupuesto, cuando los directores de proyectos cumplen los siguientes parámetros:

- Organizan el negocio en pequeños equipos autónomos multifuncionales.
- Organizan el trabajo en pequeñas partes realizables.
- Priorizan y completan la estimación y la entrega.
- Organizan la entrega de pequeños componentes de trabajo en un corto plazo de tiempo (1 a 4 semanas) en Sprints (Iteraciones).
- Consultan con los clientes / usuarios finales y organizar planes de entrega basado en resultados de la inspección de cada iteración.

- Optimizan todo el proceso basado en la revisión retrospectiva después de cada iteración.

Esta es la idea central de la que evolucionó Scrum. A medida que el mundo de los negocios se volvió más competitivo, los promotores del proyecto estaban buscando una manera de acelerar la ejecución de los proyectos para que pudieran obtener una ventaja competitiva más rápidamente al hacer lanzamiento al mercado de su producto. Especialistas en gestión de proyectos llegaron a soportar ese objetivo con la estructura de Scrum.

La estructura Scrum es, entregar un proyecto conforme con los requisitos de diversas partes interesadas (Equipo Scrum y Clientes) es una tarea desafiante. Sin embargo, Scrum ha facilitado este desafío investigando la prescripción de una estructura para llevar a cabo el proceso de gestión del proyecto. Los proyectos complejos pueden ser entregados a través de la colaboración y el trabajo eficaz en equipo. El foco principal del éxito de Scrum es tratar de gestionar con eficacia un Product Backlog (Pila de producto) de entregables que representan el todo. Esto a su vez terminó por romper el todo en pequeñas entregas conocidos como Sprint Backlog (Pila del sprint). A medida que se completa cada Sprint, añade valor a las prestaciones de los Sprints anteriores por lo que el sprint final marca la culminación del proyecto. Dentro de esta estructura, Scrum utiliza equipos que se rigen por roles predeterminados, Eventos, artefactos y Reglas.

En cuanto al modelado y análisis del sistema utilizaremos UWE el cual nos ayudará el modelado de la aplicación web, El principal objetivo del enfoque UWE es proporcionar: un lenguaje de modelado específico del dominio basado en UML; una metodología dirigida por modelos; herramientas de soporte para el diseño sistemático; y herramientas de soporte para la generación semi-automática de Aplicaciones Web.

Para desarrollar el Frontend y Backend se tienen las siguientes herramientas; como lenguaje de programación a PHP, Frameworks como Bootstrap, CSS, Ajax, Codeigniter y como gestor de base de datos a Mysql.

2.1. MARCO TEÓRICO

2.2. INGENIERÍA DE SOFTWARE

La ingeniería de software es la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento de software, y el estudio de estos enfoques, es decir, el estudio de las aplicaciones de la ingeniería al software. Integra matemáticas, ciencias de la computación y prácticas cuyos orígenes se encuentran en la ingeniería.

Se citan las definiciones más reconocidas, formuladas por prestigiosos autores:

Ingeniería de software es el estudio de los principios y metodologías para el desarrollo y mantenimiento de sistemas software (Zelkovitz, 1978).

Ingeniería de software es la aplicación práctica del conocimiento científico al diseño y construcción de programas de computadora y a la documentación asociada requerida para desarrollar, operar y mantenerlos. Se conoce también como desarrollo de software o producción de software (Bohem, 1976).

La ingeniería de software trata del establecimiento de los principios y métodos de la ingeniería a fin de obtener software de modo rentable, que sea fiable y trabaje en máquinas reales (Bauer, 1972).

La ingeniería de software es la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación, y mantenimiento del software.

En 2004, la U. S. Bureau of Labor Statistics (Oficina de Estadísticas del Trabajo de Estados Unidos) contó 760 840 ingenieros de software de computadora.

El término "Ingeniero de Software", sin embargo, se utiliza de manera genérica en el ambiente empresarial, y no todos los que se desempeñan en el puesto de ingeniero de software poseen realmente títulos de ingeniería de universidades reconocidas.

Algunos autores consideran que "desarrollo de software" es un término más apropiado que "ingeniería de software" para el proceso de crear software. Personas como Pete McBreen (autor de Software Craftmanship) cree que el término IS implica niveles de rigor y prueba de procesos que no son apropiados para todo tipo de desarrollo de software.

Indistintamente se utilizan los términos "Ingeniería de Software" o "ingeniería del software"; aunque menos común también se suele referenciar como "ingeniería en software". En Hispanoamérica los términos más comúnmente usados son los dos primeros.

La creación del software es un proceso intrínsecamente creativo y la ingeniería del software trata de sistematizar este proceso con el fin de acotar el riesgo de fracaso en la consecución del objetivo, por medio de diversas técnicas que se han demostrado adecuadas sobre la base de la experiencia previa.

2.3. METODOLOGÍAS DE DESARROLLO ÁGILES

Las metodologías de desarrollo ágiles nacen en respuesta a los problemas de las metodologías tradicionales y se basa en dos aspectos, el retrasar las decisiones y la planificación adaptativa; potenciando más el desarrollo del software a gran escala (Cabrera, Figueroa, y Solís, 2008).

Como resultado de esta nueva teoría se crea un Manifiesto Ágil cuyos principios son:

- Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor.
- Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos ágiles aprovechan el cambio para proporcionar ventaja al cliente.
- Entregamos software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible.

- Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.
- La atención continua a la excelencia técnica y al buen diseño mejora la Agilidad.
- Los proyectos se desarrollan en torno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo.
- El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.
- El software funcionando es la medida principal de progreso.
- Los procesos Ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de forma indefinida.
- La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.
- Las mejores arquitecturas, requisitos y diseños emergen de equipos auto-organizados.
- A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia.

2.4. METODOLOGÍA SCRUM

Scrum es un marco de trabajo que define un conjunto de prácticas y roles, y que puede tomarse como punto de partida para definir el proceso de desarrollo que se ejecutará durante un proyecto.

Los roles principales en Scrum son el Scrum Master, que procura facilitar la aplicación de Scrum y gestionar cambios, el Product Owner, que representa a los stakeholders (interesados externos o internos), y el Team (equipo) que ejecuta el desarrollo y demás elementos relacionados con él.

Durante cada sprint, un periodo entre una y cuatro semanas (la magnitud es definida por el equipo y debe ser lo más corta posible), el equipo crea un incremento de software potencialmente entregable (utilizable). El conjunto de características que forma parte de cada sprint viene del Product Backlog, que es un conjunto de requisitos de alto

nivel priorizados que definen el trabajo a realizar (PBI, Product Backlog Item). Los elementos del Product Backlog que forman parte del sprint se determinan durante la reunión de Sprint Planning. Durante esta reunión, el Product Owner identifica los elementos del Product Backlog que quiere ver completados y los da a conocer al equipo. Entonces, el equipo conversará con el Product Owner buscando la claridad y magnitud adecuadas (Cumpliendo el INVEST) para luego determinar la cantidad de ese trabajo que puede comprometerse a completar durante el siguiente sprint. Durante el sprint, nadie puede cambiar el Sprint Backlog, lo que significa que los requisitos están congelados durante el sprint.

Scrum permite la creación de equipos auto organizados impulsando la co-localización de todos los miembros del equipo, y la comunicación verbal entre todos los miembros y disciplinas involucrados en el proyecto.

La metodología se basa en:

- El desarrollo incremental de los requisitos del proyecto en bloques temporales cortos y fijos.
- Se da prioridad a lo que tiene más valor para el cliente.
- El equipo se sincroniza diariamente y se realizan las adaptaciones necesarias.
- Tras cada iteración (un mes o menos entre cada una) se muestra al cliente el resultado real obtenido, para que este tome las decisiones necesarias en relación a lo observado.
- Se le da la autoridad necesaria al equipo para poder cumplir los requisitos.
- Fijar tiempos máximos para lograr objetivos.
- Equipos pequeños (de 3 a 9 personas cada uno).

PRINCIPALES CARACTERÍSTICAS DE SCRUM

- Gestión regular de las expectativas del cliente, resultados anticipados, flexibilidad y adaptación, retorno de inversión, mitigación de riesgos, productividad y calidad, o equipo motivado.
- Se hace uso de equipos auto-dirigidos y auto-organizados.

- Se realiza a diario una reunión de Scrum, que es una reunión de avance diaria que no dura más de 15 minutos con el objetivo de obtener realimentación sobre las tareas del equipo y los obstáculos que se presentan.

Cada uno de estos puntos mencionados hacen que el Scrum sea utilizado de manera regular en un conjunto de buenas prácticas para el trabajo en equipo y de esa manera obtener resultados posibles.

Existen varias implementaciones de sistemas para gestionar el proceso de Scrum, que van desde notas amarillas "post-it" y pizarras hasta paquetes de software; requiere muy poco esfuerzo para comenzarse a utilizar. Así, si se utiliza una pizarra con notas autoadhesivas cualquier miembro del equipo podrá ver tres columnas: trabajo pendiente ("backlog"), tareas en proceso ("in progress") y hecho ("done"). De un solo vistazo, una persona puede ver en qué están trabajando los demás en un momento determinado.

ROLES PRINCIPALES

- **Product Owner**, se asegura de que el equipo Scrum trabaje de forma adecuada desde la perspectiva del negocio. El Product Owner ayuda al usuario a escribir las historias de usuario, las prioriza, y las coloca en el Product Backlog.
- **ScrumMaster (o Facilitador)**, su trabajo primario es eliminar los obstáculos que impiden que el equipo alcance el objetivo del sprint. El ScrumMaster no es el líder del equipo (porque ellos se auto-organizan), sino que actúa como una protección entre el equipo y cualquier influencia que le distraiga. El ScrumMaster se asegura de que el proceso Scrum se utiliza como es debido. El ScrumMaster es el que hace que las reglas se cumplan.
- **Equipo de desarrollo**, tiene la responsabilidad de entregar el producto. Es recomendable un pequeño equipo de 3 a 9 personas con las habilidades transversales necesarias para realizar el trabajo (análisis, diseño, desarrollo, pruebas, documentación).

ROLES AUXILIARES

Los roles auxiliares en los "equipos Scrum" son aquellos que no tienen un rol formal y no se involucran frecuentemente en el "proceso Scrum", sin embargo, deben ser tomados en cuenta. Un aspecto importante de una aproximación ágil es la práctica de involucrar en el proceso a los usuarios, expertos del negocio y otros interesados ("stakeholders"). Es importante que esa gente participe y entregue retroalimentación con respecto a la salida del proceso a fin de revisar y planear cada sprint.

- **Stakeholders (Clientes, Proveedores, Vendedores y otros similares)**

Son las personas que hacen posible el proyecto y para quienes el proyecto producirá el beneficio acordado que justifica su desarrollo. Solo participan directamente durante las revisiones del "sprint".

FLUJO DE TRABAJO

SPRINT

El Sprint es el período en el cual se lleva a cabo el trabajo en sí. Es recomendado que la duración de los sprints sea constante y definida por el equipo con base en su propia experiencia. Se puede comenzar con una duración de sprint en particular (2 o 3 semanas) e ir ajustándolo con base en el ritmo del equipo, aunque sin relajarlo demasiado. Al final de cada sprint, el equipo deberá presentar los avances logrados, y el resultado obtenido es un producto que, potencialmente, se puede entregar al cliente.

Así mismo, se recomienda no agregar objetivos al sprint o *sprint backlog* a menos que su falta amenace al éxito del proyecto. La constancia permite la concentración y mejora la productividad del equipo de trabajo.

El tiempo mínimo de un Sprint es de dos (2) semanas y el máximo es de cuatro (4) semanas.

PLANIFICACIÓN DEL SPRINT

Al comienzo de un sprint, el equipo de scrum tiene un evento de planificación de sprint.

- Uno de los objetivos de la reunión es identificar y comunicar cuánto del trabajo es probable que se realice durante el actual Sprint.

SCRUM DIARIO

También llamado Daily Standup. Cada día durante la iteración, tiene lugar una reunión de estado del proyecto. Su objetivo es que los miembros del equipo se mantengan actualizados unos a otros sobre el trabajo de cada uno desde el último standup, qué problemas han encontrado o prevén encontrar, y qué planean hacer.²

- La reunión tiene una duración fija de entre 5 y 15 minutos.
- Se recomienda hacerla de pie para recordar que debe ser una reunión breve y centrada en su objetivo, sin divagaciones. Es obligatorio parar todo lo que se está haciendo para concentrarse en la reunión.
- Si se requiere ampliar un tema, se hará tras el Daily Standup, pero no se interrumpe la dinámica del Standup para elaborar una discusión.
- Se hace siempre a la misma hora y en el mismo lugar. Si falta alguien, no se pospone la reunión.

REVISIÓN DE SPRINT

Al final de un sprint, el equipo realiza dos eventos: la revisión del sprint y la retrospectiva del Sprint.

En la reunión de revisión de sprint se presentan los trabajos completados y su duración no debería ser superior a 4 horas para un Sprint de 1 mes.

RETROSPECTIVA DEL SPRINT

Después de cada sprint, se lleva a cabo una retrospectiva del sprint, en la cual todos los miembros del equipo dejan sus impresiones sobre el sprint recién superado. El propósito de la retrospectiva es realizar una mejora continua del proceso. Esta reunión tiene un tiempo fijo de cuatro horas.

BENEFICIOS DE SCRUM

- **Flexibilidad a cambios.** Gran capacidad de reacción ante los cambiantes requerimientos generados por las necesidades del cliente o la evolución del mercado. El marco de trabajo está diseñado para adecuarse a las nuevas exigencias que implican proyectos complejos.
- **Reducción del Time to Market.** El cliente puede empezar a utilizar las características más importantes del proyecto antes de que esté completamente terminado.
- **Mayor calidad del software.** El trabajo metódico y la necesidad de obtener una versión de trabajo funcional después de cada iteración, ayuda a la obtención de un software de alta calidad.
- **Mayor productividad.** Se logra, entre otras razones, debido a la eliminación de la burocracia y la motivación del equipo proporcionado por el hecho de que pueden estructurarse de manera autónoma.
- **Maximiza el retorno de la inversión (ROI).** Creación de software solamente con las prestaciones que contribuyen a un mayor valor de negocio gracias a la priorización por retorno de inversión.
- **Predicciones de tiempos.** A través de este marco de trabajo se conoce la velocidad media del equipo por sprint, con lo que es posible estimar de manera fácil cuando se podrá hacer uso de una determinada funcionalidad que todavía está en el Backlog.
- **Reducción de riesgos** El hecho de desarrollar, en primer lugar, las funcionalidades de mayor valor y de saber la velocidad a la que el equipo avanza en el proyecto, permite despejar riesgos efectivamente de manera anticipada.

DOCUMENTOS

PRODUCT BACKLOG

El **product backlog** se trata como un documento de alto nivel para todo el proyecto. Es el conjunto de todos los requisitos de proyecto, el cual contiene descripciones genéricas

de funcionalidades deseables, priorizadas según su retorno sobre la inversión (ROI) . Representa el *qué* va a ser construido en su totalidad. Es abierto y solo puede ser modificado por el product owner. Contiene estimaciones realizadas a grandes rasgos, tanto del valor para el negocio, como del esfuerzo de desarrollo requerido. Esta estimación ayuda al product owner a ajustar la línea temporal (KEV) y, de manera limitada, la prioridad de las diferentes tareas. Por ejemplo, si dos características tienen el mismo valor de negocio la que requiera menor tiempo de desarrollo tendrá probablemente más prioridad, debido a que su ROI será más alto.

SPRINT BACKLOG

El **sprint backlog** es el subconjunto de requisitos que serán desarrollados durante el siguiente sprint. Al definir el sprint backlog, se describe el *cómo* el equipo va a implementar los requisitos durante el sprint. Por lo general los requisitos se subdividen en tareas, a las cuales se asignan ciertas horas de trabajo, pero ninguna tarea con una duración superior a 16 horas. Si una tarea es mayor de 16 horas, deberá ser dividida en otras menores. Las tareas en el sprint backlog nunca son asignadas, son tomadas por los miembros del equipo del modo que les parezca adecuado.

BURN DOWN CHART

La burn down chart es una gráfica mostrada públicamente que mide la cantidad de requisitos en el Backlog del proyecto pendientes al comienzo de cada Sprint. Dibujando una línea que conecte los puntos de todos los Sprints completados, podremos ver el progreso del proyecto. Lo normal es que esta línea sea descendente (en casos en que todo va bien en el sentido de que los requisitos están bien definidos desde el principio y no varían nunca) hasta llegar al eje horizontal, momento en el cual el proyecto se ha terminado (no hay más requisitos pendientes de ser completados en el Backlog). Si durante el proceso se añaden nuevos requisitos la recta tendrá pendiente ascendente en determinados segmentos, y si se modifican algunos requisitos la pendiente variará o incluso valdrá cero en algunos tramos.

DEFINITION OF DONE

El Definition of Done es un documento con una serie de criterios comunes para determinar cuándo una tarea está completamente hecha.

INGENIERÍA WEB BASADA EN UML (UWE)

La Ingeniería Web basada en UML (UWE, por sus siglas en inglés UML Based Web Engineering) es una metodología detallada para el proceso de autoría de aplicaciones con una definición exhaustiva del proceso de diseño que debe ser utilizado. Este proceso, iterativo e incremental, incluye flujos de trabajo y puntos de control, y sus fases coinciden con las propuestas en el Proceso Unificado de Modelado.

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.

Otras características relevantes del proceso y método de autoría de UWE son el uso del paradigma orientado a objetos, su orientación al usuario, la definición de un metamodelo (modelo de referencia) que da soporte al método y el grado de formalismo que alcanza debido al soporte que proporciona para la definición de restricciones sobre los modelos.

La metodología UWE define vistas especiales representadas gráficamente por diagrama en UML. Los principales aspectos en los que se fundamenta UWE son los siguientes:

- Notación estándar: el uso de la metodología UML para todos los modelos.
- Métodos definidos: pasos definidos para la construcción de cada modelo.
- Especificación de restricciones: recomendables de manera escrita, para que la exactitud en cada modelo aumente.

EL PROCESO UWE

Aplicando los principios de MDA, la metodología UWE propone construir un conjunto de CIM's, PIM's y PSM's como el resultado de las fases de análisis, diseño e implementación del proceso. La fase de análisis apunta a obtener un conjunto de

requerimientos estables. Los requerimientos funcionales son capturados a través del modelo de requerimientos. El modelo de requerimientos conforma casos de uso especializados y un modelo de clases para la aplicación Web. La fase de diseño consiste en construir una serie de modelos para el contenido, navegación, proceso, presentación y aspectos adaptativos a un nivel de plataforma independiente. Las transformaciones implementan la construcción sistemática de modelos dependientes por la generación de modelos por defecto, los cuales pueden ser refinados por el diseñador. Finalmente, los modelos de diseño son transformados a una implementación de plataforma específica PSM. Este proceso se extiende con la construcción de máquinas de estado UML (conocidos como la gran foto) que integran los modelos de diseño.

La siguiente figura ilustra la dependencia existente entre los correspondientes modelos de UWE:

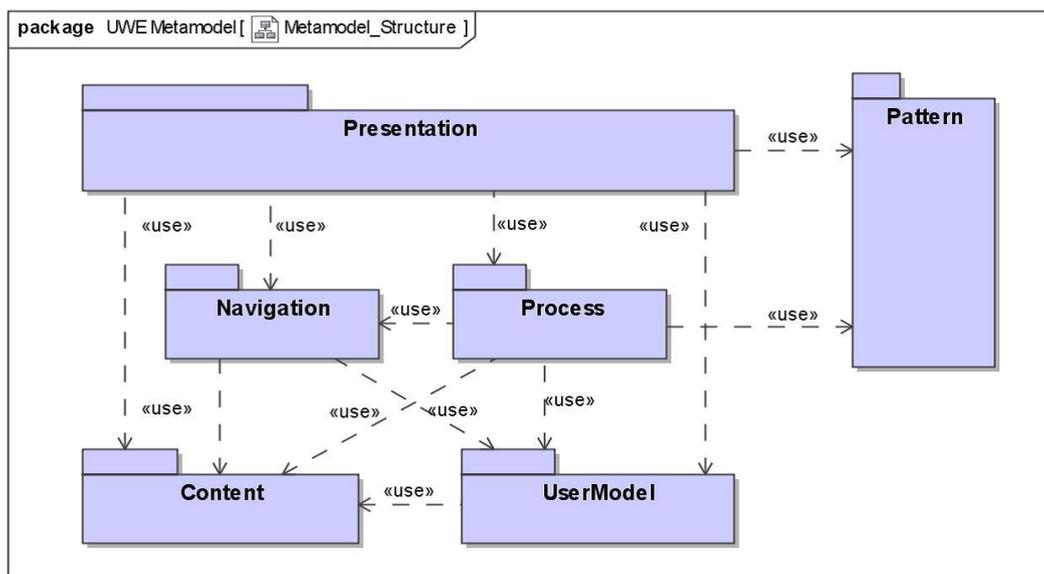


Figura 2.1.: Estructura de paquetes del metamodelo de UWE.

Fuente: (Vallecillo, 2004)

Utilizando un diagrama de actividades puede mostrarse los procesos que sigue UWE, en el mismo los modelos se representan con nodos de objetos y las transformaciones como actividades estereotipadas (círculos icónicos especiales), una cadena de transformaciones define el flujo de control.

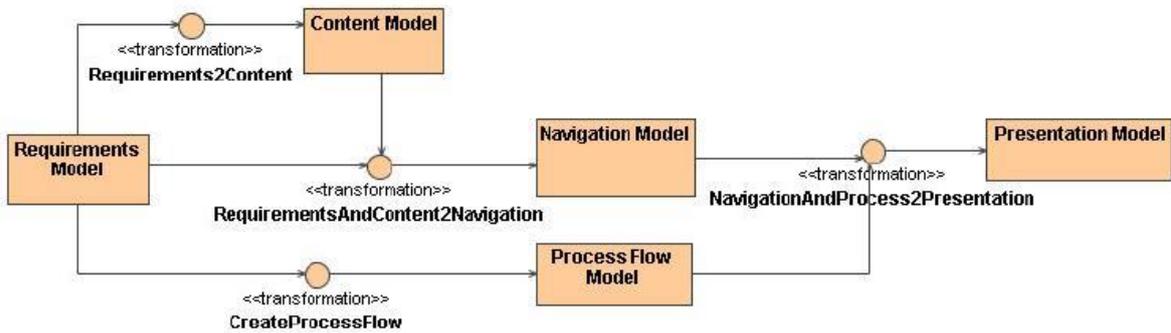


Figura 2.2.: Proceso UWE para transformaciones CIM2PIM y PIM2PIM.
Fuente: (Model Driven Development, 2017)

Un ejemplo de modelos de Contenido, Navegación y Presentación en UWE sería el siguiente:

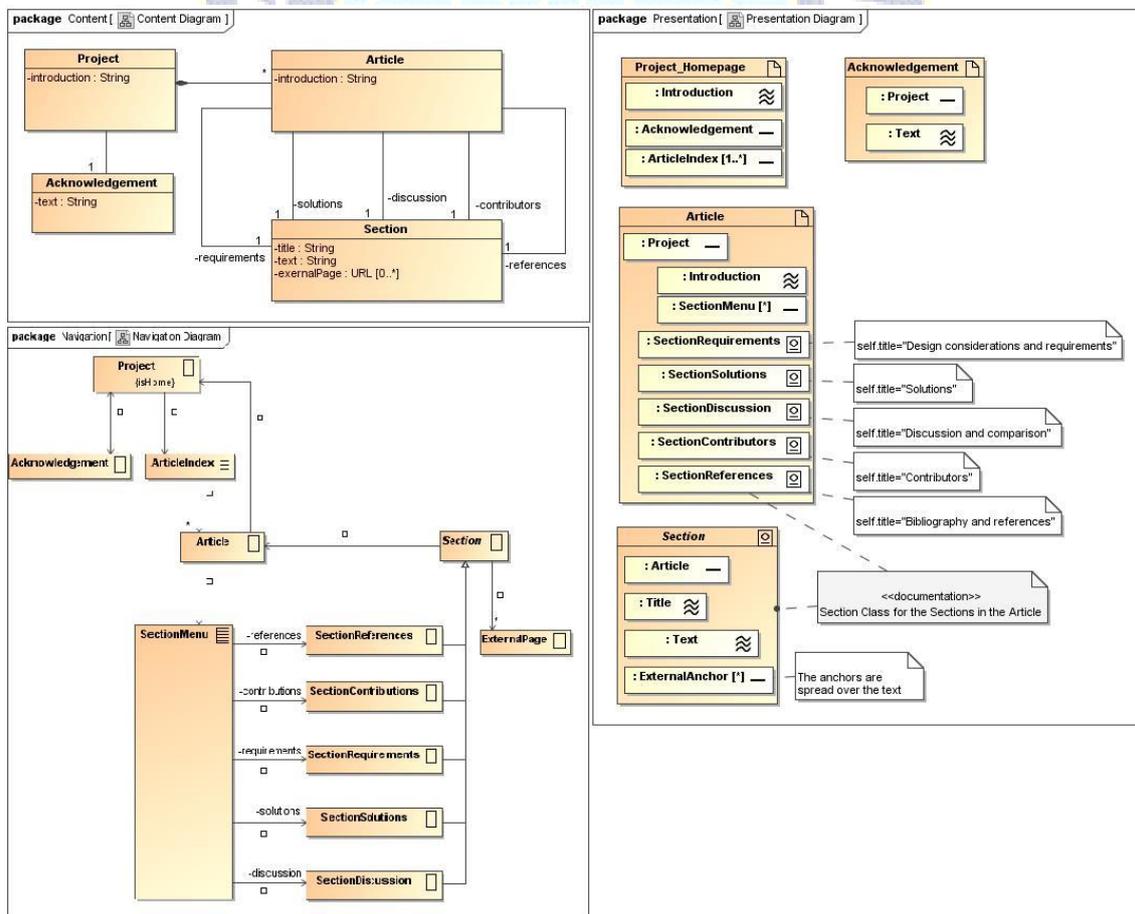


Figura 2.3.: Ejemplos de Modelos de Contenido, Navegación y Presentación.
Fuente: (marcelosalasvargas.blogspot.com, 2017)

INTEGRACIÓN DE PROCESOS

La integración de procesos es una característica que permite actuar como una interface entre los procesos y la navegación. El metamodelo de integración está representado por dos elementos ProcessClass (clase de proceso) y ProcessLink (link de proceso).

Cada proceso está representado por una actividad de proceso (ProcessActivity) y una clase de proceso está asociada a su correspondiente caso de uso de proceso Web (Web Process Use Case). Para cada proceso se designa una clase de proceso integrado en el modelo de navegación. Los links de procesos son links especiales utilizados para invocar un proceso, en el link de proceso el origen o destino del mismo (pero no ambos) se debe ejecutar una clase de proceso.

Cuando se ejecuta un link de proceso que apunta a una clase de proceso se ejecuta la actividad de proceso correspondiente, los parámetros de entrada deben ser compatibles con la clase de contenido origen, los parámetros de salida de la actividad de proceso deben ser compatibles con la clase de contenido destino. El siguiente diagrama ilustra el metamodelo de integración de procesos:

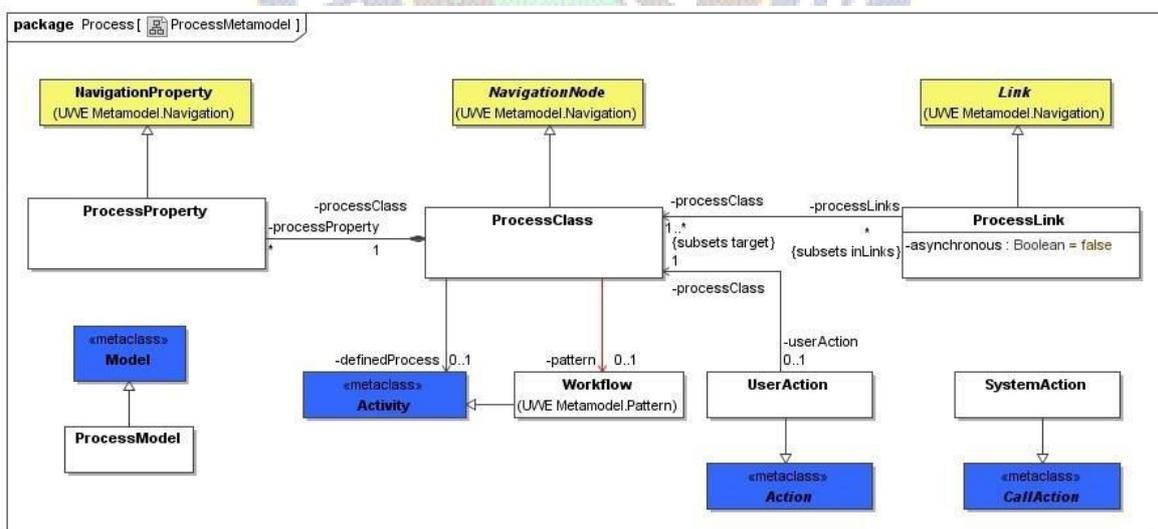


Figura 2.4.: Metamodelo de Integración de Procesos.

Fuente: (marcelosalasvargas.blogspot.com, 2017)

FLUJO DE DATOS Y DE PROCESOS

El comportamiento del proceso Web se define por un modelo de flujo, además se requiere de un modelo de procesamiento de datos que define los datos necesarios para la ejecución del modelo de flujo. Los modelos de Procesos y de Datos generalmente se elaboran concurrentemente.

Los diagramas UML de Actividades son usados para definir el modelo de Flujo. Una actividad en estos diagramas denota un comportamiento parametrizado del proceso, el flujo de ejecución se representa por nodos de actividades conectados. Los nodos de control proveen estructuras de control para el procesamiento. Los nodos de objetos representan datos que fluyen a través de los nodos de actividades, un nodo de acción representa la ejecución de un proceso, pueden definirse acciones especiales para invocar otras actividades, como también acciones compuestas. El siguiente diagrama ilustra el metamodelo del modelo de datos de un proceso:

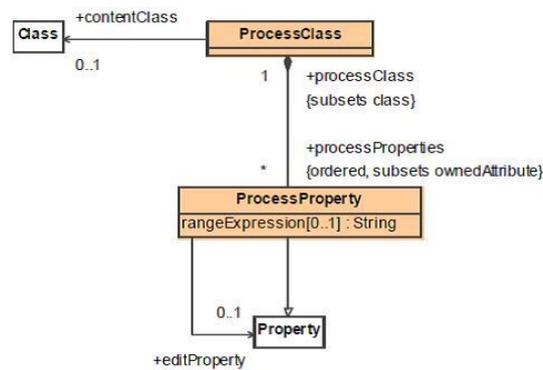


Figura 2.5.: Metamodelo para el modelo de proceso de datos.
Fuente: (marcelosalasvargas.blogspot.com, 2017)

2.5. ARQUITECTURA DEL SISTEMA

La arquitectura de sistema representa una correlación de funciones con componentes de hardware y software, es decir, una correlación de la arquitectura de software con la arquitectura de hardware, e interacción humana con estos componentes.

Para este proyecto se aplica lo que es el MVC (Model View Controller) o modelo, vista y controlador, es un patrón de diseño que separa la lógica empresarial, la lógica de presentación y los datos. MVC se utiliza para diseñar tanto aplicaciones web como aplicaciones móviles. Consta de tres capas:

1. Modelo: Representa la capa empresarial de la aplicación.
2. Vista: Define la lógica de presentación de la aplicación.
3. Controlador: administra el flujo de la aplicación y se comporta como una interfaz entre Modelo y Vista.

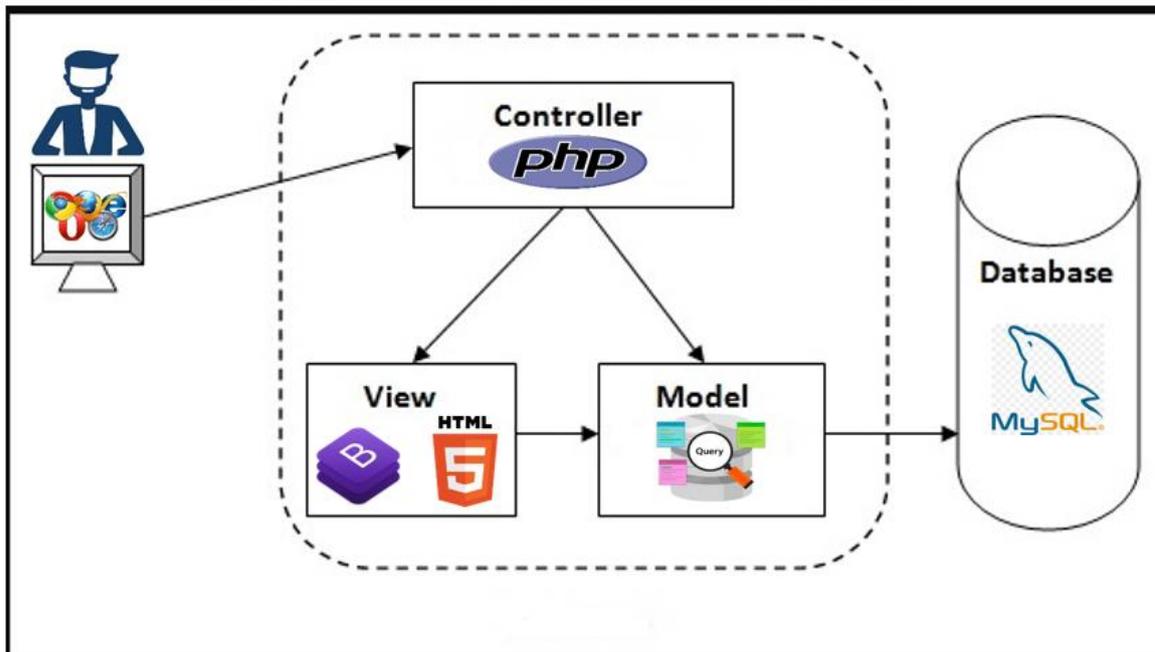


Figura 2.6.: Diagrama de arquitectura del sistema

Fuente: (Elaboración Propia,2020)

3.1. INTRODUCCIÓN

En este capítulo se describe el análisis y diseño del desarrollo del Sistema Web de Gestión de Clientes, Registro de Pedidos y Envíos Caso: Florería “Dalia”, aplicando la metodología Scrum en sus tres fases, Pre – Game, Game y Post – Game, además del modelado UWE, cuyas características y funciones se mencionaron en el capítulo anterior. Las fases de Scrum expresan un modelado iterativo de desarrollo de software, es también complementaria a la metodología UWE, al comenzar una iteración se priorizará las partes principales que se van a construir según el pedido del cliente.

3.2. PROCESO DE DESARROLLO DEL PROYECTO

En la figura 3.1. se observa el proceso que sigue el sistema teniendo las fases de la metodología Scrum al mismo tiempo se complementa con el modelado UWE

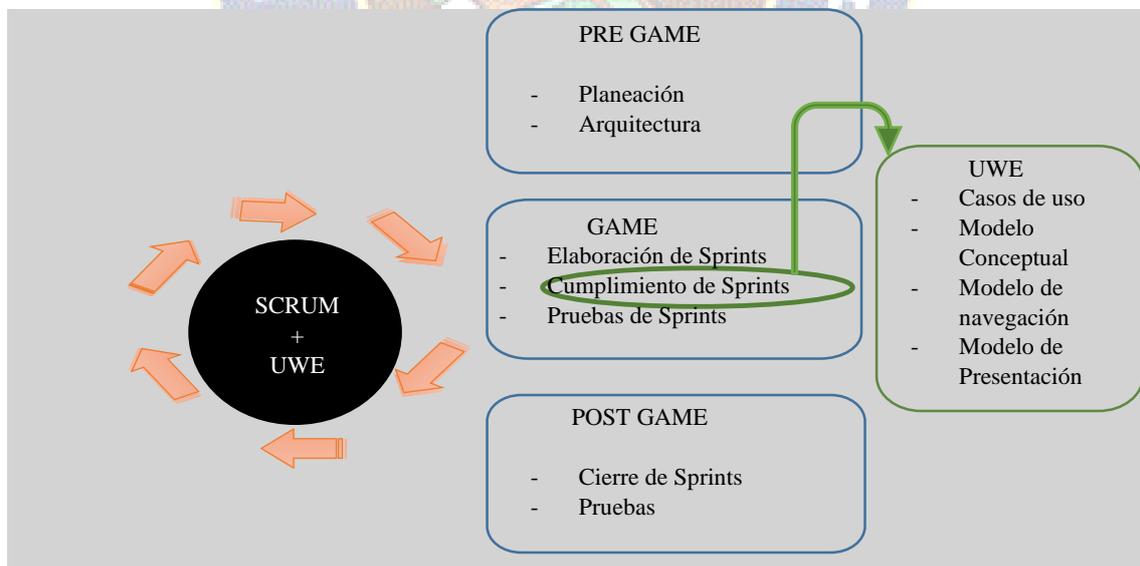


Figura 3.1.: Proceso Scrum
Fuente: (Propia, 2020)

3.3. PRE – GAME

3.3.1. ASIGNACION DE ROLES

A continuación, se detalla a las personas involucradas en el desarrollo del sistema:

ROL	NOMBRE
Product Owner	Dionicio Mamani Laura
Scrum Master	Odón Orlando Rivera Cruz
Scrum Team	Odón Orlando Rivera Cruz
Diseñador	
Desarrollador	
Testeador	
Stekeholders	Florería Dalia

Tabla 3.1.: Asignación de Roles

Fuente: (Propia, 2020)

3.3.2. RECOLECCIÓN DE REQUISITOS

En este punto se tiene las reuniones y entrevistas, con el propietario de la Florería, con el objetivo de recolectar los requisitos, pedidos, tareas y características para el Sistema, se consulta que es lo que esperan del Sistema, además de preguntar que procesos pueden llegar a ser automatizados, sacando así toda la información necesaria para poder armar una pila de requisitos inicial.

Estos requisitos solicitados por el propietario de la florería son: Bases de datos, Login de Acceso tanto para el Administrador, Gerente, usuarios y clientes, informes diarios y mensuales, Interfaces amigables de excelente diseño con colores atractivos, el cual será de gran agrado para la visión del usuario.

A continuación, se muestra la siguiente tabla de requisitos:

ID	DESCRIPCIÓN	MODULO	PRIORIDAD	ESTADO
R1	Base de Datos único para la Florería Dalia.	M1	Alta	Terminado
R2	Log In (Administrador, Gerente, Clientes, Usuarios).	M1	Alta	Terminado
R3	Guardar el historial de todas las actividades que realicen los Usuarios.	M1	Alta	Terminado
R4	Selección de Clientes Potenciales.	M2	Alta	Terminado
R5	Contacto Permanente con Clientes Potenciales.	M2	Alta	Terminado
R6	Registrar fechas de eventos importantes.	M3	Alta	Terminado
R7	Registro de Pedidos.	M3	Alta	Terminado
R8	Registro de Envíos.	M3	Alta	Terminado
R9	Informes Diarios y Mensuales.	M3	Alta	Terminado

Tabla 3.2.: Lista de Requisitos

Fuente: (Propia, 2020)

3.3.3. PRODUCT BACKLOG

Antes de conformar los Sprints, primero se necesita definir una lista de requisitos, estos mismos se realizaron en reuniones con el cliente (Product Owner) y el líder del equipo (Scrum Master), luego los requisitos son ordenados (Product Backlog) priorizando los pedidos del cliente conjunto a el equipo de desarrollo (Scrum Team).

Continuando con la elaboración del Product Backlog se entrevistó al cliente, preguntándole el proceso actual que se tiene para obtener los registros de todos los pedidos y envíos, el cliente Dionisio Mamani Laura comento que el control de pedidos y

envíos tienen cierta demora, y reconoció que todas estas falencias se deben por la falta de sistematización y actualización en su Florería.

A continuación, se muestra una tabla detallada, en donde se indica las tareas, la importancia, estimación, como se hará y algunas notas de cada módulo que tendrá el Sistema Web.

NOMBRE	IMPORTANCIA	ESTIMACIÓN	COMO PROBARLO	NOTAS
MÓDULO 1 (Administración de Usuarios y Gestión de Clientes)				
Diseño de Base de datos y Arquitectura del Sistema	Alta	20 horas	Ver el servidor la implementación de la Base de Datos y la lógica.	Que sea un diseño practico
Autenticación de Administrador.	Alta	20 horas	Entrar como administración, abrir página de inicio, ingresar un nuevo usuario, modificar al personal, eliminar al personal, mostrar personal.	Si fuera necesario, poner tu captcha.
Mostrar acciones con hora y registro de los Usuarios.	Alta	18 horas	Ver la lista de acciones que realizan los Usuarios con fecha y hora de ingreso y salida del Sistema.	Posiblemente generar un informe.

Registro de datos del Cliente.	Alta	20 horas	Registrar los datos del Cliente de manera satisfactoria.	Mandar notificación de registro.
Interfaz de gestión de Cliente.	Alta	20 horas	Ver un Entorno amigable y sencillo para el Usuario.	
Clasificación de Clientes Potenciales.	Alta	20 horas	Filtrar clientes con los parámetros de consumo de cada Cliente.	
MODULO 2 (Registro de Pedidos y Envíos)				
Registro de pedidos, realizados por el Cliente.	Alta	10 horas	Registrando todos los pedidos de acuerdo a la preferencia y demanda del Cliente.	
Registro de envíos de la empresa al Cliente.	Alta	20 horas	Registrando todo el movimiento que hacen los empleados de manera eficiente.	
Interfaz de control de pedidos y envíos en proceso.	Alta	20 horas	Viendo un entorno donde se controle el estado donde se encuentre cada	

			pedido y envío registrado.	
MODULO 3 (Reportes)				
Reportes Gastos e Ingresos.	Alta	10 horas	Generando un informe detallado de todo el movimiento contable diario y mensual de la Florería.	
Reporte mensual de pedidos y envíos para el control.	Alta	8 horas	Generando un informe detallado de todo el movimiento de pedidos y envíos diarios y mensuales de la Florería.	
Reporte de clientes con sus diferentes clasificaciones.	Alta	12 horas	Mostrando a todos aquellos clientes fieles a la Florería.	

Tabla 3.3.: Detalles de los Módulos

Fuente: (Propia, 2020)

3.3.4. REQUERIMIENTO DE HARDWARE Y SOFTWARE

Para poder desarrollar el proyecto e implementarlo se necesita hardware y software, se requiere de los siguientes componentes descritos en la siguiente tabla, con el fin del desarrollo e implementación del Sistema Web.

Requerimiento de hardware	PC de escritorio o Laptop, con las siguientes características: <ul style="list-style-type: none"> - Memoria RAM de 2GB o superior - Espacio Disponible en disco duro de 8 GB o superior - Procesador i3 o Superior a 2.5 MHZ - Pantalla, teclado y mouse.
Requerimiento de software	Se necesitará los siguientes programas <ul style="list-style-type: none"> - Microsoft Word para la documentación - Adobe Reader para los informes - Editor de Plantillas de Excel - Navegador en HTML5 - Que tenga un Servidor

Tabla 3.4.: Requerimiento de Hardware y Software

Fuente: (Propia, 2020)

3.4. GAME

Una de las muchas formas para desarrollar el Sistema Web, es empezar haciendo un análisis y diseño con los modelos propios de la metodología UWE, para después comenzar a desarrollar las interfaces Web en base a los modelos de presentación y navegación, para finalmente llegar a la fase de desarrollo de las clases y sus métodos.

Los módulos a realizar para el Sistema Web de Gestión de Clientes, Registro de Pedidos y Envíos Caso: Florería “Dalia” son: Administración de Usuarios y Gestión de Clientes, Registro de Pedidos y Control de Envíos, y Reportes.

3.4.1. SPRINTS

Son parte del núcleo central de la metodología. Se trata de una partición del proyecto que aproximadamente tiene una duración de 4 semanas, cuyo objetivo es conseguir un incremento de valor en proyecto final a desarrollarse.

3.4.1.1. PRIMER SPRINT DE ADMINISTRACIÓN DE USUARIOS Y GESTIÓN DE CLIENTES

La metodología Scrum indica el inicio del proyecto, este se realizó con la reunión de planificación para el primer Sprint. Un factor importante a considerar es que, en esta reunión, al igual que en la mayor parte del proyecto, la comunicación con los usuarios se realizó de forma irregular debido a que el propietario tiene otras sucursales y otros negocios donde también es indispensable su presencia, por lo cual se hizo difícil reunirse con él, pese a todo lo anterior dicho se pudo trabajar vía virtual y mediante llamadas telefónicas.

3.4.1.2. PLANIFICACIÓN DEL SPRINT 1: ADMINISTRACIÓN DE USUARIOS Y GESTIÓN DE CLIENTES

Durante el primer Sprint se lleva a cabo requerimientos de máxima prioridad según el cliente, que pertenecen al módulo inicial. A continuación, se muestra una tabla en donde se observa las tareas planificadas para este Sprint y que fueron concluidas.

SPRINT 1: MÓDULO DE ADMINISTRACIÓN DE USUARIOS Y GESTIÓN DE CLIENTES			
Sprint	Duración en horas		Días de trabajo
1	150		20 días hábiles
SPRINT BACKLOG			
Tareas	Tipo	Programación en horas	Estado
Planificación y análisis de los requerimientos del Sprint.	Desarrollo	20	Hecho
Diseño de la Base de Datos para el Sistema Web.	Desarrollo	20	Hecho

Diseño Conceptual.	Desarrollo	20	Hecho
Diseño de Presentación.	Desarrollo	7	Hecho
Diseño de la interfaz gráfica (Inicio de Sesión).	Desarrollo	9	Hecho
Seguridad y control de acceso al sistema.	Desarrollo	10	Hecho
Listado de Clientes exclusivos.	Desarrollo	20	Hecho
Registro seleccionado de Clientes.	Desarrollo	25	Hecho
Modificación de datos del Clientes.	Desarrollo	9	Hecho
Eliminación de datos del Clientes.	Desarrollo	7	Hecho
Búsqueda de Clientes.	Desarrollo	9	Hecho
Diseño de la interfaz gráfica (Nuevo Cliente).	Desarrollo	8	Hecho
Seguridad y control de acceso al Sistema.	Desarrollo	10	Hecho

Tabla 3.5.: Primer Sprint
Fuente: (Propia, 2020)

3.4.1.3. ESPECIFICACIÓN DE CASO DE USO DEL SPRINT 1

A continuación, se detalla el caso de uso del primer Sprint.

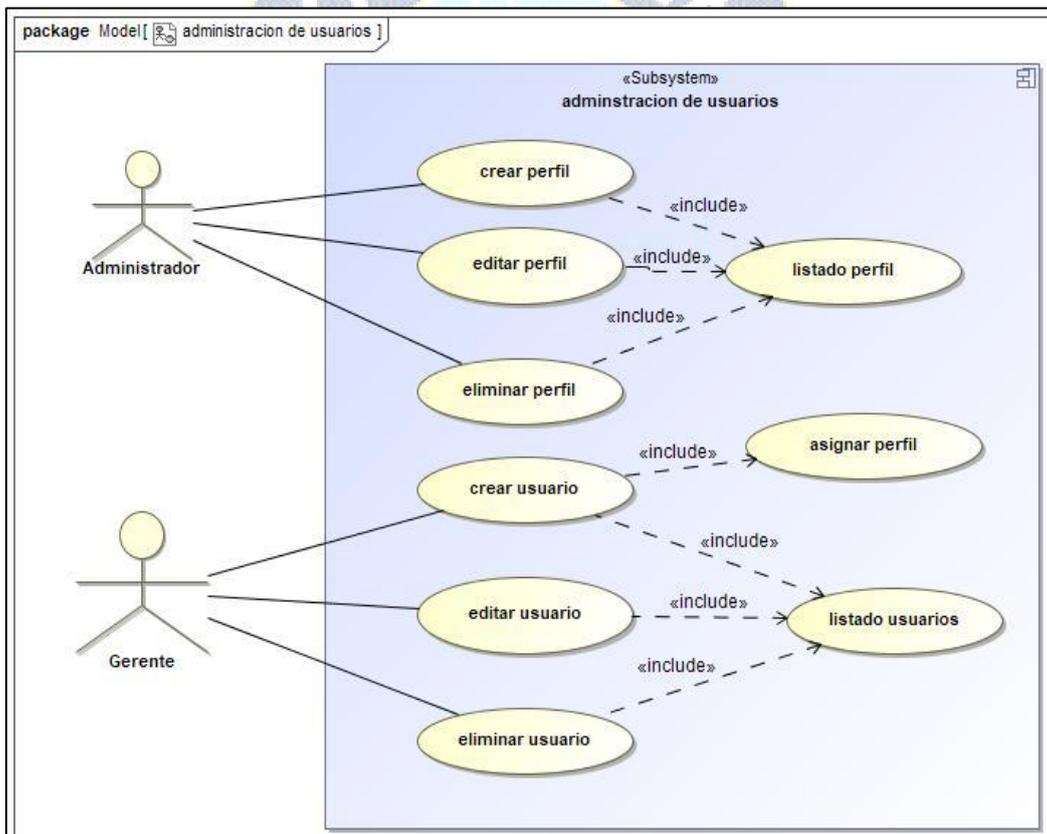


Figura 3.2.: Caso de uso para Administración de roles de Usuario
Fuente: (Elaboración Propia,2020)

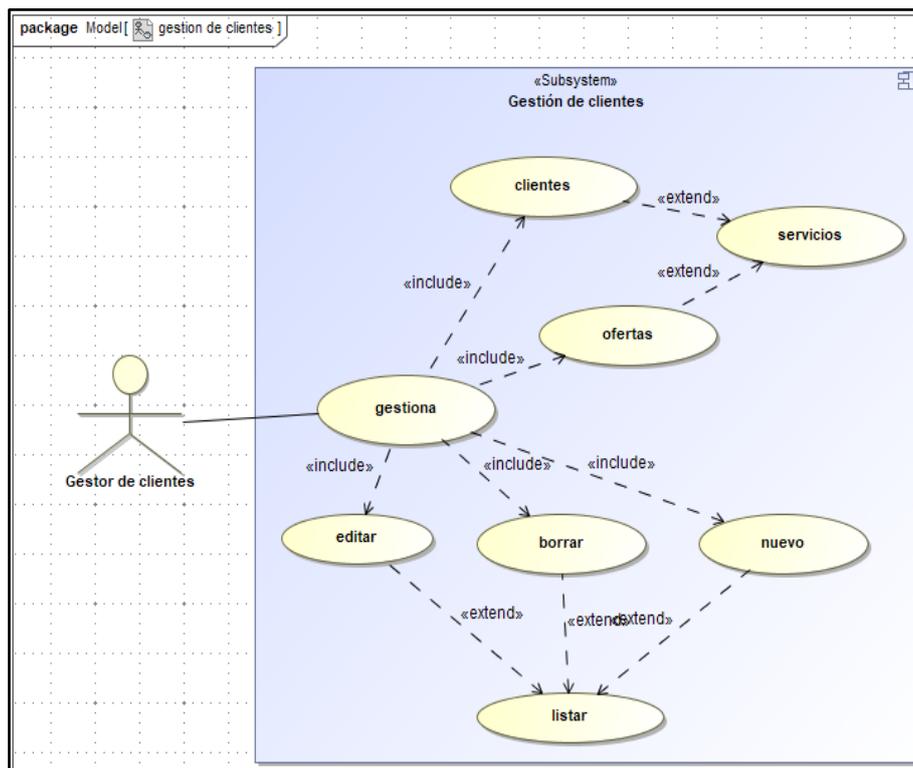


Figura 3.3.: Caso de uso para Gestión de Clientes

Fuente: (Elaboración Propia,2020)

El primer Sprint muestra el inicio de sesión del Usuario como Administrador o Gerente, en el momento de ingresar al Sistema se pedirá el número de carnet y la contraseña designada, además de la implementación del Gestor de Clientes.

NOMBRE	Módulo de Administración de Usuarios y Gestión de Clientes.	
ACTORES	Administrador, Gerente y Gestor de Clientes.	
PROPÓSITO	Administrar los usuarios involucrados en el manejo del Sistema, también Gestionar a los Usuarios	
RESUMEN	La Creación y Administración de nuevos Usuarios.	
REFERENCIAS CRUZADAS	NINGUNA	
EVENTO ACTOR	LO QUE HACE EN EL	EVENTO SISTEMA

El Actor ingresa al Sistema con una cuenta Administrativa.	→	El Sistema valida que los datos sean correctos, posteriormente, envía a la página principal.
El Actor ingresa a la interfaz donde despliega un listado de Usuarios.	→	El Sistema lista a todos los Usuarios registrados anteriormente.
El Actor realiza la acción de; crear, editar o eliminar Usuario.	→	El Sistema crea, edita o elimina Usuario.
El Actor ingresa al Sistema con una cuenta de Gestor de Clientes.	→	El sistema valida los datos una vez llenado el formulario de Log In, una vez ingresado los datos correctos, te envía a la página de inicio.
El Actor ingresa a la interfaz.	→	El Sistema despliega la pantalla de menú.
Una vez llenado los campos del formulario, hace click en el botón adicionar.	→	Almacena la información enviada y reenvía al cliente a la página de listado de administrador.
FLUJO PRINCIPAL		
PRECONDICIÓN	Debe de existir la Base de Datos con la tabla correspondiente a los Usuarios como también de Clientes.	
POS CONDICIÓN	Creación y Administración de nuevos Clientes.	
PRESUNCIÓN	Almacenamiento de datos de forma segura	

Tabla 3.6.: Administración de Usuarios y Gestión de Clientes

Fuente: (Propia, 2020)

3.4.1.4. DISEÑO NAVEGACIONAL ADMINISTRACIÓN DE USUARIOS Y GESTIÓN DE USUARIOS

El diseño navegacional muestra las opciones de navegación y procesos del módulo Administración de Usuarios y Gestión de Clientes que el sistema tendrá en su desarrollo.

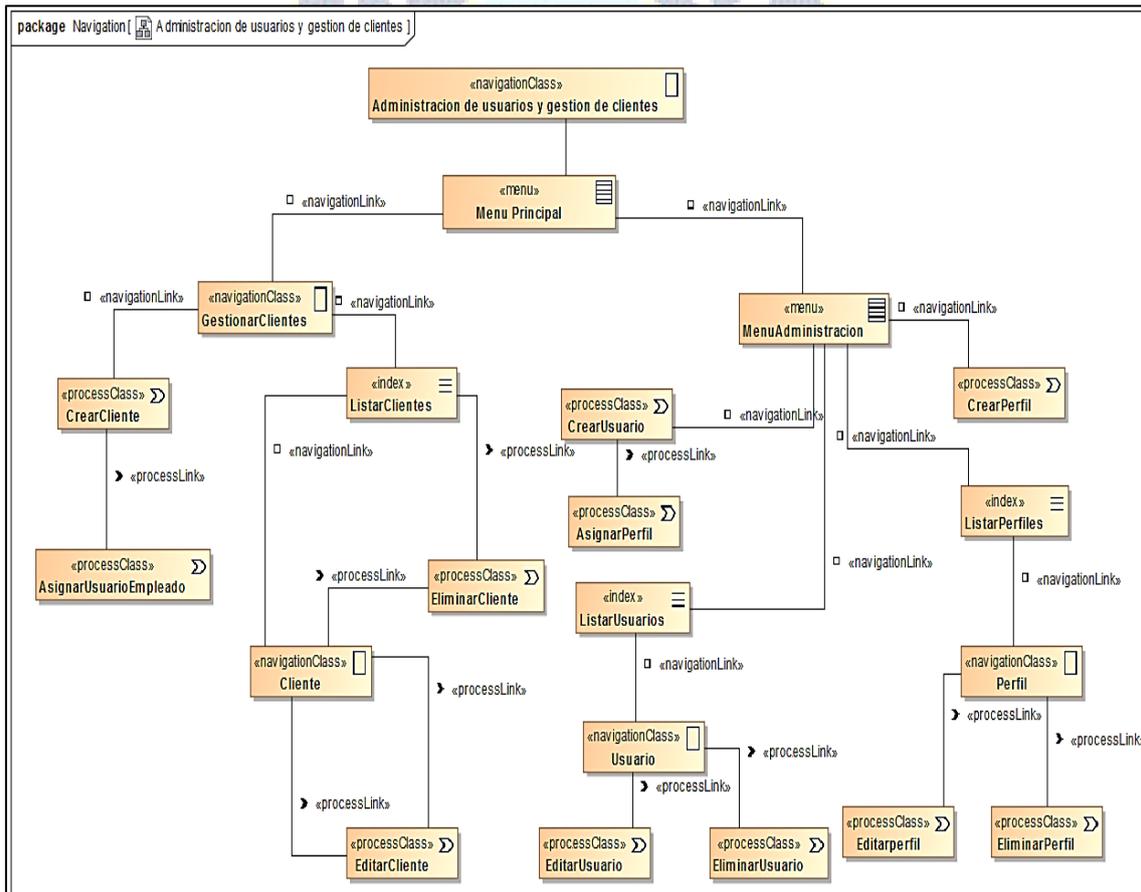


Figura 3.4.: Modelo Navegacional, Administración de Usuarios y Gestión de Clientes
Fuente: (Elaboración Propia, 2020)

3.4.1.5. DISEÑO DE PRESENTACIÓN

El modelo de presentación muestra un bosquejo de la interfaz de la vista para la Administración de Usuarios y Gestión de Clientes.

Se despliegan listas tanto de usuarios como de clientes con sus distintos atributos desplegándose de forma simultánea para hacer visual la información más importante, y

teniendo en las mismas interfaces las acciones necesarias para cumplir las tareas solicitadas.

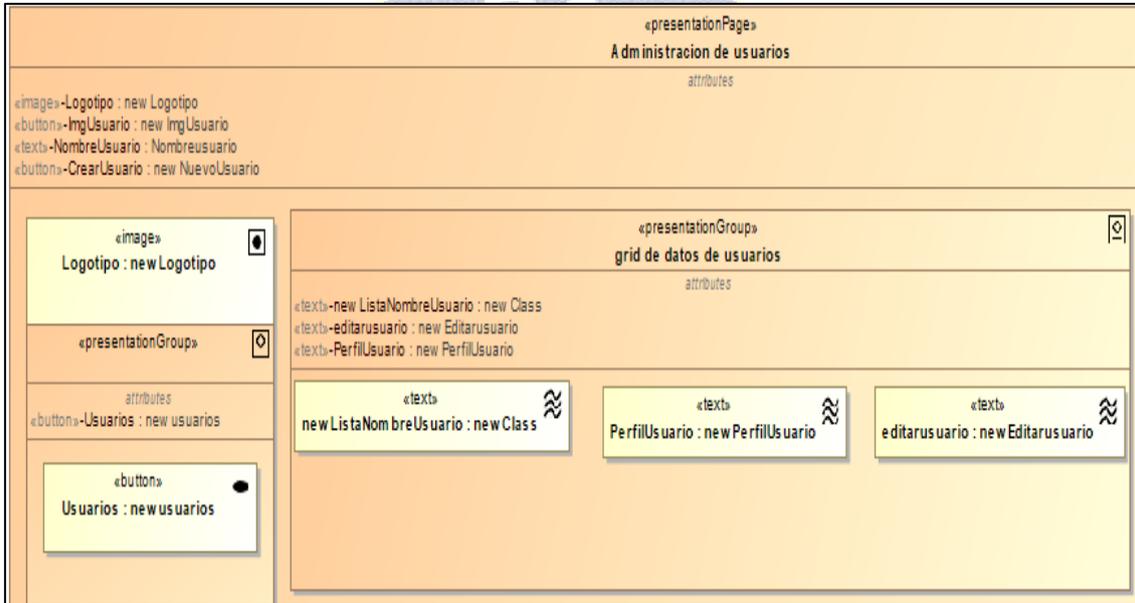


Figura 3.5.: Modelo de Presentación del Administrador de usuarios

Fuente: (Elaboración Propia, 2020)

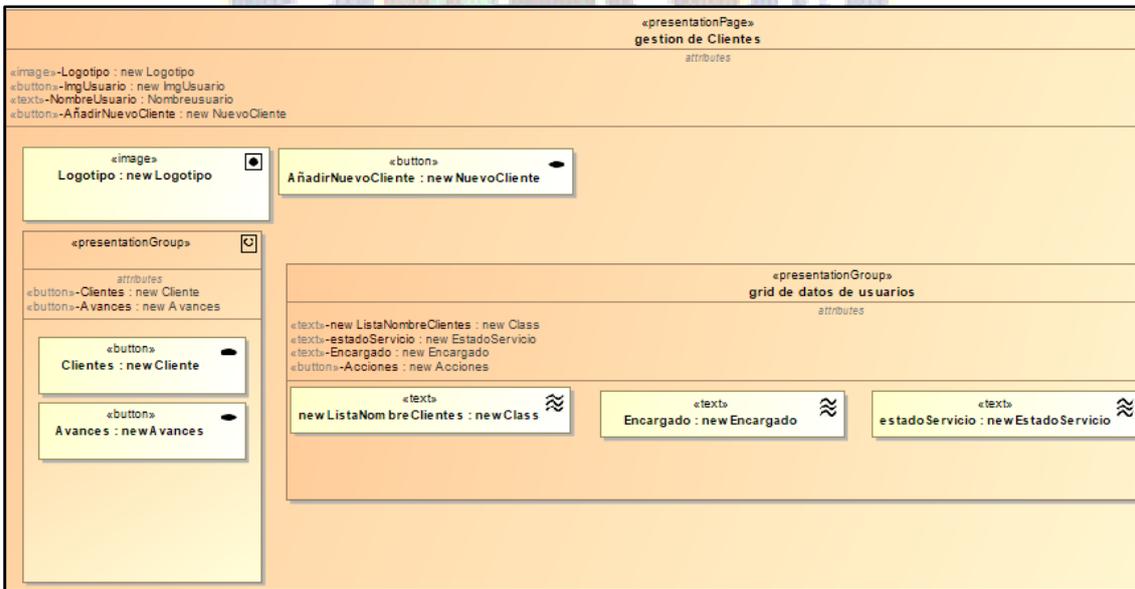


Figura 3.6.: Modelo de presentación para listar y desplegar la información

Fuente: (Elaboración Propia, 2020)

```
$item = "usuario";
$valor = $_POST["ingUsuario"];

$respuesta = ModeloUsuarios::MdlMostrarUsuarios($tabla, $i
if($respuesta["usuario"] == $_POST["ingUsuario"] && $respu
    if($respuesta["estado"] == 1){
        $_SESSION["iniciarSesion"] = "ok";
        $_SESSION["id"] = $respuesta["id"];
        $_SESSION["nombre"] = $respuesta["nombre"];
        $_SESSION["usuario"] = $respuesta["usuario"];
        $_SESSION["foto"] = $respuesta["foto"];
        $_SESSION["perfil"] = $respuesta["perfil"];
    }
}
```

Figura 3.7.: Código php, para el ingreso de usuario
Fuente: (Elaboración Propia, 2020)

3.4.1.6. PANTALLAS DEL SPRINT 1

A continuación, se puede observar la página de inicio de sesión, donde tiene acceso el administrador para ver los servicios que ofrece el sistema y administrar a los usuarios.

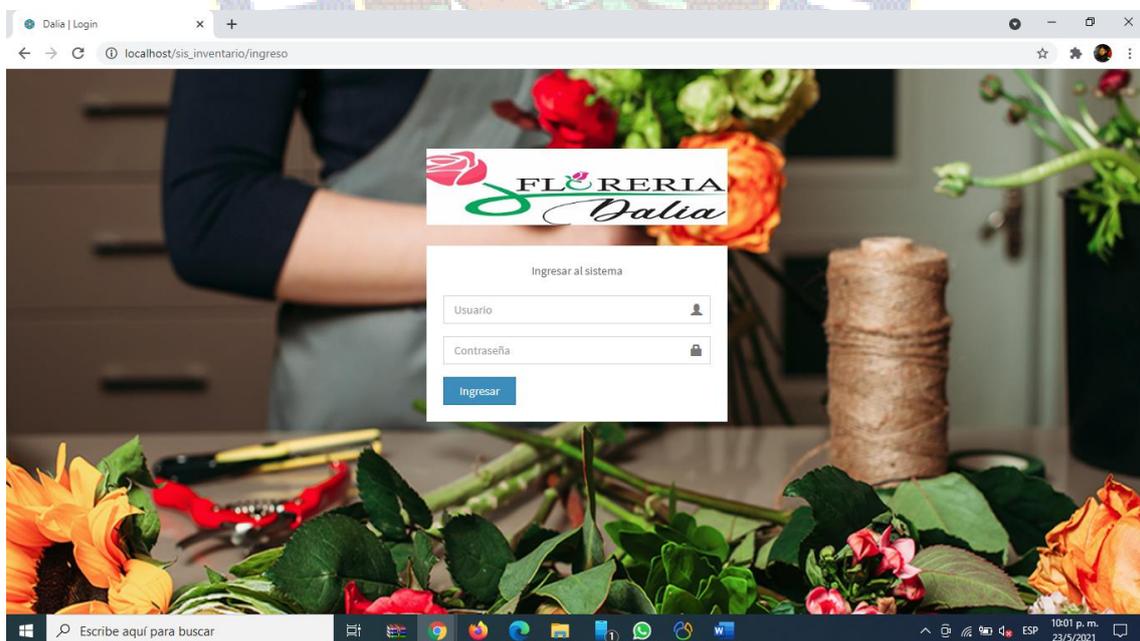


Figura 3.8.: Pantalla de Inicio
Fuente: (Elaboración Propia,2020)

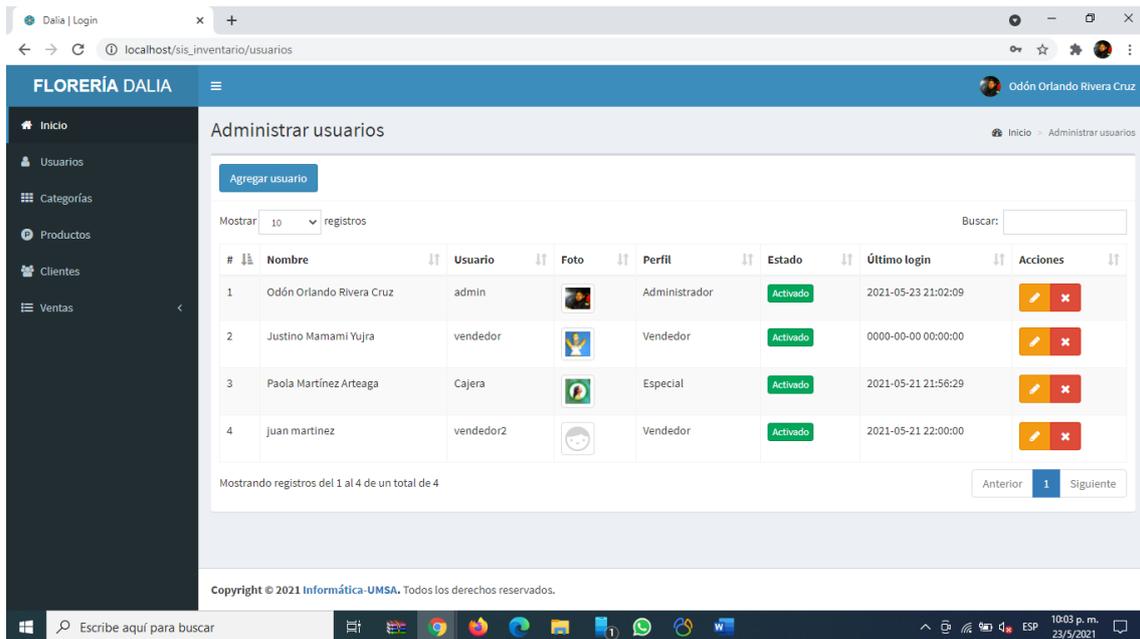


Figura 3.9.: Pantalla de Administración de Usuarios
Fuente: (Elaboración Propia,2020)

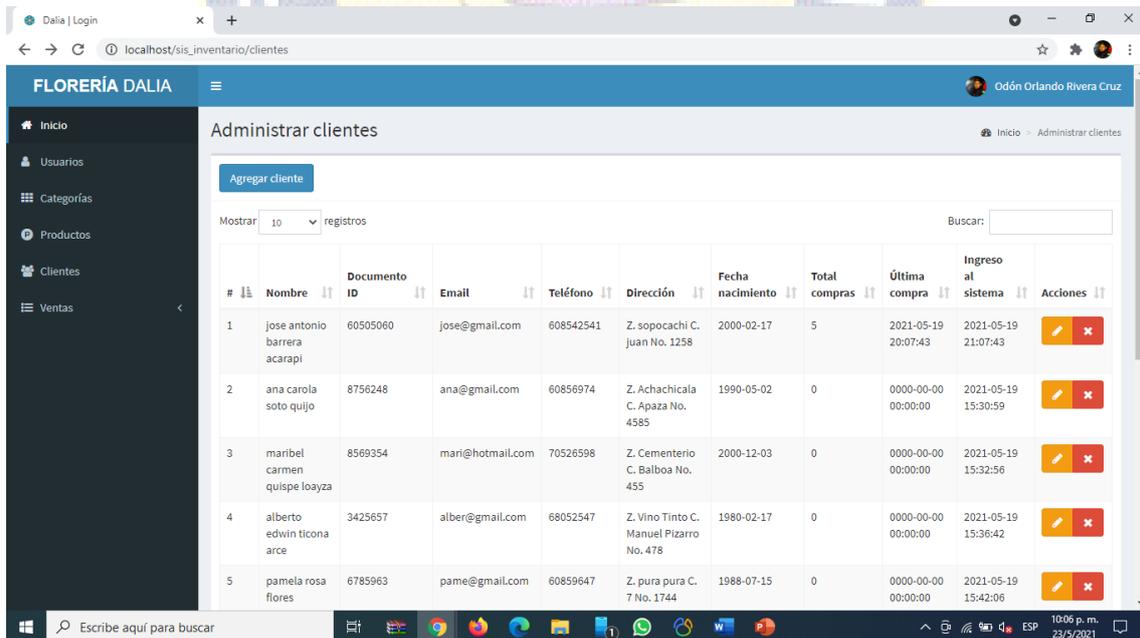


Figura 3.10.: Pantalla de Administración de Clientes
Fuente: (Elaboración Propia,2020)

3.4.1.7. PRUEBA UNITARIA DEL SPRINT 1

En la siguiente tabla, se muestra la prueba unitaria, sirve para comprobar que el módulo funcione correctamente.

Prueba No. 1 MÓDULO DE ADMINISTRACION DE USUARIOS Y GESTIÓN DE CLIENTES	
Descripción	El Administrador introducirá su carnet y contraseña para ingresar al Sistema, una vez dentro del mismo, podrá crear, modificar, eliminar y gestionar a los Usuarios y Clientes.
Objetivos	Abrir el Sistema, después de haber colocado correctamente los datos pedidos.
Condiciones	Conexión a internet desde cualquier medio (PC, Celular, iPhone, Tablet).
Resultado Esperado	Que se habrá el Sistema, sin ningún problema.
Resultado Obtenido	El Sistema inicio correctamente.

Tabla 3.7.: Prueba unitario del Sprint 1

Fuente: (Propia, 2020)

3.4.2. DESARROLLO DEL SEGUNDO SPRINT, REGISTRO DE PEDIDOS Y ENVÍOS

En este módulo vemos un listado amplio que tiene diversos envíos y pedidos a la hora de ser registradas, en las reuniones obtenidas se dijo que, el proceso y tiempo de preparación, moldeado y control de calidad será acorde a la importancia, a continuación, veremos el registro que estas tendrán en su proceso.

3.4.2.1. PLANIFICACIÓN DEL SPRINT 2 REGISTRO DE PEDIDOS Y ENVÍOS

La planificación que dispone y el intervalo que tenga pedidos y envíos siempre se tomaran al pie de la letra como el usuario lo propuso, además de ser distribuido hacia los diferentes tipos de clientes de manera correcta. La entrevista dio como fruto al orden de registro de cada una de ellas que primaban a la hora de su pedido.

En la siguiente tabla se muestra detalladamente las tareas y el tiempo de duración que tardaran en realizar para el segundo Sprint.

SPRINT 2: MÓDULO REGISTRO DE PEDIDOS Y ENVÍOS			
Sprint	Duración en horas		Días de trabajo
2	53		12 días hábiles
SPRINT BACKLOG			
Tareas	Tipo	Programación en horas	Estado
Listado de Pedidos	Desarrollo	8	Hecho
Listado de Envíos	Desarrollo	8	Hecho
Modificación de Pedidos	Desarrollo	7	Hecho
Modificación de Envíos	Desarrollo	8	Hecho
Eliminación de Pedidos	Desarrollo	10	Hecho
Eliminación de Envíos	Desarrollo	10	Hecho
Búsqueda de Pedidos y Envíos	Desarrollo	8	Hecho
Diseño de la interfaz gráfica (Nuevo Pedido)	Desarrollo	9	Hecho
Seguridad y control de acceso al sistema	Desarrollo	10	Hecho

Tabla 3.8.: Segundo Sprint

Fuente: (Propia, 2020)

3.4.2.2. ESPECIFICACIÓN DE CASO DE USO DEL SPRINT 2

A continuación, se detalla el caso de uso del segundo Sprint.

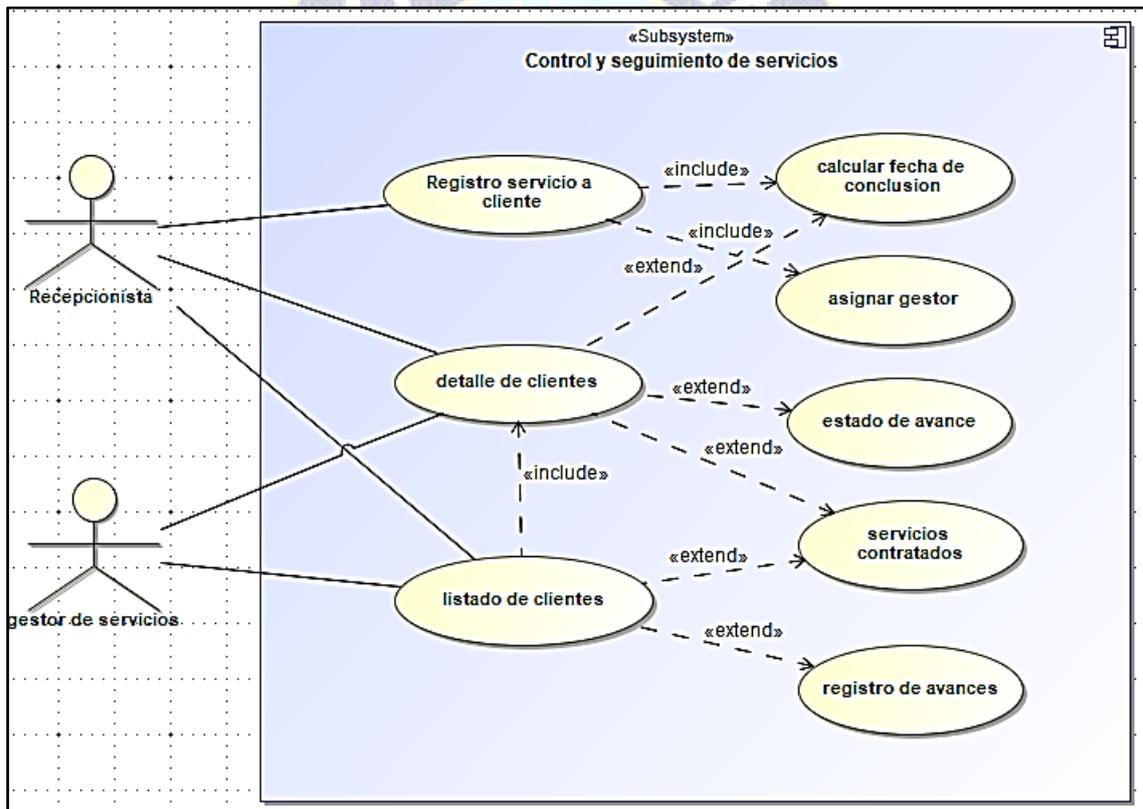


Figura 3.11.: Caso de uso para el Registro de Pedidos y Envíos

Fuente: (Elaboración Propia, 2020)

El segundo Sprint muestra el Registro de Pedidos y Envíos en este caso en el momento de solicitar un producto cuando se ingresa al Sistema.

NOMBRE	Registro de Pedidos y Envíos.
ACTORES	Administrador y Usuario.
PROPÓSITO	Interacción con el Sistema.

RESUMEN	Registro de Pedidos y Envíos de manera correcta, ordenada y puntual.	
REFERENCIAS CRUZADAS	NINGUNA	
EVENTO ACTOR	LO QUE HACE EN EL	EVENTO SISTEMA
El Cliente ingresa al sistema para realizar su pedido, seleccionando el arreglo de su preferencia, indicando el lugar y fecha de entrega.	→	El sistema valida los datos ingresados por el usuario e inmediatamente manda el registro a el administrador, este mismo despliega un mensaje, para que un empleado realice el trabajo de envío del pedido.
En la página de pedidos, se ingresa dentro de un formulario donde se pide todos los datos más relevantes del usuario, para que este sea registrado en su pedido.	→	Direcciona a la página que contiene un formulario de registro con un botón de adicionar, para la creación de un nuevo Usuario.
Una vez llenado los campos del formulario, hace click en el botón registrar.	→	Almacena la información enviada y reenvía al cliente un mensaje “su pedido se ha realizado exitosamente”.
FLUJO PRINCIPAL		
PRECONDICIÓN	Debe de existir la Base de Datos con la tabla correspondiente a los pedidos del Cliente.	
POS CONDICIÓN	Creación de Pedidos.	
PRESUNCIÓN	Almacenamiento de datos de forma segura	

Tabla 3.9.: Registro de Pedidos y Envíos

Fuente: (Propia, 2020)

3.4.2.3. DISEÑO NAVEGACIONAL

El diseño navegacional muestra las opciones de navegación y procesos que el sistema tendrá en su desarrollo.

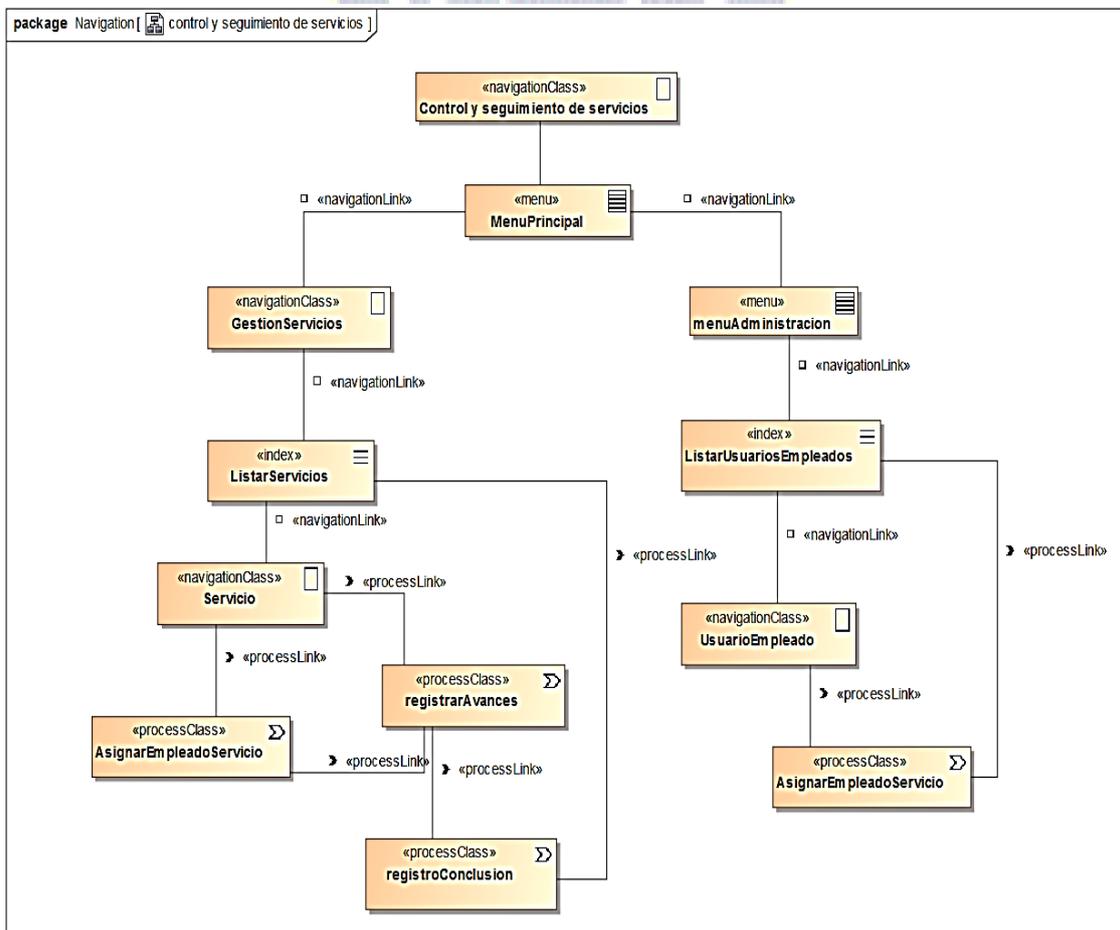


Figura 3.12.: Diseño Navegacional Registro de Pedidos y Envíos

Fuente: (Elaboración Propia, 2020)

3.4.2.4. DISEÑO DE PRESENTACIÓN

En el diagrama de presentación para el módulo de inicio de sesión se muestra la interfaz donde el Cliente introduce los datos respectivos para ingresar al sistema. Proceso a realizando.

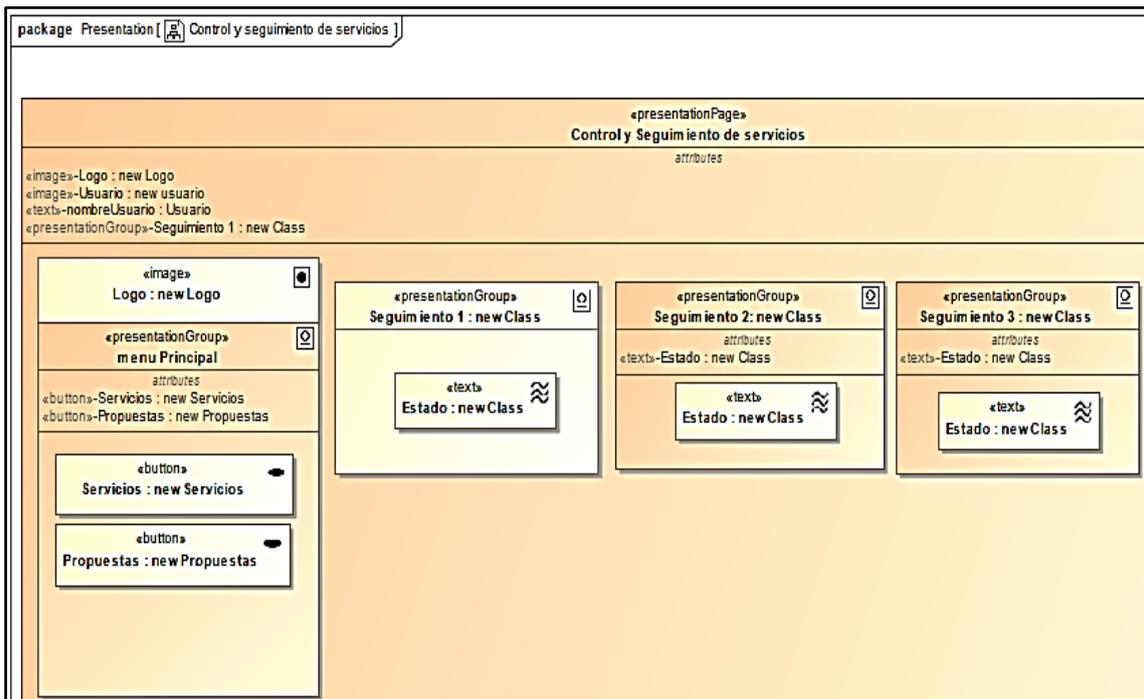


Figura 3.13.: Diseño de Presentación Registro de Pedidos y Envíos
Fuente: (Elaboración Propia, 2020)

```

$tabla = "categorias";

$datos = $_POST["nuevaCategoria"];

$respuesta = ModeloCategorias::mdlIngresarCategori

if($respuesta == "ok"){

    echo '<script>

        swal({
            type: "success",
            title: "La categoría ha sido guardada co
            showConfirmButton: true,
            confirmButtonText: "Cerrar"
        }).then(function(result){

```

Figura 3.14.: Código php para registrar una nueva categoría
Fuente: (Elaboración Propia, 2020)

3.4.2.5. PANTALLA DEL SPRINT 2

A continuación, se puede observar la página de Registro de Pedidos a la cual tendrá acceso el administrador para ver los servicios requeridos por el Cliente.

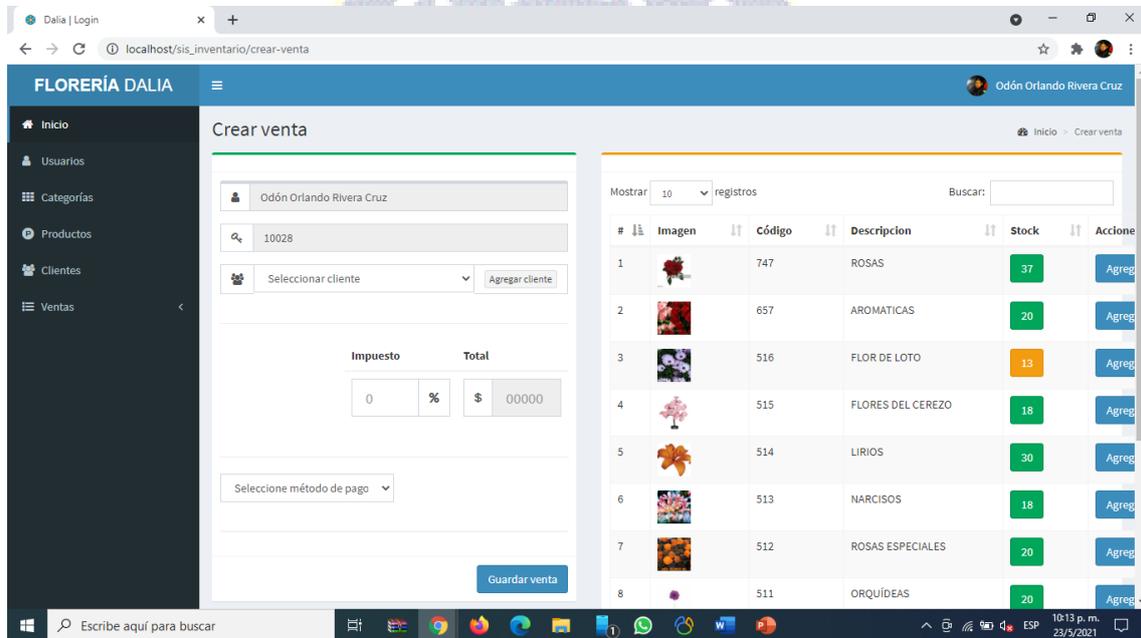


Figura 3.15.: Pantalla Crear Venta
Fuente: (Elaboración Propia, 2020)

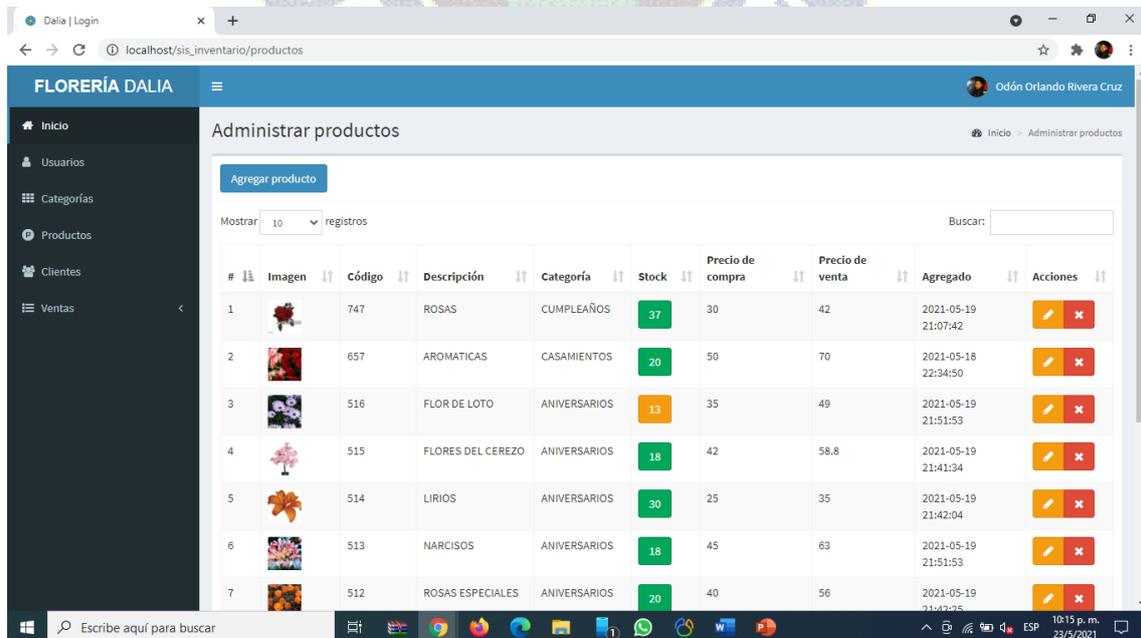


Figura 3.16.: Pantalla Administración de Productos
Fuente: (Elaboración Propia, 2020)

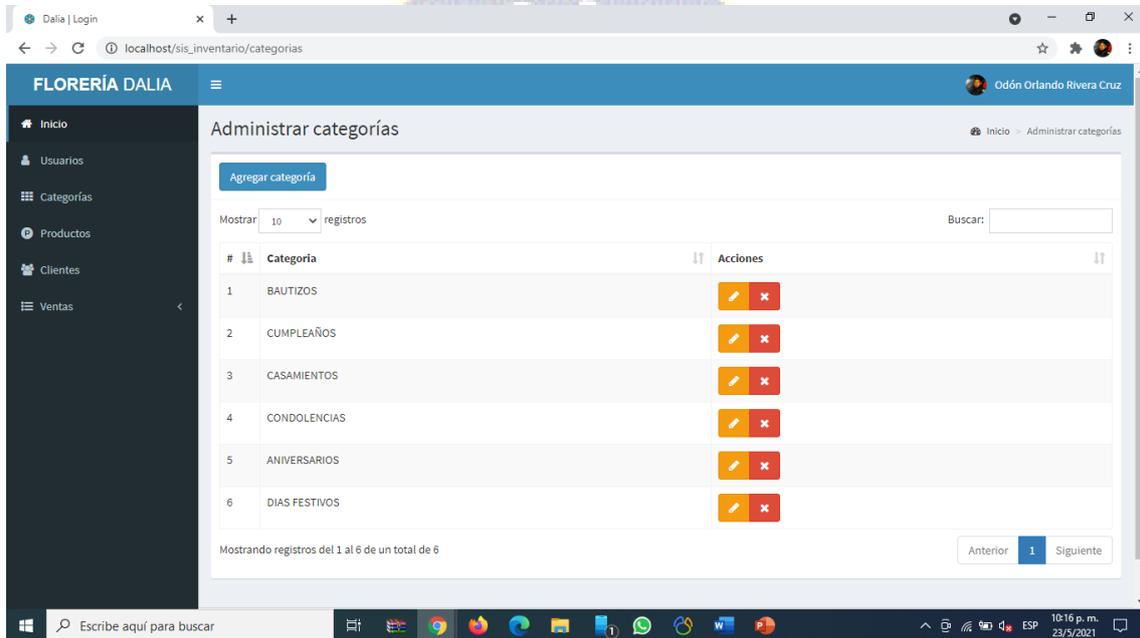


Figura 3.17.: Pantalla Categorías
Fuente: (Elaboración Propia, 2020)

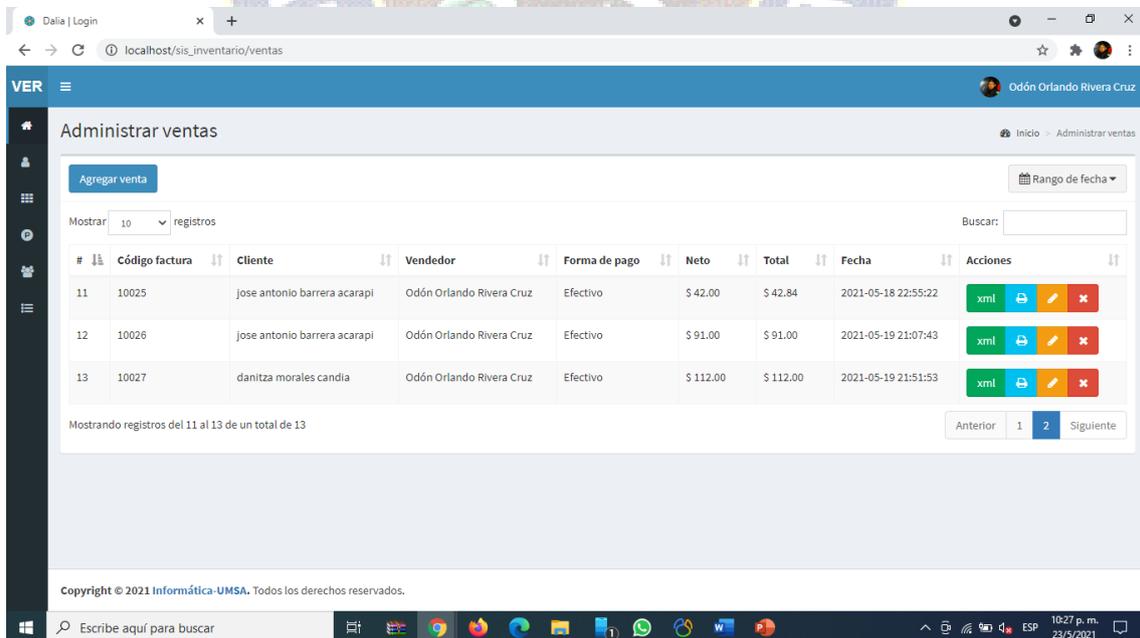


Figura 3.18.: Pantalla Registro de Pedidos y Envíos
Fuente: (Elaboración Propia, 2020)

3.4.2.6. PRUEBA UNITARIA DEL SPRINT 2

La prueba unitaria a continuación, sirve para comprobar que el módulo funcione correctamente.

Prueba No. 2 MÓDULO DE REGISTRO DE PEDIDOS Y ENVÍOS	
Descripción	El Cliente seleccionará el pedido de su preferencia, el Sistema guardará todos los datos seleccionados por el Cliente y finalmente pasará a la cola de envíos.
Objetivos	Que ingrese al Sistema y realice el pedido, posteriormente se despliegue el envío.
Condiciones	Conexión a internet desde cualquier medio (PC, Celular, iPhone, Tablet).
Resultado Esperado	Que se habrá el Sistema, sin ningún problema.
Resultado Obtenido	El Sistema inicio correctamente.

Tabla 3.10.: Prueba unitario del Sprint 2

Fuente: (Propia, 2020)

3.4.3. DESARROLLO DEL TERCER SPRINT, REPORTES

Este desarrollo tuvo reuniones acertadas ya que conocimos más de la producción y pedidos de último momento por parte del Cliente, estos cambios finales fueron programados y realizados acorde a los tiempos ya definidos anteriormente, así la culminación del Sprint de Reportes, no tuvo contratiempos.

3.4.3.1. PLANIFICACIÓN DEL SPRINT 3 REPORTES

La planificación del módulo 3 dio lugar a un listado general de informes que ponían fin a su transparencia y comprobante del arreglo floral, ya que estaban detalladamente descritos por el Sistema de la Florería. A continuación, en la siguiente tabla se detalla cómo se dio lugar al armado del módulo 3.

SPRINT 3: MÓDULO REPORTES			
Sprint	Duración en horas		Días de trabajo
3	52		10 días hábiles
SPRINT BACKLOG			
Tareas	Tipo	Programación en horas	Estado
Detalle de informes diarios y mensuales.	Desarrollo	9	Hecho
Listado de todos los pedidos y envíos realizados diariamente con lugar y fecha	Desarrollo	9	Hecho
Registro de pedidos con alta demanda según fechas de alta demanda	Desarrollo	7	Hecho
Detalle básico del monto de dinero generado diariamente y mensualmente	Desarrollo	8	Hecho
Diseño de la interfaz gráfica (Reporte Generado)	Desarrollo	9	Hecho
Seguridad y control de acceso al sistema	Desarrollo	10	Hecho

Tabla 3.11.: Tercer Sprint

Fuente: (Propia, 2020)

3.4.3.2. ESPECIFICACIÓN DE CASO DE USO DEL SPRINT 3

A continuación, se detalla el caso de uso del tercer Sprint.

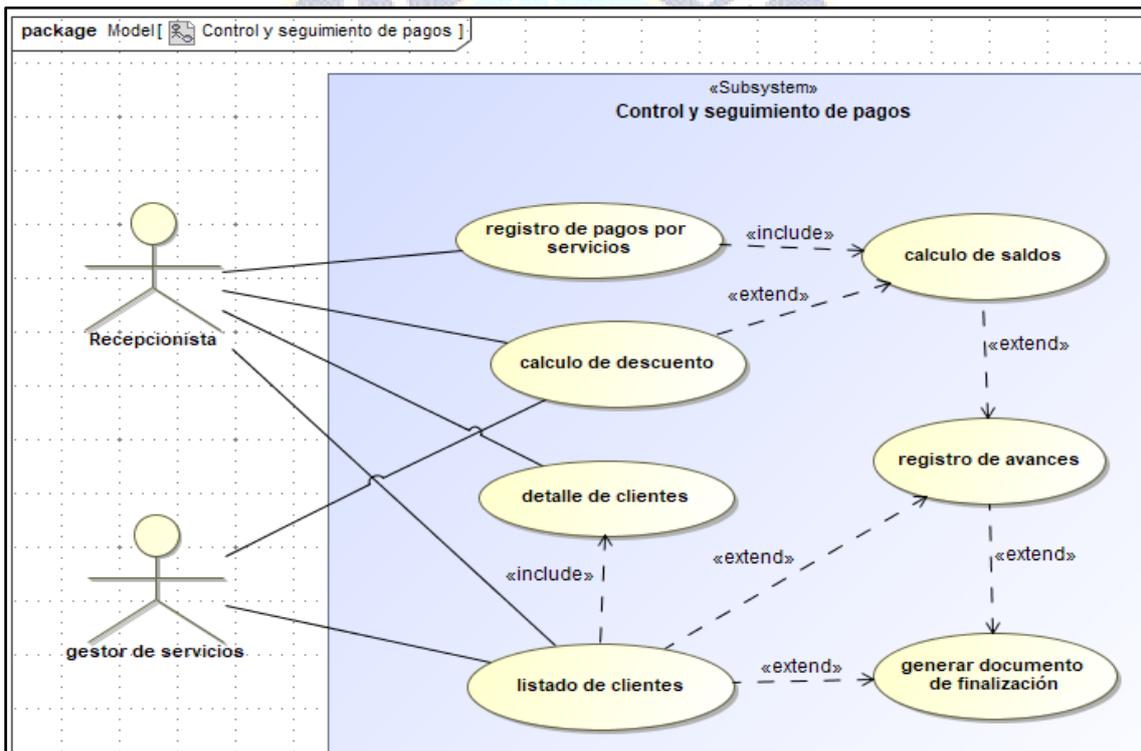


Figura 3.19.: Caso de uso Reportes

Fuente: (Elaboración Propia, 2020)

El tercer Sprint muestra una interfaz de Reportes en este caso en el momento de hacer una consulta al Sistema.

NOMBRE	Reportes
ACTORES	Administrador
PROPÓSITO	Interacción con el Sistema
RESUMEN	Elaboración de Reportes diarios y mensuales.

REFERENCIAS CRUZADAS	NINGUNA	
EVENTO ACTOR	LO QUE HACE EN EL	EVENTO SISTEMA
El Administrador ingresa al sistema para generar el reporte diario de todos los Pedidos y Envíos que realizaron los Clientes.	→	El sistema valida los datos ingresados por el Administrador e inmediatamente manda el reporte del día, así mismo despliega un mensaje, puede imprimirse.
El Administrador ingresa al sistema para generar el reporte mensual de todos los Pedidos y Envíos que realizaron los Clientes.	→	El sistema valida los datos ingresados por el Administrador e inmediatamente manda el reporte del día, así mismo despliega un mensaje, puede imprimirse.
FLUJO PRINCIPAL		
PRECONDICIÓN	Debe de existir la Base de Datos con la tabla correspondiente de reportes.	
POS CONDICIÓN	Despliegue de Reportes.	
PRESUNCIÓN	Almacenamiento de datos de forma segura	

Tabla 3.12.: Reportes

Fuente: (Propia, 2020)

3.4.3.3. DISEÑO NAVEGACIONAL

El diseño navegacional muestra las opciones de navegación y procesos que el Sistema tendrá en su desarrollo.



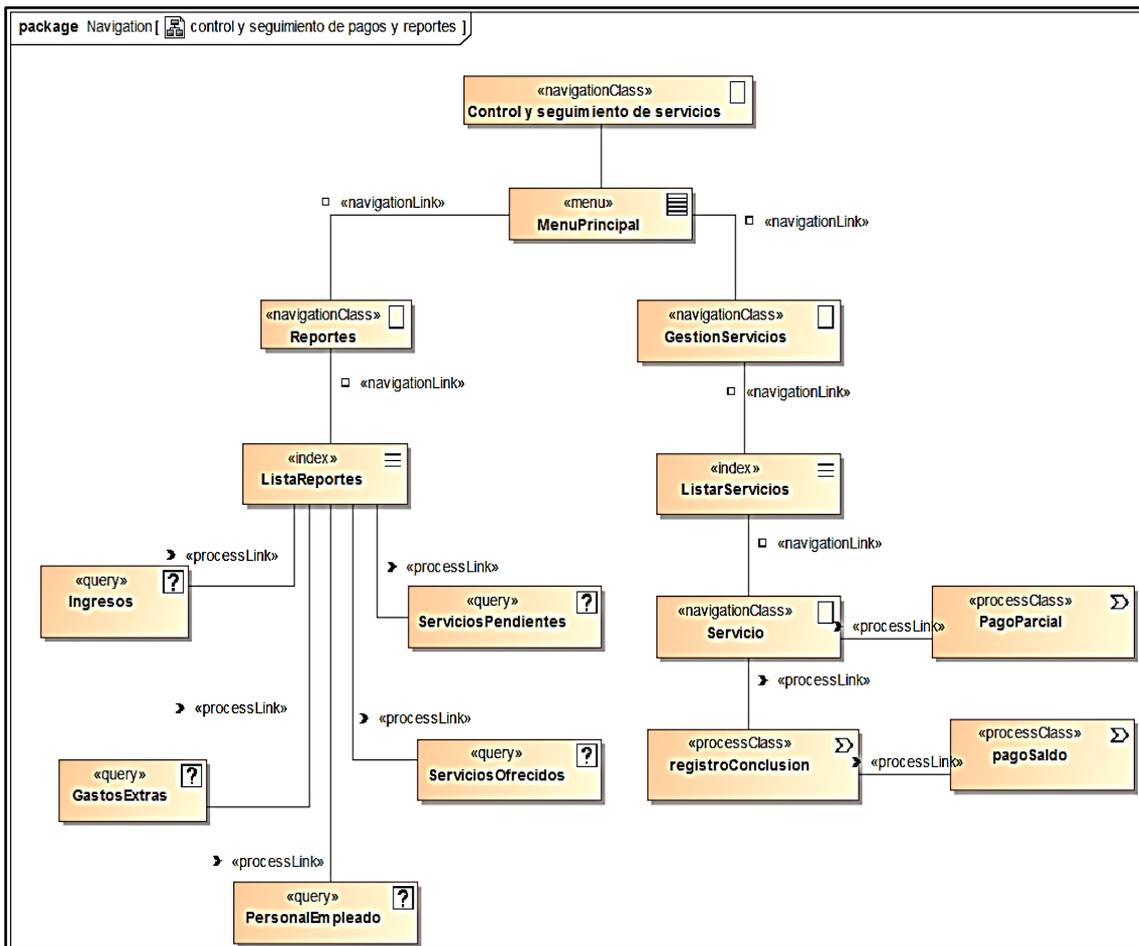


Figura 3.20.: Diseño Navegacional Reportes

Fuente: (Elaboración Propia, 2020)

3.4.3.4. DISEÑO DE PRESENTACIÓN

En el diagrama de presentación para el módulo de Reportes se muestra la interfaz donde el Administrador introduce los datos respectivos para ingresar al Sistema.

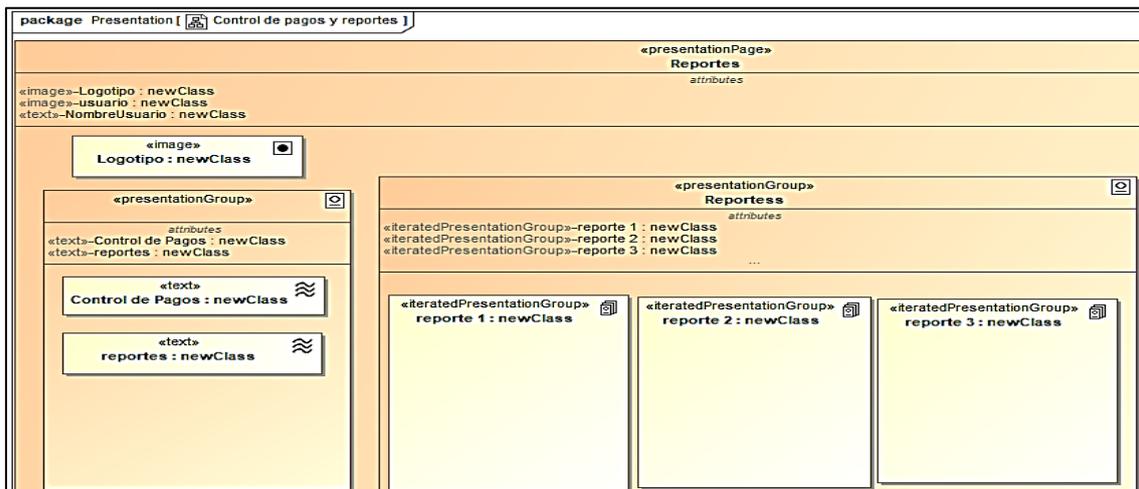


Figura 3.21.: Diseño de Presentación Reportes

Fuente: (Elaboración Propia, 2020)

```

class ControladorVentas{
    static public function ctrMostrarVentas($item, $valor){
        $tabla = "ventas";
        $respuesta = ModeloVentas::mdlMostrarVentas($tabla,
        return $respuesta;
    }
}

```

Figura 3.22.: Código php para mostrar las ventas

Fuente: (Elaboración Propia, 2020)

3.4.3.5. PANTALLA DEL SPRINT 3

A continuación, se puede observar la página de Reportes a la cual tendrá acceso el administrador para ver el todo el movimiento realizado en el día o el en mes.

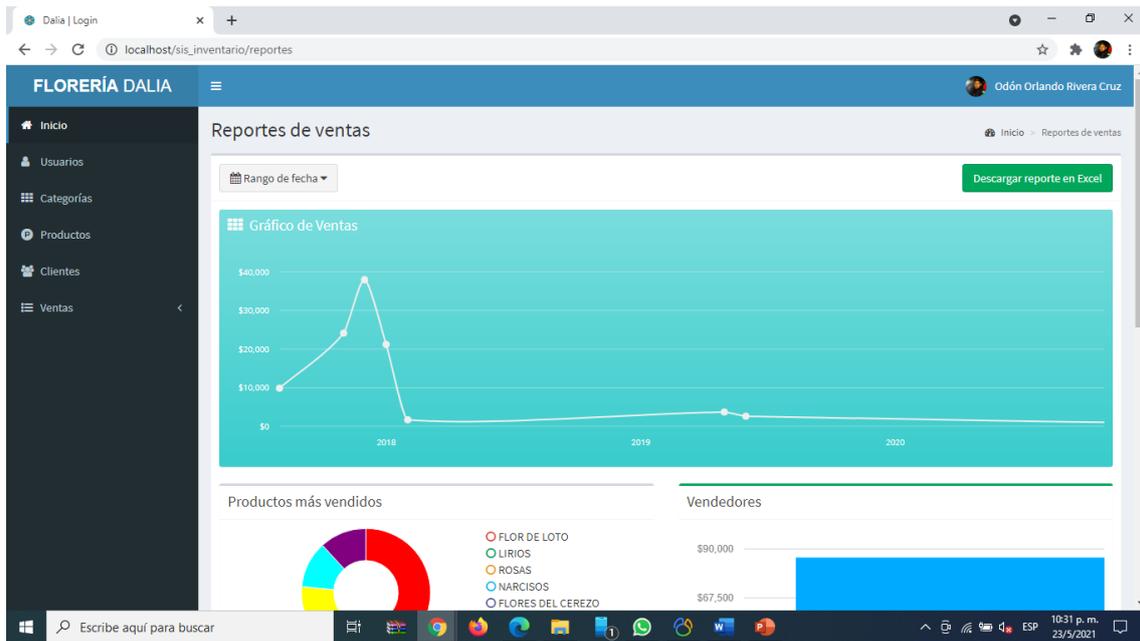


Figura 3.23.: Pantalla Reportes, grafico de ventas
Fuente: (Elaboración Propia, 2020)

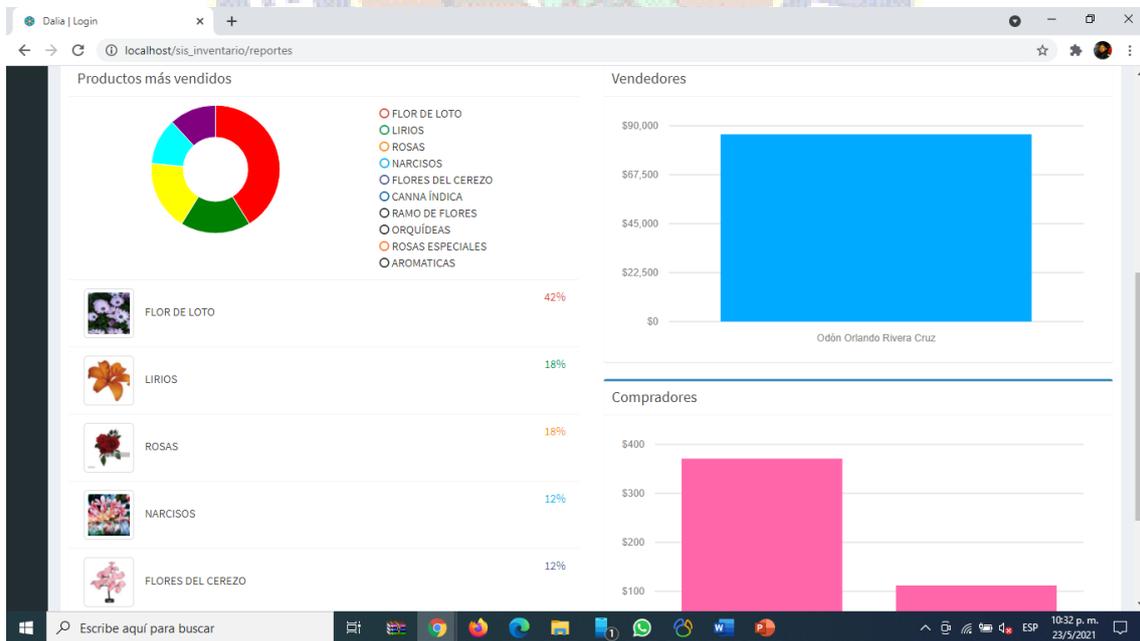


Figura 3.24.: Pantalla Gráficos, productos más vendidos y clientes potenciales
Fuente: (Elaboración Propia, 2020)

3.4.3.6. PRUEBA UNITARIA DEL SPRINT 3

La prueba unitaria a continuación, sirve para comprobar que el módulo funcione correctamente.

Prueba No. 3 REPORTES	
Descripción	El Administrador hará click en, “servicios realizados hoy” y el Sistema desplegará la información requerida.
Objetivos	Que ingrese al Sistema para ver los Reportes al hacer click, exitosamente.
Condiciones	Conexión a internet desde cualquier medio (PC, Celular, iPhone, Tablet).
Resultado Esperado	Que se habrá el Sistema, sin ningún problema.
Resultado Obtenido	El Sistema inicio correctamente.

Tabla 3.13.: Prueba Unitaria del Sprint 3

Fuente: (Propia, 2020)

3.5. DIAGRAMA ENTIDAD RELACIÓN

En el diagrama entidad relación se toma en cuenta las tablas y relaciones para hacer una representación de lo que será la base de datos.

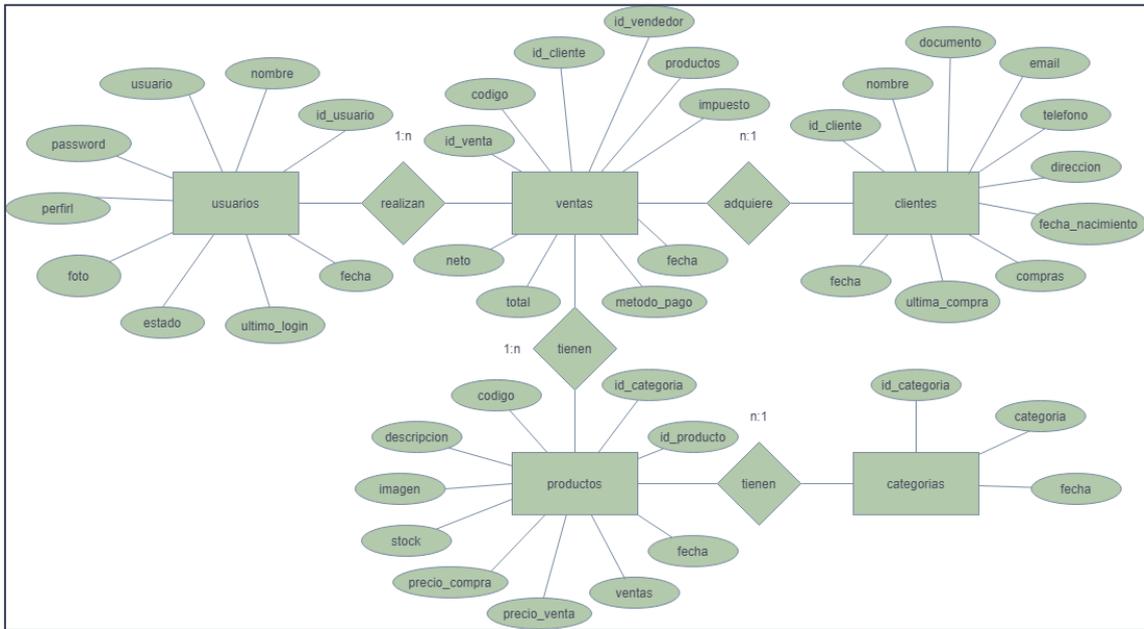


Figura 3.25.: Diagrama entidad relación

Fuente: (Elaboración Propia, 2020)

3.6. MODELO FÍSICO

El modelo físico muestra el diseño de la base de datos basándose en el modelo entidad relación.

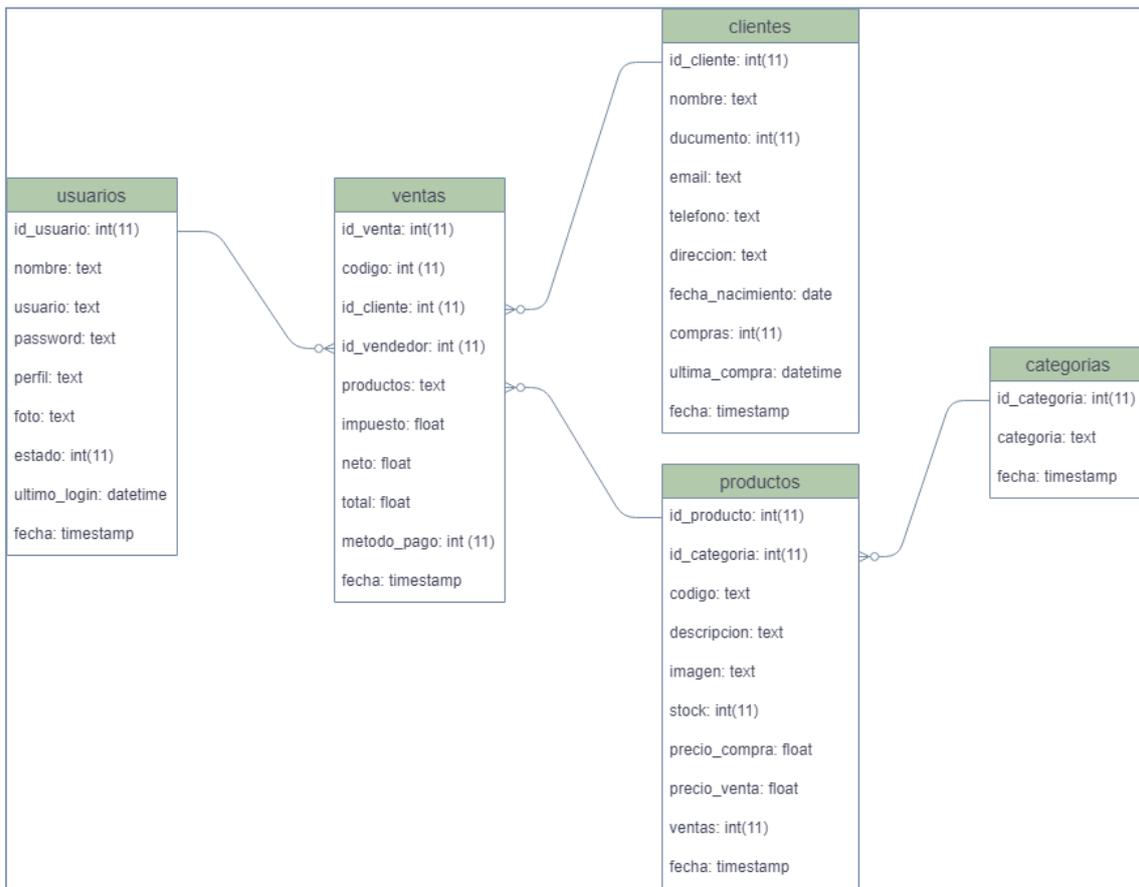


Figura 3.26.: Modelo físico
Fuente: (Elaboración Propia, 2020)

4.1. INTRODUCCIÓN

En el desarrollo de este capítulo se determina la calidad del sistema web en base a los parámetros de medición de la norma ISO 9126 y aplicando Web QEM el cual es una metodología cuantitativa que permite determinar un factor como calidad (o costo) de sitios y aplicaciones Web, donde hallaremos el punto función que nos servirá para el capítulo de costo beneficio, y las medidas de seguridad que se deben adoptar del lado del cliente y lado del servidor.

4.2. CALIDAD DE SOFTWARE

La ingeniería de software se diferencia de otras áreas, al no estar basada en leyes cuantitativas básicas, en su lugar se realiza un conjunto de medidas conocidas como métricas, las cuales proporcionan una referencia de la calidad algún producto de software (Pressman, 2002).

Para valorar la calidad de los productos de software o sistemas que se desarrollan se proporcionan información adecuada sobre los datos referentes de la misma a la calidad del producto, permitiendo una visión más profunda sobre el cumplimiento de los objetivos del proyecto (Pressman, 2002).

Medir la calidad de un software determina una de las tareas más complicadas que se presenta en el desarrollo de un sistema. Pero gracias a esta necesidad se fueron creando diferentes formas de medición de las mismas. Para el presente proyecto implementado utilizaremos el modelo de calidad de la ISO 9126, donde se medirá aspectos como la funcionalidad, fiabilidad, usabilidad y mantenibilidad (Pressman, 2002).

4.3. NORMA ISO 9126

La norma ISO 9126 (*International Standard Organization – Organización Internacional de Normalización*) es un estándar internacional para la evaluación de software, que nos ayudará a medir la calidad del sistema siguiendo los correspondientes criterios.

4.4. METODOLOGÍA WEB QEM

El principal objetivo de esta metodología cuantitativa consiste en evaluar y determinar el nivel de cumplimiento de las características especificadas para lo cual se analizan las preferencias elementales, parciales y globales. El resultado del proceso de evaluación (y eventualmente de comparación) puede ser interpretado como el grado de requerimientos de calidad satisfechos.

La metodología comprende una serie de fases y actividades que los evaluadores deben llevar a cabo en el proceso; entre las que podemos citar las siguientes actividades técnicas:

4.4. FASES DE WEB QEM

4.4.1.1. DEFINICIÓN DE LAS METAS DE EVALUACIÓN Y SELECCIÓN DEL PERFIL DE USUARIO

Los evaluadores deben definir metas y establecer alcances del proyecto de evaluación web. La evaluación se lleva a cabo tanto en la fase de desarrollo como en la fase operativa de un proyecto web, y así se puede valorar la calidad de un producto completo o bien se puede valorar la calidad de un conjunto de características y atributos de un componente. Los resultados podrán ser utilizados para comprender, mejorar, controlar o predecir la calidad de los productos.

Para propósitos de evaluación de dominios web, hemos considerados tres perfiles de usuario a un alto nivel de abstracción, a saber: visitantes, desarrolladores, y gerenciados.

4.4.1.2. DEFINICIÓN DE LOS REQUERIMIENTOS DE CALIDAD Y/O COSTO

Los evaluadores deben especificar atributos y características de calidad que estarán presentes en el proceso, agrupándolos en un árbol de requerimientos. Toma en cuenta las características de ISO.

- Metas de evaluación
- Perfil de usuario (gerente, desarrollador, visitante)
- Definición de requisitos no-funcionales

4.4.1.3. DEFINICIÓN DE CRITERIOS BÁSICOS Y PROCEDIMIENTOS DE MEDICIÓN

Los evaluadores deben definir una base una base de criterios para evaluación elemental, y realizar el proceso de medición y puntaje elemental.

Un criterio de evaluación elemental clara y específica de cómo medir atributos cuantificables. El resultado final es una preferencia o indicador elemental, el cual puede ser interpretado como el grado o resultado del requerimiento.

Por lo tanto, para cada métrica de un atributo necesitamos establecer un rango de valores aceptables y definir la función de criterio elemental, que producirá una correspondencia entre el valor de la métrica con el nuevo valor que representa la preferencia elemental.

4.4.1.4. DEFINICIÓN DE ESTRUCTURAS DE AGREGACIÓN E IMPLEMENTACIÓN DE LA EVALUACIÓN GLOBAL

En los pasos anteriores se definieron preferencias de calidad elemental realizados en el árbol de requerimientos. Por lo tanto, aplicando un mecanismo de agregación paso a paso, las preferencias elementales se pueden agrupar convenientemente para producir al final un esquema de agregación.

4.4.1.5. ANÁLISIS DE RESULTADOS Y RECOMENDACIONES

Una vez diseñado e implementado el proyecto de evaluación, el proceso culmina con la documentación de las conclusiones y recomendaciones. Los evaluadores analizan los resultados considerando las metas y el perfil de usuario establecidos.

A continuación de muestra las fases de WebQEM:

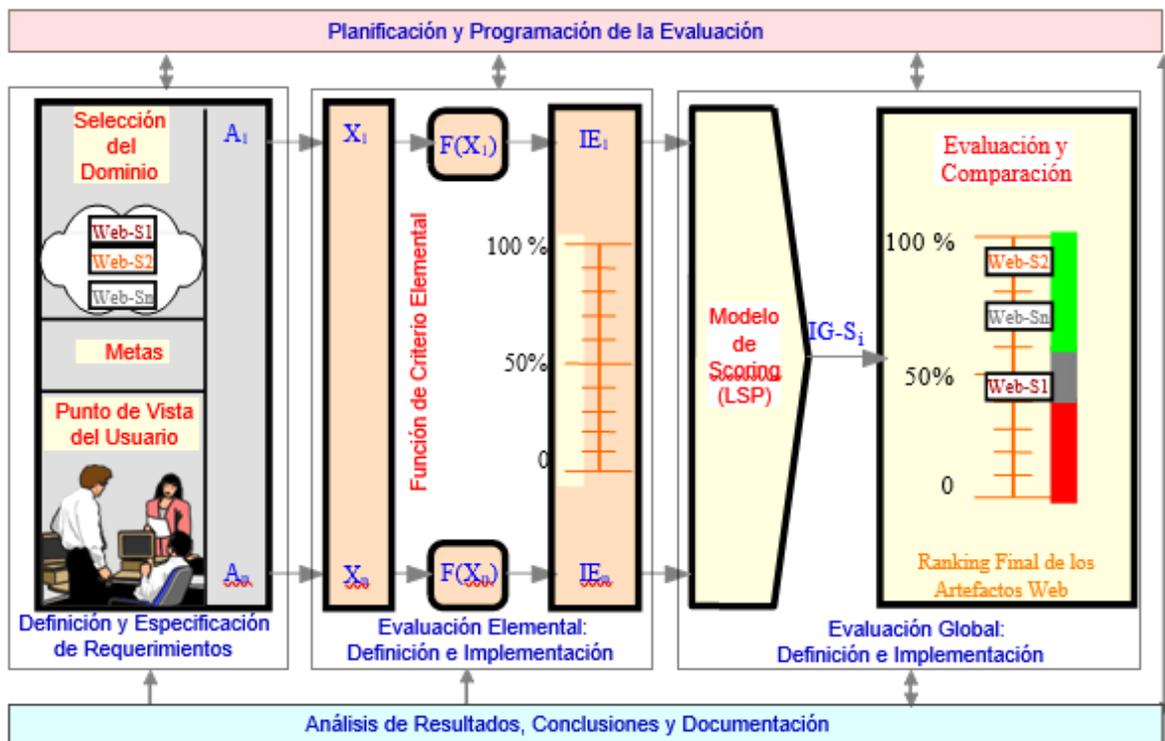


Figura 4.1.: Fases de WebQEM

Fuente: (OLSINA, 2004)

4.4.2. DESARROLLO DE CARACTERÍSTICAS DE WEB QEM

WebQem toma las métricas del modelo de calidad ISO 9126-1 la cual da referencia a las siguientes características:

4.4.2.1. FUNCIONALIDAD

Métrica para obtener una valoración mediante el cálculo del punto función en base a la evaluación de un conjunto de características y capacidades que debe cumplir el sistema:

FACTORES DE COMPLEJIDAD	Sin influencia	Incidental	Moderado	Medio	Significativo	Esencial	Fi
	0	1	2	3	4	5	

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

Tabla 4.1.: Factor de ajuste de complejidad.

Fuente: (Elaboración propia, 2020)

La funcionalidad es medida a través del punto función (PF), que proporciona una medida objetiva, cuantitativa y auditable del tamaño de la aplicación, basada en la visión del usuario de la aplicación (Pressman, 2002).

Para calcular el punto función se utiliza la siguiente relación:

$$PF = Cuenta\ Total * (X + Min(Y) * \sum Fi)$$

Dónde:

PF: Medida de funcionalidad

Cuenta Total: Es la suma de los siguientes datos (Nro. de entradas, Nro. de salidas, Nro. de peticiones, Nro. de archivos, Nro. de interfaces externas).

X: Confiabilidad del proyecto, varía entre 1 a 100%.

Min (Y): Error mínimo aceptable al de la complejidad.

$\sum Fi$: Son los valores de ajuste de complejidad, donde ($1 \leq i \leq 14$).

Para calcular el PF se usa la siguiente ecuación:

$$PF = \text{cuenta total} * (X + \text{Min} (Y) * \sum Fi)$$

$$PF = \text{cuenta total} * [0.65 + (0.01 * \sum Fi)]$$

Reemplazando los valores obtenidos en la tabla 31, se obtiene el siguiente resultado:

$$PF = 154.98$$

A continuación, calculamos el PF ideal:

$$PF_{ideal} = 126 * [0.65 + (0.01 * 70)]$$

$$PF_{ideal} = 170.1$$

Entonces la funcionalidad del sistema es:

$$Funcionalidad = 91.1\%$$

Con el resultado obtenido se puede interpretar que **9 de cada 10 personas**, consideran que el sistema responde de manera óptima a las funcionalidades requeridas por la empresa.

4.4.2.2. CONFIABILIDAD

Para determinar la confiabilidad de un sistema, se toma en cuenta las fallas que puedan ocurrir en el sistema en un tiempo determinado. En el desarrollo de software las fallas son más que todo por diseño e implementación. Para medir el tiempo medio entre fallos (TMEF) se usará la siguiente fórmula:

$$\text{TMEF} = \text{TMDF} + \text{TMDR}$$

Donde:

TMDF: Tiempo medio de fallo.

TMDR: Tiempo medio de reparación.

Se estima que un fallo puede ocurrir cada 20 días hábiles y su reparación en promedio pueda tomar 1 hora después de haber entregado una nueva funcionalidad del sistema, entonces:

$$\text{TMEF} = (20 * 8) + 1$$

$$\text{TMEF} = 161 \text{ horas}$$

Por lo que la disponibilidad es un buen indicador de fiabilidad, en base de la siguiente fórmula se tiene:

$$\text{Disponibilidad} = 99,4\%$$

Con lo que se llega a la conclusión de que el sistema tiene un **99.4% de confiabilidad.**

4.4.2.3. USABILIDAD

La usabilidad representa facilidad de uso que el usuario final percibirá del sistema. Esta métrica nos muestra el esfuerzo necesario para aprender a manipular el sistema (Pressman, 2002).

La tabla 4.2., muestra los resultados obtenidos en base a preguntas propuestas a los usuarios del sistema.

Nro.	Pregunta	Valor 0 - 100
1	¿Es entendible?	91
2	¿Las pantallas son agradables a la vista del usuario?	95
3	¿Es fácil de aprender?	92
4	¿Contiene información necesaria?	90
5	¿Facilita su trabajo?	86
6	¿La navegabilidad es fluida?	94
Promedio		91.3

Tabla 4.2.: Preguntas para obtener el grado de usabilidad

Fuente: (Elaboración propia, 2020)

Entonces la **usabilidad del sistema seria del 91.3%**, lo que indica que 9 de cada 10 usuarios pueden utilizar el sistema con facilidad.

4.4.2.4. MANTENIBILIDAD

Para la evaluación de la mantenibilidad, se desarrolló algunas preguntas, estas preguntas son valoradas en porcentaje por el desarrollador del sistema al momento de la culminación del proyecto. Este valor tiene consideración por la experiencia y la forma de trabajo de cada programador, el mismo puede ser relativo respecto a otros desarrolladores (Largo y Marín, 2005).

La tabla 4.3. muestra las preguntas y los resultados obtenidos en la evaluación de mantenibilidad, estas preguntas se las hizo a todo el equipo de desarrollo del sistema.

FACTOR DE AJUSTE	VALOR
¿Se puede modificar el sistema?	97%
¿Deja identificar las partes que deben ser modificadas?	95%
¿Permite implementar una modificación específica?	93%

¿Presenta efectos inesperados como posibles errores?	98%
Total	95.75%

Tabla 4.3.: Evaluación de mantenibilidad

Fuente: (Elaboración propia, 2020)

El resultado de la **mantenibilidad es de 95.75%**, lo que significa que el esfuerzo necesario para realizar mantenimiento al sistema es mínimo.

4.4.2.5. PORTABILIDAD

La portabilidad es la capacidad con que un software puede ser llevado de un entorno a otro, considera la facilidad de instalación, ajuste y adaptación al cambio. Para medir la portabilidad del sistema usaremos la siguiente relación:

Si:

- $GP > 0$, la portabilidad es más rentable que el re-desarrollo.
- $GP < 0$, el re-desarrollo es más rentable que la portabilidad.
- $GP = 0$, la portabilidad es perfecta.

Los recursos necesarios para llevar el sistema a otro entorno son: servicio de hosting para alojar el código fuente, la base de datos, dominio para la URL, conexión FTP, conexión a internet, conexión intranet, responsivo, espacio almacenamiento, y los recursos necesarios para crear el sistema son: IDE de desarrollo, html5 y frameworks (codeigniter v3, jquery, bootstrap 3, css3,).

Por lo que los valores obtenidos son:

$$ET = 9 \text{ y } ER = 10$$

$$GP = 0.90$$

Con el resultado obtenido sabemos que el grado de portabilidad es del 90%, entonces la portabilidad del sistema es más rentable que en el caso se desee volver a desarrollar.

4.4.2.6. CALIDAD GLOBAL

Para poder obtener la calidad global del sistema, se saca la media de todas las medidas expresadas en porcentaje hasta el momento, funcionalidad, confiabilidad, usabilidad, mantenibilidad y portabilidad (Pressman, 2002).

CRITERIOS	RESULTADO
Funcionalidad	91.1%
Confiabilidad	99.4%
Usabilidad	91.3%
Mantenibilidad	95.75%
Portabilidad	90%
Calidad Global	93.51%

Tabla 4.4.: Calidad global
Fuente: (Elaboración propia, 2020)

El resultado de la **calidad global es de 93.51%**, con ese resultado concluimos que 9 de cada 10 usuarios consideran al sistema web de calidad.

4.5. SEGURIDAD

En este apartado se explica el uso del algoritmo de encriptación (o cifrado) sha-2 “secure Hash Algorithm” como medio para cifrar la información del cliente, como característica de seguridad en la base de datos, además se explica el uso de los SSL “Secure Sockets Layer” como certificado de seguridad, como característica de seguridad en el servidor web.

4.5.1. ALGORITMO DE ENCRIPCIÓN SHA-2

SHA-2 es un conjunto de funciones hash criptográficas (SHA-224, SHA-256, SHA-384, SHA-512) diseñadas por la Agencia de Seguridad Nacional (NSA) y publicada en 2001 por el Instituto Nacional de Estándares y Tecnología (NIST) como un Estándar Federal de Procesamiento de la Información (FIPS).

Una función hash es un algoritmo que transforma ("digiere") un conjunto arbitrario de elementos de datos, como puede ser un fichero de texto, en un único valor de longitud fija (el "hash"). El valor hash calculado puede ser utilizado para la verificación de la integridad de copias de un dato original sin la necesidad de proveer el dato original. Esta irreversibilidad significa que un valor hash puede ser libremente distribuido o almacenado, ya que sólo se utiliza para fines de comparación. SHA significa algoritmo de hash seguro. SHA-2 incluye un significativo número de cambios respecto a su predecesor, SHA-1; y consiste en un conjunto de cuatro funciones hash de 224, 256, 384 o 512 bits.

4.5.2. VARIANTES DEL SHA-2

Valores hash para una cadena de caracteres vacía.

SHA224("")

0x d14a028c2a3a2bc9476102bb288234c415a2b01f828ea62ac5b3e42f

SHA256("")

0x e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855

Incluso un pequeño cambio en el mensaje (con una apabullante probabilidad) dará como resultado un hash completamente distinto, debido al efecto avalancha. Por ejemplo, añadiendo un punto al final de la frase:

SHA224("The quick brown fox jumps over the lazy dog")

0x 730e109bd7a8a32b1cb9d9a09aa2325d2430587ddbc0c38bad911525

SHA224("The quick brown fox jumps over the lazy dog.")

0x 619cba8e8e05826e9b8c519c0a5c68f4fb653e8a3d8aa04bb2c8cd4c

4.5.3. CAPA DE CONEXIÓN SEGURA SSL

SSL "Secure Sockets Layer" es un protocolo diseñado para permitir que las aplicaciones

transmitan información de manera segura. Las aplicaciones que utilizan el protocolo Secure Sockets Layer conocen el flujo de envío y recepción de claves cifradas con otras aplicaciones, así como la manera de cifrar y descifrar los datos enviados entre las mismas.

4.5.3.1 IMPLEMENTACIÓN DE CERTIFICADO SSL CON LET'S ENCRYPT Y CERBOT

Let's Encrypt es una autoridad de certificación (CA) automatizada y abierta operada por Internet Security Research Group (ISRG) y fundada por la Electronic Frontier Foundation (EFF), la Mozilla Foundation y otros.

Proporciona certificados SSL / TLS gratuitos que se utilizan comúnmente para cifrar las comunicaciones con fines de seguridad y privacidad, el caso de uso más notable es HTTPS.

Let's Encrypt se basa en el protocolo ACME (Entorno automático de gestión de certificados) para emitir, revocar y renovar certificados. Certbot es una utilidad gratuita y de código abierto que se utiliza principalmente para administrar certificados SSL / TLS de la autoridad de certificación Let's Encrypt. Está disponible para la mayoría de los sistemas operativos UNIX y similares a UNIX, incluidos GNU / Linux, FreeBSD, OpenBSD y OS X.

La mejor forma de utilizar Let's Encrypt sin acceso shell es usando el soporte incorporado del proveedor de hospedaje. Si el proveedor de hospedaje ofrece soporte para Let's Encrypt, es posible solicitar un certificado gratis en su nombre, instalarlo, y mantenerlo actualizado automáticamente.

Para algunos proveedores de hospedaje, esto es un ajuste de configuración que tienes que prender. Otros proveedores automáticamente solicitan e instalan certificados para todos sus clientes.

A continuación, se muestra un ejemplo de certificado activo.

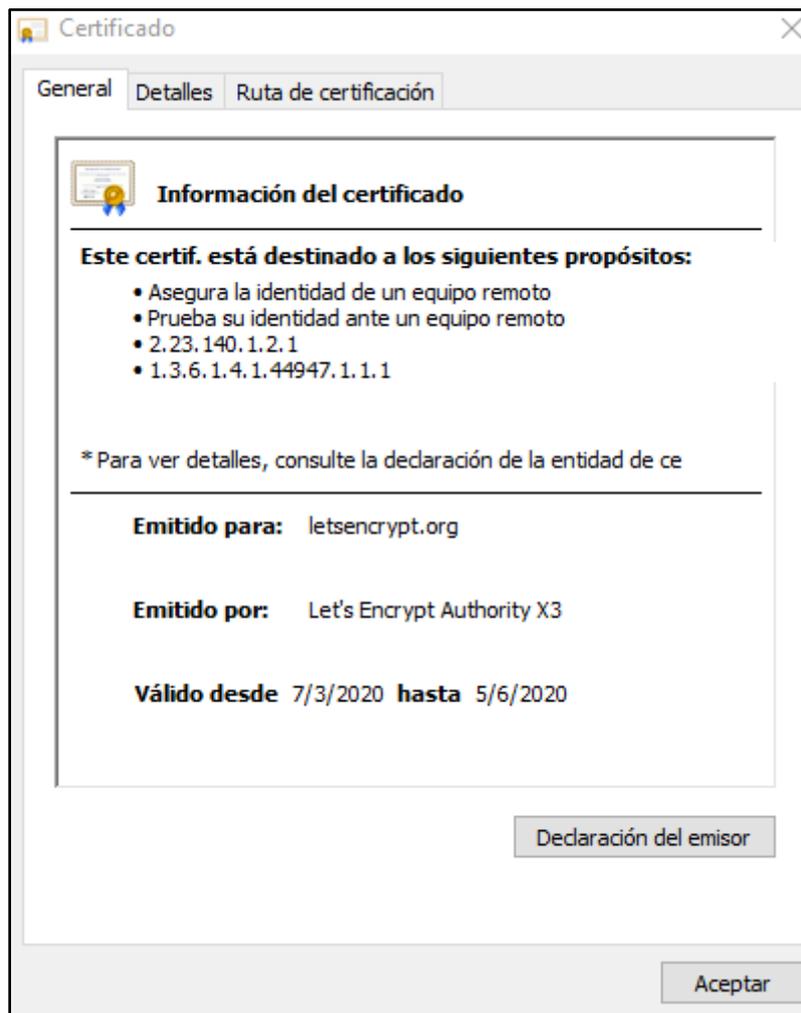


Figura 4.2.: Certificación *Let's Encrypt* activa
Fuente: (Elaboración propia, 2020)

5.1. INTRODUCCIÓN

En el desarrollo del presente capítulo, se dará a conocer a la Florería “Dalia”, que el desarrollo y la posterior implementación del sistema es sustentable y genera beneficios mediante la evaluación efectiva de la relación costo-beneficio, con el fin de tomar la mejor decisión.

Para tal efecto se realiza el análisis de costo del sistema usando el método COCOMO II y en base al costo del sistema y otros gastos se determina la rentabilidad del sistema con el cálculo del valor neto actual (VAN) y la tasa interna de retorno (TIR).

Posterior a los análisis necesarios para la obtención de los resultados esperados estaremos en la capacidad de afirmar si el proyecto es viable, redituable y comprobar que es buena opción invertir en el proyecto.

5.2. COCOMO II

El Modelo Constructivo de Costes (COCOMO) es un modelo matemático de base empírica, utilizando para la estimación de costes de software. Incluye tres submodelos, cada uno ofrece un nivel de detalle y aproximación, cada vez mayor, a medida que avanza el proceso de desarrollo del software:

a) **Básico**

estima el esfuerzo y el tiempo empleado en el desarrollo de un proyecto de software usando dos variables predictivas denominadas factores de costo (*costdrivers*): el tamaño del software y el modo de desarrollo.

b) **Intermedio**

Comparado con el modelo anterior, éste provee un nivel de detalle y precisión superior, el cual es más apropiado para la estimación de costos en etapas de

mayor especificación. COCOMO Intermedio incorpora un conjunto de quince variables de predicción que toman en cuenta las variaciones de costos no consideradas por COCOMO Básico.

c) Detallado:

El Modelo Detallado provee los medios para generar estimaciones con mayor grado de precisión y detalle. Difiere del Modelo Intermedio en dos aspectos principales que ayudan a superarlas limitaciones: jerarquía de niveles del producto y multiplicadores de esfuerzo.

COCOMO II es aquel que permite realizar estimaciones en función del tamaño del software, y de un conjunto de factores de costo y beneficio. Los factores de costo describen aspectos relacionados con la naturaleza del producto, hardware utilizado, personal involucrado, y características propias del proyecto. Cada uno de ellos orientados a sectores específicos del mercado de desarrollo de software y las distintas etapas del desarrollo de software.

Por otro lado, también se define tres modos de desarrollo o tipo de proyectos:

Modos de Desarrollo	A	b	c	d
Orgánico	3,2	1,05	2,5	0,38
Semi- acoplado	3	1,12	2,5	0,35
Empotrado	2,8	1,2	2,5	0,32

Tabla 5.1: Tabla de modos de desarrollo

Fuente: (COCOMO II, 2000)

a) **Orgánico**

En esta clasificación se encuentran proyectos desarrollados en un ambiente familiar y estable. El producto a elaborar es relativamente pequeño y requiere pocas innovaciones tecnológicas en lo que refiere a algoritmos, estructuras de datos e integración de hardware.

b) **Semi-acoplado**

Es un modelo para productos de software de tamaño y complejidad media. Las características de los proyectos se consideran intermedias a las de los modos Orgánico y Empotrado.

c) **Empotrado**

En esta clasificación están incluidos proyectos de gran envergadura que operan en un ambiente complejo con altas restricciones de hardware, software y procedimientos operacionales, tales como los sistemas de tráfico aéreo.

Estos métodos de estimación de costos se engloban en el grupo de modelos algorítmicos que tratan de establecer una relación matemática la cual permite estimar el esfuerzo y tiempo requerido para desarrollar un producto.

Para nuestro caso particular usaremos el modelo intermedio, pues se adecua a nuestro proyecto para lograr una buena precisión, además que se aplicara el modo de desarrollo semi-acoplado para proyecto de complejidad intermedia es decir con líneas de código menores a los 300 KDLC.

5.3. COSTO DEL SISTEMA

En el proceso de estimación se emplearán las siguientes formulas:

$$E = (a)(KLDC)^b * (FAE)$$

$$T = (c) * E^d$$

$$P = E/T$$

Siendo:

- ✓ E: Esfuerzo
- ✓ T: Tiempo
- ✓ P: Personal
- ✓ KLDC: Kilo líneas de código (Miles de líneas de código)
- ✓ FAE: Factor de Ajuste de Esfuerzo

- ✓ a, b, c, d: Dependientes del modo de desarrollo, según tabla 5.1
- ✓ PF: puntos de función

Hallando la variable KLDC:

Con PF = 261,36 y las líneas por cada PF equivalen a 29 para el lenguaje PHP, según la tabla 13

Lenguaje	LDC/PF
Ensamblador	320
C	150
COBOL	105
Pascal	91
PHP	29
C++	64
Visual Basic	32
SQL	12

Tabla 5.2.: Puntos fusión COCOMO II
Fuente: (COCOMO II, 2000)

Por tanto:

$$KLDC = (PF * \text{líneas de código por cada PF}) / 1000$$

$$KLDC = (261.36 * 29) / 1000$$

$$KLDC = 7.58$$

Y por otro lado también hemos de hallar la variable FAE, la cual se obtiene mediante la multiplicación de los valores evaluados en los diferentes 15 conductores de costos que se observan en la tabla 5.3.

CONDUCTORES DE COSTO	VALORACIÓN					
	<i>Muy bajo</i>	<i>Bajo</i>	<i>Nominal</i>	<i>Alto</i>	<i>Muy Alto</i>	<i>Extra alto</i>
Fiabilidad requerida del software	0,75	0,88	1.00	1,15	1,4	-
Tamaño de la base de datos	-	0,94	1.00	1,08	1,16	-
Complejidad del Producto	0,7	0,85	1.00	1,15	1,3	1,65
Restricciones del tiempo de ejecución	-	-	1.00	1,11	1,3	1,66
Restricciones del almacenamiento principal	-	-	1.00	1,06	1,21	1,56
Volatilidad de la máquina virtual		0,87	1.00	1,15	1,3	-
Tiempo de respuesta del Ordenador	-	0,87	1.00	1,07	1,15	-
Capacidad del analista	1,46	1,19	1.00	0,86	0,71	-
Experiencia en la aplicación	1,29	1,13	1.00	0,91	0,82	-
Capacidad de los programadores	1,42	1,17	1.00	0,86	0,7	-
Experiencia en S.O. Utilizado	1,21	1,1	1.00	0,9	-	-
Experiencia en el lenguaje de Programación	1,14	1,07	1.00	0,95	-	-
Prácticas de programación Modernas	1,24	1,1	1.00	0,91	0,82	-
Utilización de herramientas software	1,24	1,1	1.00	0,91	0,83	-
Limitaciones de planificación del proyecto	1,23	1,08	1.00	1,04	1,1	-

Tabla 5.3.: Tabla modo de conductores de costo

Fuente: (COCOMO II, 2000)

$$FAE = 1,15 * 1,00 * 0,85 * 1,11 * 1,00 * 1,07 * 0,86 * 0,82 * 0,70$$

$$* 1,00 * 0,91 * 1,08$$

$$FAE = 0,54$$

Cálculos del esfuerzo de desarrollo:

$$E = 3 * 7,58^{1,12} * 0,54$$

$$E = 14,74 \text{ personas/mes}$$

En este caso las personas se toman como a los desarrolladores que hicieron los esfuerzos por mes.

A continuación, se realiza el cálculo del tiempo de desarrollo:

$$T = 2,5 * (14,74)^{0,35}$$

$$T = 6,41 \text{ meses}$$

A continuación, se realiza el cálculo del personal promedio de desarrollo:

$$P = 14,74/6,41$$

$$P = 2,3 \text{ personas}$$

A continuación, se realiza el cálculo de productividad que desarrolla las líneas de código con el esfuerzo en este caso también personas se toman como desarrolladores.

$$PR = LDC/E, \text{ donde PR: productividad}$$

$$PR = 2314/14,74$$

$$PR = 156,99 \text{ LDC/personas mes}$$

La interpretación de este resultado da a conocer que es necesario un equipo de tres personas alrededor de seis a siete meses.

5.4. ESTIMACIÓN DEL COSTO

Se realiza la estimación de COCOMO II mediante su software. A continuación, se muestra la siguiente figura sobre COCOMO II:

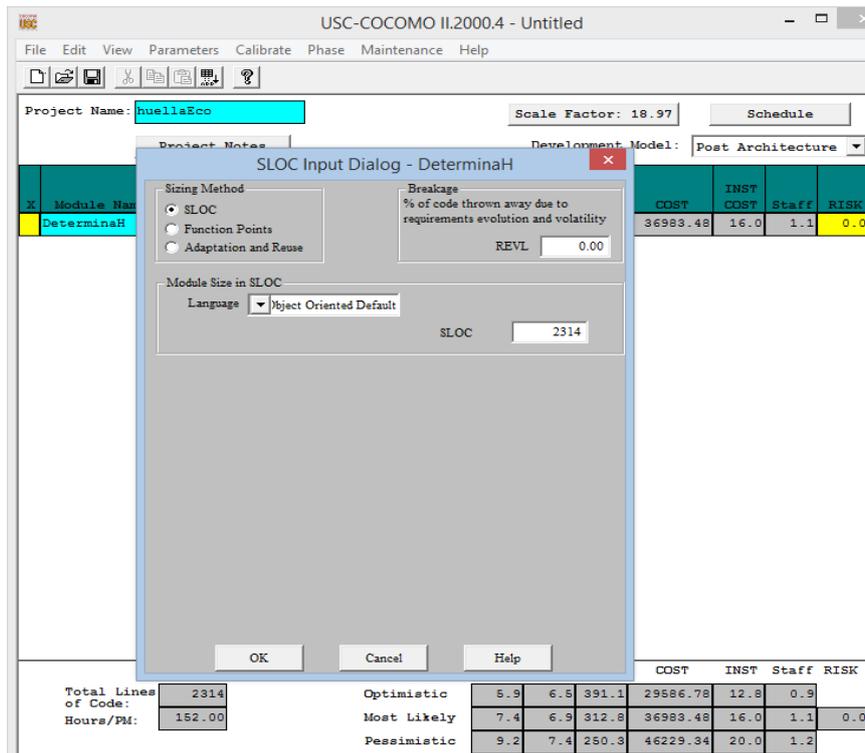


Figura 5.1.: Líneas de código software COCOMO

Fuente: (COCOMO II, 2000)

Una vez puesta las líneas de código nos sale unas tablas las cuales podemos modificar: module name (nombre del proyecto), module size (líneas de código), labor rate (Costo del producto), lo cual nos ayudará a definir la tabla mostrada en la parte superior de la figura. Que calcula: effort (esfuerzo), sched (tiempo), prod (productividad), cost (costo), Inst (costo intermedio), staff (cosas). Para lo otro nos define el estimado para ello existen tres estimaciones las cuales son: optimistic (óptimo), most likely (lo más probable) y pessimistic (pésimo).

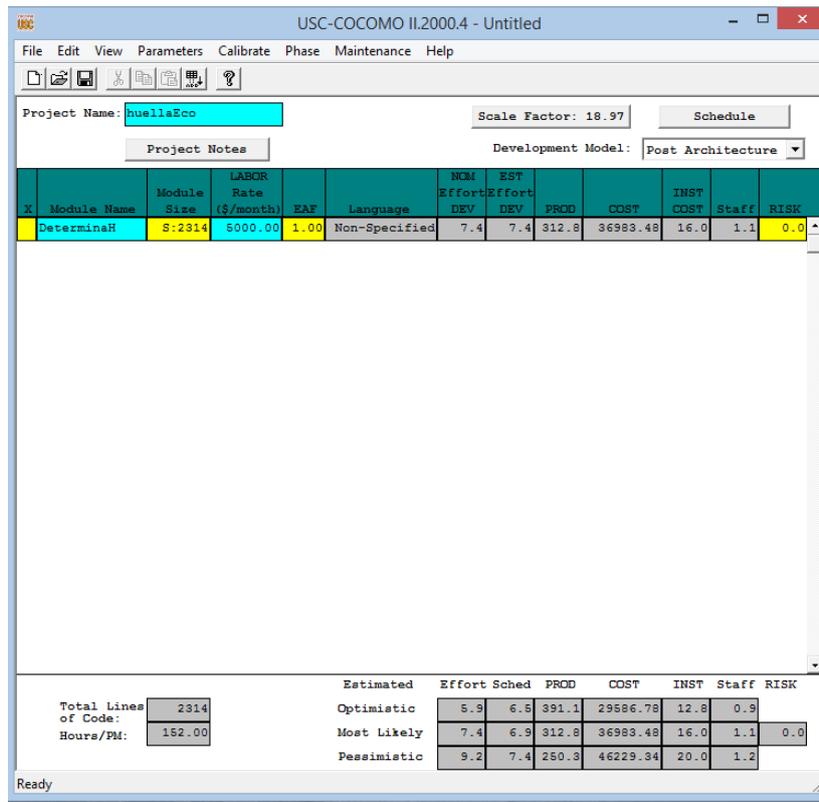
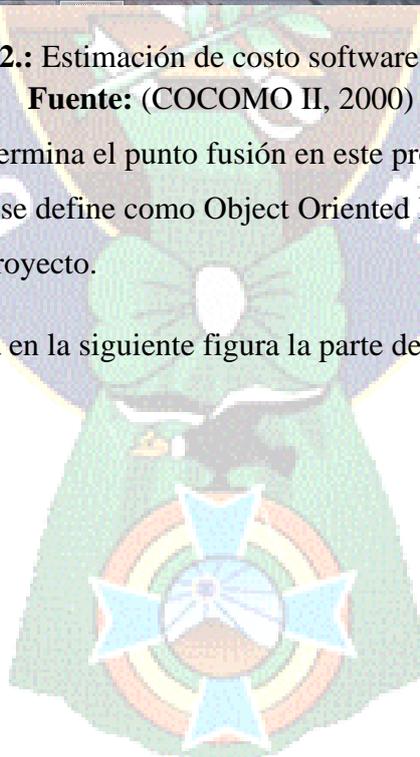


Figura 5.2.: Estimación de costo software COCOMO

Fuente: (COCOMO II, 2000)

Una vez sacado esto se determina el punto fusión en este proyecto se manejó el lenguaje PHP el cual en el software se define como Object Oriented Default y tenemos que llenar los demás datos según al proyecto.

A continuación, se muestra en la siguiente figura la parte de punto fusión.



SLOC Input Dialog - DeterminaH

Sizing Method:
 SLOC
 Function Points
 Adaptation and Reuse

Breakage
 % of code thrown away due to requirements evolution and volatility
 REVL: 0.00

Module Size in Function Points
 Language: Object Oriented Default Change Multiplier: 29

Ratio Type:
 Jones David

Calculation Method:
 Using Table Input Calculated Function Point

Function Type	# of Function Points			SubTotal
	Low	Average	High	
Inputs	0	1	0	4
Outputs	0	3	0	15
Files	0	1	0	10
Interfaces	0	3	0	21
Queries	0	4	0	16
Total Unadjusted Function Points				66
Equivalent Total in SLOC				1914

OK Cancel Help

Figura 5.3.: Punto fusión software COCOMO
Fuente: (COCOMO II, 2000)

Una vez puesto punto fusión volvemos al proyecto y vemos que algunos datos cambiaron en cuanto al costo. A continuación, se muestra la figura sobre estimación de costos.

USC-COCOMO II.2000.4 - Untitled

Project Name: **huelia&co** Scale Factor: 18.97 Schedule

Development Model: Post Architecture

X	Module Name	Module Size	LABOR Rate (\$/month)	KAF	Language	HEM Effort DEV	ESV Effort DEV	PREC	COST	INST COST	Staff	RISK
	DeterminaR	F:2320	5000.00	1.00	Object-Orient	7.4	7.4	\$12.8	\$7088.94	16.0	1.1	0.0

	Estimated	Effort Sched	PREC	COST	INST	Staff	RISK
Total Lines of Code:	2320						
Hours/PM:	152.00						
	Optimistic	6.9	6.5	\$91.0	29671.16	12.8	0.9
	Most Likely	7.4	6.9	\$12.8	\$7088.94	16.0	1.1
	Pessimistic	9.3	7.5	\$20.2	46361.18	20.0	1.2

PREC: Precedentedness

Figura 5.4.: Estimación de costo software COCOMO
Fuente: (COCOMO II, 2000)

Ya teniendo la estimación de costos se puede modificar la parte de *Scale Factor* (factores de escala). Esto depende del desarrollo del modelo en este caso se utiliza *post architecture* en los factores de escala se realiza los ajustes finos.

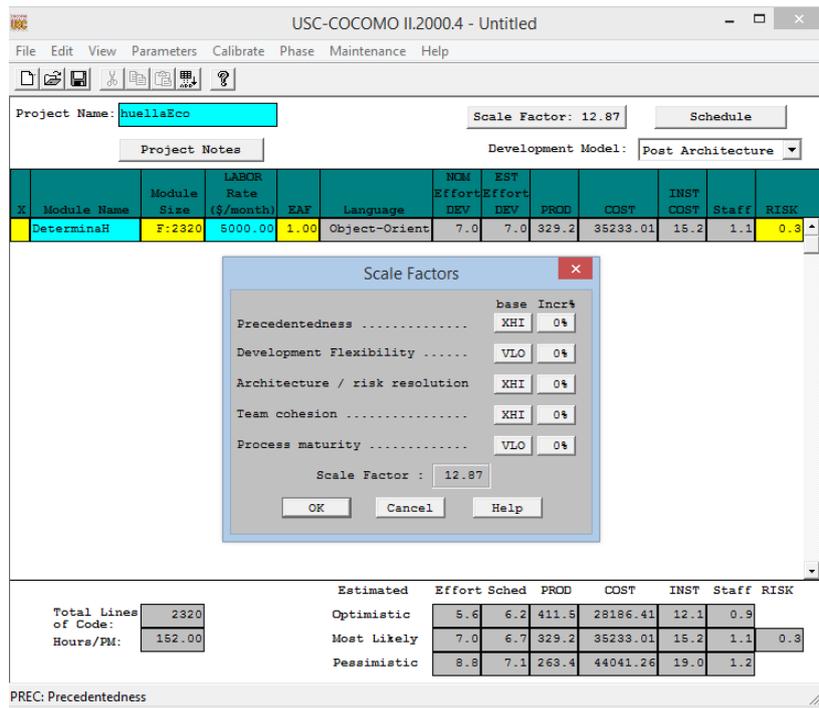


Figura 5.5.: Factores de escala software COCOMO

Fuente: (COCOMO II, 2000)

A continuación, se muestra una tabla que hace referencia al monto total.

MATERIAL	COSTO (USD)	COSTO (BOLIVIANOS)
Material de escritorio	30	208,8
Computadora	800	5568
Desarrollo del software	5000	34800
Internet	30	208,8
Servidor web	120	835,2
Mantenimiento	1000	6960

Tabla 5.4.: Costo total desarrollo de software

Fuente: (Elaboración propia,2020)

5.5. CÁLCULO BENEFICIO VAN Y TIR

El VAN y el TIR son dos herramientas financieras procedentes de las matemáticas financieras que nos permiten evaluar la rentabilidad de un proyecto de inversión,

entendiéndose por proyecto de inversión no solo como la creación de un nuevo negocio, sino también, como inversiones que podemos hacer en un negocio en marcha, tales como el desarrollo de un nuevo producto, la adquisición de nueva maquinaria, el ingreso en un nuevo rubro de negocio, etc.

5.5.1. VALOR ACTUAL NETO (VAN)

Es el valor actual de los beneficios netos que genera el proyecto durante toda su vida. El valor actual neto (VAN) es un método de valoración de inversiones que puede definirse como la diferencia entre el valor actualizado de los cobros y de los pagos generados por una inversión. Proporciona una medida de la rentabilidad del proyecto analizado en valor absoluto, es decir expresa la diferencia entre el valor actualizado de las unidades monetarias cobradas y pagadas.

A continuación, se muestra la tabla de la caja de flujo en años con una inversión de 48580.

AÑO	FLUJO DE CAJA (BS)
2016	6000
2017	9000
2018	10000
2019	15000
2020	30000

Tabla 5.5.: Flujo de caja por años
Fuente: (Elaboración propia,2020)

formula:

$$VAN = -A + \sum_{i=1}^n \frac{Q_i}{(1 + K)^i}$$

Donde:

Q₁, Q₂,..., Q_n: Flujos de caja

k: Tasa de descuento seleccionada

A: desembolso inicial o inversión inicial n: vida útil del proyecto

i: período

Desarrollando:

$$\begin{aligned}VAN &= -A + \frac{Q_1}{(1+K)^1} + \frac{Q_2}{(1+K)^2} + \dots + \frac{Q_n}{(1+K)^n} \\VAN &= -48580 + \frac{6000}{(1+0,1)^1} + \frac{9000}{(1+0,1)^2} + \frac{10000}{(1+0,1)^3} + \frac{15000}{(1+0,1)^4} \\&\quad + \frac{30000}{(1+0,1)^5} \\VAN &= -48580 + 5454,54 + 7438,02 + 7513,15 + 10245,20 \\&\quad + 18627,64 \\VAN &= 698,55\end{aligned}$$

Interpretación VAN:

VAN > 0; se recomienda pasar a la siguiente etapa del proyecto

VAN = 0; es indiferente realizar la inversión

VAN < 0; se recomienda desecharlo o postergarlo

Como el VAN tiene un resultado mayor a cero, entonces el proyecto es aceptable.

5.5.2. TASA INTERNA DE RETORNO (TIR)

Es una tasa porcentual que indica la rentabilidad promedio anual que genera el capital que permanece invertido en el proyecto. También se define como la tasa de descuento que hace que el VAN = 0, su valor no depende del tiempo y representa el máximo costo que el inversionista podría pagar por el capital prestado.

Desarrollando:

con $VAN = 0$

$$0 = -48580 + \frac{6000}{(1 + 0,1)^1} + \frac{9000}{(1 + 0,1)^2} + \frac{10000}{(1 + 0,1)^3} + \frac{15000}{(1 + 0,1)^4} + \frac{30000}{(1 + 0,1)^5}$$

Se logra calcular el TIR con el siguiente valor TIR (V1:V6)

$K=0,1$, esto equivale al 10%

Interpretación TIR:

$TIR > k$ se recomienda pasar a la siguiente etapa

$TIR = k$ es indiferente invertir

$TIR < k$ se recomienda su rechazo o postergación

Notamos que el resultado es igual a cero por lo tanto según los parámetros, el proyecto es indiferente invertir.

5.6. COSTO BENEFICIO

Tomando en cuenta los resultados anteriores (VAN y TIR) se toma en cuenta los valores de $k = 0,1$ y $VAN = 698,55$.

Formula:

$$\text{Costo Beneficio} = \frac{C}{B}$$

Donde:

C: Costo

B: Beneficio

Desarrollando:

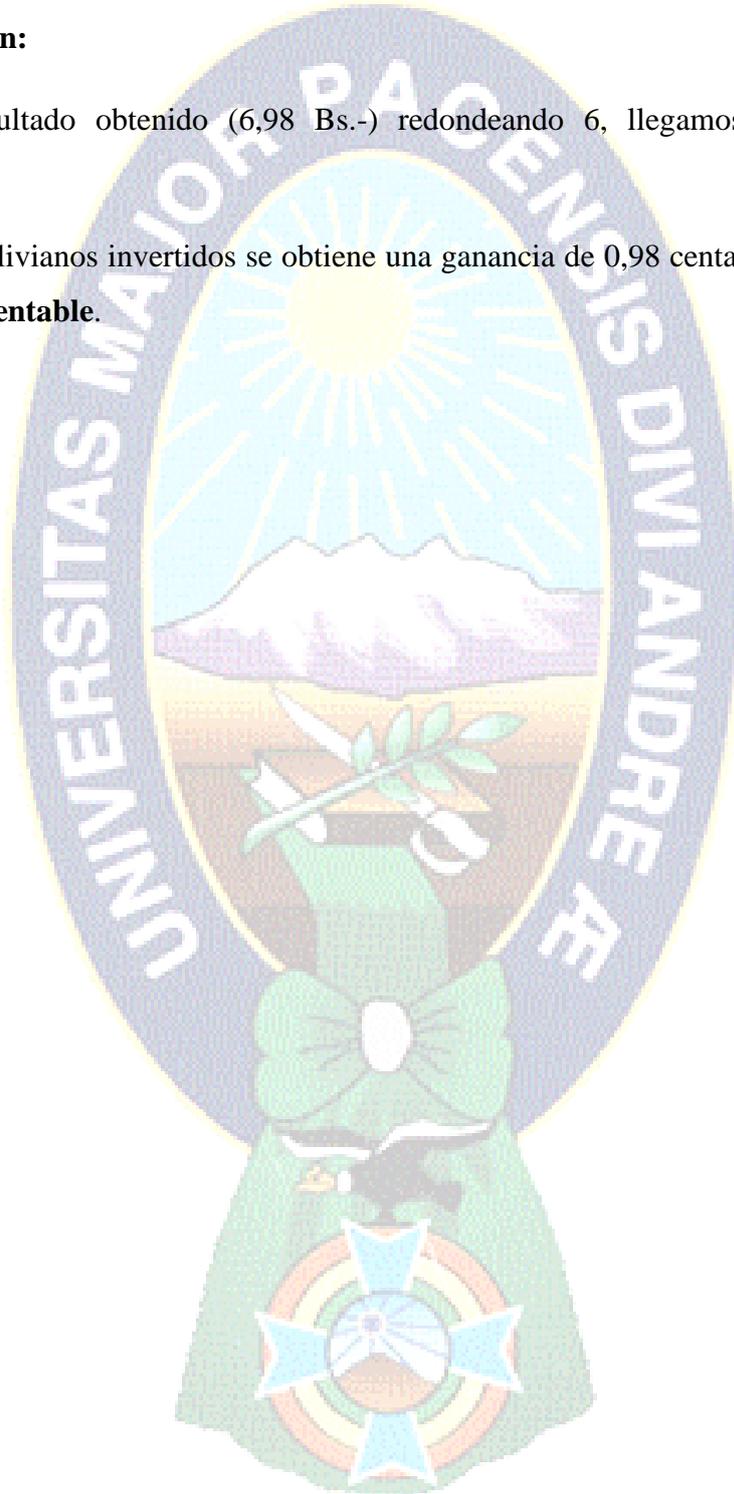
$$\text{Costo Beneficio} = \frac{698,55}{0,1}$$

Costo Beneficio = 6,98 Bolivianos

Interpretación:

Como el resultado obtenido (6,98 Bs.-) redondeando 6, llegamos a la siguiente conclusión:

Por cada 6 bolivianos invertidos se obtiene una ganancia de 0,98 centavos, por tanto, el **proyecto es rentable.**



CONCLUSIONES Y RECOMENDACIONES

6.1. CONCLUSIONES

Habiendo culminado el desarrollo y la implementación del sistema de para la Florería “Dalia”, donde se aplicó todo el conocimiento de metodologías de análisis y diseño de software, todo en base a los requerimientos planteados por el interesado, se puede concluir que se cumplió con los objetivos planteados de la siguiente manera:

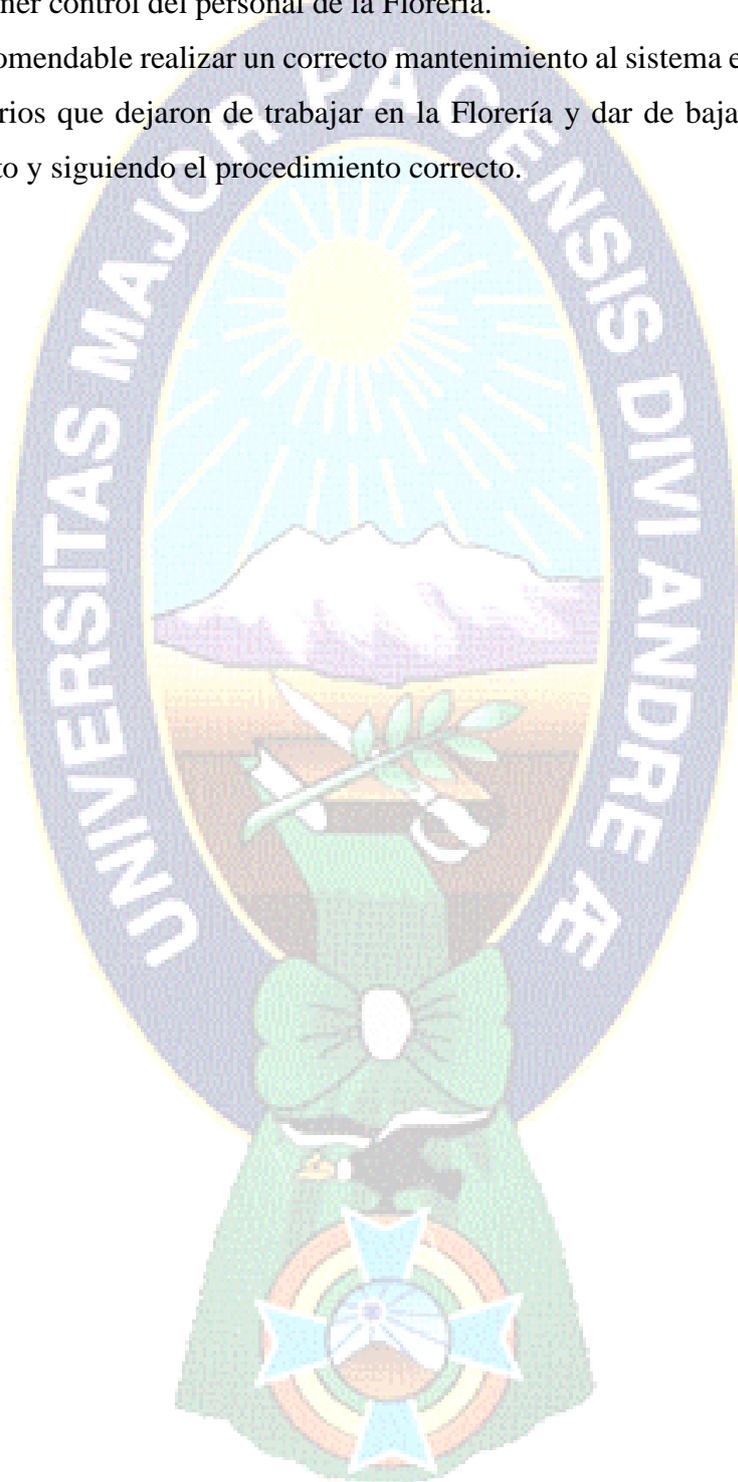
- Se realizo el proceso para el control y seguimiento de servicios en la Florería.
- Se implementó el control y permisos sobre la administración del sistema, de acuerdo al rol asignado a cada usuario.
- Se redujo costos o gastos rutinarios que generaban perdidas.
- Se implementó el módulo de clientes y servicios donde los usuarios registrados según los roles asignados pueden acceder al detalle de información de clientes, donde pueden conocer sus formas de pago y promociones.
- Se implementó el módulo de reportes, en base a diferentes criterios para la obtención de información donde los usuarios pueden obtener información diaria o mensual de los servicios a los clientes.

6.2. RECOMENDACIONES

Se recomiendan varios aspectos para potenciar el uso y aprovechamiento del sistema web, bajo lo siguiente:

- Se recomienda utilizar la misma versión del *framework Codeigniter* a momento de mejorar el sistema pues el *framework* es escalable y se adecua a diferentes contextos para su aplicación.
- La Florería “Dalia”, aún tiene necesidades en cuanto a registros contables y facturación electrónica, lo cual se recomienda no dejar pendiente y continuar automatizando los procesos de la Florería.

- Se recomienda implementar y adaptar al sistema el módulo de recursos humanos para tener control del personal de la Florería.
- Es recomendable realizar un correcto mantenimiento al sistema en lo que respecta a usuarios que dejaron de trabajar en la Florería y dar de baja en un momento correcto y siguiendo el procedimiento correcto.



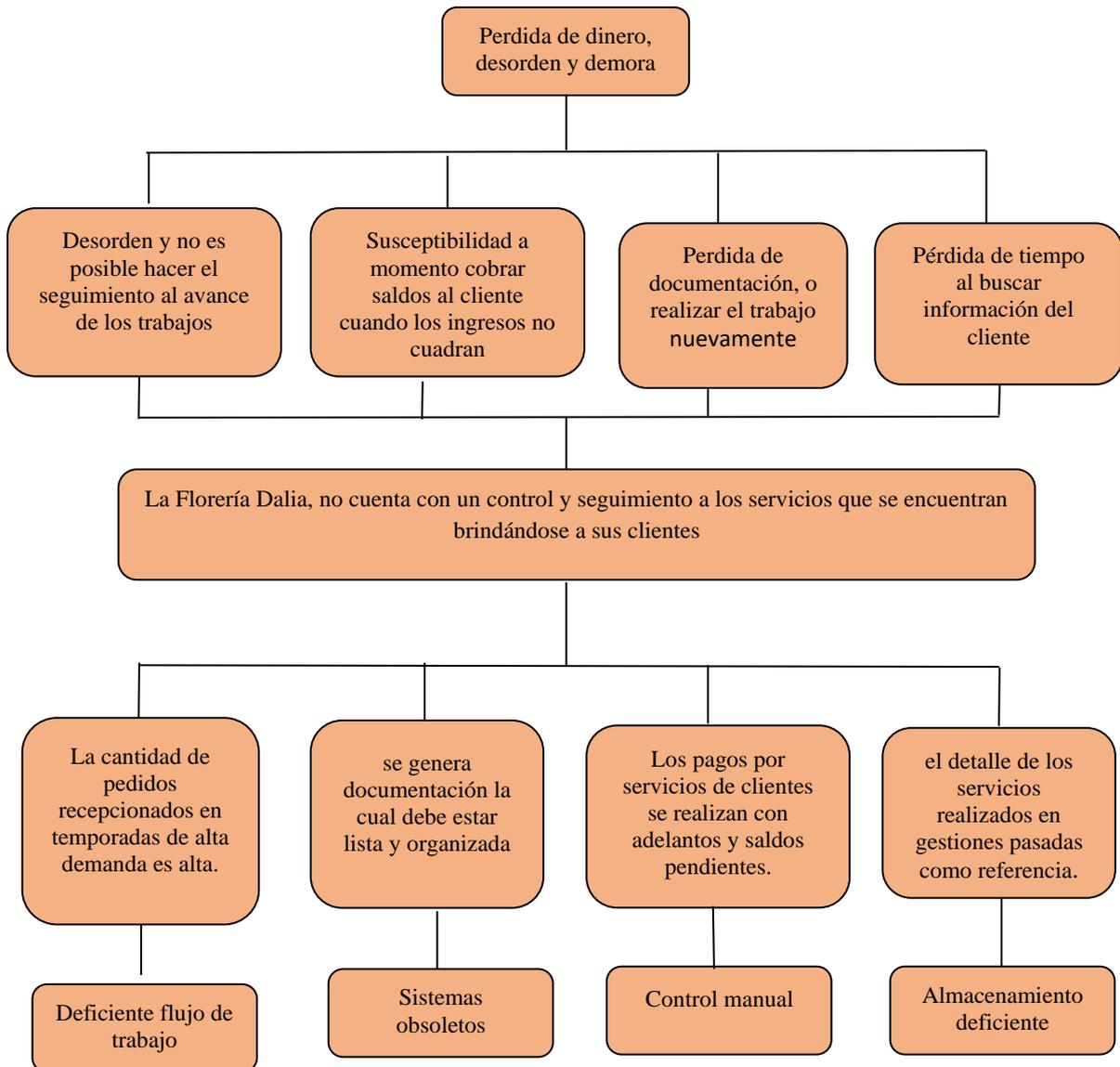
BIBLIOGRAFIA

- Aduviri, E. (2016).: **“SISTEMA WEB DE CONTROL DE VENTAS E INVENTARIOS CASO: MICHELLINE”**. Ingeniería de sistemas informáticos. La Paz, Universidad Mayor de San Andrés, Carrera de Informática.
- (Bauer, 1972). Introducción a la Ingeniería de Software
<<http://ocw.uc3m.es/ingenieria-informatica/principios-de-ingenieria-informatica/introduccion-a-la-ingenieria-del-software>> (consulta: 15 de octubre de 2019)
- (Bohem, 1976). Ingeniería de Software – Técnicas ágiles de desarrollo
<<http://umh2818.edu.umh.es/wp/content/uploads/sites/884/2016/02/Ingenieri%C3%81a-del-Software.pdf>> (consulta: 7 de septiembre de 2019)
- (Cabrera, Figueroa, y Solís, 2008). Metodologías Tradicionales Vs. Agiles
<<https://es.scribd.com/document/372156782/Articulo-Metodologia-de-Sw-Formato>> (consulta: 2 de octubre de 2019)
- Cueto, L. (2012). Software boliviano de exportación, a un BIT del negocio. Revista Emprendedores: 23-24.
- Duran, R. (2016). **“SISTEMA DE CONTROL DE INVENTARIOS CON ALERTAS TEMPRANAS PARA PEDIDOS EN LINEA CASO: ASOCIACIÓN MIXTA DE ORFEBRES YRELOJEROS A.M.O.R.”**. Ingeniería de sistemas informáticos. La Paz, Universidad Mayor de San Andrés, Carrera de Informática.
- Gutiérrez, A. (2016). **“SISTEMA WEB DE COMERCIO ELECTRÓNICO CASO: EMPRESA DE DISTRIBUCIÓN DE EQUIPOS DE COMPUTACIÓN HSB”**. Ingeniería de sistemas informáticos. La Paz, Universidad Mayor de San Andrés, Carrera de Informática.
- Hinojosa, R. (2016). **“APLICACIÓN ANDROID PARA LA GESTIÓN Y CONTROL DE LA ENTREGA Y RECOJO DE CORRESPONDENCIA CASO: TNT EXPRESS”**. Ingeniería de sistemas informáticos. La Paz, Universidad Mayor de San Andrés, Carrera de Informática.

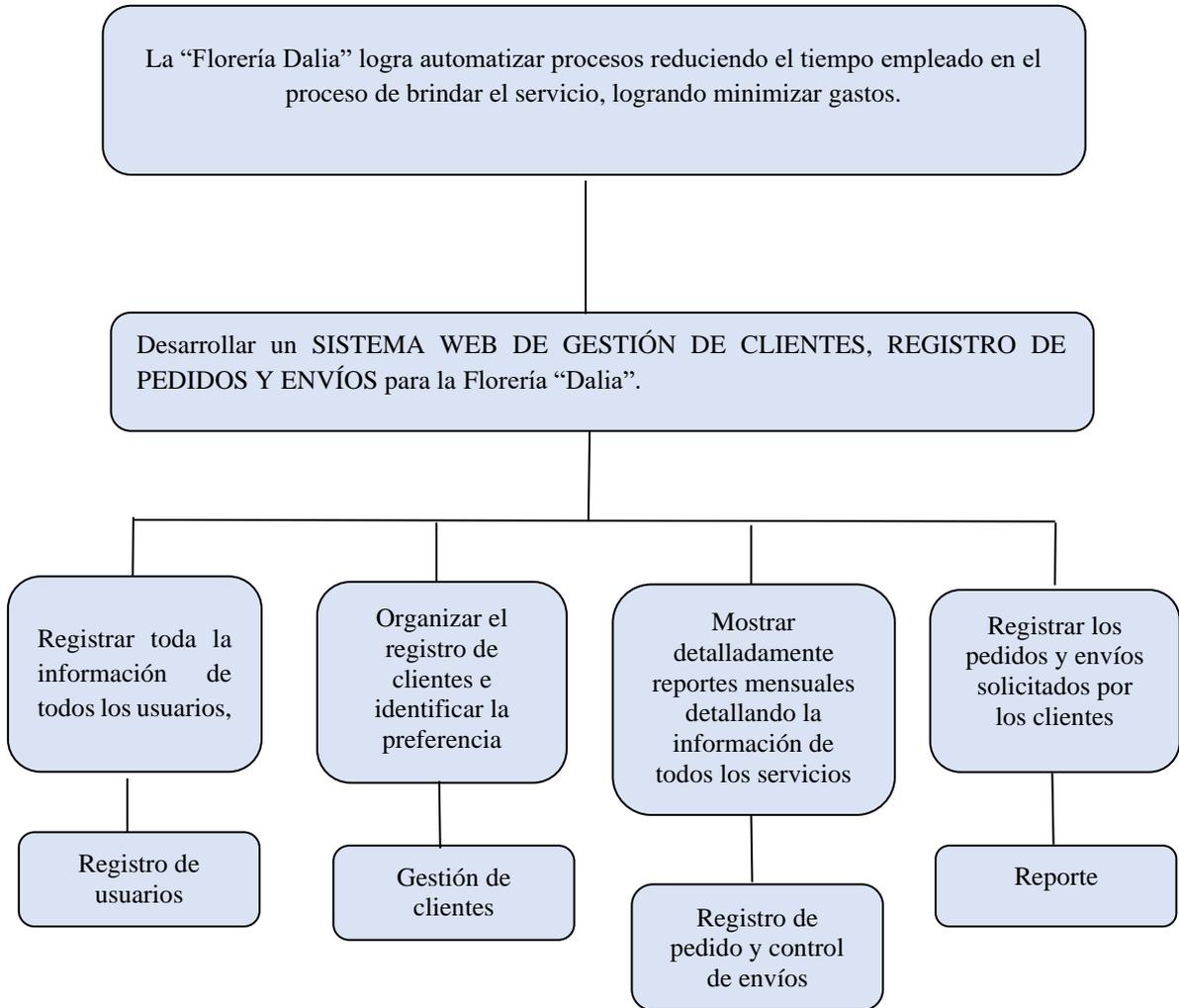
- Limachi, I. (2018). **“SISTEMA WEB PARA EL CONTROL Y SEGUIMIENTO DE VENTAS DE PRODUCTOS ARTESANALES CASO: BOLIVIA TECH HUB”**. Ingeniería de sistemas informáticos. La Paz, Universidad Mayor de San Andrés, Carrera de Informática.
- Mateo.A. (2018). ¿Qué es un sistema Web? [en línea].
<<https://www.neosoft.es/blog/que-es-un-sistema--web>> (consulta: 04 de junio de 2019)
- Pressman, F. A. (2012). Ingeniería del software un enfoque práctico. Quinta edición. México. McGraw-Hill Interamericana Editores S.A de C.V.
- Rafael.M. (2016). Desarrollo Scrum.
<<https://colombianos.com.co/articulos/desarrollo-scrum-.html>> [consulta: 25 de agosto de 2020]
- Suaznabar, R. (2017). Empresas de desarrollo de software en Cochabamba. Primera edición. Cochabamba. Talleres Gráficos “Kipus”
- Wolf. D.(2013). Metodología UWE [en línea].
<<https://metodologiauwe.wordpress.com/author/danielthewolf1993>> (consulta: 28 de agosto de 2019)
- (Zelkovitz, 1978). Ingeniería de Software
<<http://dis.um.es/~barzana/Informatica/IAGP/IAGP>> (consulta: 3 de septiembre de 2019)

ANEXOS

ANEXO A – ARBOL DE PROBLEMAS



ANEXO B – ARBOL DE OBJETIVOS



ANEXO C – MARCO LÓGICO

RESUMEN NARRATIVO	INDICADORES OBJETIVAMENTE VERIFICABLES	MEDIOS DE VERIFICACIÓN	SUPUESTOS
<p>FIN:</p> <p>La Florería “Dalia” logra automatizar procesos reduciendo el tiempo empleado en el proceso de brindar el servicio, logrando minimizar gastos.</p>	<p>Comparación de estados de resultados de la Florería, demostrando la diferencia en las ganancias.</p>	<p>Informes y reportes (estados financieros) a partir de la implementación del sistema web dentro de la Florería.</p>	<p>Los usuarios de sistema logran asimilar el flujo de trabajo con el sistema y lo aplican a la rutina de la Florería.</p>
<p>PROPOSITO:</p> <p>Desarrollar un sistema web de gestión de clientes registro de pedidos y envíos, para la Florería “Dalia”</p>	<p>Cumplimiento de los requisitos solicitados. Sistema web completamente funcional</p>	<p>Aval de la Florería “Dalia”, una vez que el sistema esté terminado.</p>	<p>Se cuenta con la aprobación del representante legal, Sr Dionicio Mamani Laura, de la Florería para la implementación del sistema.</p>
<p>PRODUCTOS:</p> <p>Realizar el control y seguimiento de trabajos recepcionados. Realizar el control y seguimiento de pagos por servicios. Almacenar documentos en un repositorio de archivos clasificados por clientes. Facilitar la administración de documentación la cual se genera cuando el servicio al usuario culmino. Generar reportes sobre la información de los clientes.</p>	<p>Registro de todos los trabajos recibidos. Registro de todos los pagos realizados por los clientes. Archivos digitales almacenados por cada servicio. Información centralizada</p>	<p>Los usuarios pueden dar de crear, modificar y borrar clientes. Los usuarios obtienen reportes Para obtener información centralizada. El sistema registra cada trabajo recibido (servicio) y permite al usuario realizar su correspondiente seguimiento.</p>	<p>Se cuenta con toda la información de los clientes registrados manualmente y se los importe a la base de datos del sistema web desarrollado. El personal de la Florería conoce de su rol asignado en el sistema. Se cuenta con todas las herramientas necesarias para el proceso de desarrollo del sistema web.</p>
<p>ACTIVIDADES:</p> <p>-Coordinar requerimientos con el personal de la Florería. -Implementar los controladores del sistema. -Implementar los modelos del sistema. -Diseñar e implementar las vistas del sistema.</p>		<p>Revisión de la documentación donde se encuentra todas las fases realizadas en el proceso de desarrollo del sistema web.</p>	<p>Se cuenta con el tiempo planeado dentro del cronograma de actividades. Se cuenta con reuniones de capacitación para los usuarios que manipularan el sistema.</p>

<ul style="list-style-type: none"> -Clasificar y asignar roles de usuario del sistema. -Ordenar los servicios de la Florería -Generar reportes de servicios de clientes. -Implementar el sistema web. -Realizar pruebas al sistema 			<p>El servidor web se encuentre disponible para su implementación.</p>
---	--	--	--

ANEXO D – CRONOGRAMA DE ACTIVIDADES

ACTIVIDADES	DURACIÓN EN DÍAS	DEL 1 DE JULIO AL 30 DE NOVIEMBRE																							
		JULIO				AGOSTO				SEPTIEMBRE				OCTUBRE				NOVIEMBRE				DIC			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2		
Redacción del Capítulo II - Marco Teórico	7																								
Desarrollo del Capítulo III - Marco Aplicativo	91																								
MÓDULO ADMINISTRACIÓN DE USUARIOS Y GESTIÓN DE CLIENTES.	21																								
Fase de análisis	7																								
Análisis de requerimientos	1																								
Casos de uso	1																								
Modelo de contenidos	5																								
Fase de diseño	7																								
Modelo navegacional	3																								
Modelo de representación	4																								
Fase de desarrollo	6																								
Modelo de flujo de procesos	6																								
Fase de implementación	1																								
MODULO DE REGISTRO DE PEDIDOS Y ENVIOS	21																								
Fase de análisis	7																								
Análisis de requerimientos	1																								
Casos de uso	1																								
Modelo de contenidos	5																								
Fase de diseño	7																								

