

UNIVERSIDAD PÚBLICA DE EL ALTO

CARRERA INGENIERÍA DE SISTEMAS



PROYECTO DE GRADO

“SISTEMA WEB PARA LA GESTION DE ALMACENES Y DISTRIBUCION”

CASO: ADIPHARM S.R.L.

Para optar al título de Licenciatura en Ingeniería de Sistemas

Mención: GESTION Y PRODUCCION

Postulante: José Ronald Tambo Arcani

Tutor Metodológico: Ing. Marisol Arguedas Balladares

Tutor Especialista: Lic. Freddy Salgueiro Trujillo

Tutor Revisor: Lic Rosa Patricia Nina Chura

EL ALTO – BOLIVIA

2020

DEDICATORIA

El presente proyecto va dedicado en primer lugar a dios, por haberme dado las fuerzas para seguir y poder concluir este proceso ya es uno de los anhelos más deseados.

A mis padres por su amor, su apoyo y la paciencia en todos estos años, que gracias a sus enseñanzas eh podido llegar hasta aquí. Ha sido un orgullo y privilegio ser su hijo.

A mi esposa Patricia, gracias a su apoyo incondicional, que me dio fuerzas y ánimos en todo momento.

A mis hermanas (os) por estar siempre presentes, acompañándome y por el apoyo moral, que nos brindaron a lo largo de esta etapa de nuestras vidas. A todas las personas que nos han apoyado y han hecho que el trabajo se realice con éxito en especial a aquellos que nos abrieron las puertas y compartieron sus conocimientos.

José Ronald Tambo Arcani

AGRADECIMIENTOS

Primero y antes que nada quisiera agradecer a Dios por la oportunidad que me brindo para realizar este proyecto y aprender de él.

A mis padres ya que me brindaron apoyo incondicional en mis estudios y a lo largo de mi vida.

A mi esposa por el apoyo y la paciencia en cada momento a lo largo del desarrollo del proyecto.

A mis tutores que fueron un apoyo incondicional Lic. Freddy Salgueiro Trujillo, Ing. Marisol Arguedas Balladares, Lic. Rosa Patricia Chura por toda la sabiduría y experiencia profesional, por sus consejos, colaboración y paciencia a lo largo del desarrollo de éste Proyecto, que es un gran pasó en la vida de un estudiante Universitario.

RESUMEN

En la actualidad el contar con un sistema de información es vital para las empresas ya que estas para su crecimiento hacen uso de las tecnologías de información que ayudan a resolver los problemas optimizar procesos y eliminar los tiempos muertos a través de sistemas innovadores.

La empresa ADIPHARM S.R.L. tiene como actividad principal la venta y distribución de productos alimenticios para mascotas domesticas (Gatos y Perros), una empresa que va a la vanguardia implementando herramientas tecnológicas de acuerdo a sus necesidades y requerimientos.

El presente proyecto tiene como principal objetivo desarrollar un sistema web para la gestión de almacenes y distribución, de manera que pueda mejorar el flujo de información y la administración logística dentro de la empresa.

El proyecto es desarrollado utilizando la metodología UWE (UML-BASED WEB ENGINEERING – INGENIERIA WEB BASADA EN UML) que va enfocado al desarrollo exclusivo de sistemas web, aplicando las actividades UWE en cada una de las fases que contempla la metodología.

Se empleó los factores de calidad (Métodos de evaluación de calidad). Para evaluar y medir la calidad del producto se aplicó los puntos de vista que plantea el modelo McCall tomando en cuenta sus criterios de evaluación como ser: la facilidad de uso, integridad, corrección, fiabilidad, eficiencia, facilidad de mantenimiento facilidad de prueba, flexibilidad, reusabilidad, interoperabilidad y portabilidad. Las cuales nos permiten medir la calidad del producto final.

Finalmente, los objetivos planteados en base a la problemática existente, han sido alcanzados de manera satisfactoria, de manera que se produjo un sistema de calidad que permite tener un control de inventarios, planificación y distribución a través del sistema.

INDICE GENERAL

1.1 INTRODUCCION	1
1.1 ANTECEDENTES	2
1.1.1 Antecedentes institucionales.	2
1.2.2 Antecedentes Académicos.....	5
1.3 PLANTEAMIENTO DEL PROBLEMA	6
1.3.1 PROBLEMA GENERAL	8
1.3.2 PROBLEMAS ESPECIFICOS	8
1.4 OBJETIVOS	9
1.4.1 OBJETIVO GENERAL.....	9
1.4.2 OBJETIVOS ESPECIFICOS.....	9
1.5 JUSTIFICACION	10
1.5.1 Justificación Técnica	10
1.5.2 Justificación Económica.....	10
1.5.3 Justificación Social	10
1.6 METODOLOGÍA.....	11
1.6.1 METODO CIENTIFICO	11
Metodología UWE.....	11
Métodos de control de Inventarios ABC.....	13
Factores de calidad modelo McCall	14
Modelo constructivo de costos Cocomo II	15
1.6.2 METODO DE INGENIERIA	17
Método KANBAN.....	17
Arquitectura de desarrollo en (N) capas.....	18
Sistemas de información Geográfica	19
1.7 HERRAMIENTAS.....	20
1.8 LIMITES Y ALCANCES	24
1.8.1 Alcances.....	24
1.8.2 Limites	24
1.9 APORTES	24
1.10 ESQUEMA	¡Error! Marcador no definido.
1.11 CRONOGRAMA DE EJECUCION.....	¡Error! Marcador no definido.

CAPITULO II MARCO TEORICO	25
2.1 METODOLOGIA UWE	25
2.1.1 Actividades de modelado de UWE.	26
2.1.2 Fases o etapas de la metodología UWE.	30
2.2 ARQUITECTURA DE DESARROLLO EN (N) CAPAS	31
2.2.1 Capa de Presentación.....	32
2.2.2 Capa de Reglas de Negocio (Empresarial).....	32
2.2.3 Capa de Datos.....	33
2.2.4 Capa de Implementación del Servicio:.....	33
2.3 METODO DE CONTROL DE INVENTARIOS	33
2.3.1 Entrada o ingreso de productos	34
2.3.2 Salida o egreso de productos.....	34
2.3.3 Importancia del control de la administración de almacenes.....	35
2.3.4 Método de control de inventarios ABC.....	35
2.3.5 Clasificación del método ABC para inventarios	36
2.3.6 Ventajas del método ABC	36
2.3.7 Desventajas del método ABC	37
2.4 METODOLOGIA KANBAN	37
2.4.1 Kanban en el desarrollo de software	38
2.4.2 Tablero de kanban.	38
2.4.3 Principios de Kanban.....	40
2.4.4 Características de la metodología kanban.....	40
2.5 FACTORES DE CALIDAD MODELO McCall.....	41
2.5.1 Calidad de software.....	42
2.5.2 El modelo de McCall.....	45
2.6 MODELO CONSTRUCTIVO DE COSTOS COCOMO II	50
2.6.1 Definición del modelo.....	50
2.6.2 Estimación del Esfuerzo	52
2.6.3 Modelo composición de aplicación	52
2.6.4 Modelo Diseño Temprano.....	53
2.6.5 Modelo Post-Arquitectura.....	56
2.6.6 Estimación del tamaño de software COCOMO II	60

2.7 SISTEMAS DE INFORMACION GEOGRAFICA	66
2.7.1 Funcionamiento de un SIG	67
2.7.1 La tecnología GIS, al servicio de las empresas de suministros.....	69
CAPITULO III MARCO APLICATIVO	71
3.1 FASE DE ANÁLISIS Y ESPESIFICACION DE REQUISITOS.....	71
3.1.2 Estado actual de procesos operativos de la empresa ADHIPARM S.R.L	71
3.2.2 determinación requerimientos funcionales.....	73
3.2 FASE DE MODELADO Y DISEÑO DEL SISTEMA	76
3.4 FASE DE CODIFICACION O CONSTRUCCION	95
3.4.1 Pantallas del sistema	95
3.5 FASE DE PRUEBAS	102
3.6 FASE DE INSTALACION O FASE DE IMPLEMENTACIÓN	108
3.7 EVALUACIÓN DE CALIDAD DEL SISTEMA.....	109
3.8 ANÁLISIS DE COSTOS.....	112
3.9 SEGURIDAD DEL SISTEMA Y BASE DE DATOS.....	110
CAPITULO IV CONCLUSIONES Y RECOMENDACIONES	116
4.1 CONCLUSIONES.....	117
4.2 RECOMENDACIONES	118
ANEXOS	125

INDICE DE FIGURAS

Figura 1.1 Organigrama de la empresa ADIPHARM S.R.L.....	3
Figura 1.2 Localización de la empresa ADIPHARM S.R.L.	5
Figura 1.3 Logotipo de la metodología UWE.....	12
Figura 1.4 Fases de la metodología UWE.....	13
Figura 2.1 Descripción general de los modelos UWE	26
Figura 2.2 Nombres de estereotipos y sus iconos de modelo de navegación.	28
Figura 2.3 Nombres de estereotipos y sus iconos del modelo de presentación.....	29
Figura 2.4 Arquitectura de software (n) capas.....	32
Figura 2.5 Sistema de Administración de Almacenes	34
Figura 2.6 Tablero básico de Kanban	39
Figura 2.7 Tablero de kankan con más columnas.....	39
Figura 2.8 Modelo de evaluación de McCall	45
Figura 2.9 Factores y métricas de calidad según mccall.....	46
Figura 2.10. Tabla de valores de los factores de escala.	55
Figura 2.11 Tabla de parámetros de producto	57
Figura 2.12 Tabla de valores de plataforma.....	57
Figura 2.13 Tabla de valores de Personal.....	58
Figura 2.14 Tabla de valores de Proyecto.....	58
Figura. 2.15 El uso del SIG en la planeación de rutas.	70
Figura 3.1 Aplicación KANBAN- tablero de tareas programadas	75
Figura 3.2 Modelo de casos de uso de pedidos.....	76
Figura 3.3 Modelo de contenido de pedido	76
Figura 3.4 Modelo de navegación de pedido	77
Figura 3.5 Modelo de presentación de pedido	77
Figura 3.6 Modelo de proceso de pedido- registro de cliente.....	78
Figura 3.7 Modelo de proceso de pedido- eliminación de cliente.....	78
Figura 3.8 Modelo de proceso de pedido- eliminación de cliente.....	79
Figura 3.9 Modelo de proceso de pedido- registro de pedido	79
Figura 3.10 Modelo de proceso de pedido- elimina pedido.....	80
Figura 3.11 Modelo de proceso de pedido- actualiza pedido	80

Figura 3.12 Modelo caso de uso de gestión de almacén	81
Figura 3.13 Modelo de contenido gestión de almacén	81
Figura 3.14 Modelo de navegación de gestión de almacén	82
Figura 3.14 Modelo de presentación gestión de almacén 1	83
Figura 3.15 Modelo de presentación gestión de almacén 2	84
Figura 3.16 Modelo de procesos gestión de almacén-registro de almacén	84
Figura 3.17 Modelo de procesos gestión de almacén-elimina almacén	85
Figura 3.18 Modelo de procesos gestión de almacén-actualiza almacén	85
Figura 3.19 Modelo de procesos gestión de almacén-registra movimiento.....	86
Figura 3.20 Modelo de procesos gestión de almacén-elimina movimiento	86
Figura 3.21 Modelo de procesos gestión de almacén-actualiza movimiento ...	87
Figura 3.22 Modelo de procesos gestión de almacén-registra producto	87
Figura 3.23 Modelo de procesos gestión de almacén-elimina producto.....	88
Figura 3.24 Modelo de procesos gestión de almacén-actualiza producto	89
Figura 3.25 Modelo de procesos gestión de almacén-actualiza proveedor.....	90
Figura 3.26 Modelo de procesos gestión de almacén-elimina proveedor	91
Figura 3.27 Modelo de procesos gestión de almacén-actualiza proveedor.....	91
Figura 3.28 Modelo de casos de uso en distribución	92
Figura 3.29 Modelo de contenido de distribución	92
Figura 3.30 Modelo de navegación de distribución	93
Figura 3.31 Modelo de presentación de distribución	93
Figura 3.32 Modelo de datos.....	94
Figura 3.33 Pantalla de inicio de sesión.....	95
Figura 3.34 Pantalla principal.....	95
Figura 3.35 Pantalla gestión de usuarios	96
Figura 3.36 Pantalla gestión de usuarios	96
Figura 3.37 Pantalla gestión de usuarios	97
Figura 3.38 Pantalla gestión de clientes.....	97
Figura 3.39 Pantalla gestión de clientes ubicación geográfica	98
Figura 3.40 Pantalla gestión de proveedores	98
Figura 3.41 Pantalla gestión de productos	99

Figura 3.42 Pantalla gestión de pedidos	99
Figura 3.43 Pantalla gestión de almacenes.....	100
Figura 3.44 Pantalla gestión de movimientos.....	100
Figura 3.45 Pantalla gestión de planificación de ruta	101
Figura 3.46 Pantalla gestión de planificación, ruta generada.....	101
Figura 3.47 Pantalla gestión de distribución, lista de pedidos	102
Figura 3.48 Pantalla gestión de distribución, instrucciones de conducción y ruta	102
Figura:3.49 diagrama de flujo y grafo de ingreso de sesión.....	106
Fuente: Elaboración propia	106
Figura 3.50 Interfaz de método y petición en pruebas de caja negra.....	108
Figura 3.51 Interfaz de resultado de pruebas de caja negra	108
Figura 3.52 Aplicación web COCOMO II.....	112
Figura 3.53 Interfaz COCOMO II web- Selección de valores y ponderacion respectiva.....	113
Figura 3.53 COCOMO II Web resultados.....	115
Figura 3.54 Distribución de esfuerzo de software (persona-meses)	115

INDICE DE TABLAS

Tabla 1.1_	Tabla comparativa de lenguajes de programación	20
Tabla.1.2_	Cuadro comparativo de Bases de datos.....	22
Tabla 2.1_	Modelo de Referencia de Medición de la Calidad del Producto Software.	48
Tabla 2.2_	Distribución del Mercado de Software Actual y Futuro.	51
Tabla 2.3_	Productividad para el modelo Composición de Aplicación	53
Tabla 2.4_	Esquema de Clasificación de Puntos Objetos	61
Tabla 2.5_	Peso de un Punto Objeto	61
Tabla 2.6_	Equivalencia puntos de función en líneas de código	62
Tabla 2.7_	Puntos Función. Determinación del Peso.....	63
Tabla 2.8_	Peso del Factor de Complejidad.....	63
Tabla 3.1_	Product BackLog del Sistema web	75
Tabla 3.2_	Caso de prueba para la gestión de módulos Usuarios, Productos, Proveedores, Pedidos, Almacenes y movimientos.....	103
Tabla 3.3_	Caso de prueba- Gestión de clientes.	104
Tabla 3.4_	Caso de prueba- gestión de planificación.....	104
Tabla 3.5_	Caso de prueba – gestión de distribución	105
Tabla 3.6_	Tabla de evaluación Operación del Producto	109
Tabla 3.7_	Tabla de evaluación Operación del Producto	110
Tabla 3.8_	Tabla de evaluación Operación del Producto	111
Tabla 3.9_	Ponderación de valores en escala de software	113
Tabla 3.10_	Ponderación de valores en producto a desarrollar	114
Tabla 3.11_	Ponderación de valores de plataforma	114
Tabla 3.12_	Ponderación de valores en evaluacion de personal	114
Tabla 3.13_	Ponderación de valores en proyecto	114

CAPITULO I MARCO PRELIMINAR

1.1 INTRODUCCION

Los sistemas de Información son una parte de la organización de recursos que captura, procesa, almacena y distribuye la Información, ayudan a mejorar el control y la gestión de una empresa ya que tienen un fuerte impacto y crean ventaja competitiva.

Una gran parte de las empresas comerciales de importación y venta de productos presentan dificultades en el control de gestión de procesos logísticos lo que provoca la pérdida de la eficiencia en las operaciones, disminuyendo de cierta manera la credibilidad ante sus clientes y su entorno general, en donde se requiere la implementación de un sistema que optimice los procesos en cuanto a gestión empresarial.

En la necesidad de poder realizar un control a la gestión de procesos y rutas de distribución se plantea el desarrollo e implementación de un sistema que optimizara y mejorara y facilitara dicho control de procesos en la empresa ADIPHARM S.R.L.

Es fundamental evaluar las técnicas actuales y la tecnología disponible para desarrollar sistemas que brinden eficiencia y eficacia de la gestión de la información relevante.

El sistema se desarrollará utilizando las metodologías de análisis y diseño UWE el cual nos permite hacer una propuesta orientada por disciplinas para lograr las tareas y responsabilidades de una organización, como metodología de desarrollo se plantea KANBAN el cual nos permite realizar el desarrollo del sistema de acuerdo a las necesidades especificadas con el método de análisis y diseño. Las herramientas de desarrollo planteadas son: Base de datos Maria DB, para desarrollo del back-end lenguaje del lado de servidor PHP con el framework Codeigniter, Angular CLI para el lado del cliente. Las cuales permitirán que el desarrollo se lleve a cabo de acuerdo a los requerimientos del sistema.

Por todo ello, es importante destacar que el desarrollo de un sistema de información implica un cambio organizativo, ya que no sólo afecta a la administración de la empresa, sino también a sus empleados y habilidades, con el

fin de crear una plataforma acorde a las responsabilidades que se deben tener frente a este tipo de sistema.

Con el desarrollo del sistema se pretende automatizar los procesos operativos proporcionando información de apoyo y lograr ventajas competitivas.

1.1 ANTECEDENTES

1.1.1 Antecedentes institucionales.

La empresa ADIPHARM S.R.L. Comienza sus funciones el año 2015 dedicada IMPORTACION Y DISTRIBUCION DE PRODUCTOS AGROPECUARIOS y VETERINARIOS con oficinas centrales en la ciudad de La Paz el cual tuvo una buena aceptación con la distribución de alimento balanceado para mascotas el cual es su principal actividad.

La demanda del mercado impulsa a una nueva necesidad comercial consolidando sucursales a nivel nacional, tomando como principal actividad la distribución y comercialización de productos como alimento balanceado de mascotas tanto para felinos y caninos por mayor y menor que son más requeridos por la población.

La empresa cuenta con una cobertura nacional con sede principal en la ciudad de La Paz el cual tiene una cantidad aproximada de 500 clientes de los cuales en su mayoría son veterinarias, petshop`s, puestos de venta de alimentos para mascotas.

Organigrama de la empresa ADIPHARM S.R.L.

El organigrama nos muestra la estructura organizacional de la empresa ADIPHARM S.R.L.

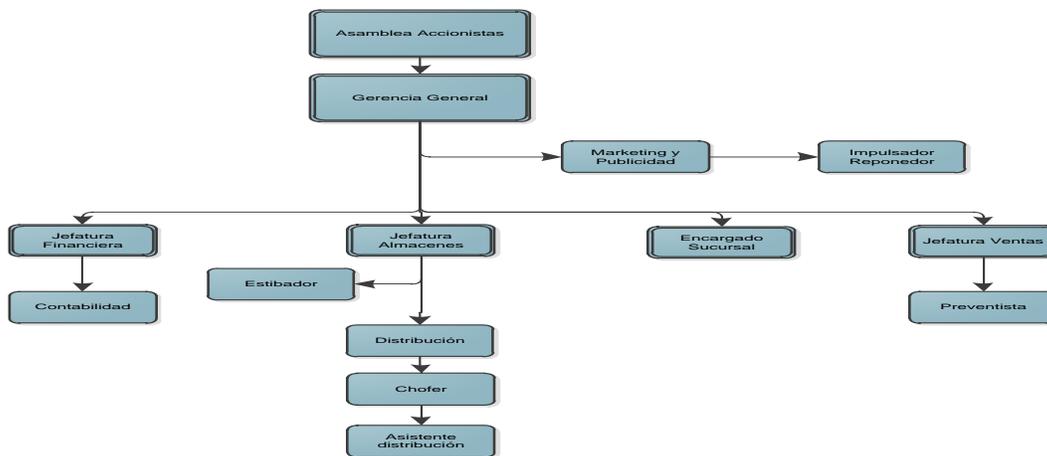


Figura 1.1 Organigrama de la empresa ADIPHARM S.R.L.
Fuente: Elaboración propia

A continuación, se describe las partes que conforman el organigrama de la empresa.

Asamblea Accionistas. - Son las personas asociadas con una parte de las acciones las cuales definen la orientación y administración de la empresa.

Gerencia General: Es el responsable de todas las acciones y compromisos que adquiera la empresa. Deberá coordinar que las funciones realizadas sean las correctas, y velar por el cumplimiento de los objetivos y normas.

Jefatura Financiera. - Encargado de supervisar toda el área administrativa contable.

Contabilidad: Encargado de manejar los flujos de dinero de la empresa, controlar las inversiones, los gastos y el cobro de las deudas pendientes. Además, deberá controlar el cumplimiento de los compromisos financieros adquiridos.

Jefatura de almacenes: Cumple la función sobre el manejo de inventarios y balances periódicos, los cuales ayudarán a determinar la rentabilidad del negocio en cuanto a producto existente.

También se encuentra a cargo del acomodo de mercancías, del inventario, órdenes de pedidos etc. deberá presentar reportes con la finalidad de que el

gerente general encargado de la administración tenga conocimiento de la parte operacional.

Jefatura de control de ventas: Estos contactan por teléfono o personalmente a clientes, promueven y solicitan ventas, así como también procuran la satisfacción de los clientes, buscando asegurar ventas a cada uno ellos.

Distribución. - Encargados de realizar la adecuada distribución de productos a cliente final, el cual cuenta con dos personas Chofer y asistente de distribución.

Personal de distribución: Conductor: Se le asignará una ruta a cubrir, de él depende la entrega oportuna de nuestro producto en los distintos establecimientos. Además, bajo su cargo se encuentra los ayudantes que es el personal que descarga el producto desde el camión y lo entrega en perfectas condiciones.

Encargado de Sucursal. - Esta encargado de coordinar la distribución de productos a nivel departamental excluyendo la central de la empresa.

Jefatura de Ventas: Esta área es el pilar principal de nuestra empresa, ya que de él dependen los ingresos que está obtendrá en base a las estrategias implementadas.

Preventista. - Personal encargado a realizar la preventa de productos, como también está encargado de realizar visitas personales a los clientes.

Marketing y publicidad. - Encargado del marketing y de atraer nuevos clientes como también gestionar promociones. A la vez, éste deberá mantener una cartera adecuada de clientes, velando por la satisfacción y lealtad de éstos hacia la empresa.

La empresa ADIPHARM S.R.L. se cataloga como mediana empresa que está ubicada en el departamento de La Paz, teniendo una oficina central en la ciudad de El Alto con una cantidad de 57 empleados en total los cuales son operan en la ciudad de La Paz y en los distintos departamentos.

La localización de la oficina central de la empresa ADIPHARM S.R.L. se encuentra ubicado en la ciudad de El Alto-Rio Seco Manzano P, Patio 10 N° 1040, telf. 2104175.

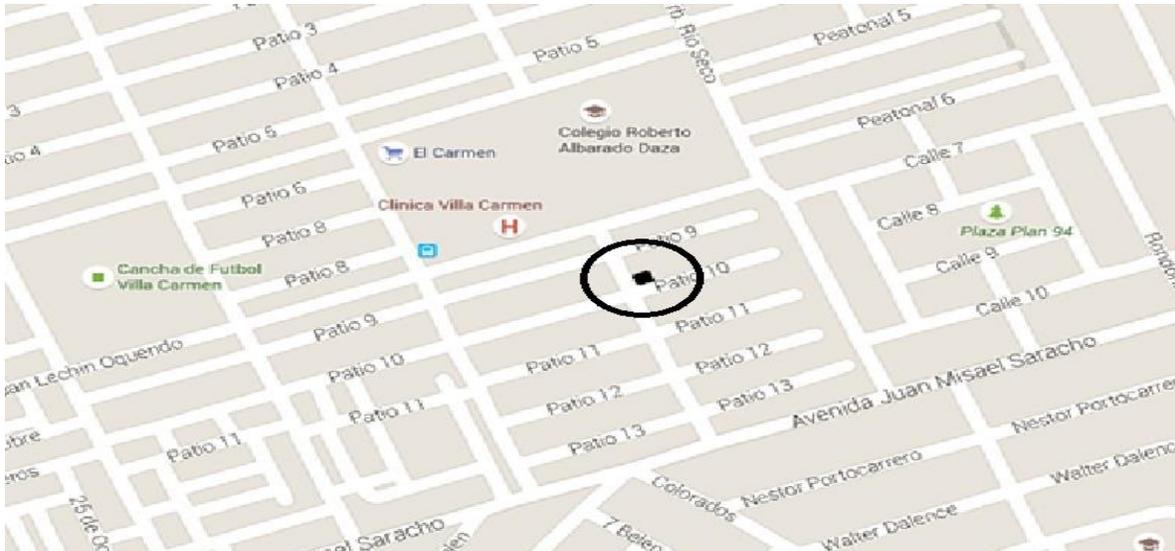


Figura 1.2 Localización de la empresa ADIPHARM S.R.L.
Fuente: elaboración propia

1.2.2 Antecedentes Académicos

En la Universidad Pública de El Alto, Iván Fernando Mujica Mamani en la gestión 2012 presentó el proyecto de grado con el título de “SISTEMA DE GESTION Y CONTROL DE PRODUCCION EN LA EMPRESA SAMI S.R.L.”, enfocado a administración de la información confiable y oportuna además por su importancia que ayuda a gestionar recursos con mayor exactitud y controlar de manera eficiente los procesos de producción y distribución en cuanto a ventas y adquisición de productos, presentada en la gestión 2012 desarrollada bajo la metodología RUP y UML para el modelado del sistema que brinda una herramienta completa para las actividades de análisis y diseño.

Sergio Ramiro Rojas Saire en la gestión 2015 presento el proyecto titulado SISTEMA DE GESTION Y ADMINISTRACION INTEGRAL DE INVENTARIOS PARA LA EMPRESA TERACORP S.R.L. TECNOLOGIA Y CONSULTORIA. Realiza un sistema de gestión y administración integral de inventarios desde

diferentes plataformas tanto para empleados como para el personal ejecutivo de la empresa, desarrollada con una metodología ágil como es el **Adaptive Software Development (ASD)** para el desarrollo del software.

En la Universidad Mayor de San Andrés presento el proyecto “CONTROL DE PRODUCCION, PEDIDOS ENTREGAS E INVENTARIO CASO: PHOVIDA”, realizado por Maddizon Mashiel Camacho Lugones en la gestión 2017. El proyecto enfoca un tratamiento de la información optimizando los procesos de producción y distribución de mercadería, mejorando y facilitando tareas que ocasionaban la pérdida de la información, desarrollado con la metodología UWE para el ciclo de desarrollo de software.

En Ecuador la Universidad Católica del Ecuador el 2014 se presentó el proyecto “SISTEMA DE CONTROL DE RUTAS DE TRANSPORTE DE MERCADERIA PARA LA CASA DEL TOLDO UTILIZANDO SISTEMAS DE POSICIONAMIENTO GEOGRAFICO”, realizado por Benjamin Ronaldo Aguilar Rodrigues y Maria Belen Jumbo Lopez, el cual describe un sistema de control de transporte con (GPS) para determinar posiciones para asegurar un mejor desempeño y control de sus recursos.

En Perú la Universidad Ricardo Palma el 2012 se presentó el proyecto “DESARROLLO DE UNA APLICACIÓN WEB PARA EL MONITOREO DE VEHICULOS CON DISPOSITIVOS GPS QUECOMERCIALIZA UNA EMPRESA DE TELECOMUNICACIONES”, realizado por Rubén Gabriel Bocanegra Urueta, el cual describe un sistema GPS para monitorización vehicular visualizando la localización vía google maps.

1.3 PLANTEAMIENTO DEL PROBLEMA

En la empresa ADIPHARM S.R.L. se ha visto necesario optimizar los procesos de distribución de productos (rutas de distribución) y la gestión en almacenes de una manera ordenada para poder alcanzar resultados favorables en la empresa con una adecuada administración del personal operativo. La actividad actual en su mayoría tiene dificultades de organización y control como ser:

- El registro de entradas y salidas de productos son realizadas en conjunto una vez por semana en hojas de cálculo Excel. (información destiempo), se realizan el registro de todas las notas de ventas registrando el detalle del pedido entregado
- El stock de inventario es controlado semanalmente ya que no existe un control diario, retrasando la información de las cantidades existentes.
- Existe variaciones en los controles de inventarios físicos y lógicos en la mayoría de los casos existen productos faltantes y sobrantes, se invierte mucho tiempo en verificar y corregir errores.
- En el control de inventarios muchas veces existe perdidas por productos de las cuales el responsable es el encargado de almacenes el cual tiene descuentos por los productos faltantes.
- Existe un desorden de productos en los almacenes y no existe un método para el almacenamiento, los productos son almacenados donde haya espacio.
- No existe un registro de despacho en almacenes, solo la hoja de pedidos con las cantidades totales de pedido, ya que todo el personal es designado para cargar el producto en el menor tiempo posible y pueda distribuirse de forma inmediata.
- Las planificaciones de rutas son realizadas de manera manual y desequilibrada porque se las realiza una vez que la planilla de pedidos este elaborada.
- Los productos son extraídos de diferentes almacenes, ya que no existe un método adecuado de despachos retrasando la distribución ya que se va de almacén en almacén para cargar productos a distribuir.
- No existe un control de recorrido del camión de distribución, muchas veces no se cumple o no se logra cubrir y entregar a todos los clientes, provocando llamadas de atención por sus superiores a los distribuidores.
- No existen informes de productos entregados y no entregados a clientes, por la falta de tiempo al final del día, muchas veces por esa razón existe descontrol y pérdida de productos.

- No se realiza la devolución de productos no entregados en almacenes.

El control de los procesos de la gestión de distribución y almacenes es un aspecto esencial ya que un mal control y funcionamiento puede provocar o prestarse a hurtos, mermas y desperdicios generando pérdidas económicas a la empresa.

1.3.1 PROBLEMA GENERAL

En la empresa ADIPHARM S.R.L. no existe una centralización y control óptimo de la información de los procesos en gestión de almacenes y rutas de distribución, lo que genera demoras y diversos errores, desactualización de la información, pérdida de tiempo y pérdidas económicas.

1.3.2 PROBLEMAS ESPECIFICOS

1. El control de ingreso y salida de productos no es registrado oportunamente ya que se actualiza una vez por semana, provocando incertidumbre e indisponibilidad de esta información.
2. No existe un método de almacenamiento de productos, lo que provoca que los productos se encuentren en varios depósitos dificultando su acceso y control.
3. El registro de clientes no es confiable ya que no cuenta con la información actualizada, generando problemas de ubicación y pérdida de tiempo.
4. La planificación de rutas de distribución se las realiza de manera manual y apresurada lo que provoca una planificación no optima y tardanza en distribución.
5. La distribución y entrega de pedido es deficiente, ya que muchas veces no se cuenta con una ruta ordenada, lo que genera desajustes en tiempos de entrega y conflictos con los clientes.
6. El cotejamiento del inventario y productos existentes se maneja de forma manual (papelería) y descentralizada, lo que implica pérdida de tiempo excesivo, informes des oportunos y desconfiables.

7. No se cuenta con información útil, actualizada, concentrada y veraz de la gestión de almacenes y distribución, lo que provoca demora en la toma oportuna de decisiones.

1.4 OBJETIVOS

1.4.1 OBJETIVO GENERAL

Desarrollar un sistema web para la gestión de almacenes y distribución, el cual permitirá facilitar la planificación de rutas, distribución y el control de los procesos operativos, reduciendo tiempos de ejecución, logrando que el flujo de información sea oportuna y confiable para la correcta toma de decisiones, agilizando tareas y reduciendo pérdidas económicas.

1.4.2 OBJETIVOS ESPECIFICOS

1. Automatizar los registros del proceso de movimiento (ingreso y salida) de productos en almacenes, para reducir las pérdidas de productos brindando información actualizada, verídica y oportuna cada que se ejecute este proceso.
2. Aplicar un método de almacenamiento, para mejorar la accesibilidad y el control del producto en almacenes.
3. Automatizar la información de los clientes tomando en cuenta su ubicación geográfica, reduciendo pérdida de tiempo y resolviendo problemas de ubicación.
4. Aplicar un servicio de ruteo con la API directions google maps en la planeación de distribución de productos, para brindar rutas de distribución óptimas.
5. Aplicar un servicio de ruteo con la API directions google maps plataforma, para automatizar las rutas de entrega o distribución de pedido de manera óptima.
6. Facilitar el cotejamiento de productos en almacenes cuando se genere la información del stock de productos, reduciendo tiempo, evitando pérdidas,

hurtos, mermas por detección temprana de errores y brindando información verídica oportuna y confiable para la toma de decisiones.

7. Generar reportes centralizados, verídico, actualizada y disponible de la gestión de almacenes y distribución, para la toma de decisiones oportuna.

1.5 JUSTIFICACION

1.5.1 Justificación Técnica

El proyecto se justifica técnicamente porque se utilizará tecnología que sean aplicables al actual proyecto ya que resulta necesaria la integración de sistemas informáticos y tecnologías que faciliten las tareas operativas para brindar un servicio más rápido y consistente, viabilizando de esta manera una mejora en la administración.

Así también, se justifica técnicamente porque en la empresa ADIPHARM S.R.L. en sus oficinas cuenta con equipos computacionales de cuarta generación. La cual hace que las herramientas planteadas para el desarrollo del sistema ofrezcan interfaces amigables, tomando en cuenta el entorno de las herramientas que nos ofrecen una adaptación adecuada para el desarrollo del sistema, como también se realizara la aplicación de una base de datos que proporcionara una mayor seguridad y disponibilidad de la información.

1.5.2 Justificación Económica

Se justifica económicamente porque el sistema web optimizara los diferentes procesos de control que son realizados de manera manual, disminuyendo los costos en papelería, impresiones y tiempo de realización, como también ayudara a la rápida elaboración de la información.

Se facilitará el seguimiento a los almacenes en cuanto a inventarios ayudando a la detección temprana de errores el cual permitirá evitar malversaciones, hurtos y perdidas económicas a la empresa.

1.5.3 Justificación Social

Se justifica socialmente ya que dentro la empresa ADIPHARM S.R.L. beneficia al personal administrativo dando paso a una mejora en la gestión de planificación y seguimiento del producto comercializado como también el control de inventarios.

Beneficiará al jefe de almacenes, al personal encargados de distribución, para realizar las diferentes operaciones con mayor facilidad permitiendo una mejora en la atención a los clientes con respecto a, distribución de producto.

El sistema web no incidirá en la reducción del personal, si no capacitara al mismo para tener mejores resultados en los procesos de control y gestión de almacenes.

1.6 METODOLOGÍA

El desarrollo de software no es una tarea fácil prueba de ello es que existen numerosas propuestas metodológicas que inciden en distintas dimensiones del proceso de desarrollo.

De hecho, la metodología destinada al desarrollo de software se considera como una estructura utilizada para planificar y controlar el procedimiento de creación de un sistema de información.

1.6.1 METODO CIENTIFICO

Se realiza la elección de la metodología UWE ya que va orientado al desarrollo de sistemas web, proporcionando un modelo el cual permita facilitar el desarrollo de manera específica.

Metodología UWE

Esta metodología nace con la finalidad de controlar el caos que han provocado en el pasado procesos de desarrollo y así proporcionar un proceso sistemático orientado una aplicación final de calidad.

UWE es una metodología muy detallada para el desarrollo de aplicaciones que tiene una definición exacta del proceso de diseño que debe ser utilizado. Este proceso presenta flujos de trabajo y puntos de control, y sus fases son las mismas a las propuestas en el Proceso Unificado de Modelado. UWE está dirigido específicamente a aplicaciones adaptativas, basándose en características de personalización como: la definición de un modelo de usuario en función de las preferencias, conocimiento o tareas de usuario.

La notación de UWE se define como una ligera extensión de UML, proporcionando un perfil UML para el dominio específico de la web.



Figura 1.3 Logotipo de la metodología UWE

Fuente: <http://uwe.pst.ifi.lmu.de/>

Actividades de modelado UWE.

Las actividades de modelado UWE son el análisis de requerimientos, el modelado conceptual, el modelo navegacional y el modelo de presentación.

- **Modelo lógico-conceptual**
- **Modelo de navegación.**
- **Modelo de presentación.**
- **Interacción Temporal.**
- **Escenarios Web.**
- **Diagramas.**

Fases de la metodología UWE.

UWE cubre todo el ciclo de vida de este tipo de aplicaciones centrandose además su atención en aplicaciones personalizadas o adaptivas.

Las fases de UWE son:

- 1) **Captura, análisis y especificación de requisitos:** En simples palabras y básicamente durante esta fase se adquieren, reúnen y especifican las características funcionales y no funcionales que deberá cumplir el sistema o aplicación web.
- 2) **Diseño del sistema:** Se basa en la especificación de requisitos producido por el análisis de requerimientos, e diseño define como estos requisitos se cumplirán, la estructura que debe darse a la aplicación o sistema Web.
- 3) **Codificación del software:** Durante esta etapa se realizan las tareas que comúnmente conocemos como programación que consiste en llevar a cabo

toda la codificación en lenguaje de programación con todas las características del diseño realizado.

- 4) **Pruebas:** Las pruebas se utilizan para asegurar el correcto funcionamiento de secciones de código.
- 5) **Instalación o fase de implementación:** Es el proceso por el cual los programas desarrollados son transferidos apropiadamente en servidores para ser puestos en producción y luego ser utilizados por el usuario final.
- 6) **El mantenimiento:** Es el proceso de control, mejora y optimización del software ya desarrollado e instalado, en donde también incluye la corrección de errores y defectos que puedan haberse filtrado de la fase de pruebas de control.



Figura 1.4 Fases de la metodología UWE

Recuperado de:

<http://evangellyscarolinacabellorodriguez.blogspot.com/2013/10/fases-o-etapas-de-la-metodologia-uwe.html>

Métodos de control de Inventarios ABC

El análisis ABC es una sistemática de clasificación muy sencilla usada frecuentemente a la hora de diseñar la distribución óptima de inventarios en almacenes. Esta metodología es usada sobre todo en el sector logístico, tiendas y almacenes de stock de todo tipo. Su propósito es optimizar la organización de los

productos de forma que los más solicitados se encuentren al alcance de esta forma reducir tiempos y aumentar la eficiencia.

Análisis del método ABC

Para realizar un análisis ABC primeramente hay que determinar cuáles son los artículos más importantes que tenemos en el almacén. Posteriormente los diferenciamos en 3 grupos:

- **Artículos de tipo A:** Se refieren a los más importantes (los más usados, más vendidos o más urgentes). Suelen ser los que más ingresos dan.
- **Artículos de tipo B:** Son aquellos de menor importancia o de una importancia secundaria.
- **Artículos de tipo C:** Estos son aquellos que carecen de importancia. Muchas veces tenerlos en el almacén cuesta más dinero que el beneficio que aportan.

Una vez hecha la asignación se procederá a colocar los artículos de Tipo A en las zonas más alcanzables: en la entrada del almacén, en la parte delantera de las zonas más transitadas del almacén y del mismo modo los artículos Tipo B y C que son los menos solicitados estarán colocados en las zonas menos accesibles, ya que la necesidad de disponer de ellos es menor

El método ABC permite aumentar la eficiencia de los almacenes al ahorrar tiempo a los encargados a la hora de coger y dejar los artículos, puesto que pueden tener mejor controlados los ítems más solicitados y requerir menos movimientos para gestionarlos. Por último, se puede mejorar aún más esta sistemática con una buena Gestión de stocks que contemple más unidades almacenadas de los productos que tengan más demanda.

Factores de calidad modelo McCall

Este modelo busca reducir la brecha entre usuarios y desarrolladores enfocándose en un número de factores de calidad que reflejen las prioridades de ambos.

El modelo establece tres perspectivas para definir e identificar la calidad de un producto software:

1. Revisión del producto
 - Mantenibilidad.

- Flexibilidad.
 - Verificabilidad (testability).
2. Transición del producto
- Portabilidad.
 - Reusabilidad (en otro contexto).
 - Interoperabilidad.
3. Operaciones del producto
- Corrección (cumplimiento con la especificación).
 - Confiabilidad.
 - Eficiencia (De ejecución y de almacenamiento).
 - Integridad (protección contra accesos no autorizados).
 - Usabilidad.

Modelo constructivo de costos Cocomo II

COCOMO II permite realizar estimaciones en función del tamaño del software, y de un conjunto de factores de coste y de escala.

En los factores de coste se incluyen aspectos relacionados con la naturaleza del sistema, equipo, y características propias del proyecto.

El factor de escala incluye la parte de escala producida a medida que un proyecto de software incrementa su tamaño.

COCOMO II posee tres modelos:

- Composición de Aplicación,
- Diseño Temprano
- Post-Arquitectura.

Cada uno de estos modelos está orientado a sectores específicos del mercado de desarrollo de software y a las distintas etapas del desarrollo de software.

La sectorización de aplicaciones en COCOMO II es esta:

- **Aplicaciones desarrolladas por Usuarios Finales:** En este sector se encuentran las aplicaciones generadas directamente por usuarios finales, mediante la utilización de generadores de aplicaciones tales como hojas de cálculo, sistemas de consultas, etc. Estas aplicaciones

surgen debido al uso masivo de estas herramientas, conjuntamente con la presión actual para obtener soluciones rápidas y flexibles.

- **Generadores de Aplicaciones:** En este sector están los módulos pre-empaquetados que serán usados por usuarios finales y programadores.
- **Aplicaciones con Componentes:** Sector en el que se están aquellas aplicaciones que se resuelven por soluciones pre-empaquetadas, pero son lo suficientemente simples para ser construidas a partir de componentes interoperables. Por ejemplo: interfaces gráficos, administradores de bases de datos, buscadores inteligentes de datos, ... Estas aplicaciones son generadas por un equipo reducido de personas, en pocas semanas o meses.
- **Sistemas Integrados:** Sistemas de gran escala, con un alto grado de integración entre sus componentes, sin antecedentes en el mercado que se puedan tomar como base. Partes de estos sistemas pueden ser desarrolladas a través de la composición de aplicaciones.
- **Infraestructura:** Área que comprende el desarrollo de sistemas operativos, protocolos de redes, sistemas administradores de bases de datos.

Los modelos de COCOMO II se adaptan tanto a los sectores descritos como al tipo y cantidad de información disponible.

- El modelo Composición de Aplicación se emplea en desarrollos de software durante la etapa de prototipado.
- El modelo Diseño Temprano se utiliza en las primeras etapas del desarrollo en las cuales se evalúan las alternativas de hardware y software de un proyecto. En estas etapas se tiene poca información, lo que concuerda con el uso de Puntos Función, para estimar tamaño y el uso de un número reducido de factores de coste.
- El modelo Post-Arquitectura se aplica en la etapa de desarrollo, después de definir la arquitectura del sistema, y en la etapa de mantenimiento.

Modelos de estimación

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

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

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

1.6.2 METODO DE INGENIERIA

Método KANBAN

El método Kanban formulado por David J. Anderson es una aproximación al proceso gradual, evolutivo y al cambio de sistemas para las organizaciones. Utiliza un sistema de extracción limitada del trabajo en curso como mecanismo básico para exponer los problemas de funcionamiento del sistema (o proceso) y estimular la colaboración para la mejora continua del sistema. Un ejemplo del sistema de extracción es el sistema Kanban, y es después de esta popular forma de trabajo en curso, que se ha denominado el método.

Principios de KANBAN

El método contempla 4 principios básicos:

- Comience por lo que a hacer ahora.
- Se acuerda perseguir el cambio incremental y evolutivo.
- Respetar el proceso actual, los roles, las responsabilidades y los cargos.
- Liderazgo en todos los niveles.

Prácticas centrales del método kanban.

Anderson identificó cinco características básicas que habían sido observadas en cada implementación correcta del método Kanban. Posteriormente fueron etiquetadas como prácticas y se ampliaron con la adición de una sexta característica.

1. Visualizar

Visualizar el flujo de trabajo y hacerlo visible es la base para comprender cómo avanza el trabajo.

2. Limitar el trabajo en curso

Limitar el trabajo en curso implica que un sistema de extracción se aplica en la totalidad o parte del flujo de trabajo.

3. Dirigir y gestionar el flujo

Se debe supervisar, medir y reportar el flujo de trabajo a través de cada estado. Al gestionar activamente el flujo, los cambios continuos, graduales y evolutivos del sistema pueden ser evaluados para tener efectos positivos o negativos.

4. Hacer las Políticas de Proceso Explícitas

Configure las reglas y directrices de su trabajo. Entienda las necesidades y asegúrese de seguir las reglas.

5. Utilizar modelos para reconocer oportunidades de mejora

Cuando los equipos tienen un entendimiento común de las teorías sobre el trabajo, el flujo de trabajo, el proceso y el riesgo, es más probable que sea capaz de construir una comprensión compartida de un problema y proponer acciones de mejora que puedan ser aprobadas por consenso.

Arquitectura de desarrollo en (N) capas

La programación por capas es una arquitectura cliente-servidor en el que el objetivo primordial es la separación de la lógica de negocios de la lógica de diseño; un ejemplo básico de esto consiste en separar la capa de datos de la capa de presentación al usuario.

La ventaja principal de este estilo es que el desarrollo se puede llevar a cabo en varios niveles y, en caso de que sobrevenga algún cambio, solo se ataca al nivel requerido sin tener que revisar entre código mezclado.

El más utilizado actualmente es el diseño en tres niveles (o en tres capas), las cuales son:

1. **Capa de presentación:** la que ve el usuario (también se la denomina "capa de usuario"), presenta el sistema al usuario, le comunica la información y captura la información del usuario en un mínimo de proceso (realiza un filtrado previo para comprobar que no hay errores de formato). También es conocida como interfaz gráfica y debe tener la característica de ser "amigable" (entendible y fácil de

usar) para el usuario. Esta capa se comunica únicamente con la capa de negocio.

2. **Capa de negocio:** es donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio (e incluso de lógica del negocio) porque es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos almacenar o recuperar datos de él. También se consideran aquí los programas de aplicación.
3. **Capa de datos:** es donde residen los datos y es la encargada de acceder a los mismos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

Sistemas de información Geográfica

Un sistema de información geográfica (SIG) es un sistema empleado para describir y categorizar la Tierra y otras geografías con el objetivo de mostrar y analizar la información a la que se hace referencia espacialmente. Este trabajo se realiza fundamentalmente con los mapas.

El objetivo de SIG consiste en crear, compartir y aplicar útiles productos de información basada en mapas que respaldan el trabajo de las organizaciones, así como crear y administrar la información geográfica pertinente.

Los mapas representan colecciones lógicas de información geográfica como capas de mapa. Constituyen una metáfora eficaz para modelar y organizar la información geográfica en forma de capas temáticas. Asimismo, los mapas SIG interactivos ofrecen la interfaz de usuario principal con la que se utiliza la información geográfica.

Un mapa SIG es una ventana interactiva a toda la información geográfica y datos descriptivos, y a ricos modelos de análisis espacial creados por profesionales de SIG.

Los mapas SIG se usan para:

- Dar a conocer y compartir el SIG
- Compilar y mantener el contenido del SIG
- Diseñar y organizar la información geográfica por medio de capas temáticas
- Obtener nueva información mediante geoprocesamiento y, posteriormente, visualizar, resumir, analizar, comparar e interpretar los resultados analíticos
- Compartir la información geográfica para su uso en la Web

En un SIG, el mapa es la interfaz.

1.7 HERRAMIENTAS

Las herramientas para el ámbito del desarrollo de software son variadas presentando innovaciones y ventajas dependiendo el tipo de herramienta.

Se realiza una comparación de lenguajes de programación con una descripción general de dicho lenguaje como se muestra a continuación.

Tabla 1.1
Tabla comparativa de lenguajes de programación

Tabla comparativa de lenguajes de programación.		
LENGUAJE	VENTAJAS	DEBILIDADES
PHP	<ul style="list-style-type: none"> • Es un lenguaje multiplataforma. • Completamente orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una Base de Datos. • El código fuente escrito en PHP es invisible al navegador y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable. • Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL. • Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones). 	<ul style="list-style-type: none"> • Debes saber cuándo menos HTML para poder hacer un trabajo medianamente funcional • Si no lo configuras correctamente dejas abiertas muchas brechas de seguridad • Se necesita instalar un servidor web. • La ofuscación es una técnica que puede dificultar la lectura del código pero no la impide y, en ciertos casos, representa un costo en tiempos de ejecución.

<p>REACT</p>	<ul style="list-style-type: none"> • Es fácil saber cómo se representa un componente, simplemente mira la función de renderizado. • JSX hace que sea fácil leer el código de tus componentes. También es realmente fácil ver el <i>layout</i> o cómo los componentes se enchufan / combinan entre sí. • puedes renderizar React en el lado del server. • es fácil de probar, y también puede integrar algunas herramientas como broma. • asegura la legibilidad y facilita el mantenimiento. • puede usar Reaccionar con cualquier marco (Backbone.js, Angular.js) ya que solo es una capa de vista. 	<ul style="list-style-type: none"> • Es solo una capa de vista, aún tiene que conectar su código para request es de Ajax, events, etc. Algunas personas se sorprenden por eso. • la biblioteca en sí es bastante grande. • la curva de aprendizaje puede ser pronunciada.
<p>ANGULAR CLI ANGULAR JS</p>	<ul style="list-style-type: none"> • Angular menciona que el tamaño (en kb) de un componente se ha reducido un mínimo de un 60% y dependiendo de lo complejo que sea el componente aún mayor será la mejora, lo cual es un avance sustancial. • Se trata de un súper conjunto de JavaScript. Es decir, añade sobre el lenguaje JavaScript una serie de nuevas funcionalidades que le aplican mayor potencia y facilidad en el desarrollo. • Cualquier código desarrollado en TypeScript, una vez compilado, se traduce en código JavaScript. 	<ul style="list-style-type: none"> • Por lo general debe combinarse con otros lenguajes para poder diseñar un sitio web completo. • Algunos buscadores no son capaces de acceder al contenido Javascript de la página lo que dificulta su posicionamiento. • Necesita descargarse en su totalidad para ser interpretado adecuadamente. • Tiene algunos riesgos de seguridad
<p>PYTHON</p>	<ul style="list-style-type: none"> • Simplificado y rápido • Elegante y flexible • Programación sana y productiva • Ordenado y limpio • Portable • Comunidad 	<ul style="list-style-type: none"> • Curva de aprendizaje La “curva de aprendizaje cuando ya estás en la parte web no es tan sencilla”. • Hosting La mayoría de los servidores no tienen soporte a Python, y si lo soportan, la configuración es un poco difícil.

		<ul style="list-style-type: none"> • Librerías incluidas Algunas librerías que trae por defecto no son del gusto de amplio de la comunidad
ASP.NET	<ul style="list-style-type: none"> • Admite la programación con Visual Basic Script lo que facilita su implementación. • Soporta el lenguaje JScript de Microsoft (una especie de Javascript). • El lenguaje ASP.NET soporta programación C#, VB.NET y J#. Es un lenguaje completamente orientado a objetos. • Se comunica de forma impecable con las bases de datos SQL Server. • ASP.NET ha incrementado notablemente la velocidad y seguridad de los sistemas frente a su hermano mayor ASP. 	<ul style="list-style-type: none"> • Es un código propietario de Microsoft no abierto. • No es fácil de leer e interpretar, se necesita escribir más código para hacer lo mismo que con otros lenguajes como el PHP. • Muchos hostings y alojamientos web no lo soportan por su alto coste. ASP.NET necesita tener instalado IIS con el Framework .Net. • El consumo de recursos de ASP.NET es importante por lo que se requieren servidores de mayor capacidad.

Fuente: Elaboración propia

Para el desarrollo del proyecto se utilizarán dos tipos de lenguaje para el Back-End el lenguaje PHP con el framework codeigniter, para el lado del Front-End el lenguaje Angular CLI, los cuales se adaptan al proyecto en base a sus características que ofrecen los mismos.

El siguiente cuadro nos muestra una descripción de las bases de datos, de los cuales se evaluarán para realizar el desarrollo adecuado de acuerdo a las necesidades que presenta el proyecto.

**Tabla.1.2
Cuadro comparativo de Bases de datos.**

Tabla comparativa de bases de datos		
Base de datos	Ventajas	Desventajas
MariaDB	<ul style="list-style-type: none"> • Mejoras de velocidad sobre todo en consultas complejas cuando se usa el motor de almacenamiento. • Se añaden nuevas tablas de sistema (INFORMATION_SCHEMA) para almacenar estadísticas que nos 	<ul style="list-style-type: none"> • La migración de un sistema muy testado y fiable como es MySQL. Aunque se garantiza una absoluta compatibilidad, y un gran sistema estable y potente, hay que valorar si a

	<p>pueden ayudar a optimizar las bases de datos.</p> <ul style="list-style-type: none"> • En general se han hecho muchas modificaciones para mejorar el rendimiento, velocidad e incluso implementar características nuevas. 	<p>día de hoy merece pasar todo mi sistema a MariaDB.</p>
MySQL	<ul style="list-style-type: none"> • Tablas para almacenar los datos. • Consultas para buscar y recuperar únicamente los datos que necesita. • Formularios para ver, agregar y actualizar los datos de las tablas. • Informes para analizar o imprimir los datos con un diseño * específico. • Páginas de acceso a datos para ver, actualizar o analizar los datos de la base de datos desde Internet. • Almacenar los datos una vez en una tabla y verlos desde varios lugares. 	<ul style="list-style-type: none"> • Tiene limitaciones en el procesamiento de las búsquedas, además que si la quieres usar para ambientes corporativo no te va a servir por su poca estabilidad.
SQL server	<ul style="list-style-type: none"> • Es un sistema de gestión de base de datos. • Es útil para manejar y obtener datos de la red de redes. • Nos permite olvidarnos de los ficheros que forman la base de datos. • Si trabajamos en una red social nos permite agregar otros servidores de SQL Server. Por ejemplo, dos personas que trabajan con SQL Server, uno de ellos se puede conectar al servidor de su otro compañero y así se puede ver las bases de datos del otro compañero con SQL Server. • SQL permite administrar permisos a todo. 	<ul style="list-style-type: none"> • Utiliza mucho la memoria RAM para las instalaciones y utilización de software. • No se puede utilizar como practicas porque se prohíben muchas cosas, tiene restricciones en lo particular. • La relación, calidad y el precio está muy debajo comparado con oracle. • Tiene muchos bloqueos a nivel de página, un tamaño de página fijo y demasiado pequeño, una pésima implementación de los tipos de datos variables.

Fuente: Elaboración propia.

De acuerdo al análisis de cuadro comparativo se ara el uso de MariaDB como base de datos ya que se adecua a las necesidades del proyecto como de la empresa.

1.8 LIMITES Y ALCANCES

1.8.1 Alcances

Los alcances se limitan a los siguientes módulos:

- Módulo de empleados.
- Módulo de clientes.
- Módulo de proveedores.
- Módulo de productos.
- Módulo de inventario (Almacén, Movimiento).
- Módulo de operaciones (planeación, distribución)
- Implementar políticas de accesos y resguardo de la información.

1.8.2 Limites

De acuerdo a los requerimientos y necesidades de la empresa, el presente proyecto se limitará en la construcción de módulos de control operativo, los cuales son descritos en los alcances.

El proyecto en cuanto a empleados solo se limita a realizar el control de asistencia, y brindar información básica del personal, el módulo de comisiones solo se limita a mostrar la información solicitada no así implicando planillas de pago, afp's seguro de salud, ya que solo es un apoyo a gestionar y organizar el personal. El sistema no contempla aspectos de administración financiera.

1.9 APORTES

El aporte que se realiza a la empresa ADIPHARM S.R.L. es el sistema el cual permitirá una mejora en cuanto a la gestión operativa optimizando tiempos y costos así elevar su productividad con la finalidad de proporcionar mejoras económicas a la empresa.

También permitiéndole ser más competitivo en el área en cuanto al mercado.

No solo el aporte va a orientado a la funcionalidad, sino que también al software ya que el diseño y arquitectura de software permite ser altamente escalable ya que el sistema en una segunda versión puede llegar a ser un sistema ERP.

CAPITULO II MARCO TEORICO

2.1 METODOLOGIA UWE

Partiendo de páginas estáticas, la Web ha evolucionado incorporando elementos de seguridad, optimización, concurrencia y demás requerimientos que son necesarios para crear soluciones sólidas. Sin embargo, el desarrollo de una aplicación Web incluye elementos que no son comunes a una aplicación de escritorio.

Esto requiere cambios importantes en la forma de realizar y controlar el proceso de desarrollo. Es decir, pasar de una Ingeniería de Software a una Ingeniería Web. UWE es una metodología que permite especificar de mejor manera una aplicación Web en su proceso de creación, mantiene una notación estándar basada en el uso de UML (*Unified Modeling Language*) para sus modelos y sus métodos, lo que facilita la transición. La metodología define claramente la construcción de cada uno de los elementos del modelo.

UWE es un Método de Ingeniería de Software para el desarrollo de Web basado en UML, es decir es una herramienta basada en UML: una metodología dirigida por modelos, pero para aplicaciones Web, esto conlleva a que cualquier diagrama UML puede ser usado, debido a que es una extensión de UML. Esta metodología nace con la finalidad de controlar el caos que han provocado en el pasado procesos de desarrollo y así proporcionar un proceso sistemático orientado a una aplicación final de calidad. UWE es una metodología muy detallada para el desarrollo de aplicaciones que tiene una definición exacta del proceso de diseño que debe ser utilizado.

Este proceso presenta flujos de trabajo y puntos de control, y sus fases son las mismas a las propuestas en el Proceso Unificado de Modelado. UWE está dirigido específicamente a aplicaciones adaptativas, basándose en características de personalización como: la definición de un modelo de usuario en función de las preferencias, conocimiento o tareas de usuario.

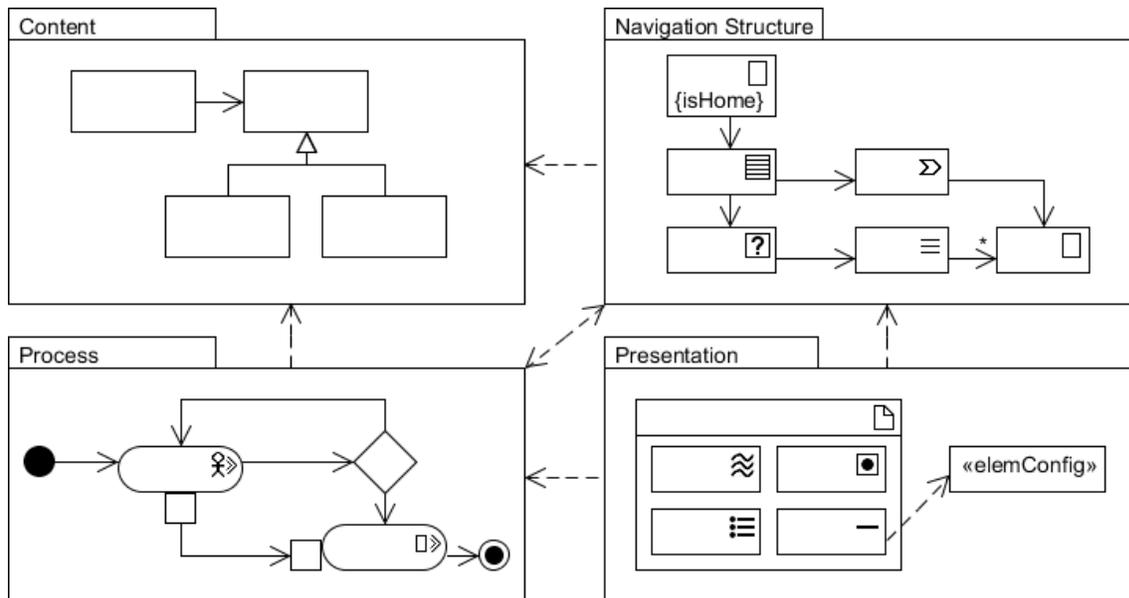


Figura 2.1 Descripción general de los modelos UWE
 Recuperado de: <https://uwe.pst.ifi.lmu.de/aboutUwe.html>

2.1.1 Actividades de modelado de UWE.

Las actividades base de modelado de UWE son el análisis de requerimientos, el modelo conceptual, el modelo navegacional y el modelo de presentación. A estos modelos se pueden sumar otros modelos como lo son el modelo de interacción y la visualización de Escenarios Web. El modelo que propone UWE está compuesto por etapas o sub-modelos:

- Modelo de Casos de Uso
- Modelo de Contenido
- Modelo de Usuario
- Modelo de estructura o presentación.
- Modelo Abstracto
- Modelo de Adaptación

Modelo de requerimientos. UWE apunta a construir un modelo conceptual de una aplicación Web, procura no hacer caso en la medida de lo posible de cuestiones relacionadas con la navegación, y de los aspectos de interacción de la aplicación Web. La construcción de este modelo lógico-conceptual se debe llevar

a cabo de acuerdo con los casos de uso que se definen en la especificación de requerimientos. El modelo conceptual incluye los objetos implicados en las actividades típicas que los usuarios realizarán en la aplicación Web.

- **Casos de uso:** En UWE, los casos de uso se distinguen por los estereotipos y para indicar si una aplicación modifica los datos persistentes de la aplicación o no.
- **Ocupaciones:** Dado que solo se puede capturar poca información con casos de uso, cada caso de uso se puede describir con mayor precisión mediante un flujo de proceso. Por lo tanto, se pueden modelar las acciones dentro de un caso de uso, así como los datos presentados al usuario y la entrada requerida.

Durante la ingeniería de requisitos, generalmente también es útil especificar qué datos se presentan dónde y cuándo. Para el modelado de grupos de presentación en UWE se usa el estereotipo “Action”, mientras que los dos estereotipos de pin de acción “display Pin” y “pin de interacción” se utilizan para modelar los elementos de salida y entrada. Finalmente está el estereotipo “acción de navegación”, que puede usarse para modelar las opciones de navegación y los elementos de presentación asociados.

- **Transformaciones:** Una vez que se modelan los requisitos, hay dos formas en que se puede simplificar el desarrollo de otros modelos (contenido, navegación, presentación y proceso).
 - 1.- En lugar de crear un modelo y el diagrama correspondiente manualmente, también se puede generar con una transformación a partir de los datos del modelo de requisitos.
 - 2.- Además, un modelo generado previamente puede extenderse mediante nuevas clases transformadas a partir del modelo de requisitos o agregando nuevos datos dependientes del modelo a las clases existentes.

Modelo de navegación. Consta de la construcción de dos modelos de navegación, el modelo del espacio de navegación y el modelo de la estructura de navegación. El primero especifica que objetos serán visitados por el navegador a través de la aplicación. El segundo define como se relacionarán.

En una aplicación para la Web es útil saber cómo están enlazadas las páginas. Ello significa que se requiere un diagrama de navegación con nodos y enlaces. Este diagrama se modela con base en el análisis de los requisitos y el modelo de contenido. UWE provee diferentes estereotipos para el modelado de navegación, en la figura 5 se presentan los estereotipos de estructura de navegación, muestran vínculos directos entre las clases de navegación y representan posibles pasos a seguir por el usuario y, por lo tanto, estos vínculos tienen que ser dirigidos.



Figura 2.2 Nombres de estereotipos y sus iconos de modelo de navegación

Recuperado de: <https://uwe.pst.ifi.lmu.de/>

Modelo de presentación. Describe dónde y cómo los objetos de navegación y accesos primitivos serán presentados al usuario, es decir, una representación esquemática de los objetos visibles al usuario.

El modelo de presentación ofrece una visión abstracta de la interfaz de usuario de una aplicación Web. Se basa en el modelo de navegación y en los aspectos concretos de la interfaz de usuario (IU). Describe la estructura básica de la IU, es decir, ¿qué elementos de interfaz de usuario (por ejemplo, texto, imágenes, enlaces, formularios) se utilizan para presentar los nodos de navegación? Su ventaja es que es independiente de las técnicas actuales que se utilizan para implementar un sitio Web, lo que permite a las partes interesadas discutir la conveniencia de la presentación antes de que realmente se aplique.



Figura 2.3 Nombres de estereotipos y sus iconos del modelo de presentación

Recuperado de: <https://uwe.pst.ifi.lmu.de/teachingTutorialPresentation.html>

Modelo de procesos. La estructura de navegación puede ser extendida mediante clases de procesos que representan la entrada y la salida de procesos de negocio. El modelo del proceso representa el aspecto que tienen las acciones de las clases de proceso. En este modelo se tienen dos tipos de modelos:

1. **Modelo de estructura del proceso**, que describe las relaciones entre las diferentes clases de proceso,
2. **Modelo de flujo del proceso**, que especifica las actividades conectadas con cada clase de procesos.

Modelo de estructura del proceso. Es representado por un diagrama de clases donde se describen las relaciones entre las diferentes clases de proceso.

Modelo del flujo del proceso. Siguiendo el principio de la utilización de UML se han refinado los requisitos con los diagramas de actividad UML. Los diagramas de actividades incluyen actividades, actores responsables de estas actividades (opcional) y elementos de flujo de control. Ellos pueden ser enriquecidos con flujos de objetos que muestran objetos relevantes para la entrada o salida de esas actividades.

Interacción temporal. Presenta los objetos que participan en la interacción y la secuencia de los mensajes enviados entre ellos.

Escenarios Web. Permiten detallar la parte dinámica del modelo de navegación, especificando los eventos que disparan las situaciones, definen condiciones y explícitamente incluyen las acciones que son realizadas. Junto con el modelo de

interacción temporal, los escenarios Web proveen la representación funcional dinámica del modelo de navegación.

Diagramas. Los diagramas usados por UWE, son diagramas UML puro. Entre los más importantes tenemos: Diagramas de estado, de Secuencia, de colaboración y diagramas de Actividad.

2.1.2 Fases o etapas de la metodología UWE.

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

- **Captura, análisis y especificación de requisitos:** En simple palabras y básicamente, durante esta fase, se adquieren, reúnen y especifican las características funcionales y no funcionales que deberá cumplir la aplicación web.
- Trata de diferente forma las necesidades de información, las necesidades de navegación, las necesidades de adaptación y las de interfaz de usuario, así como algunos requisitos adicionales. Centra el trabajo en el estudio de los casos de uso, la generación de los glosarios y el prototipo de la interfaz de usuario.
- **Diseño del sistema:** Se basa en la especificación de requisitos producido por el análisis de los requerimientos (fase de análisis), el diseño define cómo estos requisitos se cumplirán, la estructura que debe darse a la aplicación web.
- **Codificación del software:** Durante esta etapa se realizan las tareas que comúnmente se conocen como programación; que consiste, esencialmente, en llevar a código fuente, en el lenguaje de programación elegido, todo lo diseñado en la fase anterior.
- **Pruebas:** Las pruebas se utilizan para asegurar el correcto funcionamiento de secciones de código.
- **La Instalación o Fase de Implementación:** Es el proceso por el cual los programas desarrollados son transferidos apropiadamente al computador

destino, inicializados, y, eventualmente, configurados; todo ello con el propósito de ser ya utilizados por el usuario final.

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

- **El Mantenimiento:** es el proceso de control, mejora y optimización del software ya desarrollado e instalado, que también incluye depuración de errores y defectos que puedan haberse filtrado de la fase de pruebas de control.

2.2 ARQUITECTURA DE DESARROLLO EN (N) CAPAS

El Patrón de arquitectura por capas es una de las técnicas más comunes que los arquitectos de software utilizan para dividir sistemas de software complicados. Al pensar en un sistema en términos de capas, se imaginan los principales subsistemas de software ubicados de la misma forma que las capas de un pastel, donde cada capa descansa sobre la inferior. En este esquema la capa más alta utiliza varios servicios definidos por la inferior, pero la última es inconsciente de la superior. Además, normalmente cada capa oculta las capas inferiores de las siguientes superiores a esta.

Los beneficios de trabajar un sistema en capas son:

- Se puede entender una capa como un todo, sin considerar las otras.
- Las capas se pueden sustituir con implementaciones alternativas de los mismos servicios básicos
- Se minimizan dependencias entre capas.
- Las capas posibilitan la estandarización de servicios
- Luego de tener una capa construida, puede ser utilizada por muchos servicios de mayor nivel.

La imagen que se muestra a continuación presenta el esquema de una arquitectura siguiendo este patrón:

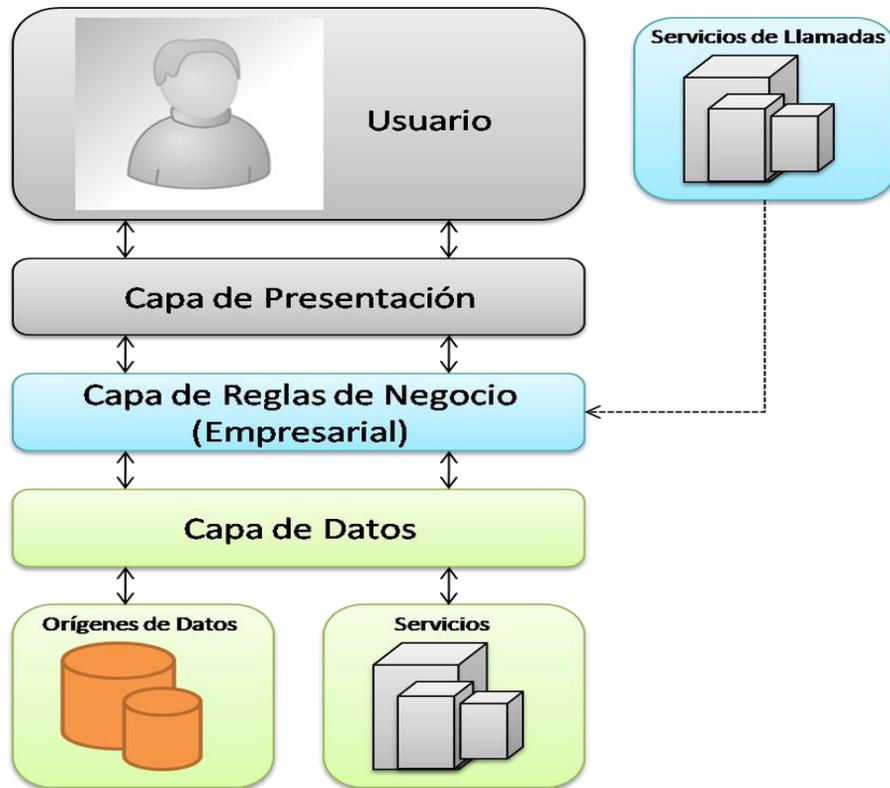


Figura 2.4 Arquitectura de software (n) capas

Recuperado de: <https://arevalomaria.wordpress.com/2010/12/02/introduccion-al-patron-de-arquitectura-por-capas/>

A continuación, se describen las tres capas principales de un patrón de arquitectura por capas:

2.2.1 Capa de Presentación

Referente a la interacción entre el usuario y el software. Puede ser tan simple como un menú basado en líneas de comando o tan complejo como una aplicación basada en formas. Su principal responsabilidad es mostrar información al usuario, interpretar los comandos de este y realizar algunas validaciones simples de los datos ingresados.

2.2.2 Capa de Reglas de Negocio (Empresarial)

También denominada Lógica de Dominio, esta capa contiene la funcionalidad que implementa la aplicación. Involucra cálculos basados en la información dada por el usuario y datos almacenados y validaciones. Controla la ejecución de la capa de acceso a datos y servicios externos. Se puede diseñar la lógica de la capa de

negocios para uso directo por parte de componentes de presentación o su encapsulamiento como servicio y llamada a través de una interfaz de servicios que coordina la conversación con los clientes del servicio o invoca cualquier flujo o componente de negocio.

2.2.3 Capa de Datos

Esta capa contiene la lógica de comunicación con otros sistemas que llevan a cabo tareas por la aplicación. Estos pueden ser monitores transaccionales, otras aplicaciones, sistemas de mensajerías, etc. Para el caso de aplicaciones empresariales, generalmente está representado por una base de datos, que es responsable por el almacenamiento persistente de información. Esta capa debe abstraer completamente a las capas superiores (negocio) del dialecto utilizado para comunicarse con los repositorios de datos (PL/SQL, *Transact-SQL*, etc.).

2.2.4 Capa de Implementación del Servicio:

En esta capa se tienen la clase adaptador que es la encargada de intermediar la comunicación entre la capa de presentación y la de lógica de negocio. Como la lógica de negocio solo maneja entidades de negocio y la capa de presentación solo maneja tipos de datos, existe una clase traductor que se encarga de traducir las entidades en tipos de datos y viceversa

Partiendo de que un servicio Web o *Web Service* es una colección de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Los estándares abiertos y el foco en la comunicación y colaboración entre las personas y aplicaciones han creado un entorno donde los servicios Web se están convirtiendo en la plataforma para la integración de aplicaciones. Estas aplicaciones de software pueden ser desarrolladas en lenguajes de programación diferentes y ejecutadas sobre cualquier plataforma, y mediante los servicios poder intercambiar datos entre ellas a través de la Internet.

2.3 METODO DE CONTROL DE INVENTARIOS

La Administración de Almacenes, es la función administrativa que comprende actividades y procedimientos relativos al ingreso, salida, almacenamiento, distribución, medidas de salvaguarda y control de los bienes.

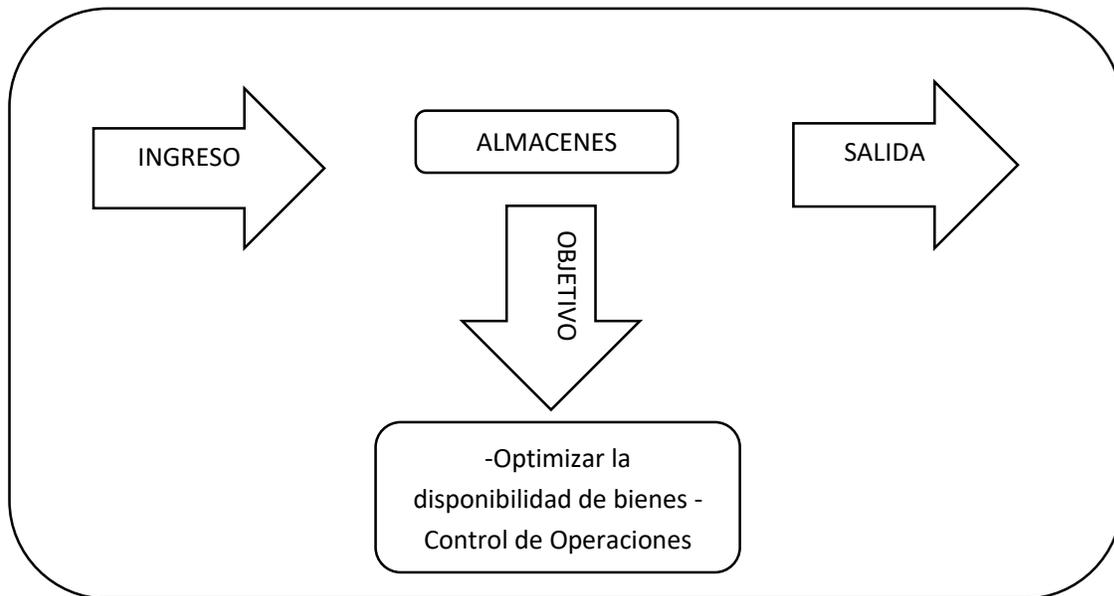


Figura 2.5 Sistema de Administración de Almacenes
Fuente: Propia

2.3.1 Entrada o ingreso de productos

La primera fase del ingreso de bienes a la entidad es la recepción. Toda recepción de bienes estará basada en documentos que autoricen su ingreso, emitidos por la autoridad competente o respaldo por la solicitud de su adquisición. La recepción comprende las siguientes tareas:

- El cotejamiento de la documentación pertinente con lo efectivamente solicitado por la entidad.
- La verificación de la cantidad y de los atributos técnicos, físicos, funcionales o de volumen de los bienes.

2.3.2 Salida o egreso de productos

La salida de almacenes es la distribución o entrega física de los bienes, que implica:

- Atender las solicitudes de bienes, cuando se encuentren debidamente autorizadas por una solicitud o autorización escrita por la autoridad competente.

- Comprobar que la calidad, cantidad y características de los bienes a entregar correspondan a lo solicitado

2.3.3 Importancia del control de la administración de almacenes

Toda empresa debe tener un inventario, es eso que cuando se tiene un negocio con una gran cantidad de productos, una de las cosas más importantes es tener el control de ésta, qué se tiene, qué falta, qué excedente hay, y aun si nuestro negocio no maneja una gran cantidad de productos, el control de inventarios se vuelve indispensable para que pueda fluir la empresa sin retrasos y, sobre todo, sin pérdidas.

Los 3 aspectos más importantes a tener en cuenta son: “saber qué tienes, cuánto tienes y cuánto te falta”. Al saber qué se tiene, se puede determinar las cantidades de productos suficientes para abastecer a la demanda, si se está en riesgo de quedarse sin producto, si es necesario reforzar los pedidos para equilibrar tus productos, etc. Al saber qué hace falta, se toman las medidas para volver a abastecerse, para hacer los pedidos correspondientes, para saber qué decirle al cliente cuando pregunte por algo que no se tenga.

2.3.4 Método de control de inventarios ABC.

El método ABC es la clasificación frecuente que se utiliza en la gestión de un determinado inventario, que parte del resultado del Principio de Pareto. Este análisis puede identificar los artículos que impactan significativamente el valor global de inventario, de costos y de ventas, permitiendo crear categorías de productos que necesitan modalidades diferentes.

Es de gran utilidad y tiene la capacidad de agilizar diversos procesos de almacenamiento de mercancías en las empresas sean pequeñas o grandes. Es una herramienta de análisis de inventarios favorable para los empleados pertenecientes al departamento de logística y transporte, los cuales se encargan de organizar y gestionar el almacén, donde se puede especificar la relación desigual que existe entre las entradas y salidas.

2.3.5 Clasificación del método ABC para inventarios

Con este método la presencia del inventario, se puede llegar a dividir en tres tipos de clases, las que a continuación se detallarán:

Clase A

Estas se pueden ubicar en los productos de máxima importancia, por lo tanto, se les da mayor atención que a las otras zonas. Generalmente estas presentan el 80% de la valoración del consumo anual de la empresa y el 15% de la totalidad de unidades que se encuentra en el inventario.

Clase B

En este caso se habla del segundo artículo de consumo medio. Aquí se encuentran los productos que suelen representar el 15% del valor del consumo anual y también constituye el 30% o 40% de la totalidad de los artículos de inventario. A pesar de no son tan atendidos de la forma que se atienden los de la clase A, la existencia y los costos de los mismos se controlan constantemente.

Clase C

En esta zona se encuentran los productos de menor importancia, así que se les hace muy poca supervisión. Estos representan el 15% de la valoración total de consumo y el 40% o 50% de la valoración de los artículos que están en el almacén.

2.3.6 Ventajas del método ABC

Entre las ventajas más importantes de este método, se encuentran:

- Jerarquiza diversos costos de los productos y corrige los beneficios previos de los atributos de los productos que tienen bajo volumen.
- El análisis de los beneficios, predice nuevas perspectivas vinculadas con el comportamiento de los costos.
- Se aumenta la utilidad y la credibilidad de la información de los costos, en la toma de decisiones.
- Implanta el total de calidad.
- Elimina las actividades que no tengan valor de producto.
- Utiliza la cadena de valor, como una herramienta de competencia.

2.3.7 Desventajas del método ABC

Entre las desventajas de este método están:

- No es sencillo poder seleccionar las actividades que se reflejen en el comportamiento de los costos.
- Su implantación suele ser costosa.
- Generalmente no distingue con claridad la causa de una determinada actividad.

2.4 METODOLOGIA KANBAN

Kanban es una palabra japonesa que viene a significar cartel o panel, elemento clave de este método productivo. El sistema Kanban como tal surgió en Toyota, el fabricante japonés de automóviles, para organizar mejor su producción de vehículos dividiendo el proceso en fases bien delimitadas que se tenían que cubrir correctamente para pasar a la siguiente fase, garantizando así un producto de calidad. De este sistema, aplicado a la industria de la automoción, surgió el método Kanban, ideado por David J. Anderson y que adapta la idea original al desarrollo de software, un proceso con muchos puntos en común con el industrial, con diferentes fases, equipos de trabajo y el requisito de que cada pieza del programa a crear funcione correctamente y sea de la mejor calidad posible. El método Kanban en su versión moderna aplicada al software se usó por primera vez en Microsoft, y desde entonces ha sido aplicado en cientos de proyectos de todo el mundo.

Kanban es un administrador de proyectos sencillo y elegante, que ayuda a mejorar y controlar los procesos de desarrollo de software tanto como trabajos de equipo o solo para una persona, es bastante adaptable dependiendo las necesidades del proyecto.

Kanban sobre todo es un sistema de gestión que te indica que producir, cuando producir y cuanto producir. En la actualidad se convertido en una metodología crucial en el mundo del desarrollo de software.

¿Qué es Kanban? ¿En qué consiste? Hay dos objetivos que rigen este método productivo: por un lado, lograr un **producto de calidad**, al obligar a cada fase del

proyecto a finalizar su tarea correctamente, y **acabar con el caos, saturación o cuello de botella** que puede darse en una fase del proyecto en condiciones normales en las que prima la rapidez por encima de la calidad del producto.

2.4.1 Kanban en el desarrollo de software

En el proceso de desarrollo de software kanban se orienta a la optimización de recursos y la optimización de trabajo que se está realizando, ya que es una metodología que fácilmente se adapta y simplifica los procesos de producción.

Kanban maneja cuatro principios para conseguir sus propósitos:

Limita el trabajo en curso: Si te multiplicas para realizar muchas tareas y las dejas a medias no tiene ningún sentido. Un principio básico de este método es terminar lo que empiezas antes de iniciar otra tarea.

Gestiona el flujo de trabajo: Se tiene que controlar su funcionamiento. Observar en todo momento si todo evoluciona correctamente o si alguna persona del equipo tiene dificultades y necesita ayuda para solucionarlas.

Marca las reglas del proceso: Si se quiere aplicar bien este método, lo primero es entenderlo. Tan importante es conocer quien hace que, como también las personas conozcan la manera de realizar su trabajo y entiendan las reglas o especificaciones.

Mejora en equipo: La mejora continua es uno de los pilares del método kanban. En ese aspecto el progreso tiene que acordar con el equipo que está haciendo el uso de la experiencia de todos los componentes.

2.4.2 Tablero de kanban.

El tablero kanban es la herramienta para visualizar y mapear el flujo de trabajo y unos de los elementos fundamentales de esta metodología. Se puede emplear desde un tablero de corcho, pizarra blanca o emplear algún software específico para esa labor. Lo importante es poder dividir un espacio en filas y columnas lo suficiente como para abarcar el proyecto.

Las columnas representan una fase del proyecto y las filas simbolizan los distintos tipos de actividades específicas (diseño, implementación, depuración de errores etc.).

El tablero muestra en tres secciones básicas el estado de las tareas:

- Por hacer o pendiente.
- En proceso o en curso.
- Terminado o hecho.

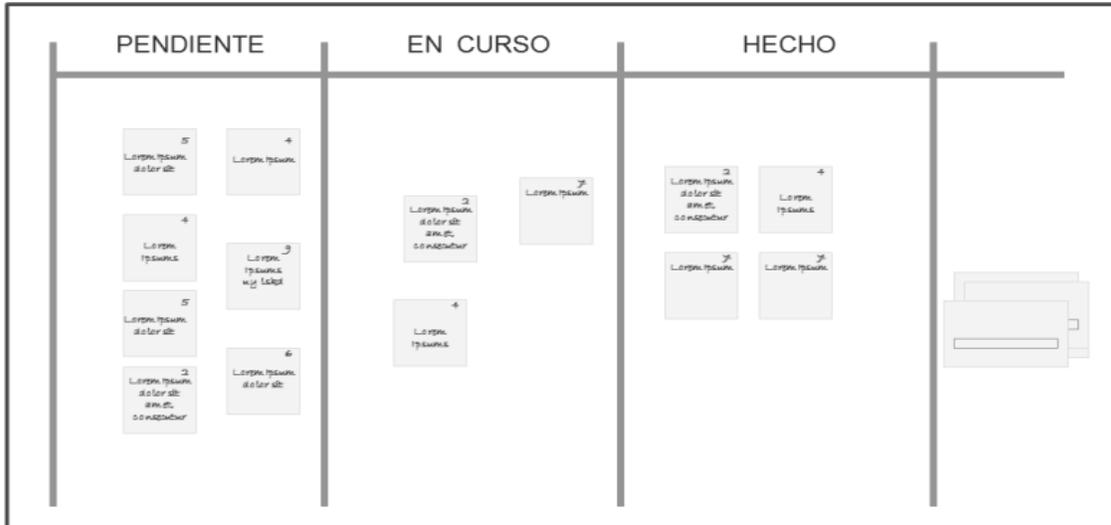


Figura 2.6 Tablero básico de Kanban
 Recuperado de: <https://www.scrummanager.net>

En algunos casos es conveniente incluir estados adicionales (por ejemplo: testado, validado). El orden de los trabajos desde el área “pendiente”, refleja la secuencia de tareas prevista, según sus prioridades. Los trabajos monitorizados pueden ser tareas, historias de usuario, según el uso al que se dedique el tablero.

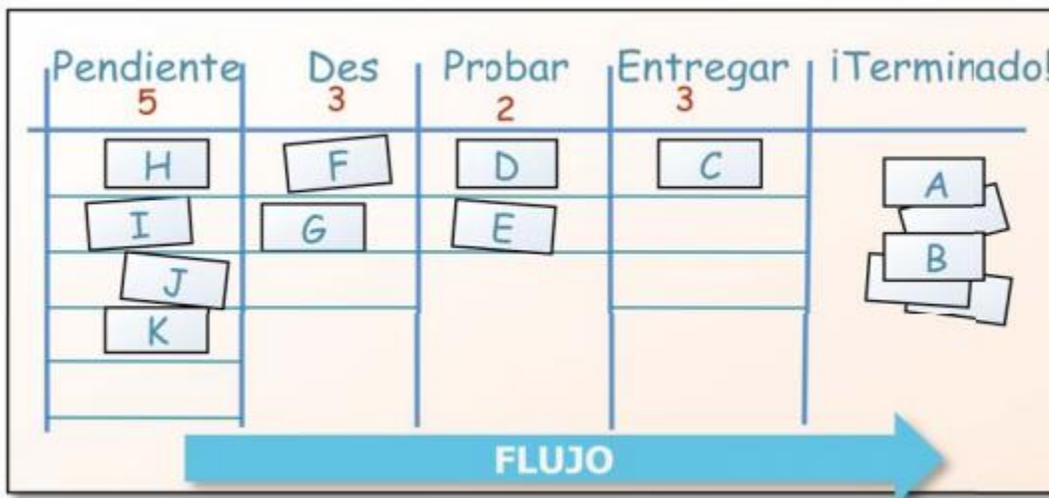


Figura 2.7 Tablero de kankan con más columnas.
 Recuperado de: *kanban vs Scrum lo mejor de ambos*
 C4Media, editores de InfoQ.com. 2010.

En el tablero kanban visualiza el flujo de trabajo o Divide el trabajo en bloques, escribe cada elemento en una tarjeta y ponlo en el muro. o Utiliza columnas con nombre para ilustrar dónde está cada elemento en el flujo de trabajo.

- Limita el WIP (Work in Progress, trabajo en proceso) - asigna límites concretos a cuántos elementos pueden estar en progreso en cada estado del flujo de trabajo.
- Mide el lead time (tiempo medio para completar un elemento, a veces llamado "tiempo de ciclo"), optimiza el proceso para que el lead time sea tan pequeño y predecible como sea posible

2.4.3 Principios de Kanban

- **Empieza con lo que haces ahora:** Kanban es un método de producción, no así un sistema que te dice cómo hacer tu trabajo. Cada integrante o persona sabe cómo realizar su trabajo, kanban te ayuda decidir si lo estas haciendo bien o si hay que cambiar algo.
- **Acepta el cambio:** Kanban apuesta por algo así “si algo no funciona, cámbialo” o “si algo puede funcionar mejor, mejóralo”. Siguiendo el método kanban todos los integrantes de equipo tienen que estar dispuestos a aplicar cambios constantes para mejora sus rutinas de trabajo.
- **Respetar el proceso en curso y los roles y responsabilidades de cada uno:** Tanto en un Proyecto de desarrollo de software o en un proceso productivo, es imprescindible que cada miembro del equipo sepa que tiene que hacer y cuáles son sus funciones. Para el método kanban funcione esto tiene que estar claro. No se trata de que todos hagan todo sino de que sepan que hacer en el momento adecuado.
- **Liderazgo en todos los niveles:** Tener iniciativa y gestionar correctamente tu tarea o a tu equipo es otro elemento básico a tener en cuenta. No se trata de crear sistemas piramidales unos dentro de otros sino de que cada subgrupo y cada miembro tenga clara su función y la ejecute correctamente.

2.4.4 Características de la metodología kanban

La metodología kanban posee las siguientes características:

- Kanban es Lean y Ágil.
- Emplea el sistema de planificación "pull".
- Establecen límites WIP(working in process, Trabajo en proceso).
- Visibilidad del proceso es la base de su mejora.
- Tiene como objetivo la entrega temprana y frecuente de software.
- Trabaja con equipos auto-organizados.
- Necesitan la división del trabajo en partes.
- Revisan y mejora de forma continua el plan del producto en base a datos empíricos (velocidad / tiempo de entrega)

El tiempo fijo de las iteraciones es opcional ya que puede estar sujeto en función al plan del producto y a la mejora del proceso.

La métrica por defecto para la planificación y la mejora de procesos es el **Lead Time** (tiempo de entrega) o tiempo medio que tarda una petición en salir del ciclo.

Los equipos pueden ser multifuncionales o especializados, o incluso solo puede una persona que realiza el proyecto, ya que no existe ninguna prescripción en cuanto al tamaño o número de personal que debe involucrarse en el proyecto.

Se implementa una limitación de proceso de trabajo ya que normalmente el número máximo de procesos simultáneos es de 2 entonces se puede decir que primero debe acabarse una para poder pasar a otra tarea.

Se pueden añadir tareas siempre que haya capacidad disponible en base al número de personas que participan en el proyecto.

El tablero kanban puede ser compartido por varios equipos, ya que no hay roles prescritos como también el tablero kanban es persistente no se hace una limpieza de tareas si no es requerida.

2.5 FACTORES DE CALIDAD MODELO McCall

Para conocer los factores de calidad se debe conocer en esencia el concepto de calidad:

- "Conjunto de esfuerzos efectivos de los diferentes grupos de una organización para la integración del desarrollo, del mantenimiento y de la superación de la calidad de un producto, con el fin de hacer posible la fabricación y servicio a satisfacción completa del consumidor y al nivel más económico" [Feigenbaun, Deming y Juran]
- "La mejor calidad que una empresa puede producir con su tecnología de producción y capacidades de proceso actuales, y que satisfará las necesidades de los clientes, en función de factores tales como el coste y el uso previsto" [Dr. Kaoru Ishikawa]
- " La gestión de calidad en la empresa es el proceso de identificar, aceptar, satisfacer y superar constantemente las expectativas y necesidades de todos los colectivos humanos relacionados con ella, clientes, empleados, directivos, propietarios, proveedores y la comunidad con respecto a los productos y servicios que esta proporciona" [consultora Arthur Andersen]

De las definiciones se puede concluir que la calidad, no es más que el servicio o producto que pueda satisfacer las necesidades del sujeto o empresa a la que van destinadas. Si cumple con el objetivo y la empresa o cliente está de acuerdo con ello entonces se puede decir que el producto es de calidad.

2.5.1 Calidad de software

La calidad del producto software se diferencia de la calidad de otros productos de fabricación industrial, ya que el software tiene sus propias características específicas:

- El software es un producto mental. Es algo abstracto, un intangible.
- Se desarrolla, no se fabrica. El coste está fundamentalmente en el proceso de diseño, no en la posterior producción en serie, y los errores se introducen también en el diseño, no en la producción.
- Los costes del desarrollo de software se concentran en las tareas de Ingeniería, mientras que en la fabricación clásica los costes se van más en las tareas de producción.
- El software no se deteriora con el tiempo. No es susceptible de los efectos del entorno y su curva de fallos es muy diferente de la del hardware. Todos

los problemas que surjan durante el mantenimiento estaban allí desde el principio y afectan a todas las copias del mismo; no se generan nuevos errores.

- Es artesanal en gran medida. El software, en su mayoría, se construye a medida, en vez de ser construido ensamblando componentes existentes y ya probados, lo que dificulta aún más el control de su calidad.
- El mantenimiento del software es mucho más complejo que el mantenimiento del hardware. Cuando un componente del hardware se deteriora se sustituye por una pieza de repuesto, pero cada fallo en el software implica un error en el diseño o en el proceso mediante el cual se tradujo el diseño en código máquina ejecutable.
- Es engañosamente fácil realizar cambios sobre un producto software, pero los efectos de estos cambios se pueden propagar de forma explosiva e incontrolada.
- Como disciplina, el desarrollo de software es aún muy joven, por lo que las técnicas de las que dispone aún no están perfeccionadas.
- El software con errores no se rechaza. Se asume que es inevitable que el software presente algunos errores de poca importancia.

También es importante destacar que la calidad de un producto software debe ser considerada en todos sus estados de evolución es decir en las especificaciones, diseño, códigos, etc.

No basta con verificar la calidad del producto una vez finalizado cuando los problemas de mala calidad ya no tienen solución o su reparación es muy costosa.

La problemática general a la que se enfrenta el software es:

- Aumento constante del tamaño y complejidad de los programas.
- Carácter dinámico e iterativo a lo largo de su ciclo de vida, es decir que los programas de software a lo largo de su vida cambian o evolucionan de una versión a otra para mejorar las prestaciones con respecto a las anteriores.
- Dificultad de conseguir productos totalmente depurados, ya que en ningún caso un programa será perfecto.

- Se dedican elevados recursos monetarios a su mantenimiento, debido a la dificultad que los proyectos de software.
- No suelen estar terminados en los plazos previstos, ni con los costes estipulados, ni cumpliendo los niveles deseables de los requisitos especificados por el usuario.
- Incrementos constantes de los costes de desarrollo debido entre otros, a unos niveles de productividad bajos.
- Los clientes tienen una alta dependencia de sus proveedores por ser en muchos casos aplicaciones a "medida".
- Insuficientes procedimientos normalizados para estipular y evaluar la productividad, costes, y calidad.

Los modelos de calidad definen la calidad como consecuencia de la evaluación de un conjunto de indicadores o métricas en diferentes etapas:

En el nivel más alto de la jerarquía se encuentran los factores de calidad definidos a partir de la visión del usuario del software, y conocidos también como atributos de calidad externos.

Cada uno de los factores se descompone en un conjunto de criterios de calidad, o sea aquéllos atributos que cuando están presentes contribuyen a obtener un software de la calidad. Se trata de una visión de la calidad técnica, desde el punto de vista del producto software que son también atributos de calidad internos. Finalmente, para cada uno de los criterios de calidad se definen un conjunto de métricas o medidas cuantitativas de ciertas características del producto que indican el grado de calidad.

De esta manera, a través de un modelo de calidad se concretan los aspectos relacionados con ella de tal manera que se puede definir, medir y planificar. Además, el empleo de un modelo de calidad permite comprender las relaciones que existen entre diferentes características de un producto software.

En contra de los modelos de calidad pesa que aún no ha quedado demostrada la validez absoluta de ninguno de ellos.

2.5.2 El modelo de McCall.

El modelo de McCall fue el primer modelo de medición de calidad del software, desde el principio de la ingeniería de software, McCall se observa que la calidad está compuesta por características, un modelo de calidad describe diferentes relaciones con modelos muy diversos con distinción de atributos internos y externos que dificultan la comprensión del concepto de calidad. El modelo de McCall focaliza su producto identificando sus atributos desde la perspectiva del usuario, estos atributos (factores de calidad) son los que realmente desarrollan e implementan calidad, ellos pueden ser abstractos para diferentes medios por lo que cada uno interviene criterios de calidad reflejando McCall que el atributo tiene efecto directo en el atributo correspondiente.

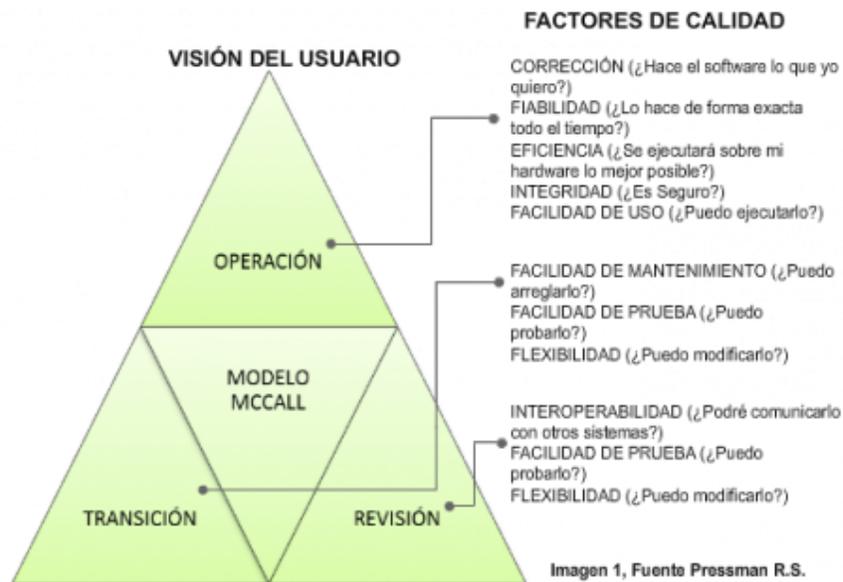


Figura 2.8 Modelo de evaluación de McCall
Recuperado de: McCall, J., Richards, P., & Walters, G. (1977).

McCall propone algunas los factores de calidad que métricas comúnmente utilizadas para evaluar la calidad del software.

McCall, Presenta la relación entre algunos de los factores de calidad y algunas métricas comúnmente utilizadas para evaluar la calidad del software.

El modelo de Calidad MCCALL está planteado desde la perspectiva del usuario presentándose cuestionamientos en cada uno de sus capacidades, cada criterio

de calidad es un conjunto de métricas de calidad, los cuales son otro atributo lo cual establece 3 perspectivas:

- Operación del Producto.
- La revisión del Producto.
- Transición del Producto.



Figura 2.9 Factores y métricas de calidad según mcall
Recuperado de: McCall, J., Richards, P., & Walters, G. (1977).

Factores de Operación del Producto:

- **Corrección:** Mide el grado en que un programa satisface sus especificaciones y consigue los objetivos del usuario.
¿El software hace lo que necesito?
- **Fiabilidad:** mide el grado en que se puede esperar que un programa lleve a cabo sus funciones esperada con la precisión requerida.
¿Lo Hace de forma exacta siempre?
- **Eficiencia:** mide la cantidad de recursos de computadora y de código requerido por un programa para que lleve a cabo las funciones especificadas.
¿Aprovecha el hardware lo mejor posible?
- **Integridad:** es el grado en que puede controlarse el acceso al software o a los datos por personal no autorizado.

¿Es seguro?

- **Facilidad de Uso:** es el esfuerzo requerido para aprender un programa e interpretar la información de entrada y de salida.

¿Lo puedo usar con facilidad?

Factores de Calidad de Revisión:

- **Mantenibilidad:** El software libre es un gran ejemplo de mantenibilidad, el cual con el paso del tiempo sigue liderando el mercado mundial del software del mundo, teniendo una muy buena mantenibilidad al momento de ser utilizado por los usuarios, sin tener mayor inconveniente en la corrección de fallas del producto hasta el momento.

¿Qué tan difícil es arreglarlo?

- **Facilidad de Prueba:** Es el esfuerzo requerido para probar un programa.

¿Se puede probar?

- **Flexibilidad:** Un software muy bien estructurado, analizado, compilado y ejecutado el cual tiene características muy particulares tiene la garantía que puede facilitarle realizar cambios mediante el paso del tiempo y no estar reestructurado a cada momento.

¿Se puede modificar?

Factores de Transición del Producto

- **Portabilidad:** es el esfuerzo requerido para transferir un software de un hardware o un entorno de sistemas a otro.

¿Se puede usar en otra máquina?

- **Reusabilidad:** es el grado en que un programa (o partes de un programa) se puede reutilizar en otro.

¿Se puede reutilizar parte del Software?

- **Facilidad de Interoperación:** es el esfuerzo requerido para asociar un programa a otro.

¿Se puede comunicar con otros sistemas?

Características de calidad

- facilidades de uso

- integridad
- corrección
- confiabilidad
- facilidad de mantenimiento
- facilidad de prueba
- flexibilidad
- facilidad de reutilización
- portabilidad
- interoperabilidad

Tabla 2.1
Modelo de Referencia de Medición de la Calidad del Producto Software.

Puntos De Vista O Ejes	Factor	Criterios
OPERACIÓN DEL PRODUCTO	Facilidad de uso	<ul style="list-style-type: none"> • Facilidad de operación: Atributos del software que determinan la facilidad de operación del software. • Facilidad de comunicación: Atributos del software que proporcionan entradas y salidas fácilmente asimilables. • Facilidad de aprendizaje: Atributos del software que facilitan la familiarización inicial del usuario con el software y la transición del modo actual de operación. • Formación: El grado en que el software ayuda para permitir que nuevos usuarios apliquen el sistema.
	Integridad	<ul style="list-style-type: none"> • Control de accesos. Atributos del software que proporcionan control de acceso al software y los datos que maneja. • Facilidad de auditoría: Atributos del software que facilitan la auditoría de los accesos al software. • Seguridad: La disponibilidad de mecanismos que controlen o protejan los programas o los datos.
	Corrección	<ul style="list-style-type: none"> • Compleitud: Atributos del software que proporcionan la implementación completa de todas las funciones requeridas. • Consistencia: Atributos del software que proporcionan uniformidad en las técnicas y notaciones de diseño e implementación. • Trazabilidad o rastreabilidad: Atributos del software que proporcionan una traza desde los requisitos a la implementación con respecto a un entorno operativo concreto.
	Fiabilidad	<ul style="list-style-type: none"> • Precisión: Atributos del software que proporcionan el grado de precisión requerido en los cálculos y los resultados.

		<ul style="list-style-type: none"> • Consistencia. • Tolerancia a fallos: Atributos del software que posibilitan la continuidad del funcionamiento bajo condiciones no usuales. • Modularidad: Atributos del software que proporcionan una estructura de módulos altamente independientes. • Simplicidad: Atributos del software que posibilitan la implementación de funciones de la forma más comprensible posible. • Exactitud: La precisión de los cálculos y del control.
	Eficiencia	<ul style="list-style-type: none"> • Eficiencia en ejecución: Atributos del software que minimizan el tiempo de procesamiento. • Eficiencia en almacenamiento: Atributos del software que minimizan el espacio de almacenamiento necesario.
REVISIÓN DEL PRODUCTO	Facilidad de mantenimiento	<ul style="list-style-type: none"> • Modularidad. • Simplicidad. • Consistencia. • Concisión: Atributos del software que posibilitan la implementación de una función con la menor cantidad de códigos posible. • Auto descripción: Atributos del software que proporcionan explicaciones sobre la implementación de las funciones.
	Facilidad de prueba	<ul style="list-style-type: none"> • Modularidad. • Simplicidad. • Auto descripción. • Instrumentación: Atributos del software que posibilitan la observación del comportamiento del software durante su ejecución para facilitar las mediciones del uso o la identificación de errores.
	Flexibilidad	<ul style="list-style-type: none"> • Auto descripción. • Capacidad de expansión: Atributos del software que posibilitan la expansión del software en cuanto a capacidades funcionales y datos. • Generalidad: Atributos del software que proporcionan amplitud a las funciones implementadas. • Modularidad.
TRANSICIÓN DEL PRODUCTO	Reusabilidad	<ul style="list-style-type: none"> • Auto descripción. • Generalidad. • Modularidad. • Independencia entre sistema y software: Atributos del software que determinan su dependencia del entorno operativo. • Independencia del hardware: Atributos del software que determinan su dependencia del hardware.

	Interoperabilidad	<ul style="list-style-type: none"> • Modularidad. • Compatibilidad de comunicaciones: Atributos del software que posibilitan el uso de protocolos de comunicación e interfaces estándar. • Compatibilidad de datos: Atributos del software que posibilitan el uso representaciones de datos estándar. • Estandarización en los datos: El uso de estructuras de datos y de tipos estándar a lo largo de todo el programa.
	Portabilidad	<ul style="list-style-type: none"> • Auto descripción. • Modularidad. • Independencia entre sistema y software. • Independencia del hardware.

Fuente: El-modelo-de-McCall. Cavano, J.P., McCall, 1997

2.6 MODELO CONSTRUCTIVO DE COSTOS COCOMO II

2.6.1 Definición del modelo

Los objetivos principales que se tuvieron en cuenta para construir el modelo COCOMO II fueron:

- Desarrollar un modelo de estimación de costo y cronograma de proyectos de software que se adaptara tanto a las prácticas de desarrollo de la década del 90 como a las futuras.
- Construir una base de datos de proyectos de software que permitiera la calibración continua del modelo, y así incrementar la precisión en la estimación.
- Implementar una herramienta de software que soportara el modelo.
- Proveer un marco analítico cuantitativo y un conjunto de herramientas y técnicas que evaluaran el impacto de las mejoras tecnológicas de software sobre los costos y tiempos en las diferentes etapas del ciclo de vida de desarrollo. COCOMO II está compuesto por tres modelos denominados: Composición de Aplicación, Diseño Temprano y Post-Arquitectura.

COCOMO II está compuesto por tres modelos denominados:

- Composición de Aplicación.
- Diseño Temprano.
- Post-Arquitectura.

Estos surgen en respuesta a la diversidad del mercado actual y futuro de desarrollo de software. Esta diversidad podría representarse con el siguiente esquema (Figura 2.11).

Tabla 2.2
Distribución del Mercado de Software Actual y Futuro.

Aplicaciones desarrolladas por usuarios finales		
Generadores de Aplicaciones	Aplicaciones con Componentes	Sistemas Integrados
Infraestructura		

Recuperado de: [Boehm 1995/1]

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

Se puede afirmar que para las aplicaciones desarrolladas por usuarios finales no se justifica la utilización de un modelo de estimación de costos. Estas aplicaciones normalmente se construyen en poco tiempo, por lo tanto, requieren solamente una estimación basada en actividades.

El modelo Composición de Aplicación, es el modelo de estimación utilizado en los proyectos de software que se construyen a partir de componentes pre-empaquetadas. En este caso, se emplean puntos objetivos para estimar el tamaño del software, lo cual está acorde al nivel de información que generalmente se tiene en la etapa de planificación.

Para los demás sectores del mercado se aplica un modelo mixto, combinación de los tres modelos:

El modelo **Composición de Aplicación** se emplea en desarrollos de software durante la etapa de elaboración de prototipos.

El modelo **Diseño Temprano** se utiliza en las primeras etapas del desarrollo en las cuales se evalúan las alternativas de hardware y software de un proyecto. En estas etapas se tiene poca información, lo que concuerda con el uso de Puntos

Función, para estimar tamaño y el uso de un número reducido de factores de costo.

El modelo **Post-Arquitectura** se aplica en la etapa de desarrollo propiamente dicho, después que se define la arquitectura del sistema, y en la etapa de mantenimiento. Este modelo utiliza:

- Puntos Función y/o Líneas de Código para estimar tamaño, con modificadores que contemplan la reutilización.
- Un conjunto de 17 atributos, denominados factores de costo , que permiten considerar características del proyecto referentes al personal, plataforma de desarrollo, etc., que tienen injerencia en los costos.
- Cinco factores que determinan un exponente, que incorpora al modelo el concepto de des-economía y economía de escala. Estos factores reemplazan los modos Orgánico, Semiacoplado y Empotrado del modelo COCOMO '81.

2.6.2 Estimación del Esfuerzo

El esfuerzo necesario para concretar un proyecto de desarrollo de software, cualquiera sea el modelo empleado, se expresa en meses/persona (PM) y representa los meses de trabajo de una persona a tiempo completo, requeridos para desarrollar el proyecto.

2.6.3 Modelo composición de aplicación

- Estimaciones realizadas con puntos de objeto y una fórmula simple para el cálculo del esfuerzo.
- Soporta proyectos con prototipado y proyectos que hacen uso intensivo de la reutilización.
- Basado en estimaciones estándar de la productividad del desarrollador en puntosobjeto/mes.

La fórmula propuesta en este modelo es la siguiente:

$$PM = NOP / PROD$$

Donde:

NOP (Nuevos Puntos Objeto): Tamaño del nuevo software a desarrollar expresado en Puntos Objeto y se calcula de la siguiente manera:

$$NOP = OP \times (100 - \%reuso)/100$$

OP (Puntos Objeto): Tamaño del software a desarrollar expresado en Puntos Objeto

%reuso: Porcentaje de reuso que se espera lograr en el proyecto

PROD: Es la productividad promedio determinada a partir del análisis de datos de proyectos en [Banker 1994], mostrada en Tabla 6.

Tabla 2.3
Productividad para el modelo Composición de Aplicación

Experiencia y capacidad de los desarrolladores	Muy Bajo	Bajo	Normal	Alto	Muy Alto
Madurez y Capacidad del ICASE	Muy Bajo	Bajo	Normal	Alto	Muy Alto
PROD	4	7	13	25	50

Fuente: Productividad para el modelo Composición de Aplicación [Boehm 1995/2]

2.6.4 Modelo Diseño Temprano

Este modelo se usa en las etapas tempranas de un proyecto de software, cuando se conoce muy poco del tamaño del producto a ser desarrollado, de la naturaleza de la plataforma, del personal a ser incorporado al proyecto o detalles específicos del proceso a utilizar. Este modelo podría emplearse tanto en productos desarrollados en sectores de Generadores de Aplicación, Sistemas Integrados o Infraestructura.

El modelo de Diseño Temprano ajusta el esfuerzo nominal usando siete factores de costo. La fórmula para el cálculo del esfuerzo es la siguiente:

$$PM_{estimado} = PM_{nominal} \times \prod_{i=1}^7 EM_i$$

Donde:

- **PM_{Estimado}** es el esfuerzo Nominal ajustado por 7 factores, que reflejan otros aspectos propios del proyecto que afectan al esfuerzo necesario para la ejecución del mismo.

- **EM_i** corresponde a los factores de costo que tienen un efecto multiplicativo sobre el esfuerzo, llamados Multiplicadores de Esfuerzo (Effort Multipliers).

$$PM_{nominal} = A \times (KSLOC)^B$$

$$B = 1.01 + 0.01 \times \sum_{j=1}^5 W_j$$

- **KSLOC** es el tamaño del software a desarrollar expresado en miles de líneas de código fuente.
- **A** es una constante que captura los efectos lineales sobre el esfuerzo de acuerdo a la variación del tamaño, (**A=2.94**).
- **B** es el factor exponencial de escala, toma en cuenta las características relacionadas con las economías y des-economías de escala producidas cuando un proyecto de software incrementa su tamaño.

B<1.0 el Proyecto exhibe economía de escala.

Si el B=1.0 las economías y deseconomías de escala están en equilibrio.

Si el B>1.0 el proyecto muestra deseconomía de escala.

El cálculo del Factor Exponencial de Escala B está basado en factores que influyen exponencialmente en la productividad y esfuerzo de un proyecto de software. Estos factores toman valores dentro de un rango que va desde un nivel

Muy Bajo hasta uno Extra Alto:

- Precedencia y Flexibilidad en el Desarrollo (PREC Y FLEX).
- Arquitectura y Determinación del Riesgo (RESL).
- Cohesión del Equipo (TEAM).
- Madurez del Proceso (PMAT).

	VLO	LO	NOM	HI	VHI	XHI
PREC	6.20	4.96	3.72	2.48	1.24	0.00
FLEX	5.07	4.05	3.04	2.03	1.01	0.00
RESL	7.07	5.65	4.24	2.83	1.41	0.00
TEAM	5.48	4.38	3.29	2.19	1.10	0.00
PMAT	7.80	6.24	4.68	3.12	1.56	0.00

Figura 2.10. Tabla de valores de los factores de escala.
Fuente: Modelo PostArquitectura.[COCOMO II.0]

Factor de Escala Wj	Muy Bajo	Bajo	Normal	Alto	Muy Alto	Extra
Precedencia PREC	Completamente sin precedentes	Ampliamente sin precedentes	Algún precedente	Generalmente Familiar	Ampliamente Familiar	Completamente Familiar
Flexibilidad en el desarrollo FLEX	Rigurosa	Relajación Ocasional	Alguna Relajación	Conformidad en general	Alguna Conformidad	Metas generales
Arquitectura/ Resolución de riesgo RESL	Poca (20%)	Alguna (40%)	Siempre (60%)	Generalmente 75%)	Principalmente (90%)	Completo (100%)
Cohesión de equipo TEAM	Interacciones difíciles	Interacciones con alguna dificultad	Interacciones básicamente cooperativas	Ampliamente Cooperativas	Altamente Cooperativas	Interacciones Sin Fisuras
Madurez del proceso PMAT	Desarrollado más adelante					

Tabla. 3.1
Factores de Escala.
Fuente: [Boehm 1995/2]

Estimación de cronograma

La versión inicial de COCOMO II provee un modelo de estimación del cronograma similar al presentado en COCOMO' 81 y ADA COCOMO. La ecuación inicial para los tres modelos de COCOMO II es:

$$TDEV = \left[3.0 \times PM_*^{(0.33 + 0.2 \times (B - 1.01))} \right] \times \frac{SCED\%}{100}$$

Dónde:

TDEV es el tiempo calendario en meses que transcurre desde la determinación de los requerimientos a la culminación de una actividad que certifique que el producto cumple con las especificaciones. PM^* es el esfuerzo expresado en meses personas, calculado sin tener en cuenta el multiplicador de esfuerzo SCED, B es el Factor de Escala, SCED% es el porcentaje de compresión/expansión del cronograma. Las futuras versiones de COCOMO II ofrecerán un modelo de estimación de cronograma más completo que refleje los diferentes modelos de procesos que se puede usar en el desarrollo de un proyecto, los efectos del reuso de software y la composición de aplicaciones

2.6.5 Modelo Post-Arquitectura

Es el modelo de estimación más detallado y se aplica cuando la arquitectura del proyecto está completamente definida. Este modelo se aplica durante el desarrollo y mantenimiento de productos de software incluidos en las áreas de Sistemas Integrados, Infraestructura y Generadores de Aplicaciones.

El esfuerzo nominal se ajusta usando 17 factores multiplicadores de esfuerzo. El mayor número de multiplicadores permite analizar con más exactitud el conocimiento disponible en las últimas etapas de desarrollo, ajustando el modelo de tal forma que refleje fielmente el producto de software bajo desarrollo. La fórmula para el cálculo del esfuerzo es la siguiente:

$$PM_{estimado} = PM_{nominal} \times \prod_{i=1}^{17} EM_i$$

Los factores se clasifican en cuatro áreas: Producto, Plataforma, Personal y Proyecto.

Factores del producto. Se refieren a las restricciones y requerimientos sobre el producto a desarrollar.

- RELY: Confiabilidad requerida
- DATA: Tamaño de la base de datos
- CPLX: Complejidad del producto
- RUSE: Requerimientos de reusabilidad
- DOCU: Documentación acorde a las diferentes etapas del ciclo de vida

	VLO	LO	NOM	HI	VHI	XHI
RELY	0.82	0.92	1.00	1.10	1.26	XXXX
DATA	XXXX	0.90	1.00	1.14	1.28	XXXX
DOCU	0.81	0.91	1.00	1.11	1.23	XXXX
CPLX	0.73	0.87	1.00	1.17	1.34	1.74
RUSE	XXXX	0.95	1.00	1.07	1.15	1.24

OK Cancel Help

Figura 2.11 Tabla de parámetros de producto
Fuente: Factores de la plataforma. Modelo Post-Arquitectura.[COCOMO II.0

Plataforma

- TIME: Limitaciones en el tiempo de ejecución.
- STOR: Limitaciones en el almacenamiento principal.
- PVOL: Volatilidad de la plataforma.

	VLO	LO	NOM	HI	VHI	XHI
TIME	XXXX	XXXX	1.00	1.11	1.29	1.63
STOR	XXXX	XXXX	1.00	1.05	1.17	1.46
PVOL	XXXX	0.87	1.00	1.15	1.30	XXXX

OK Cancel Help

Figura 2.12 Tabla de valores de plataforma
Fuente: Factores de la plataforma. Modelo Post-Arquitectura.COCOMO II.0

Personal

- ACAP: Capacidad de los analistas.
- PCAP: Capacidad del programador.
- PCON: Continuidad del personal.
- AEXP: Experiencia en las aplicaciones.
- PEXP: Experiencia en la plataforma.

- LTEX: Experiencia con el lenguaje y las herramientas.

	VLO	LO	NOM	HI	VHI	XHI
ACAP	1.42	1.19	1.00	0.85	0.71	XXXX
AEXP	1.22	1.10	1.00	0.88	0.81	XXXX
PCAP	1.34	1.15	1.00	0.88	0.76	XXXX
PEXP	1.19	1.09	1.00	0.91	0.85	XXXX
LTEX	1.20	1.09	1.00	0.91	0.84	XXXX
PCON	1.29	1.12	1.00	0.90	0.81	XXXX

Figura 2.13 Tabla de valores de Personal
Fuente: Factores de la plataforma. Modelo Post-Arquitectura.COCOMO II.0

Proyecto

- TOOL: Uso de herramientas de software.
- SITE: Desarrollo en varios sitios.

	VLO	LO	NOM	HI	VHI	XHI
TOOL	1.17	1.09	1.00	0.90	0.78	XXXX
SCED	1.43	1.14	1.00	1.00	1.00	XXXX
SITE	1.22	1.09	1.00	0.93	0.86	0.80

Figura 2.14 Tabla de valores de Proyecto
Fuente: Factores de la plataforma. Modelo Post-Arquitectura.COCOMO II.0

Estimación de la duración

La estimación del tiempo de desarrollo conociendo el esfuerzo estimado(personas/mes) se calcula con la siguiente formula:

$$TDEV = [3,67 * PM^{(0,28+0,2*(B-0,91))}] * \frac{\%SCED}{100}$$

Donde:

PM es el esfuerzo expresado en meses-hora

SCED% es el porcentaje de reducción o incremento del calendario nominal del proyecto.

B es el factor de escala para tener en cuenta las diversas economías de escala positivas o negativas, existentes en proyectos de software.

Estimación del coste

El coste estimado para el proyecto se calcula con la siguiente fórmula.

$$Coste = PM_{estimado} * Ratio$$

Donde:

PM_{estimado} es el esfuerzo estimado.

Ratio cantidad de dinero que debe cobrar por mes el desarrollador de cada módulo.

Estimación de la productividad

Para obtener la productividad del proyecto se utilizará la siguiente fórmula.

$$Productividad = Tamano_del_modulo / PM_{estimado}$$

Donde:

Tamaño_del_modulo es el tamaño del módulo en líneas de código

PM_{estimado} es el esfuerzo estimado.

Estimación del coste por instrucción

Para calcular el coste por línea de código se utiliza la siguiente fórmula.

$$Coste_instruccion = Coste / Tamano_del_modulo$$

Coste es el coste estimado para el proyecto.

Tamaño_del_modulo es el tamaño del módulo en líneas de código.

Estimación del personal

Es la estimación del número de desarrolladores necesarios para completar un módulo en tiempo de desarrollo estimado. Para calcular el personal se utilizará la siguiente fórmula.

$$\text{Personal} = \text{PM}_{\text{estimado}} / \text{Duración}$$

Duración= cantidad de meses estimada para la realización del proyecto.

PM=estimado es el esfuerzo estimado.

2.6.6 Estimación del tamaño de software COCOMO II

Cocoma II utiliza tres técnicas para poder evaluar la estimación: Puntos Objeto, Puntos Función no ajustados y líneas de código fuente. Además, se emplean otros parámetros relativos al tamaño que contemplan aspectos tales como: reuso, reingeniería, conversión y mantenimiento.

Es necesario unificar criterios de medición de tamaño, tanto para poder planificar y controlar proyectos, como para realizar estudios y análisis entre proyectos en pro de la mejora de procesos [Park 1992].

Puntos Objeto

La estimación a través de Puntos Objeto es un enfoque de medición de tamaño de software relativamente nuevo, es apropiado para las aplicaciones con componentes y para estimar esfuerzos en las etapas de prototipación. En estas circunstancias, se lo ha comparado con la estimación de Puntos Función.

A continuación, se describe el procedimiento para determinar Puntos Objeto en un proyecto de software:

- **Primero:** Determinar Cantidad de Objetos: Estimar la cantidad de pantallas, reportes, componentes de 3GL que contendrá la aplicación. **Segundo:** Clasificar cada instancia de un objeto según sus niveles de complejidad (simple, media o difícil) de acuerdo a la **Figura 2.12**.
- **Tercero:** Dar el peso a cada objeto según el nivel de complejidad. Los pesos reflejan el esfuerzo relativo requerido para implementar una instancia de ese nivel de complejidad.
- **Cuarto:** Determinar la cantidad de Puntos Objeto, sumando todos los pesos de las instancias de los tipos de objetos especificados.

Tabla 2.4
Esquema de Clasificación de Puntos Objetos

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

Recuperado de: [Boehm 1995/2]

Tabla 2.5
Peso de un Punto Objeto

Tipo de Objeto	Complejidad - Peso		
	Simple	Media	Difícil
Pantalla	1	2	3
Reporte	2	5	8
Componente 3GL			10

Recuperado de: [Boehm 1995/2]

Puntos Función no ajustados

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

Los Puntos Función procuran cuantificar la funcionalidad de un sistema de software. La meta es obtener un número que caracterice completamente al sistema.

El punto de función no ajustado tiene que ser convertidos a líneas de código fuente considerando el lenguaje de implementación. Esto se realiza para los modelos de Diseño temprano y Post Arquitectura.

Tabla 2.6
Equivalencia puntos de función en líneas de código

	Lenguaje	Tamaño LOC
1	ASP.NET	24,615
2	C	51,200
3	JAVA	53,333
4	PHP	53,333
5	PHYTON	53.333
6	HTML	160,000
7	PERL	53,333

Recuperado de: Herramientas multimodelo para la estimación de proyectos software (Maria Perez 2018)

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

La fórmula de Albretch [Albretch 1979] para calcular los puntos función, es la siguiente:

$$FP = UFP \times TCF$$

Dónde: UFP: Puntos Función no Ajustados

TCF: Factor de Complejidad Técnica

Para calcular los UFP, se deben identificar los siguientes tipos de ítems:

- Entradas Externas (Inputs): Entrada de datos del usuario o de control que ingresan desde el exterior del sistema para agregar y/o cambiar datos a un archivo lógico interno.
- Salidas Externas (Outputs): Salida de datos de usuario o de control que deja el límite del sistema de software.

- Archivo Lógicos Internos (Archivos): Incluye cada archivo lógico, es decir cada grupo lógico de datos que es generado, usado, o mantenido por el sistema de software.
- Archivos Externos de Interfase (Interfases): Archivos transferidos o compartidos entre sistemas de software.
- Solicitudes Externas (Queries): Combinación única de entrada-salida, donde una entrada causa y genera una salida inmediata, como un tipo de solicitud externa.

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

$$UFP = \sum_{i=1}^{15} (Cantidad_Items_Tipo_i) \times (Peso_i)$$

La Tabla 2.6 muestra cómo se determinan los niveles de complejidad de cada tipo de ítem en función del número y tipo de elementos de datos y archivos involucrados.

Tabla 2.7
Puntos Función. Determinación del Peso.

Para archivos lógicos internos y archivos externos de interfase				Para salidas y consultas externas				Para entradas externas			
Elementos de Registro	Elementos de datos			Tipos de archivos	Elementos de datos			Tipos de archivos	Elementos de datos		
	1-19	20-50	51+		1-5	6-19	20+		1-4	5-15	16+
1	Bajo	Bajo	Prom.	0 ó 1	Bajo	Bajo	Prom.	0 ó 1	Bajo	Bajo	Prom.
2-5	Bajo	Prom.	Alto	2-3	Bajo	Prom.	Alto	2-3	Bajo	Prom.	Alto
6+	Prom.	Alto	Alto	4+	Prom.	Alto	Alto	3+	Prom.	Alto	Alto

Recuperado de: [Boehm 1995/2]

La Tabla 10 muestra las ponderaciones asociadas a cada tipo de ítem. Estas ponderaciones han sido derivadas y validadas empíricamente mediante la observación de una gran variedad de proyectos.

Tabla 2.8
Peso del Factor de Complejidad.

Tipo de función	Peso del Factor de Complejidad		
	Bajo	Promedio	Alto
Entradas Externas (Inputs)	3	4	6
Salidas Externas (Outputs)	4	5	7
Archivo Lógicos Internos (Archivos)	7	10	15
Archivos Externos de Interfase (Interfases)	5	7	10
Consultas Externas (Queries)	3	4	6

Recuperado de: [Boehm 1995/2]

Para el cálculo del Factor de Complejidad Técnica, TCF, se considera la siguiente fórmula:

$$TCF = 0.65 + 0.01 \times \sum_{i=1}^{14} Fi$$

Donde los Fi corresponden a los pesos asignados a los siguientes factores:

F1: Mecanismos de recuperación y back-up confiables

F2: Comunicación de Datos

F3: Funciones de Procesamiento Distribuido

F4: Performance

F5: Configuración usada rigurosamente

F6: Entrada de datos on-line

F7: Factibilidad Operativa

F8: Actualización de archivos on-line

F9: Interfaces Complejas

F10: Procesamiento Interno complejo

F11: Reusabilidad

F12: Fácil Instalación

F13: Soporte de múltiples instalaciones

F14: Facilidad de cambios y amigabilidad

Los pesos se consideran dentro de una escala de 0 a 5, descrita a continuación:

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

Líneas de código

COCOMO II considera a la sentencia fuente lógica como línea standard de código. Ahora bien, definir una línea de código es difícil debido a que existen diferencias conceptuales cuando se cuentan sentencias ejecutables y de declaraciones de datos en lenguajes diferentes.

El objetivo es medir la cantidad de trabajo intelectual puesto en el desarrollo de un programa. Para minimizar esos problemas, se usa el checklist de definición desarrollado por el SEI, que permite unificar criterios en la definición de una línea de código fuente [Park 1992], [Goethert et al. 1992].

A los efectos de COCOMO II, se han efectuado cambios que consisten en eliminar las categorías de software que insumen poco esfuerzo.

Así no están incluidas librerías de soporte de lenguajes, sistemas operativos, librerías comerciales, etc., ni tampoco el código generado con generadores de código fuente.

Existen herramientas automatizadas para medir la cantidad de líneas de código fuente, como por ejemplo Amadeus [Amadeus 1994] [Selby et al. 1991]. Para realizar un análisis de mayor especificidad, Amadeus automáticamente recolecta medidas adicionales como total de líneas fuente, de comentarios, declaraciones, interfaces, anidamientos, sentencias ejecutables y otras. Esta herramienta provee varias medidas de tamaño, incluyendo métricas aplicables a tecnologías de objetos de [Chidamber and Kemerer 1994].

2.7 SISTEMAS DE INFORMACION GEOGRAFICA

Un sistema de información geográfica (SIG, también habitualmente citado como GIS por las siglas de su nombre en inglés) es un conjunto de herramientas que integra y relaciona diversos componentes que permiten la organización, almacenamiento, manipulación, análisis y modelización de grandes cantidades de datos procedentes del mundo real que están vinculados a una referencia espacial, facilitando la incorporación de aspectos sociales-culturales, económicos y ambientales que conducen a la toma de decisiones de una manera más eficaz.

En el sentido más estricto, es cualquier sistema de información capaz de integrar, almacenar, editar, analizar, compartir y mostrar la información geográficamente referenciada. En un sentido más genérico, los SIG son herramientas que permiten a los usuarios crear consultas interactivas, analizar la información espacial, editar datos, mapas y presentar los resultados de todas estas operaciones.

La tecnología de los SIG puede ser utilizada para investigaciones científicas, la gestión de los recursos, la gestión de activos, la arqueología, la evaluación del impacto ambiental, la planificación urbana, la cartografía, la sociología, la geografía histórica, el marketing, la logística por nombrar unos pocos. Por

ejemplo, un SIG podría permitir ser utilizados por una empresa para ubicar un nuevo negocio y aprovechar las ventajas de una zona de mercado con escasa competencia.

2.7.1 Funcionamiento de un SIG

El SIG funciona como una base de datos con información geográfica (datos alfanuméricos) que se encuentra asociada por un identificador común a los objetos gráficos de los mapas digitales. De esta forma, señalando un objeto se conocen sus atributos e, inversamente, preguntando por un registro de la base de datos se puede saber su localización en la cartografía.

SIG proporciona, para cada tipo de organización basada en ubicación, una plataforma para actualizar datos geográficos sin perder tiempo visitando el sitio y actualizar la base de datos manualmente.

La razón fundamental para utilizar un SIG es la gestión de información espacial. El sistema permite separar la información en diferentes capas temáticas y las almacena independientemente, permitiendo trabajar con ellas de manera rápida y sencilla, facilitando al profesional la posibilidad de relacionar la información existente a través de la topología geoespacial de los objetos, con el fin de generar otra nueva que no podríamos obtener de otra forma.

Las principales cuestiones que puede resolver un sistema de información geográfica, ordenadas de menor a mayor complejidad, son:

1. **Localización:** preguntar por las características de un lugar concreto.
2. **Condición:** el cumplimiento o no de unas condiciones impuestas al sistema. Se busca un determinado lugar que reúna ciertas condiciones
3. **Tendencia:** comparación entre situaciones temporales o espaciales distintas de alguna característica. Permite conocer la variación de algunas características a través de un determinado periodo.
4. **Rutas:** cálculo de rutas óptimas entre dos o más puntos.

5. **Pautas:** detección de pautas espaciales. Busca determinar en una zona específica, las relaciones que pudieran existir entre dos o más variables.
6. **Modelos:** generación de modelos a partir de fenómenos o actuaciones simuladas. Si un sistema planteado se somete a determinadas modificaciones de sus variables cómo queda definido el nuevo sistema, cuánto ha cambiado, etc.

Por ser tan versátiles, el campo de aplicación de los sistemas de información geográfica es muy amplio, pudiendo utilizarse en la mayoría de las actividades con un componente espacial. La profunda revolución que han provocado las nuevas tecnologías ha incidido de manera decisiva en su evolución.

Software SIG

Originalmente hasta finales de los 90, cuando los datos del SIG se localizaban principalmente en grandes ordenadores y se utilizan para mantener registros internos, el software era un producto independiente. Sin embargo con el cada vez mayor acceso a Internet/Intranet y a la demanda de datos geográficos distribuidos, el software SIG ha cambiado gradualmente su perspectiva hacia la distribución de datos a través de redes. Los SIG que en la actualidad se comercializan son combinaciones de varias aplicaciones interoperables y APIs.

Hoy por hoy dentro del software SIG se distingue en siete grandes tipos de programas informáticos:

- **SIG de escritorio.** Son aquellos que se utilizan para crear, editar, administrar, analizar y visualizar los datos geográficos.
- **Sistemas de gestión de bases de datos espaciales o geográficas** (SGBD espacial). Se emplean para almacenar la información geográfica, pero a menudo también proporcionan la funcionalidad de análisis y manipulación de los datos.
- **Servidores cartográficos.** Se utilizan para distribuir mapas a través de Internet

- **Servidores SIG.** Proporcionan básicamente la misma funcionalidad que los SIG de escritorio, pero permiten acceder a estas utilidades de geo procesamiento a través de una red informática
- **Clientes web SIG.** Permiten la visualización de datos y acceder a funcionalidades de análisis y consulta de servidores SIG a través de Internet o intranet.
- **Bibliotecas y extensiones espaciales.** Proporcionan características adicionales que no forman parte fundamental del programa ya que pueden no ser requeridas por un usuario medio de este tipo de software.
- **SIG móviles.** Se usan para la recogida de datos en campo a través de dispositivos móviles (PDA, teléfonos inteligentes, tabletas, etc.).

2.7.1 La tecnología GIS, al servicio de las empresas de suministros

los SIG han sido aplicados y utilizados en los sistemas de transporte a través de diferentes programas y sistemas, el cual integra un SIG con funciones de logística. La tecnología SIG ha revolucionado el mundo de la gestión de la información y se ha convertido en el aliado perfecto de las empresas de suministro. Eficiencia, rapidez, reducción de costes, entre otros, son los principales beneficios de su uso, de ahí que el SIG sea una tecnología con mucho futuro por delante.

Hoy en día, nos enfrentamos a un crecimiento exponencial de información que se genera a diario, de ahí que los Sistemas de Información Geográfica (SIG) jueguen un papel fundamental en tanto proporcionan la infraestructura necesaria para que las instituciones y empresas puedan mejorar la gestión y conocimiento estratégico de su entorno a partir de dicha información. Así, permite tener información sobre localización, condición, tendencia, rutas, pautas y modelos.

Por ello, la tecnología SIG se ha convertido en un aliado perfecto de las empresas de suministros. Y es que, en el sector de los proveedores de abastecimiento como empresas comerciales, el de gas, electricidad, etc. es de vital importancia que exista un aprovechamiento óptimo de sus recursos.

Todo ello, con el principal fin de alcanzar la máxima eficiencia en su explotación y mejorar la gestión de procesos y la toma de decisiones, bajando los costos de operación.

CAPITULO III MARCO APLICATIVO

La construcción del sistema se tomará las diferentes fases de la metodología UWE contemplando cada una de las actividades de modelado de sistemas web.

3.1 FASE DE ANÁLISIS Y ESPECIFICACION DE REQUISITOS.

3.1.2 Estado actual de procesos operativos de la empresa ADHIPARM S.R.L

ADIPHARM S.R.L. se ha posicionado con el tiempo en una de las empresas proveedoras de alimento balanceado para animales domésticos, con la visión de expansión nacional y sucursales actuales en la ciudad de La Paz y Cochabamba, distribuyendo desde la ciudad de La Paz a los departamentos de Oruro, Potosí y Sucre el cual ofrece una variedad de productos en diferentes formas y presentaciones.

Los alimentos balanceados con los que cuenta están dirigidos a cualquier persona que tenga un animal doméstico ya sea canino o felino al cual están orientados los productos que se comercializa, y que esta persona o dueño quiera darle un alimento completo que satisfaga las necesidades nutricionales.

El mercado es masivo y popular ya que cualquier persona de clase media puede acceder al producto, no se tiene un perfil del comprador ya que este es demasiado amplio básicamente para que una persona pueda tener acceso al producto es tener una mascota.

ADIPHARM S.R.L. es una empresa que va en crecimiento y los procesos administrativos y toda la información están centralizados en la oficina central en la ciudad de La Paz- El Alto.

La empresa tiene un comportamiento que no es claro para todos sus procesos operativos y están definidos para todos de igual manera. La principal característica a solucionar es la comunicación entre el personal que realiza las operaciones entre vendedores y almacenes, entonces debe tomarse en cuenta recursos web para poder ver la información de manera accesible.

A continuación, mostraremos detalladamente el funcionamiento de cada proceso y cómo interactúan entre ellos.

Descripción del funcionamiento actual

La empresa tiene los siguientes procesos que intervienen en el pedido, la planificación, distribución y la gestión de almacenes.

Descripción de proceso de pedido.

Los pedidos se realizan de forma verbal o telefónica por los vendedores, los cuales realizan visitas a los diferentes clientes del cual se genera una hoja de remisión con el detalle del pedido al final de día o en su defecto al inicio del día. El vendedor una vez realizado las visitas debe retornar a las oficinas de la empresa para programar su distribución, el vendedor puede visitar todo un día a sus clientes para poder realizar la mayor cantidad de pedidos para ser distribuido a los clientes de acuerdo a la fecha de entrega.

El principal problema e inconveniente está en el retorno del vendedor a la empresa, para realizar el pedido y de esta manera retrasa la distribución que se despachan en los vehículos de la empresa.

Descripción de proceso de planificación de distribución

Para el proceso de planificación de distribución se toma en cuenta solo los pedidos programados para la fecha actual de entrega, las cantidades, la ciudad a la que pertenece el cliente ya se en La Paz o El Alto.

Para poder realizar la planificación el jefe de almacenes es quien decepciona todo el detalle de pedidos el cual solo la obtiene al comienzo del día. A partir de esa orden de pedido es que entregada se genera la planificación de rutas de distribución todas las veces hechas a criterio del jefe de almacenes.

El problema principal es que al no tener disponible la orden de pedidos se generan retrasos, mala planificación, y trabajo extra al no poder tener el total de los pedidos, otro de los problemas es que el vendedor al no contar con la información de stock de productos suele tener pedidos que exceden las cantidades de productos en almacenes.

Descripción de proceso de distribución

La distribución a partir de la entrega de planificación de ruta del camión de distribución, la empresa cuenta con dos vehículos para dicho proceso.

Uno de los problemas que se encuentra a la hora de realizar la distribución es la demora de encontrar a cliente y la planificación de rutas que en su mayoría no son planificadas de manera óptima, muchas veces el camión regresa a los depósitos para volver a cargar productos. O existe exceso de entregas, lo cual no pueden cubrir en un solo día.

Descripción del proceso en control y manejo de inventarios.

El manejo y su administración en los inventarios se las realiza de manera manual y participa activamente en los procesos de pedidos y distribución.

La empresa ADIPHARM cuenta con un almacén fijo y en ocasiones cuando el pedido o compra de productos excede la capacidad del almacén entonces se suele tener o adquirir un almacén temporal.

El control de inventario físico se lo realiza una vez por semana, se realiza el conteo general de cada tipo de producto verificando con las notas de entrega de los productos ya vendidos estas deben coincidir con las cantidades existentes de productos en almacén, de esta manera controlan la cantidad y el stock de cada producto.

El principal problema que se encuentra la difícil labor de control de toda una semana revisar y analizar las cantidades despachadas a los vehículos y sus devoluciones en almacenes, el mismo control físico de productos es complicado ya que en algunos existe un producto en diferentes secciones del almacén haciendo esta tarea más tediosa.

Descripción de puntos de reposición

En cuanto a los puntos de reposición no existe un método que se aplique para las cantidades mínimas de productos que indican un nuevo pedido o reabastecimiento de productos. En muchas de las ocasiones la reposición se las realiza a destiempo y eso provoca que no haya stock disponible para distribución.

3.2.2 determinación requerimientos funcionales.

Requerimientos funcionales

Un requerimiento es una característica que el sistema DEBE tener o es una restricción que el sistema DEBE satisfacer para ser aceptada por el cliente.

Los funcionales se describen bajo el siguiente orden dependiendo de su grado de importancia:

- Usuarios (registro, lectura, actualización y eliminación)
- Clientes (registro, lectura, actualización y eliminación)
- Productos (registro, lectura, actualización y eliminación)
- Proveedores (registro, lectura, actualización y eliminación)
- Pedidos (registro, lectura, actualización y eliminación)
- Generación de reportes.
- Control de inventarios (Almacenes, Movimientos)
- Control de stock.
- Planificación de rutas optimas
- Control de distribución

Requisitos no funcionales

Describen la interacción entre el sistema y su ambiente independientemente de su implementación.

Se describen bajo el siguiente orden dependiendo de su grado de importancia:

- Tiempos de respuesta del sistema y el cliente.
- Documentos generales del sistema.
- Entorno tecnológico del sistema.
- Diseño del sistema

Planificación del sistema con la aplicación de la metodología KANBAN

Para la planificación del sistema se hace uso de la metodología de desarrollo **KANBAN** el cual permite un tipo de organización de tareas en un tablero de manera óptima y flexible recomendando.

Una vez identificado los requerimientos funcionales del sistema a desarrollar procedemos a calificar de acuerdo a la prioridad de cada tarea.

Tabla 3.1
Product BackLog del Sistema web

Nº	Tarea	Prioridad	Tiempo estimado	Fecha Inicio	Dependencia
1	Módulo de para usuarios	Media	10	12/02/20	0
2	Módulo para Clientes	Alta	10	22/02/20	0
3	Módulo para Productos	Alta	10	03/03/20	0
4	Módulo para Proveedores	Media	10	13/03/20	0
5	Módulo para Pedidos	Alta	15	18/03/20	1
6	Módulo para Almacenes	Media	10	27/03/20	2
7	Módulo para Movimientos	Media	15	13/04/20	3
8	Módulo para Planificación de rutas	Alta	15	23/04/20	2
9	Módulo de distribución	Alta	15	10/05/20	3

Fuente: Elaboración propia

La aplicación de la metodología se realizo mediante la aplicacio web trello el cual nos permite trabajar con un entorno de trabajo basado en la aplicación de la metodología KANBAN.

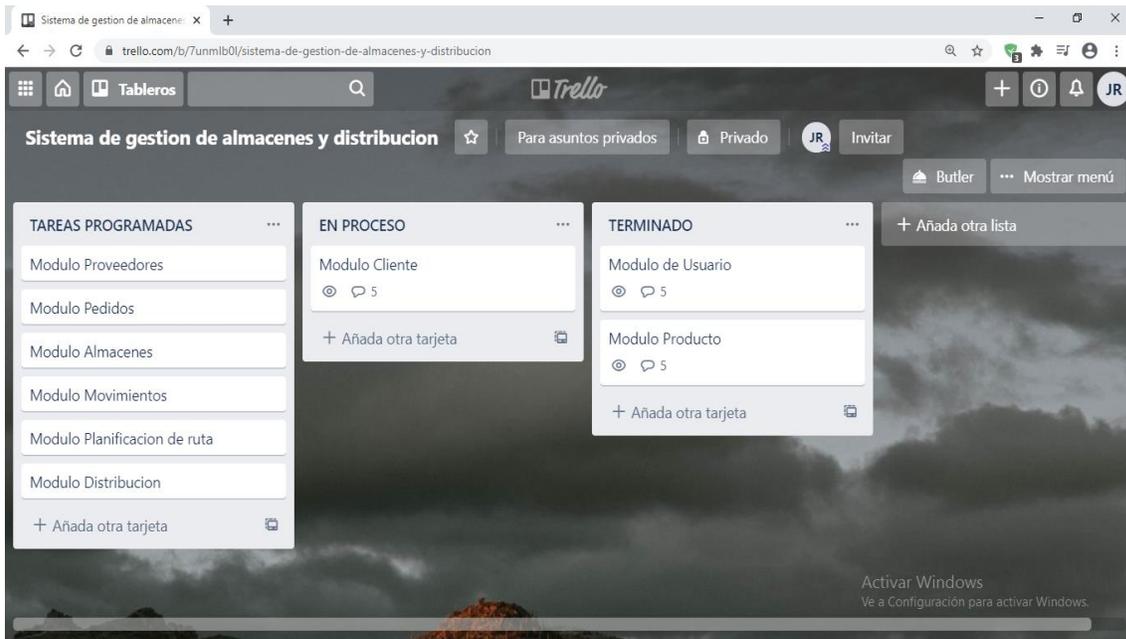


Figura 3.1 Aplicación KANBAN- tablero de tareas programadas
Fuente: elaboración propia

3.2 FASE DE MODELADO Y DISEÑO DEL SISTEMA

Para la estructuración de sistema y modelado del sistema se empleó el uso de la metodología UWE ya que permite especificar de mejor manera una aplicación Web en su proceso de creación.

Diagrama de casos de uso de pedido

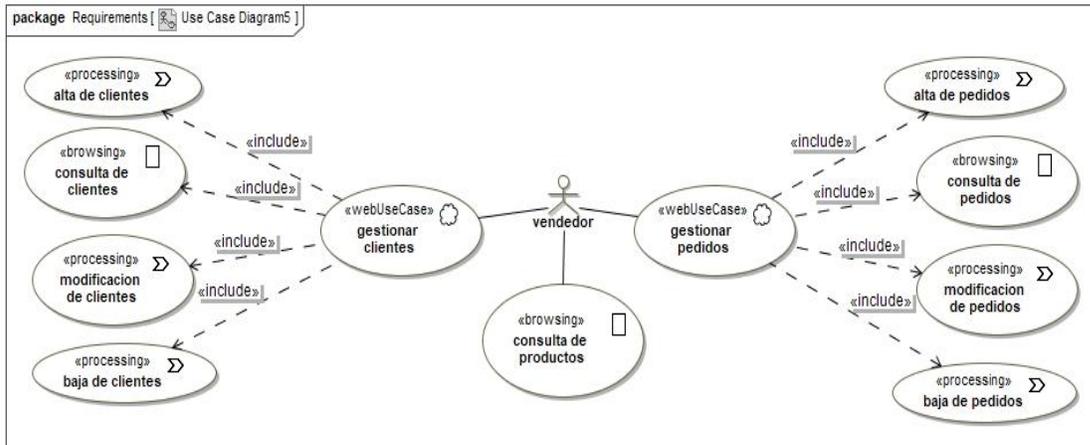


Figura 3.2 Modelo de casos de uso de pedidos
Fuente: Elaboración propia

Diagrama de contenido de pedido

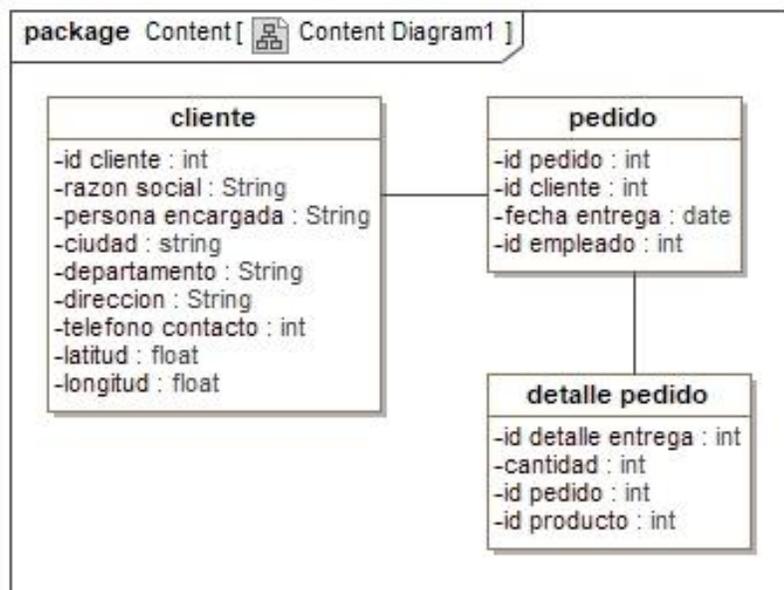


Figura 3.3 Modelo de contenido de pedido
Fuente: Elaboración propia

Diagrama de navegación de pedido

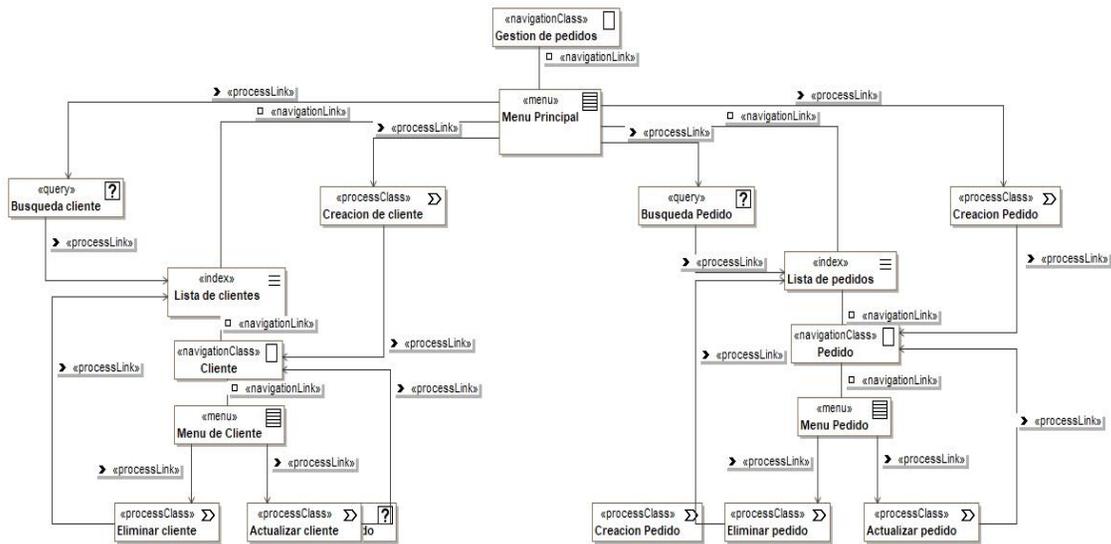


Figura 3.4 Modelo de navegación de pedido
Fuente: Elaboración propia

Diagrama de presentación de pedido

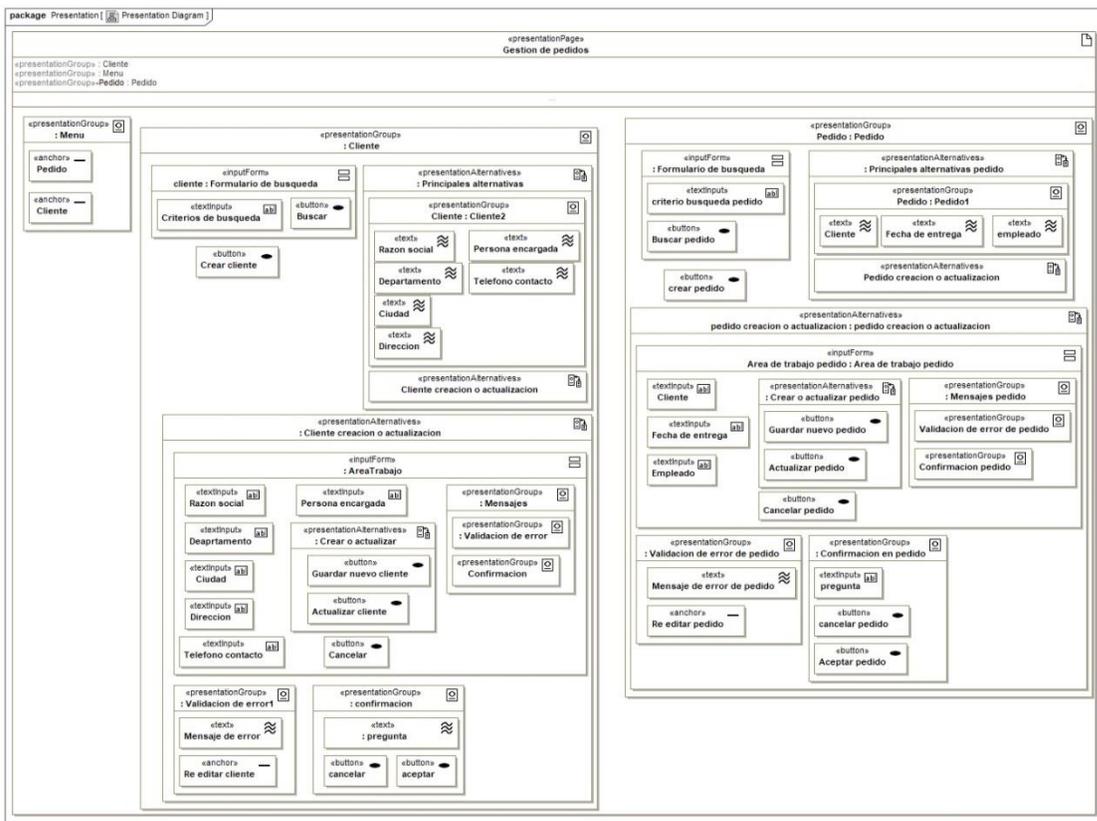


Figura 3.5 Modelo de presentación de pedido
Fuente: Elaboración propia

Diagrama de procesos de pedido

a) Diagrama de flujo de proceso creación cliente

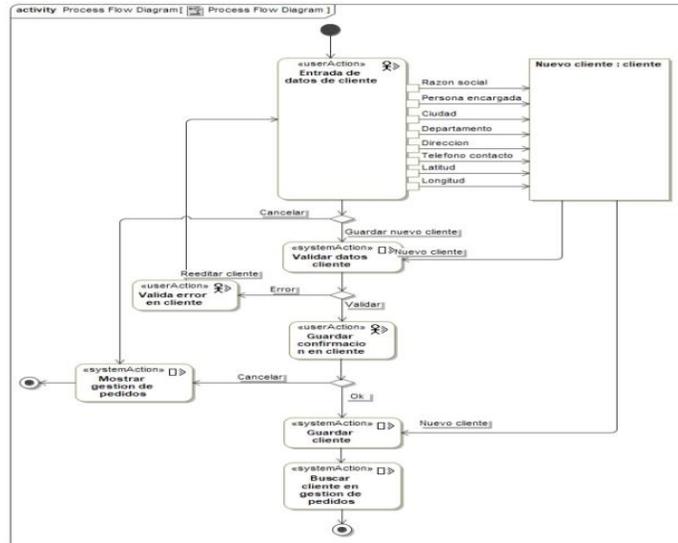


Figura 3.6 Modelo de proceso de pedido- registro de cliente
Fuente: Elaboración propia

b) Diagrama de flujo de proceso de eliminación de cliente

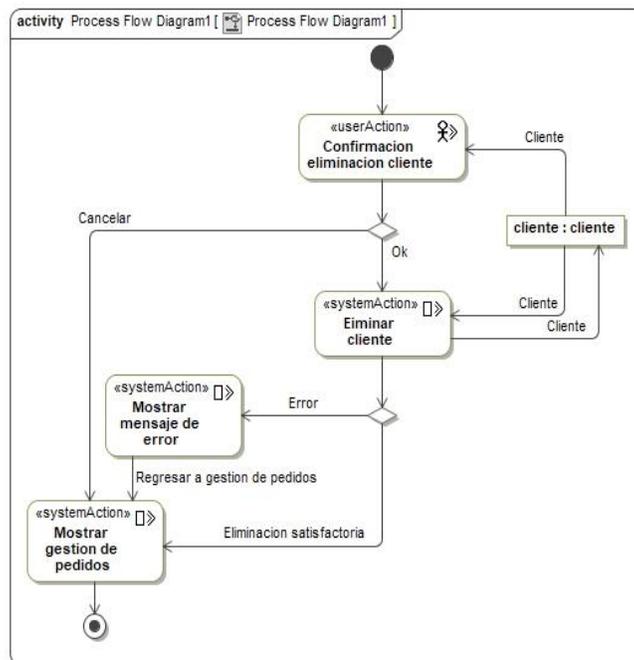


Figura 3.7 Modelo de proceso de pedido- eliminación de cliente
Fuente: Elaboración propia

c) Diagrama de flujo de proceso de actualizar cliente

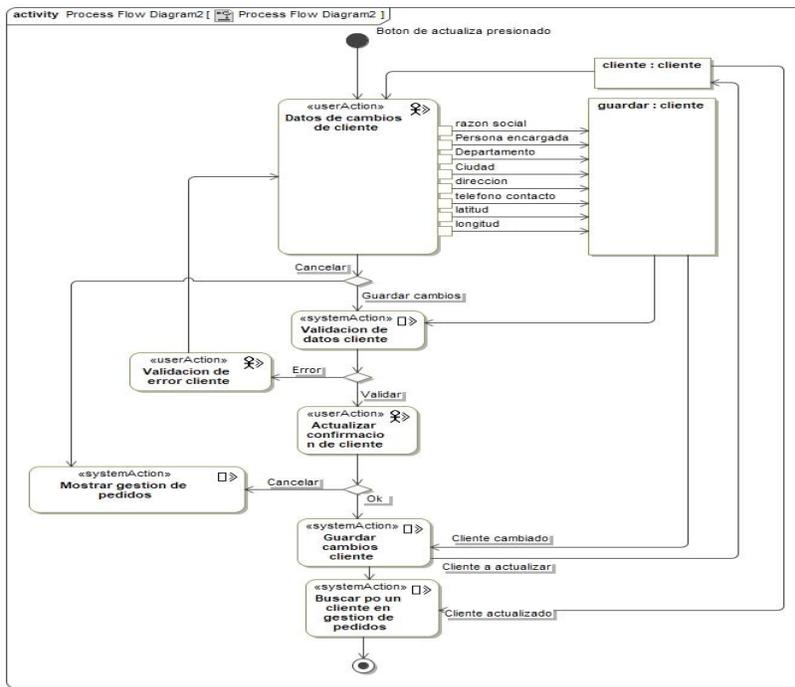


Figura 3.8 Modelo de proceso de pedido- eliminación de cliente
Fuente: Elaboración propia

a) Diagrama de flujo de proceso de pedido

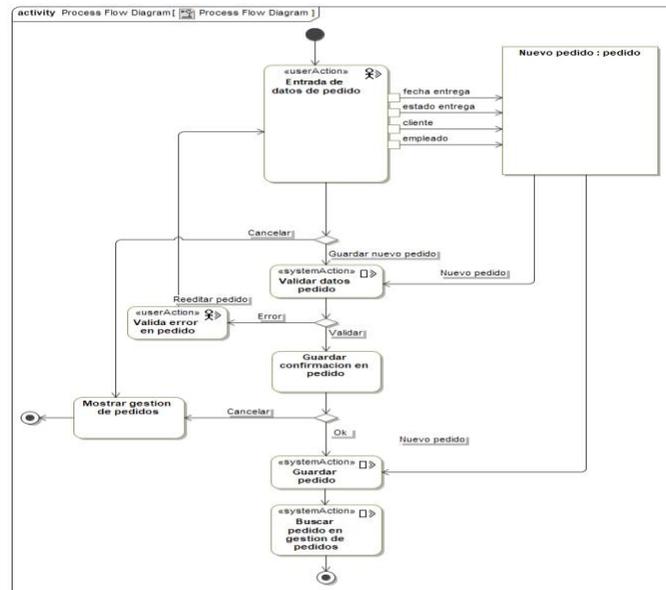


Figura 3.9 Modelo de proceso de pedido- registro de pedido
Fuente: Elaboración propia

b) Diagrama de flujo de proceso eliminar pedido

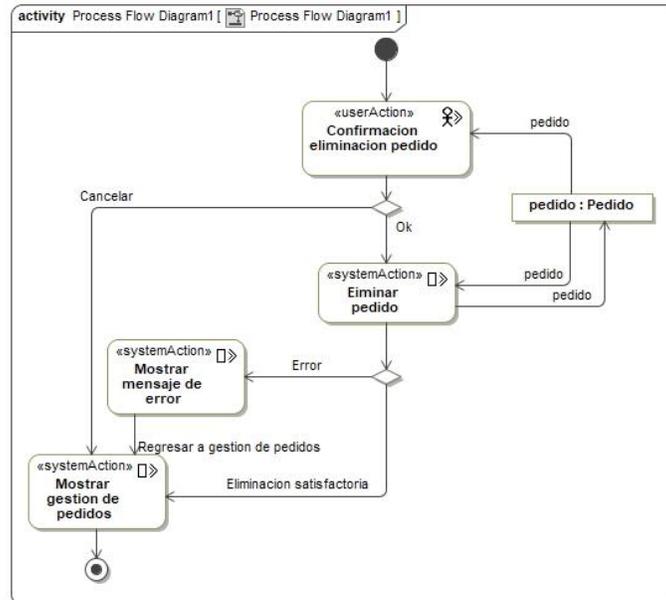


Figura 3.10 Modelo de proceso de pedido- elimina pedido
Fuente: Elaboración propia

c) Diagrama de flujo de proceso actualizar pedido

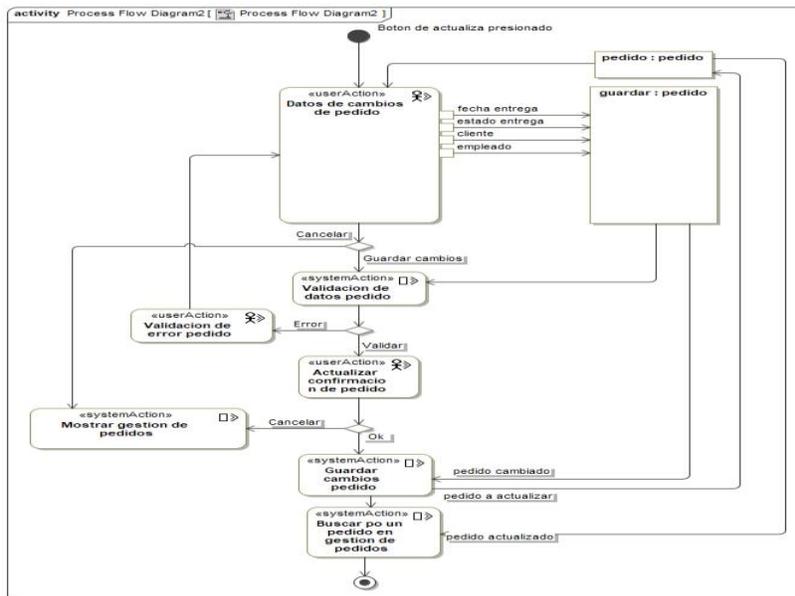


Figura 3.11 Modelo de proceso de pedido- actualiza pedido
Fuente: Elaboración propia

Diagrama de casos de uso almacén

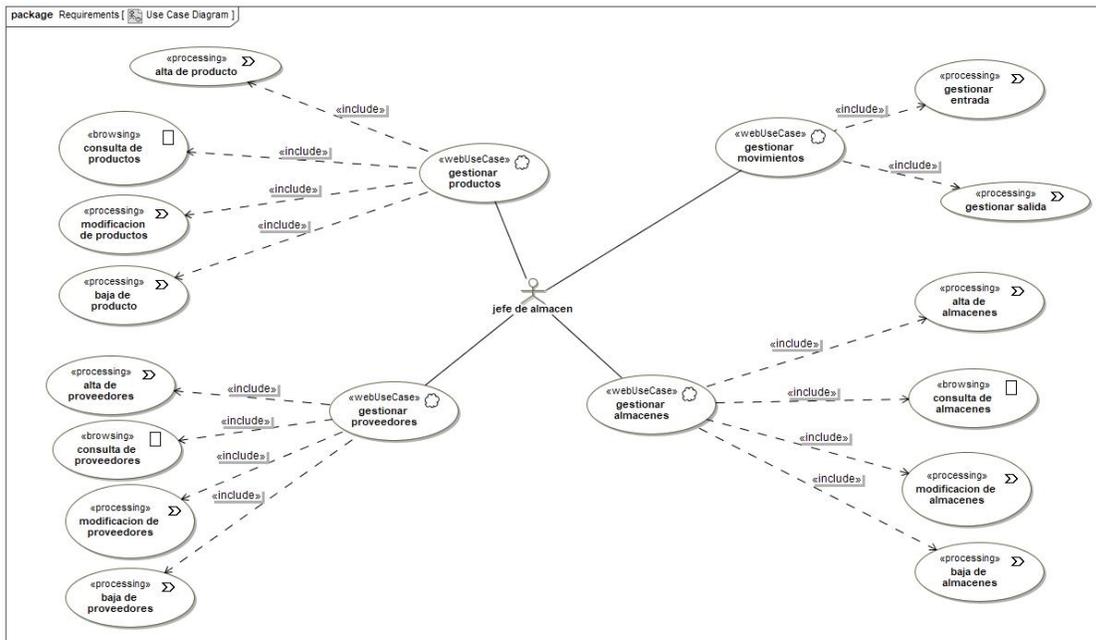


Figura 3.12 Modelo caso de uso de gestión de almacén
Fuente: Elaboración propia

Diagrama de contenido almacén

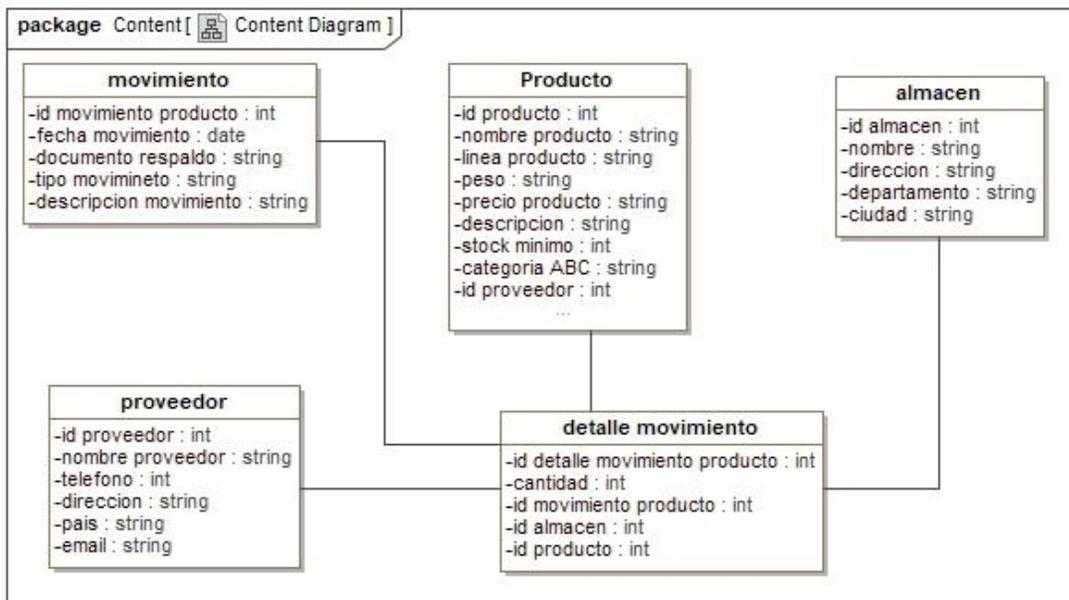


Figura 3.13 Modelo de contenido gestión de almacén
Fuente: Elaboración propia

Diagrama de navegación de almacén

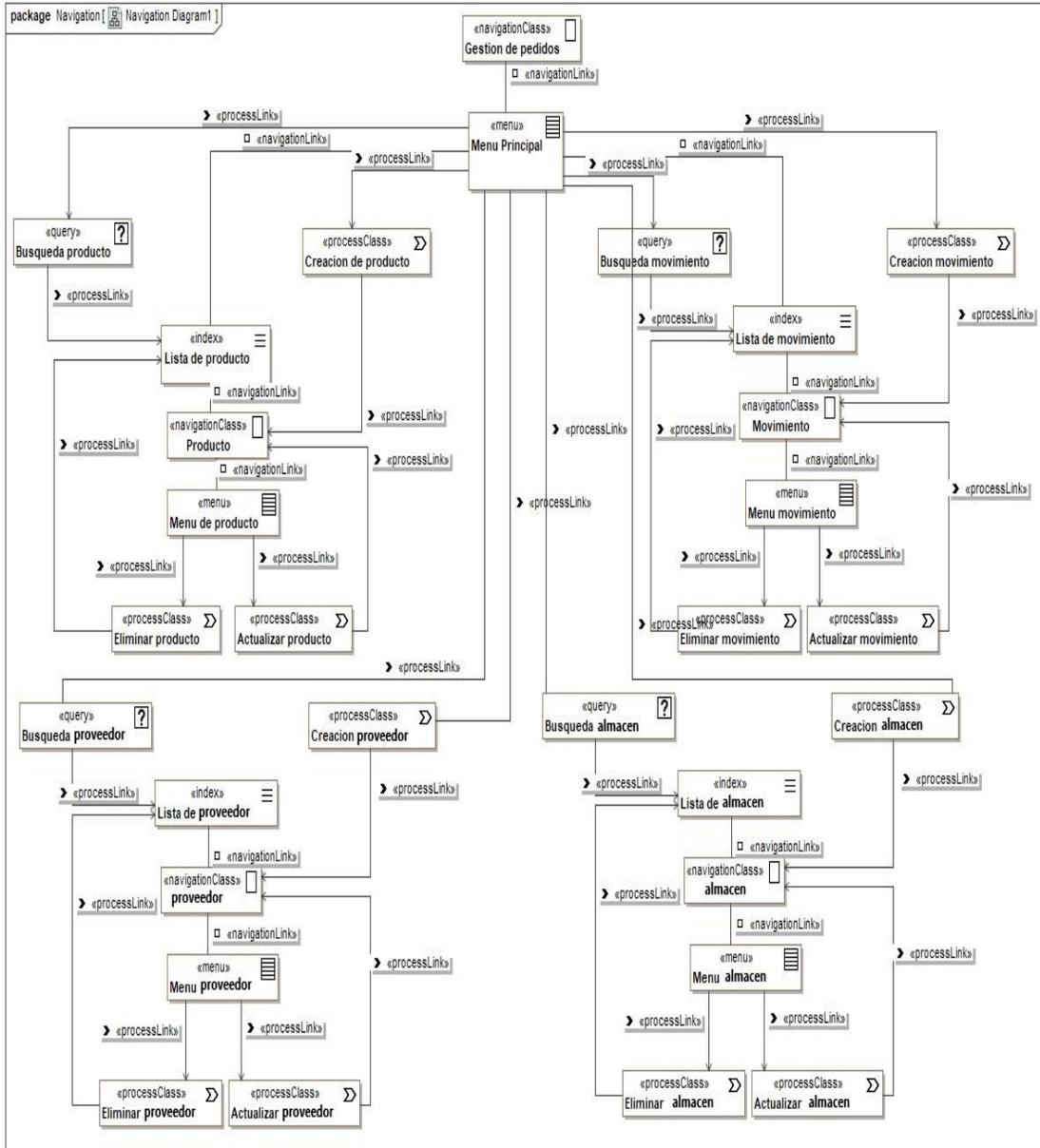


Figura 3.14 Modelo de navegación de gestión de almacén
Fuente: Elaboración propia

Diagrama de presentación de almacén

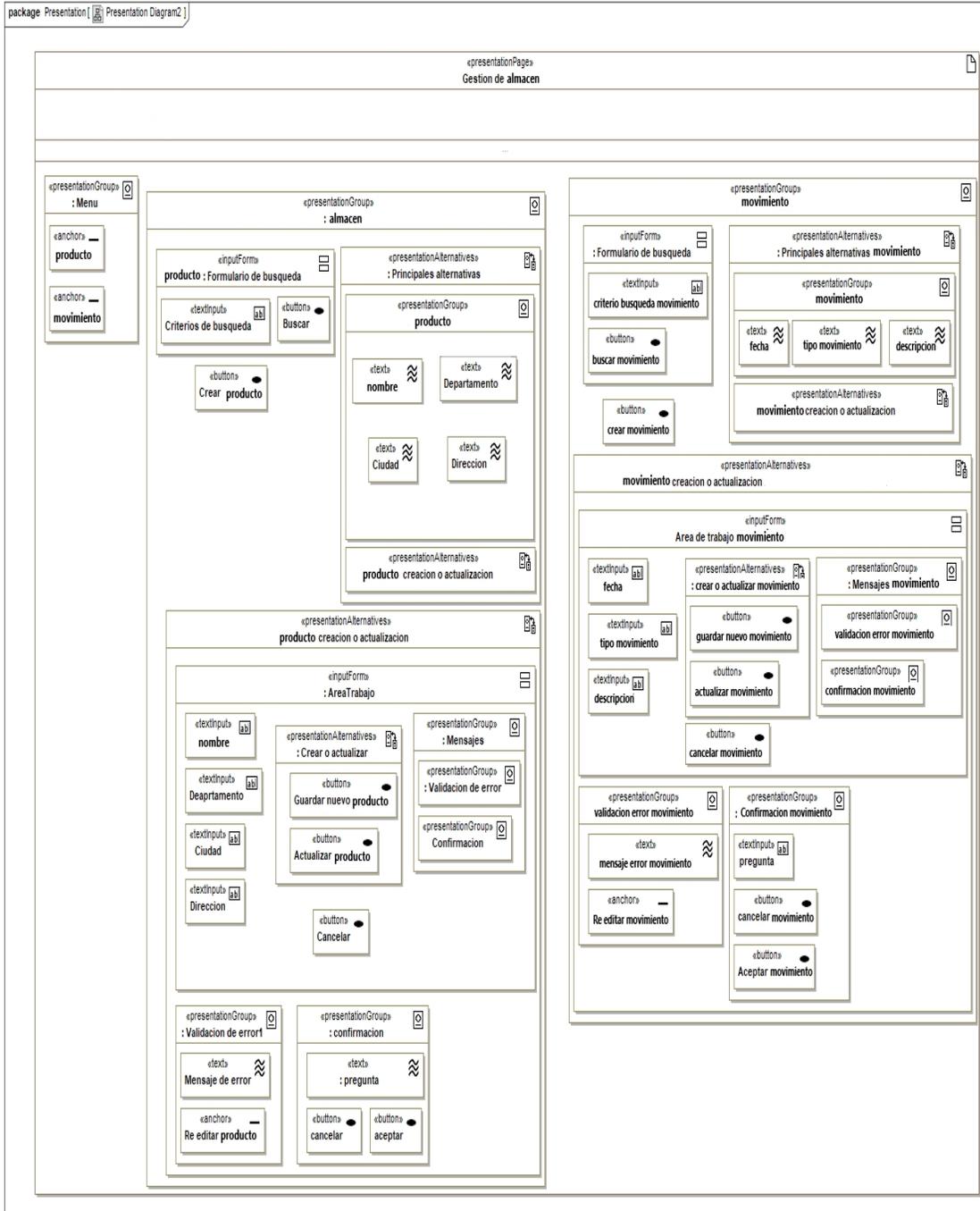


Figura 3.14 Modelo de presentación gestión de almacén 1
Fuente: Elaboración propia

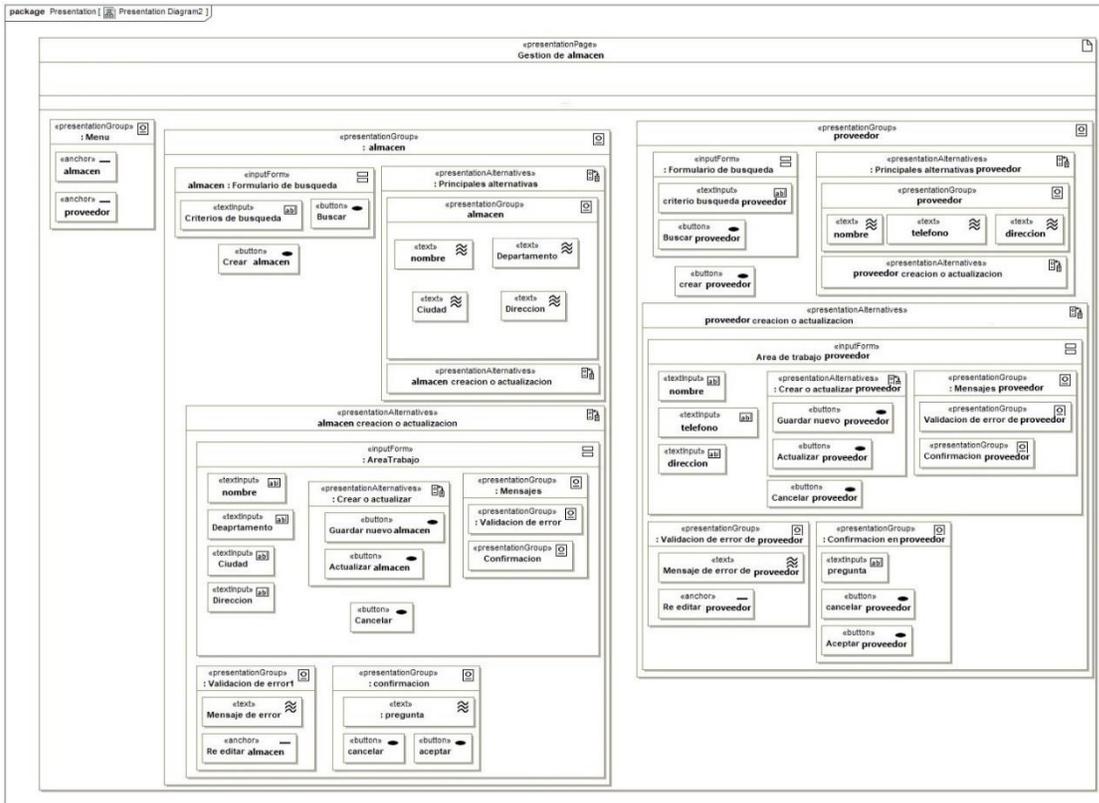


Figura 3.15 Modelo de presentación gestión de almacén 2
Fuente: Elaboración propia

Diagrama de procesos de almacén

a) Crear almacén

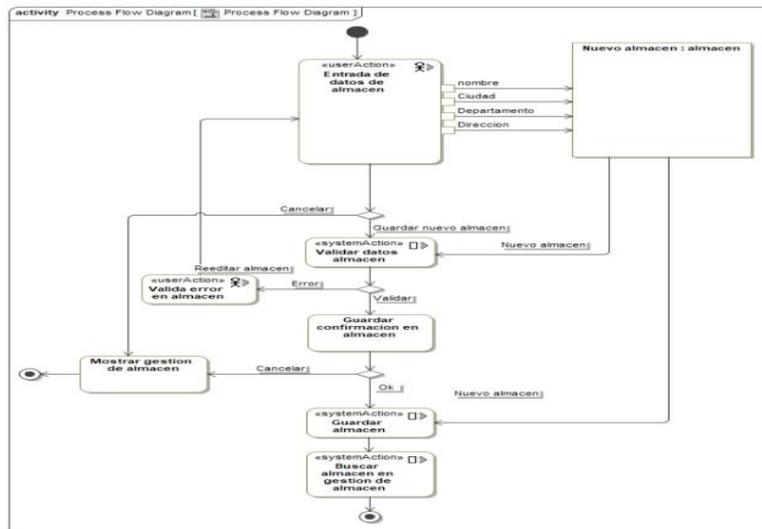


Figura 3.16 Modelo de procesos gestión de almacén-registro de almacén
Fuente: Elaboración propia

b) Eliminar almacén

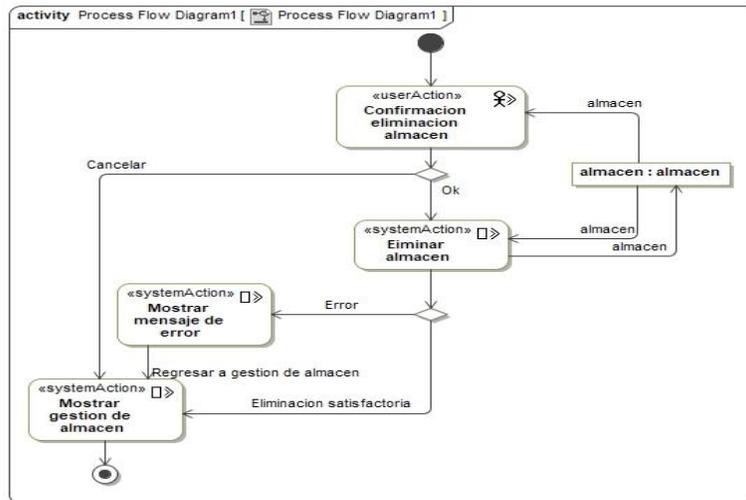


Figura 3.17 Modelo de procesos gestión de almacén-elimina almacén
Fuente: Elaboración propia

c) Actualizar almacén

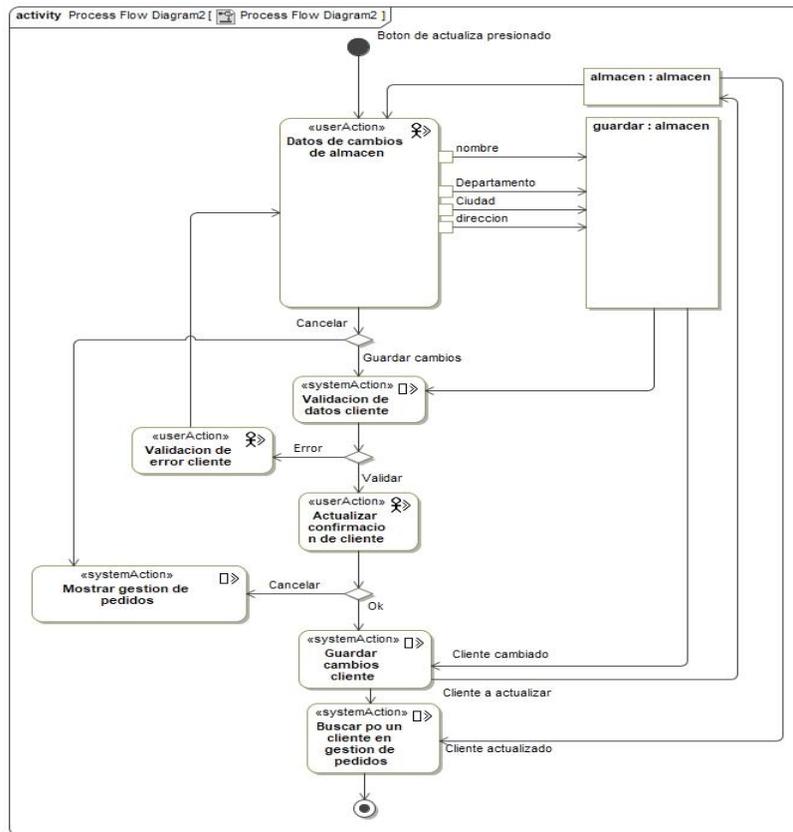


Figura 3.18 Modelo de procesos gestión de almacén-actualiza almacén
Fuente: Elaboración propia

a) Crear movimiento

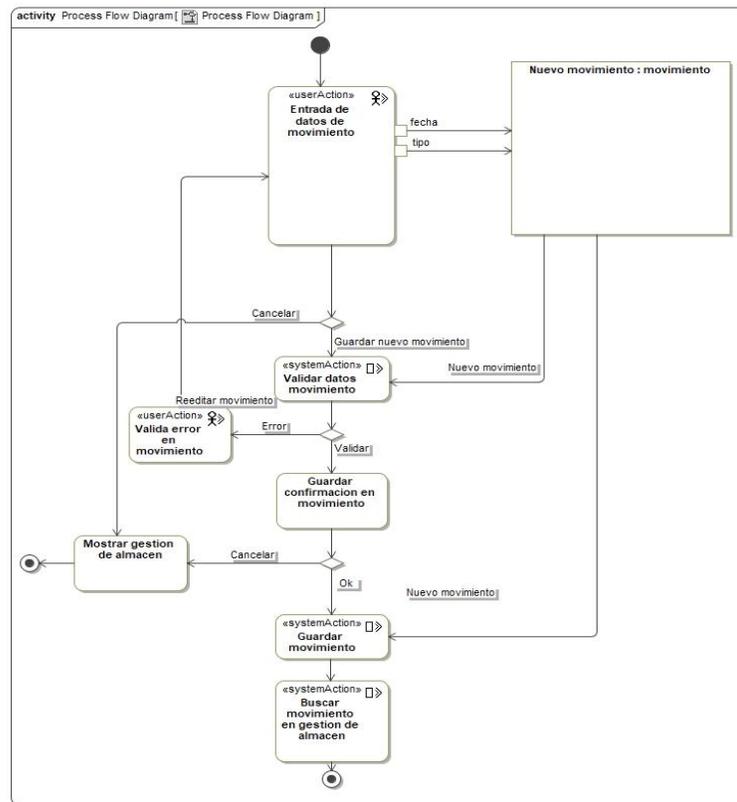


Figura 3.19 Modelo de procesos gestión de almacén-registra movimiento
Fuente: Elaboración propia

b) Eliminar movimiento

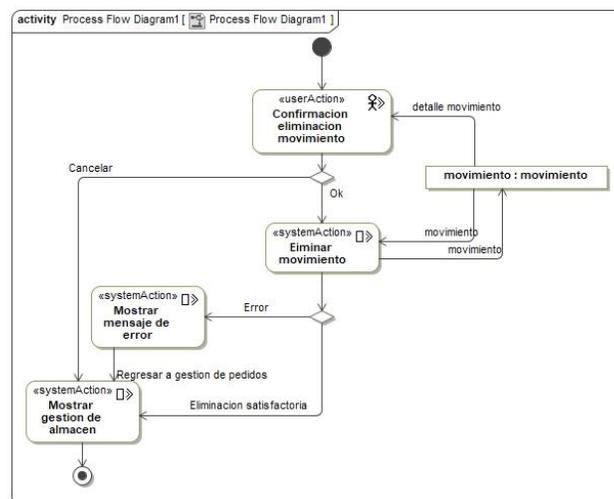


Figura 3.20 Modelo de procesos gestión de almacén-elimina movimiento
Fuente: Elaboración propia

c) Actualizar movimiento

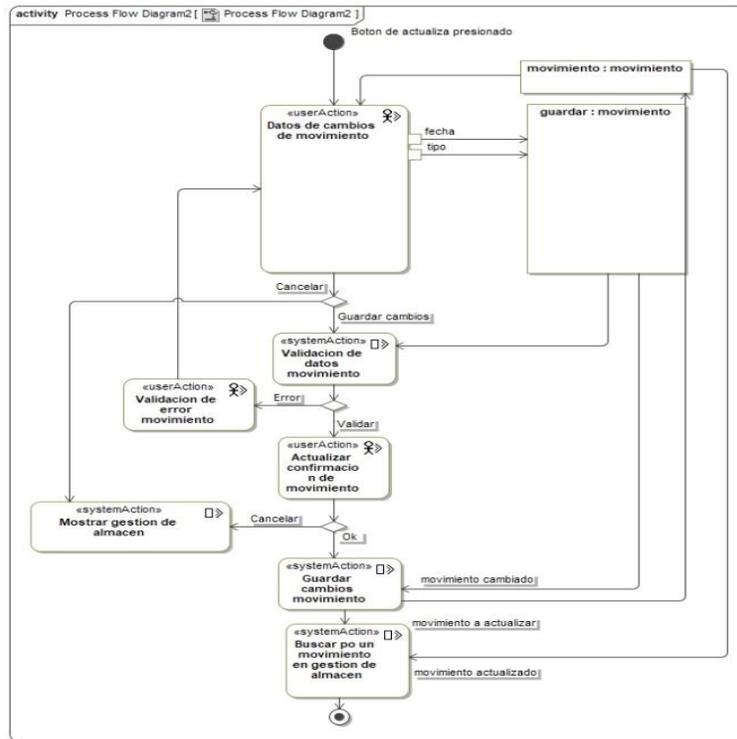


Figura 3.21 Modelo de procesos gestión de almacén-actualiza movimiento
Fuente: Elaboración propia

a) Crear producto

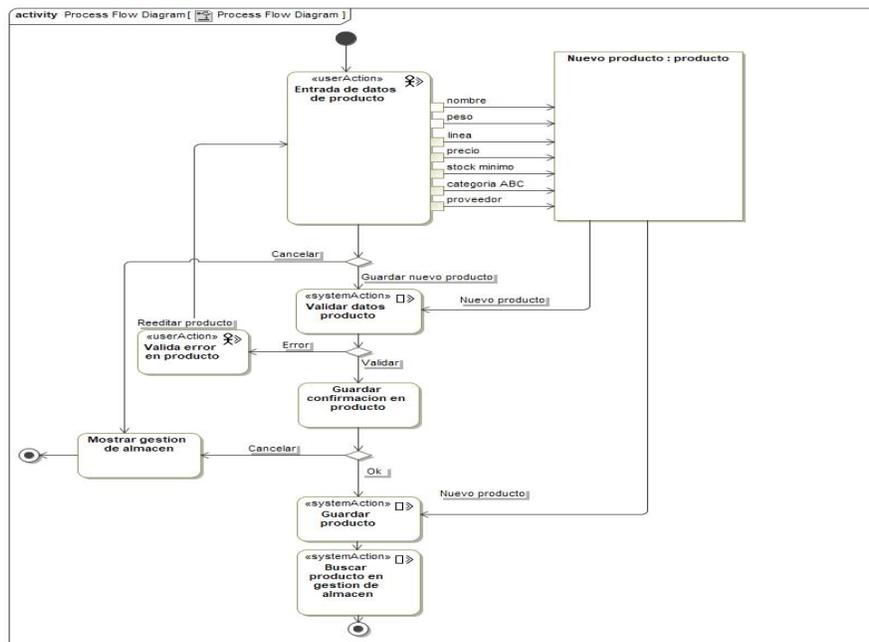


Figura 3.22 Modelo de procesos gestión de almacén-registra producto
Fuente: Elaboración propia

b) Eliminar producto

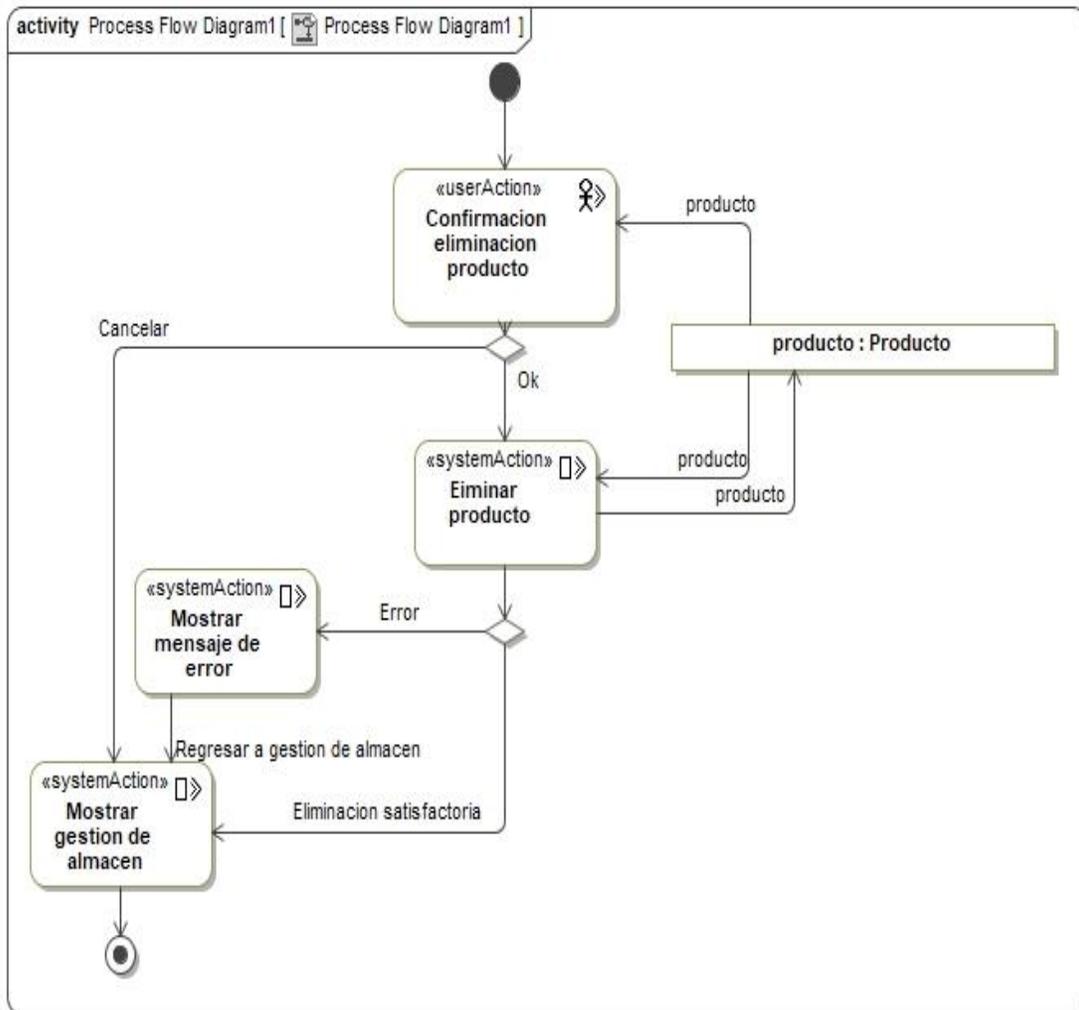


Figura 3.23 Modelo de procesos gestión de almacén-elimina producto
Fuente: Elaboración propia

c) Actualizar producto

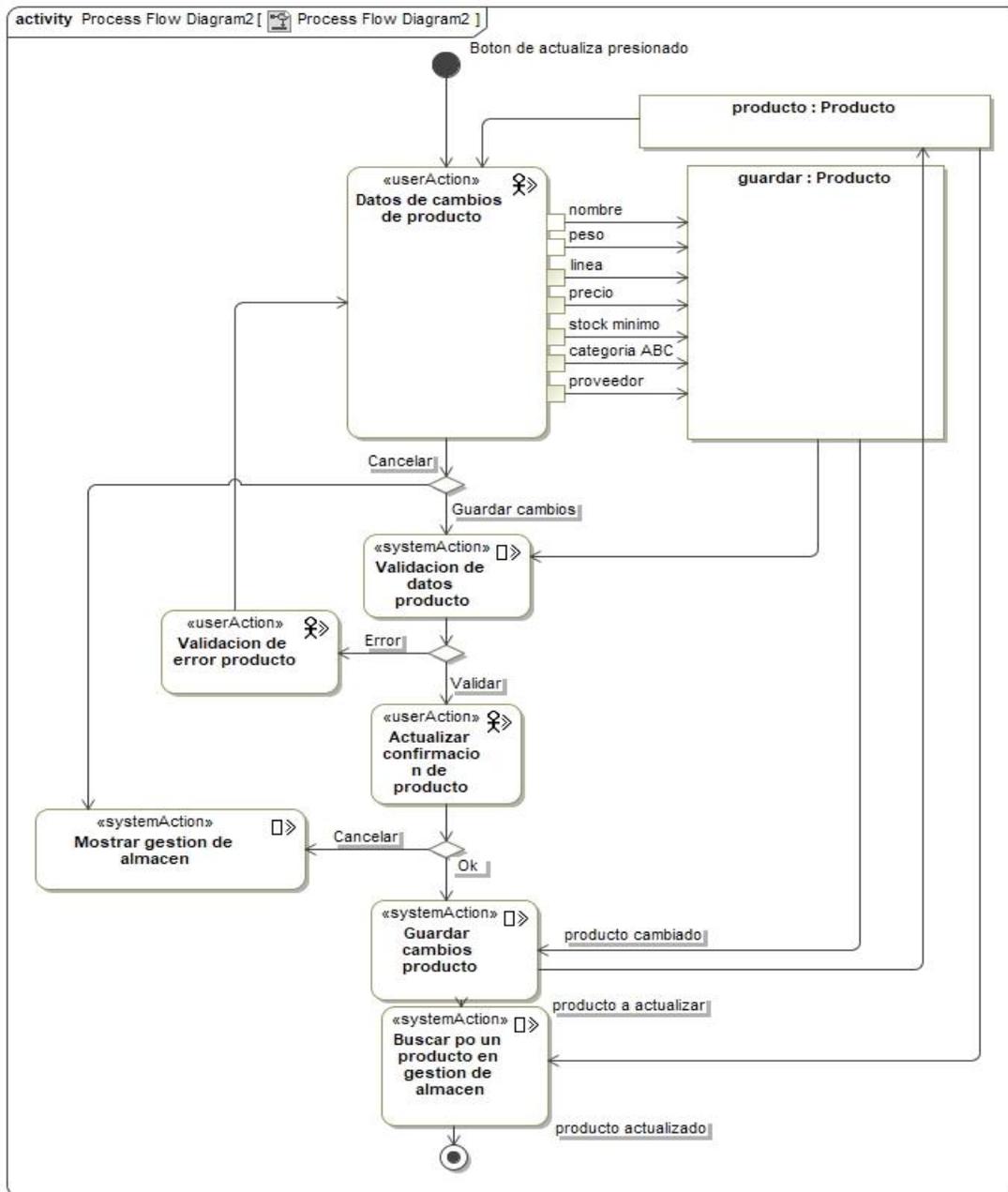


Figura 3.24 Modelo de procesos gestión de almacén-actualiza producto
Fuente: Elaboración propia

a) Crear proveedor

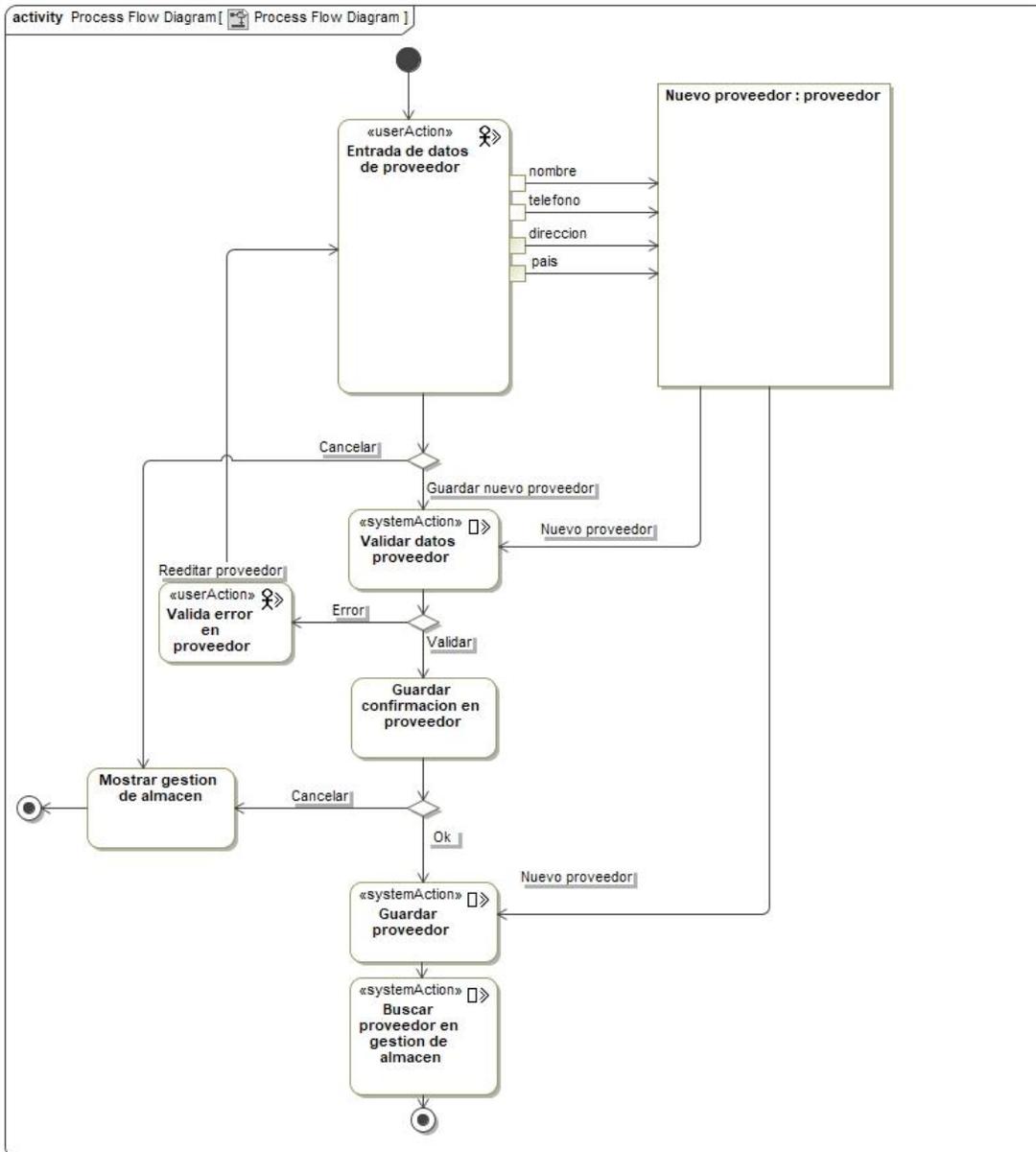


Figura 3.25 Modelo de procesos gestión de almacén-actualiza proveedor
Fuente: Elaboración propia

b) Eliminar proveedor

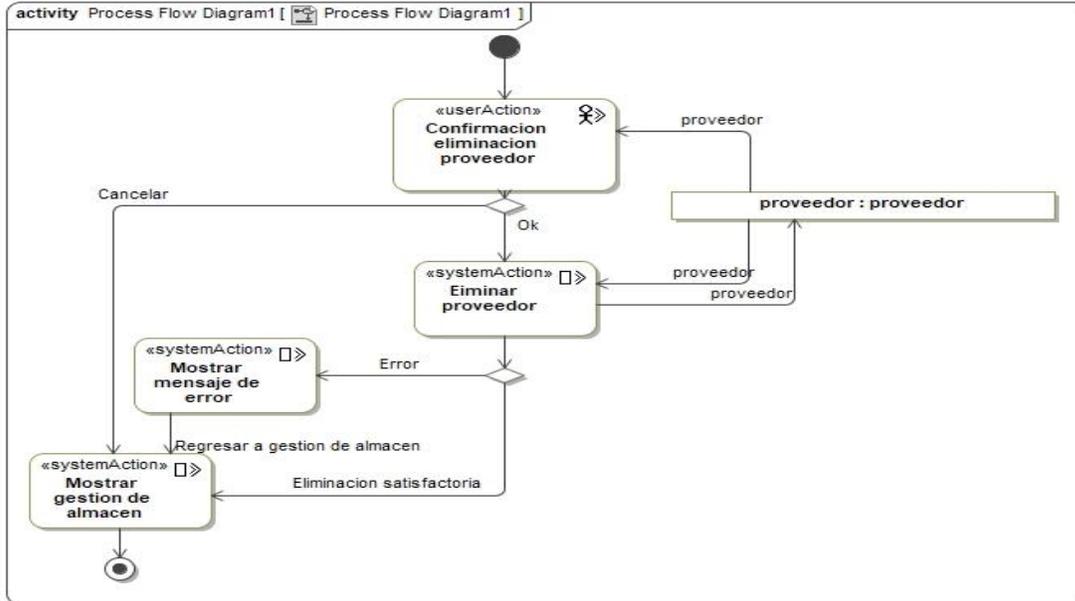


Figura 3.26 Modelo de procesos gestión de almacén-elimina proveedor
Fuente: Elaboración propia

c) Actualizar proveedor

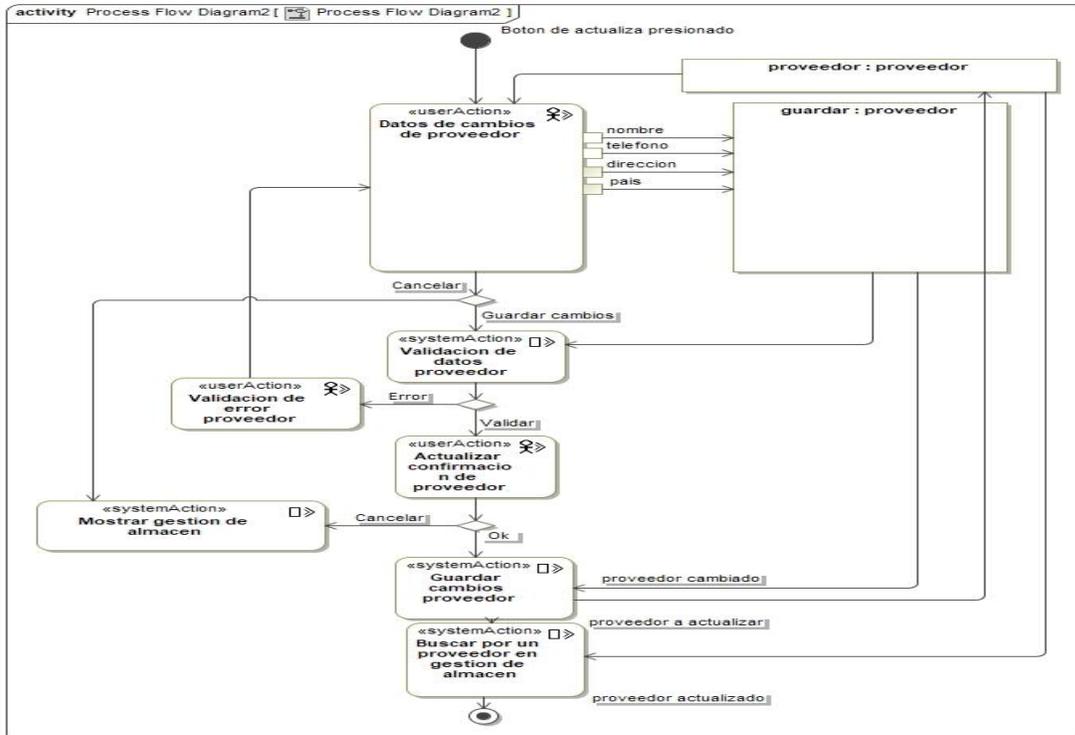


Figura 3.27 Modelo de procesos gestión de almacén-actualiza proveedor
Fuente: Elaboración propia

Diagrama de casos de uso de distribución

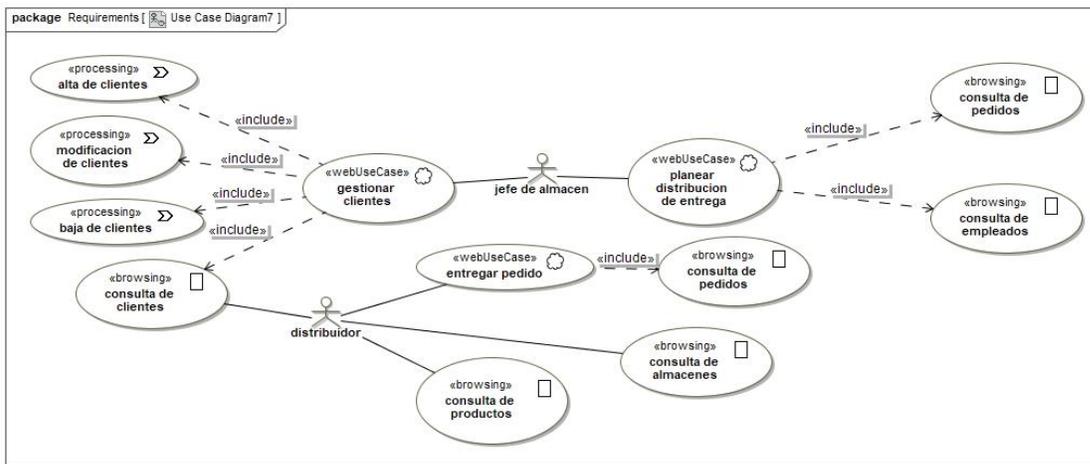


Figura 3.28 Modelo de casos de uso en distribución
Fuente: Elaboración propia

Diagrama de contenido de distribución

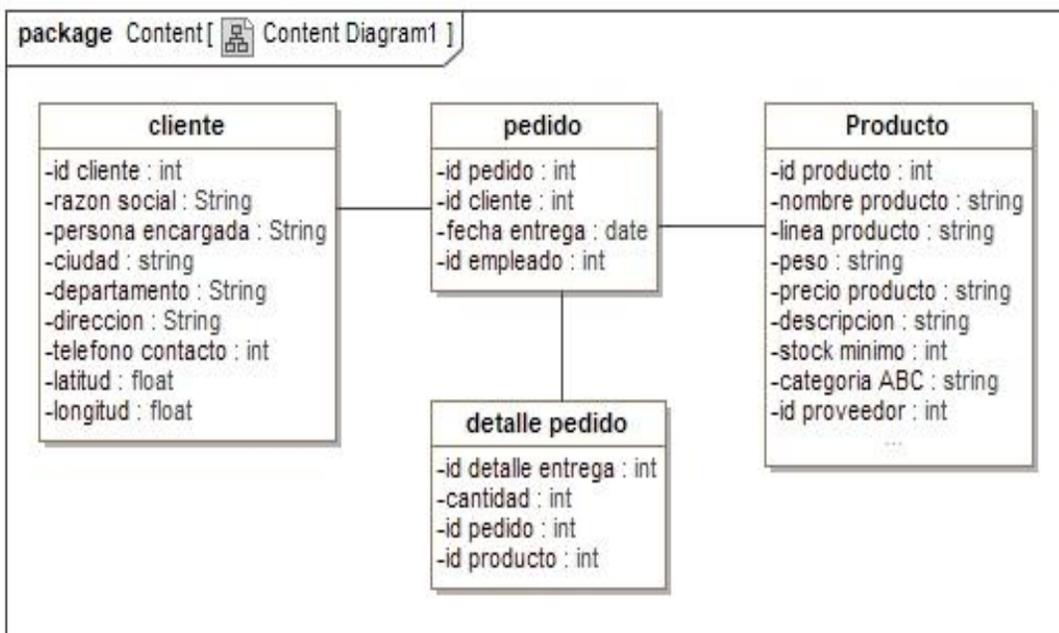


Figura 3.29 Modelo de contenido de distribución
Fuente: Elaboración propia

Diagrama de navegación de distribución

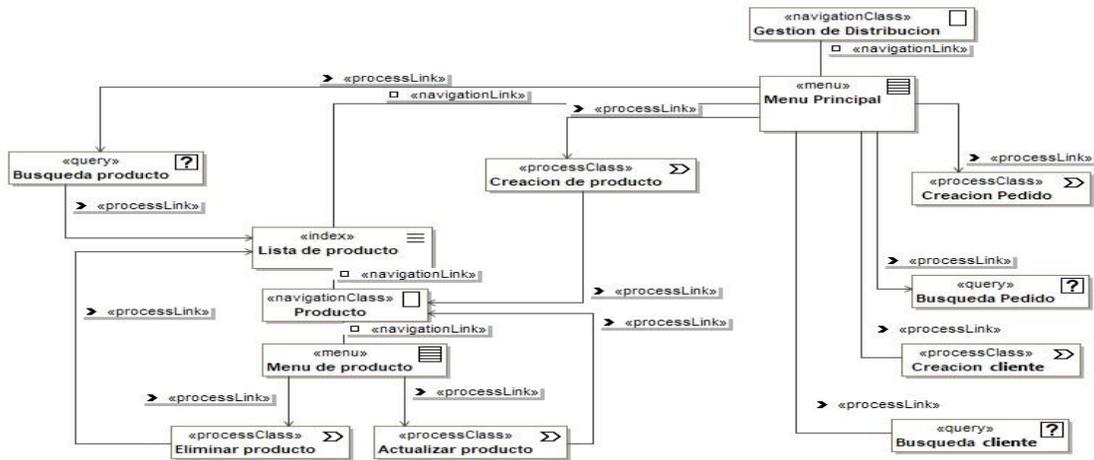


Figura 3.30 Modelo de navegación de distribución
Fuente: Elaboración propia

Diagrama de presentación de distribución

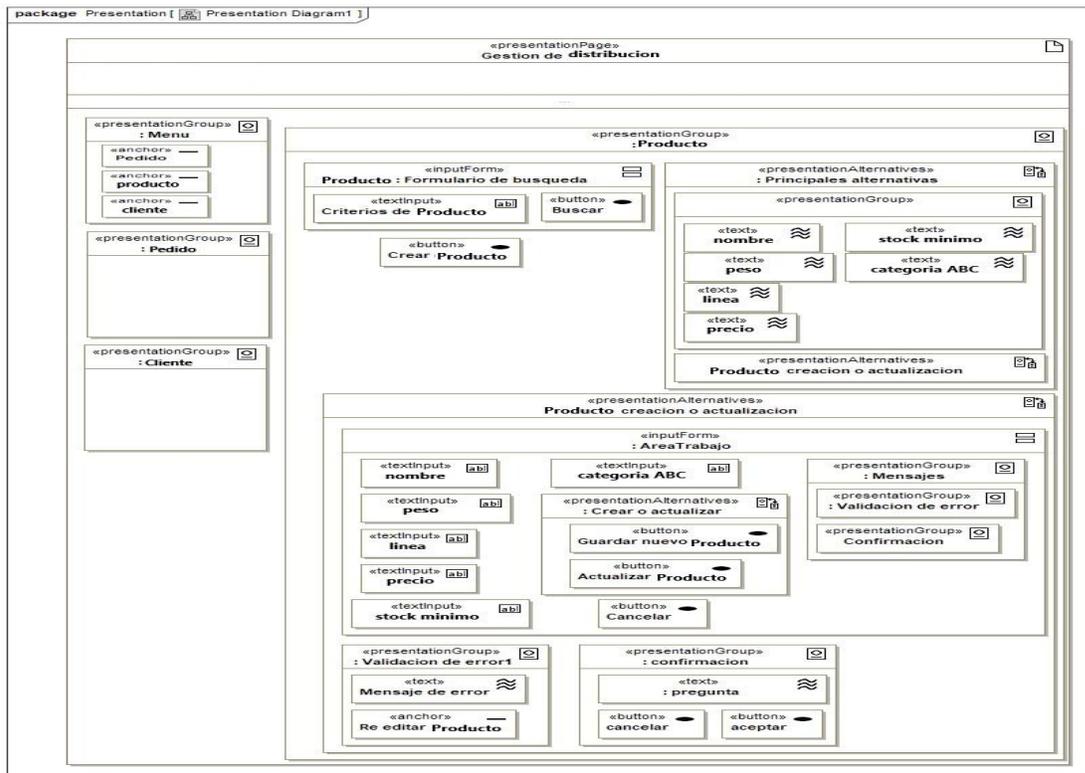


Figura 3.31 Modelo de presentación de distribución
Fuente: Elaboración propia

3.4 FASE DE CODIFICACION O CONSTRUCCION

3.4.1 Pantallas del sistema

A continuación, se muestra las pantallas del sistema web y las pantallas de operaciones esenciales que se realiza.

- **Pantalla de inicio de sesión:** Un usuario al ingresar a la ruta del sistema web se encuentra esta pantalla de inicio de sección, donde si el usuario tiene un nombre de usuario podrá ingresar al sistema.

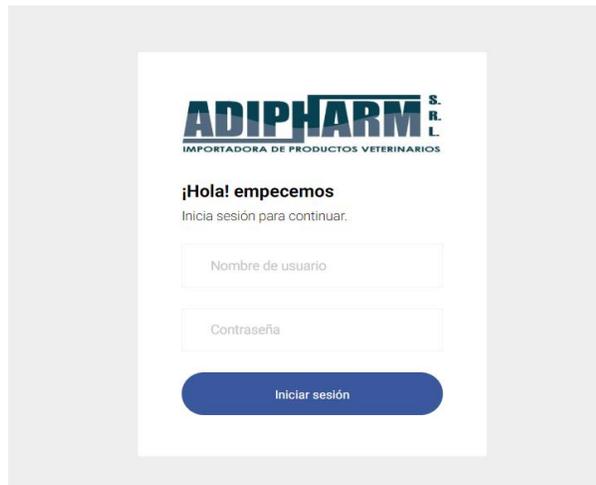


Figura 3.33 Pantalla de inicio de sesión.
Fuente: Elaboración propia.

Pantalla de inicio: Posteriormente del ingreso, de acuerdo al acceso que tiene cada usuario.

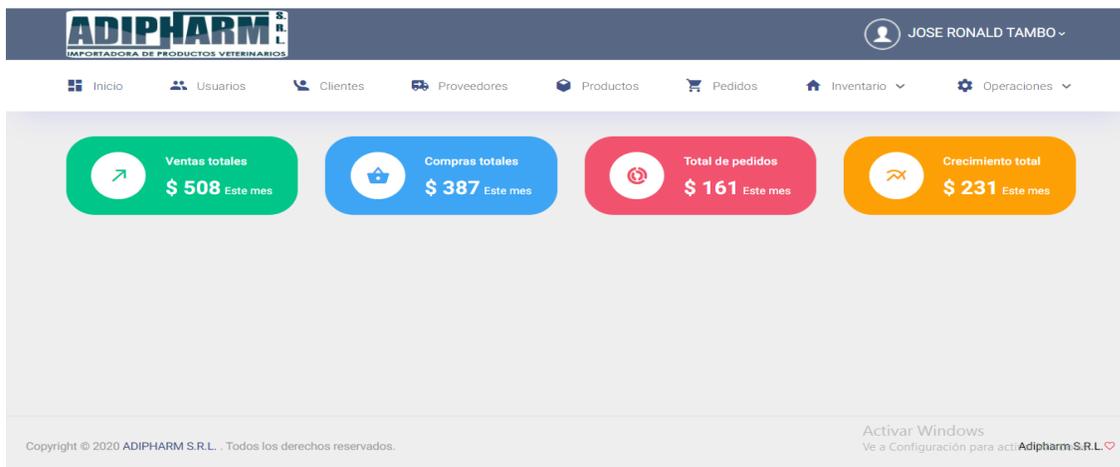


Figura 3.34 Pantalla principal.
Fuente: Elaboración propia.

Pantalla de control de usuarios: En la siguiente imagen se puede observar la lista de empleados registrados en el sistema, donde podrá modificar datos de usuario o deshabilitar el mismo, para poder realizar estas acciones debe tener los permisos necesarios.



Figura 3.35 Pantalla gestión de usuarios
Fuente: Elaboración propia.

La siguiente imagen se observa el registro de un nuevo usuario al sistema, donde el usuario puede elegir registrar al empleado usuario y la asignarle un rol de usuario, también su nombre de usuario y su contraseña.

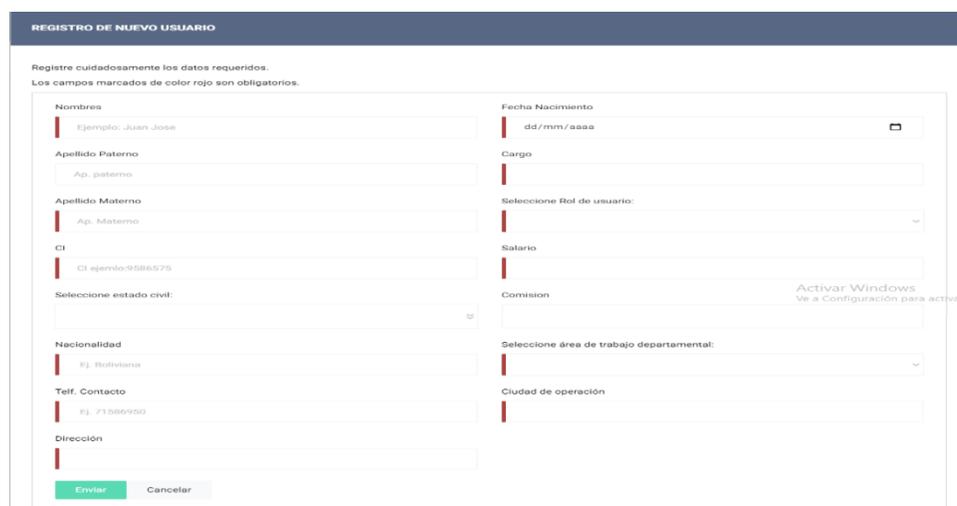


Figura 3.36 Pantalla gestión de usuarios
Fuente: Elaboración propia.

En la siguiente figura se muestra el reporte del módulo respectivo.



**SISTEMA DE CONTROL
ALMACENES Y DISTRIBUCION**

Fecha:
10/7/2020

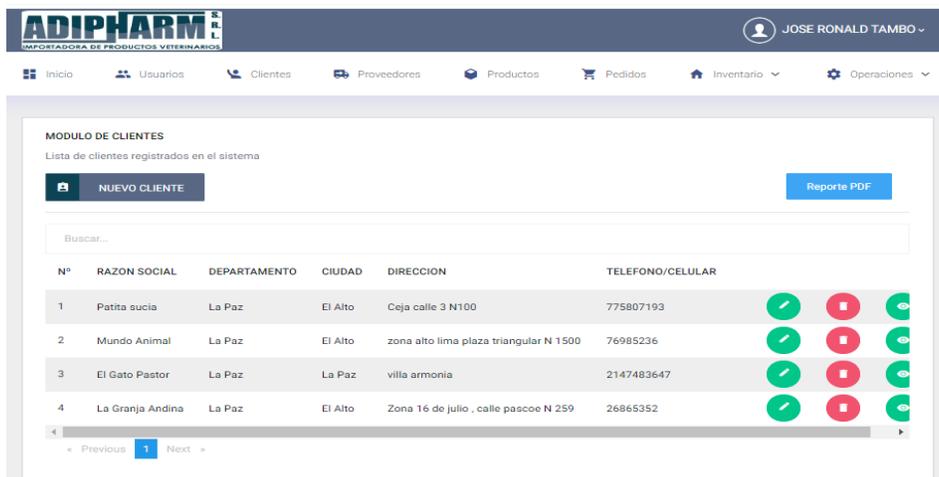
REPORTE DE USUARIOS

Lista de usuario activos

Nº	NOMBRES	AP. PATERNO	AP. MATERNO	Telf/Cel	DIRECCION	ROL USUARIO	CARGO
1	Juan Jose	Mamani	Mamani	78569832	Zona de los palotes	DISTRIBUIDOR	Distribuidor
2	JOSE ALEX	BAPTISTA	QUISPE	65987542	ZONA VILLA ADELA PLAZA SANTIDAD N 258	VENDEDOR	AYUDANTE
3	JOSE RONALD	TAMBO	ARCANI	61216486	ZONA SANTA ROSA CALLE F N 24A	ADMIN	JEFE DE ALMACENES

Figura 3.37 Pantalla gestión de usuarios
Fuente: Elaboración propia.

Pantalla de gestión de clientes: En la siguiente imagen se puede observar la lista de clientes registrados en el sistema detallando la ubicación geográfica del mismo, donde podrá modificar datos de usuario, sacar reportes o deshabilitar el mismo, para poder realizar estas acciones debe tener los permisos necesarios.



Nº	RAZON SOCIAL	DEPARTAMENTO	CIUDAD	DIRECCION	TELEFONO/CELULAR			
1	Patita sucia	La Paz	El Alto	Ceja calle 3 N100	775807193	✓	■	🗨
2	Mundo Animal	La Paz	El Alto	zona alto lima plaza triangular N 1500	76985236	✓	■	🗨
3	El Gato Pastor	La Paz	La Paz	villa armonia	2147483647	✓	■	🗨
4	La Granja Andina	La Paz	El Alto	Zona 16 de julio , calle pascoe N 259	26865352	✓	■	🗨

Figura 3.38 Pantalla gestión de clientes
Fuente: Elaboración propia.

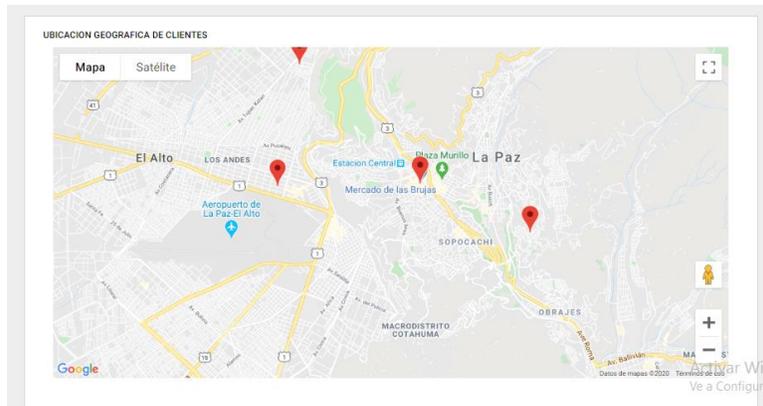


Figura 3.39 Pantalla gestión de clientes ubicación geográfica
Fuente: Elaboración propia.

Pantalla de gestión de proveedor: En la siguiente imagen se puede observar la lista de proveedores registrados en el sistema donde podrá modificar los datos de proveedor, sacar reportes o deshabilitar el mismo, para poder realizar estas acciones debe tener los permisos necesarios.



Figura 3.40 Pantalla gestión de proveedores
Fuente: Elaboración propia.

Pantalla para la gestión de productos: En la siguiente imagen se puede observar la lista de productos registrados en el sistema donde podrá modificar los datos de productos, sacar reportes o deshabilitar el mismo, para poder realizar estas acciones debe tener los permisos necesarios.

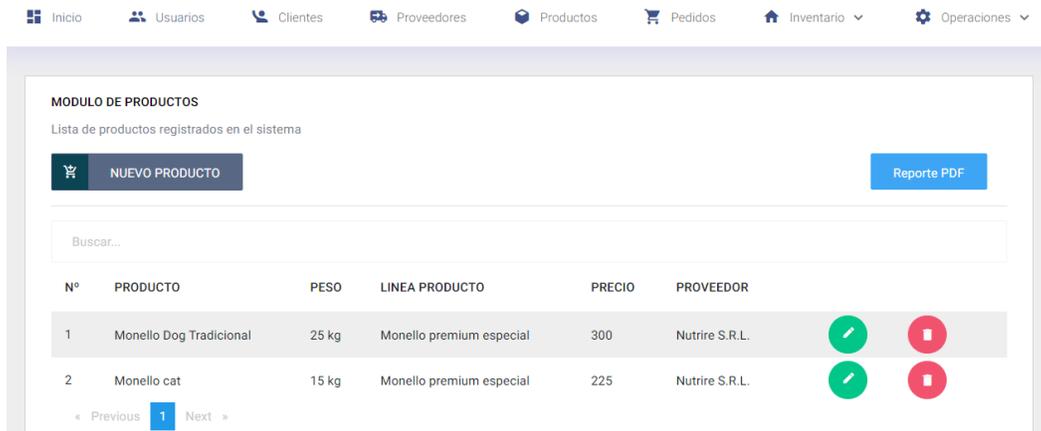


Figura 3.41 Pantalla gestión de productos
Fuente: Elaboración propia.

Pantalla para la gestión de pedidos: En la siguiente imagen se puede observar la lista de pedidos registrados en el sistema donde podrá modificar los datos de pedidos, sacar reportes o deshabilitar el mismo, para poder realizar estas acciones debe tener los permisos necesarios.

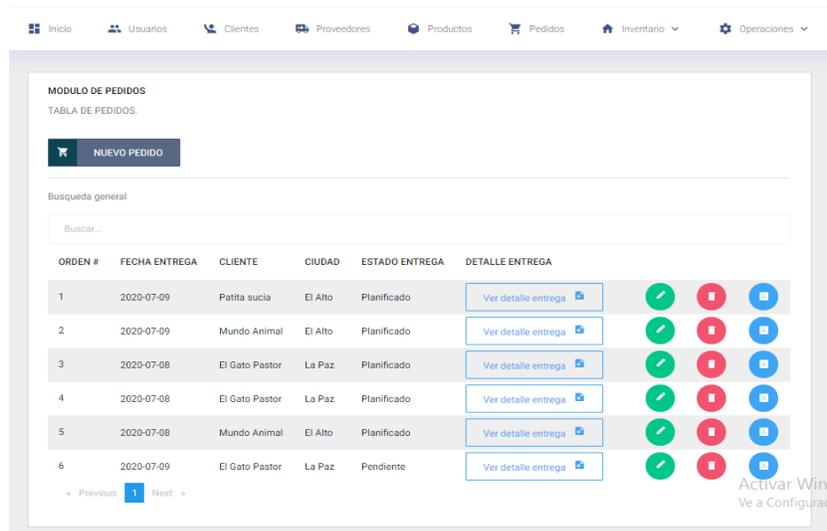


Figura 3.42 Pantalla gestión de pedidos
Fuente: Elaboración propia.

Pantalla para la gestión de almacenes: En la siguiente imagen se puede observar la lista de almacenes registrados en el sistema donde podrá modificar los datos de almacenes, sacar reportes o deshabilitar el mismo, para poder realizar estas acciones debe tener los permisos necesarios.

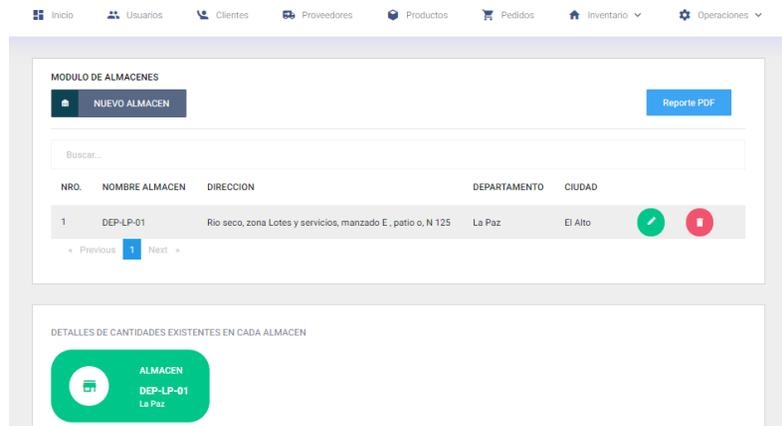


Figura 3.43 Pantalla gestión de almacenes
Fuente: Elaboración propia.

Pantalla para la gestión de movimientos: En la siguiente imagen se puede observar la lista de movimientos registrados en el sistema donde podrá modificar los datos de movimientos, sacar reportes o deshabilitar el mismo, para poder realizar estas acciones debe tener los permisos necesarios.

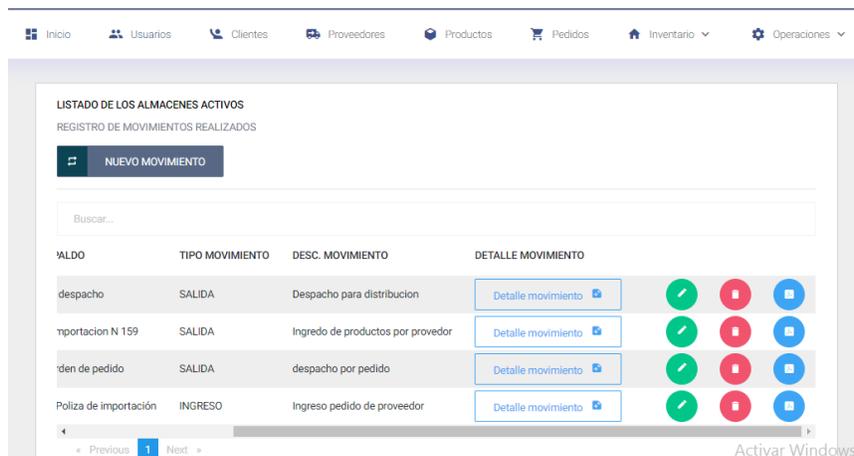


Figura 3.44 Pantalla gestión de movimientos
Fuente: Elaboración propia.

Pantalla para la gestión de planificación: En la siguiente imagen se puede observar la lista de pedidos programados para planificación.

Podrá actualizar el estado actual o deshabilitar dicho registro, visualizará la planeación, así como el recorrido visualizado en el mapa, para poder realizar estas acciones debe tener los permisos necesarios.

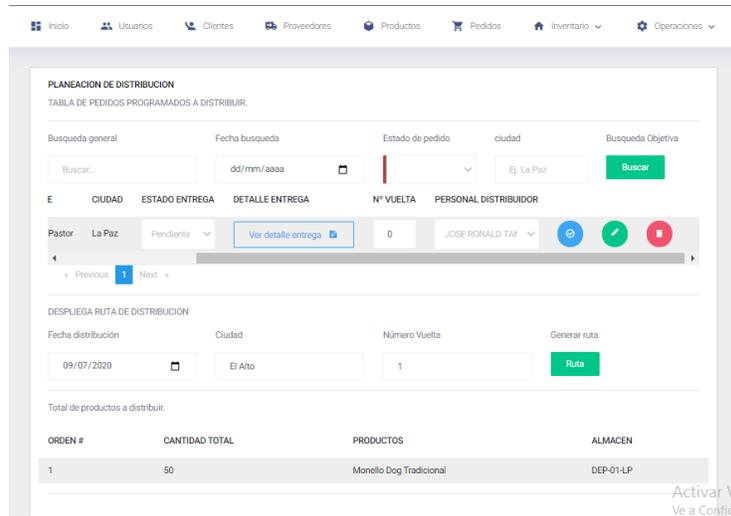


Figura 3.45 Pantalla gestión de planificación de ruta
Fuente: Elaboración propia.

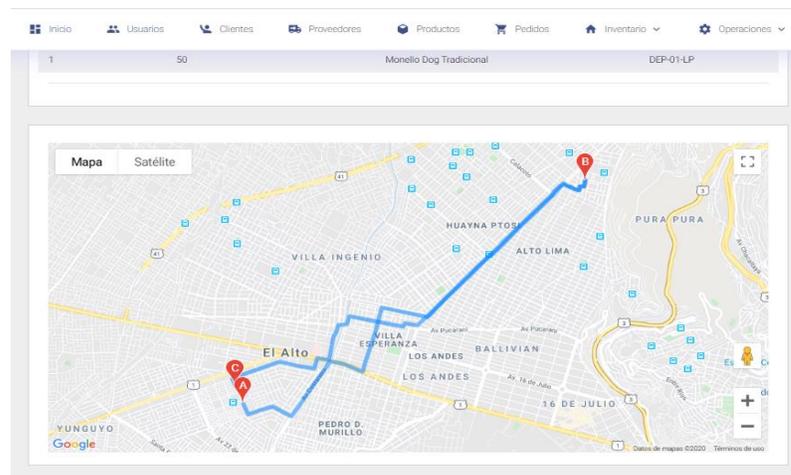
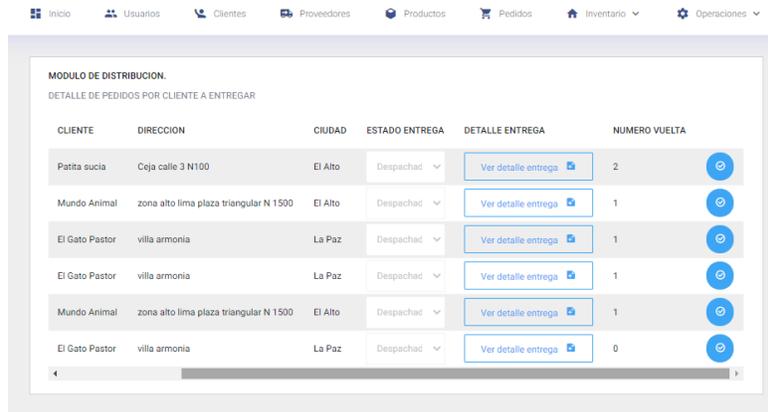


Figura 3.46 Pantalla gestión de planificación, ruta generada
Fuente: Elaboración propia.

Pantalla de gestión para distribución: En la siguiente imagen se puede observar la lista de pedidos programados para distribución.

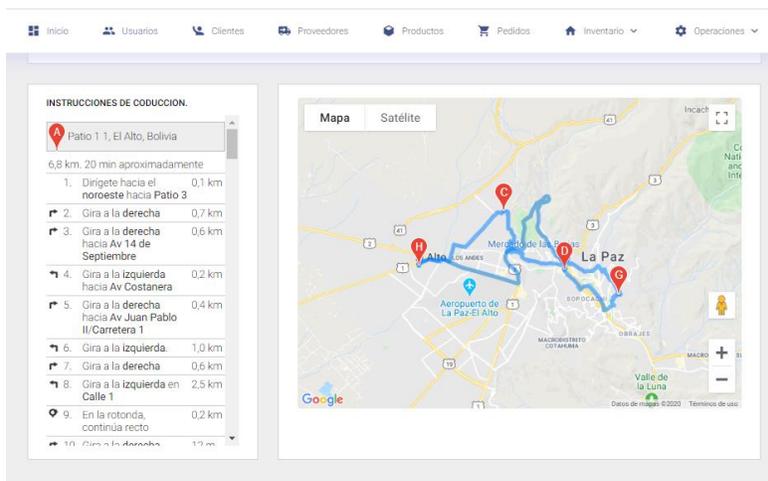
Podrá actualizar el estado actual de dicho registro, visualizará las instrucciones de conducción, así como el recorrido visualizado en el mapa, para poder realizar estas acciones debe tener los permisos necesarios.



MODULO DE DISTRIBUCION.
DETALLE DE PEDIDOS POR CLIENTE A ENTREGAR

CLIENTE	DIRECCION	CIUDAD	ESTADO ENTREGA	DETALLE ENTREGA	NUMERO VUELTA
Patita sucia	Coja calle 3 N100	El Alto	Despachad	Ver detalle entrega	2
Mundo Animal	zona alto lima plaza triangular N 1500	El Alto	Despachad	Ver detalle entrega	1
El Gato Pastor	villa armonia	La Paz	Despachad	Ver detalle entrega	1
El Gato Pastor	villa armonia	La Paz	Despachad	Ver detalle entrega	1
Mundo Animal	zona alto lima plaza triangular N 1500	El Alto	Despachad	Ver detalle entrega	1
El Gato Pastor	villa armonia	La Paz	Despachad	Ver detalle entrega	0

Figura 3.47 Pantalla gestión de distribución, lista de pedidos
Fuente: Elaboración propia.



INSTRUCCIONES DE CODUCCION.

A Patio 1 1, El Alto, Bolivia

6,8 km, 20 min aproximadamente

1. Dirigete hacia el noroeste hacia Patio 3 0,1 km
2. Gira a la derecha 0,7 km
3. Gira a la derecha hacia Av 14 de Septiembre 0,6 km
4. Gira a la izquierda hacia Av Costanera 0,2 km
5. Gira a la derecha hacia Av Juan Pablo III/Carretera 1 0,4 km
6. Gira a la izquierda. 1,0 km
7. Gira a la derecha 0,6 km
8. Gira a la izquierda en Calle 1 2,5 km
9. En la rotonda, continúa recto 0,2 km
10. Gira a la derecha 1,2 km

Mapa Satélite

Mapa de Google que muestra una ruta azul a través de El Alto y La Paz, Bolivia. Se marcan puntos de entrega con letras A, B, C, D, E y F. El mapa incluye etiquetas para 'Aeropuerto de La Paz-El Alto', 'Merced de la Paz', 'La Paz', 'Valle de la Luna' y 'MACRODISTRITO COTAHUMALI'.

Figura 3.48 Pantalla gestión de distribución, instrucciones de conducción y ruta
Fuente: Elaboración propia.

3.5 FASE DE PRUEBAS

Una vez finalizado el desarrollo de las primeras cuatro etapas, se realiza las pruebas necesarias para garantizar el funcionamiento del sistema.

El uso de las pruebas funcionales es para asegurar correcto trabajo de entrada de datos, la navegación en el sistema, procedimientos y obtención de resultados.

La siguiente tabla muestra las pruebas conjuntas ya que los funcionamientos de cada uno de módulos son similares:

Usuarios, Productos, Proveedores, Pedidos, Almacenes y movimientos.

Tabla 3.2
Caso de prueba para la gestión de módulos Usuarios, Productos, Proveedores, Pedidos, Almacenes y movimientos.

PROCEDIMIENTO	USUARIOS	MODULO	DESCRIPCION	VALOR
Prueba previa requerida.	Gerente , Jefe almacén, Distribuidor, Vendedor, Ayudante	Usuarios	Autenticado y con privilegios para el área	si
		Productos		si
		Proveedores		si
		Pedido		si
		Almacén		si
		Movimiento		si
SECUENCIA DE PRUEBA				
PROCEDIMIENTO	MODULOS	RESULTADO ESPERADO	VALOR	
Registrar datos de nuevo, modificar datos y eliminar.	Usuarios	El sistema registre los datos añadidos, modificados y eliminarlos	Si	
	Productos		Si	
	Proveedores		Si	
	Pedido		Si	
	Almacén		Si	
	Movimiento		Si	
FALLAS ENCONTRADAS	MODULOS	DESCRIPCION	GRAVEDAD	
Ninguna	Productos	Ninguna	-	
Ninguna	Proveedores	Ninguna	-	
Ninguna	Pedido	Ninguna	-	
Ninguna	Almacén	Ninguna	-	
Ninguna	Movimiento	Ninguna	-	
PASOS DE PRUEBA		RESULTADOS ESPERADOS	SI	NO
1	Se prueba el registro de un nuevo objeto	Se debe registrar un nuevo objeto y el visualizar el registro para verificar que la operación se realizó con éxito.	SI	
2	Se elige un objeto existente y se procede a editar los valores de su registro.	Posteriormente al cambio los valores son cambiado, y un mensaje de confirmación	SI	
3	Eliminación de un objeto.	Posteriormente a la eliminación los valores son cambiados y un mensaje de confirmación	SI	
COMENTARIO DE LA PRUEBA REALIZADA				

Las pruebas en los módulos se efectuaron con absoluta normalidad. Se obtuvo el resultado esperado en cuanto al registro, modificación y eliminación de registros, se mostraron las respuestas al modificar, agregar o eliminar un registro.

Fuente: Elaboración propia.

Tabla 3.3
Caso de prueba- Gestión de clientes.

PROCEDIMIENTO		DESCRIPCION	VALOR	
Prueba previa requerida		Autenticado y con privilegios para el área	Si	
Usuario		Gerente jefe de almacén y vendedor		
SECUENCIA DE PRUEBA				
PROCEDIMIENTOS		RESULTADOS ESPERADOS	VALOR	
Registro de cliente con coordenadas,		Realiza el proceso de registro	Si	
Edita datos de cliente		Realiza la edición de registro	Si	
Elimina registro de cliente		Elimina el registro	Si	
FALLAS ENCONTRADAS		DESCRIPCION	GRAVEDAD	
Ninguna		Ninguna	-	
PASOS DE PRUEBA		RESULTADO ESPERADO	SI	NO
1	Se prueba el registro de un nuevo cliente	Se debe registrar un nuevo cliente y el visualizar el registro para verificar que la operación se realizó con éxito.	SI	
2	Se elige un cliente existente y se procede a editar los valores de su registro.	Posteriormente al cambio los valores son cambiado, y un mensaje de confirmación	SI	
3	Eliminación de un cliente.	Posteriormente a la eliminación los valores son cambiados y un mensaje de confirmación	SI	
COMENTARIO DE LA PRUEBA REALIZADA				
Las pruebas en el módulo se efectuaron con absoluta normalidad. Se obtuvo el resultado esperado en cuanto al registro, modificación y eliminación de registros, se mostraron las respuestas al modificar, agregar o eliminar un registro.				

Fuente: Elaboración Propia

Tabla 3.4
Caso de prueba- gestión de planificación

PROCEDIMIENTO		DESCRIPCION	VALOR	
Prueba previa requerida		Autenticado y con privilegios para el área	Si	
Usuario		Gerente jefe de almacén y vendedor		
SECUENCIA DE PRUEBA				
PROCEDIMIENTOS		RESULTADOS ESPERADOS	VALOR	

Cambio de estado de los pedidos	Realiza el cambio de estado y asigna distribuidor	Si	
Mostrar cantidades totales de distribución.	Sumar las cantidades totales	Si	
Visualizar el recorrido de ruta	Mostrar el mapa los puntos de distribución.	Si	
FALLAS ENCONTRADAS	DESCRIPCION	GRAVEDAD	
Ninguna	Ninguna	-	
PASOS DE PRUEBA	RESULTADO ESPERADO	SI	NO
1	Se prueba el cambio de estado del pedido.	Cambio de estado del pedido para planificación y despacho	SI
2	Se prueba la suma de cantidades totales de producto.	Suma total de los productos en base a la cantidad de pedidos.	SI
3	Generar la ruta en base a los pedidos planificados	Generar la ruta óptima para la ruta de distribución	SI
COMENTARIO DE LA PRUEBA REALIZADA			
Las pruebas se realizaron con tal normalidad en el módulo de planificación. Se obtuvo el resultado esperado en cuando al cambio de estados de pedidos para planificación y despacho. Se mostraron la suma de las cantidades totales.			

Fuente: Elaboración Propia

Tabla 3.5
Caso de prueba – gestión de distribución

PROCEDIMIENTO	DESCRIPCION	VALOR	
Prueba previa requerida	Autenticado y con privilegios para el área	Si	
Usuario	Gerente jefe de almacén y distribuidor.		
SECUENCIA DE PRUEBA			
PROCEDIMIENTOS	RESULTADOS ESPERADOS	VALOR	
Lista de pedidos a distribuir	Muestra los pedidos a distribuir con fecha programada.	Si	
Cambio de estado al realizar la entrega	Realizar el cambio de estado al entregar o no el producto.	Si	
Muestra de instrucciones de conducción y visualización del mapa y ruta	Desplegar las instrucciones de conducción y ruta en el mapa.	Si	
FALLAS ENCONTRADAS	DESCRIPCION	GRAVEDAD	
Ninguna	Ninguna	-	
PASOS DE PRUEBA	RESULTADO ESPERADO	SI	NO
1	Desplegar la lista de pedidos.	Mostrar la lista de pedidos planificados a distribuir.	SI
2	Cambio de estado de pedido.	Realizar el cambio de estado al realizar una entrega o no.	SI
3	Mostrar ruta e instrucciones de conducción	Genera ruta e instrucciones de conducción en base a los pedidos	SI

COMENTARIO DE LA PRUEBA REALIZADA

La prueba realizada en el módulo se hizo con tal normalidad. Los resultados esperados son los requeridos en el despliegue de la lista de pedidos, cambio de estado y la ruta e instrucciones de conducción.

Fuente: Elaboración propia

Prueba de caja blanca

Son pruebas estructurales. Conociendo el código y siguiendo su estructura lógica, se pueden diseñar pruebas destinadas a comprobar que el código hace correctamente lo que el diseño de bajo nivel indica y otras que demuestren que no se comporta adecuadamente ante determinadas situaciones. Ejemplos típicos de ello son las pruebas unitarias. Se centran en lo que hay codificado o diseñado a bajo nivel por lo que no es necesario conocer la especificación de requisitos, que por otra parte será difícil de relacionar con partes diseñadas a muy bajo nivel.

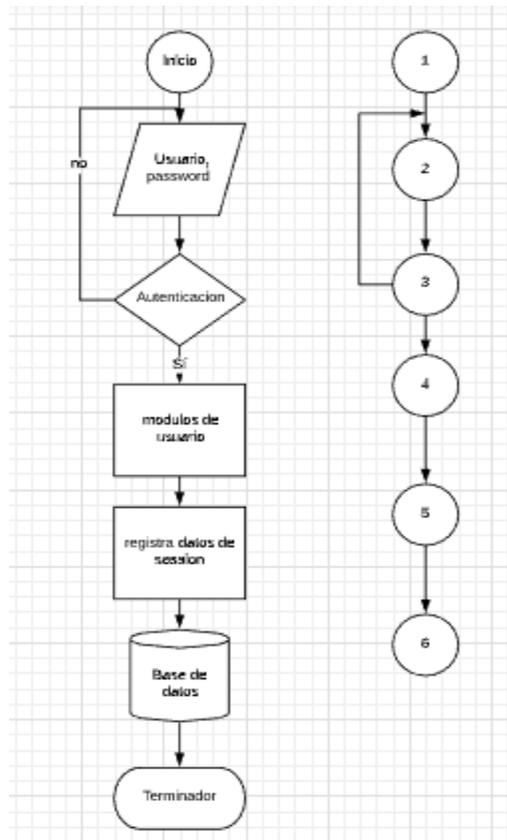


Figura:3.49 diagrama de flujo y grafo de ingreso de sesión

Fuente: Elaboración propia

La complejidad se obtiene en base al grafo de flujo se define como:

$$V(G)=E-N+2$$

Dónde: E= número de aristas y N= número de nodos.

$$V(G)=6-6+2=2$$

Por lo tanto, las existencias son de 2 caminos independientes, para los que realizamos las pruebas.

Caso de prueba camino 1

Valor (usuario=" JOSE", contraseña=" JOSEJOSE") = entrada validada ya que los datos ingresados existen en la base de datos, rol como admin.

Ingresa al sistema, visualiza los módulos. Resultados esperados correctos

Caso de prueba camino 2

Valor (usuario=" JOSE", contraseña=" ninguno") = entrada no validada ya que los datos ingresados no existen en la base de datos, vuelva a introducir datos.

Actualiza datos de entrada.

Resultados esperados correctos.

Pruebas de caja negra

Son pruebas funcionales. Se parte de los requisitos funcionales, a muy alto nivel. Las pruebas se aplican sobre el sistema empleando un determinado conjunto de datos de entrada y observando las salidas que se producen para determinar si la función se está desempeñando correctamente por el sistema bajo prueba. Las herramientas básicas son observar la funcionalidad y contrastar con la especificación.

La prueba de caja negra de caja negra se realizó en la parte de construcción de proyecto utilizando la herramienta de INSOMNIA el cual nos permite realizar las pruebas para verificar la correcta funcionalidad del sistema.

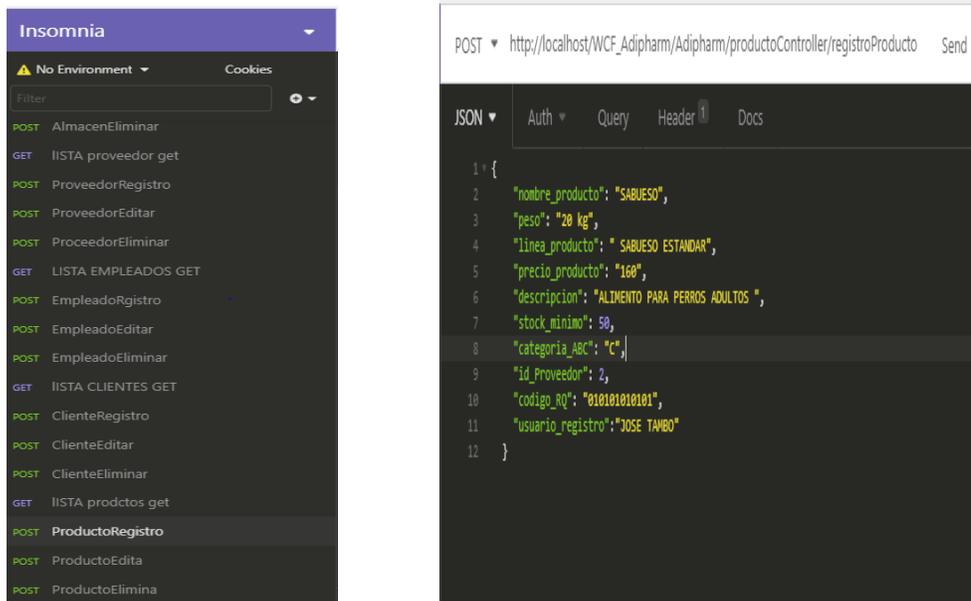


Figura 3.50 Interfaz de método y petición en pruebas de caja negra
Fuente: Elaboración propia.

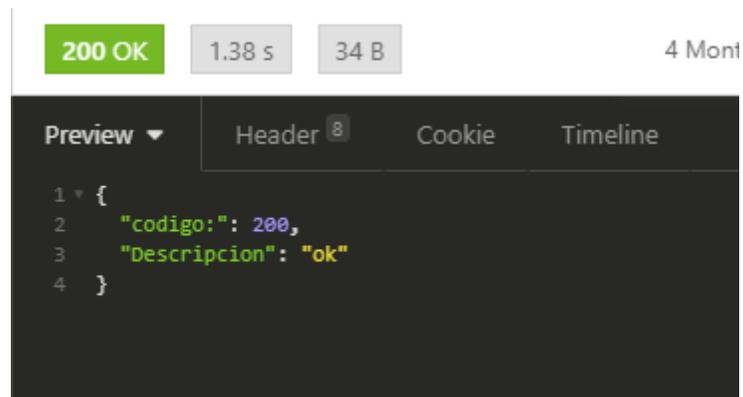


Figura 3.51 Interfaz de resultado de pruebas de caja negra
Fuente: Elaboración propia.

3.6 FASE DE INSTALACION O FASE DE IMPLEMENTACIÓN

Las actividades para la implementación del sistema.

- Instalar y/o configurar un navegador web en el sistema operativo.
- Instalar y configurar los servidores que requiere nuestro sistema (Apache).
- Instalar y configurar el gestor de base de datos (MariaDB).
- Instalar y configurar el servidor NodeJS para carga del front-end.

- Cargar el sistema para la empresa.
- Se establecen el usuario y las contraseñas en el gestor de base de datos.
- Se realiza la capacitación a los usuarios.

3.7 EVALUACIÓN DE CALIDAD DEL SISTEMA

Para evaluar la calidad del sistema se utilizará los factores de calidad McCall, que propone factores y métricas comúnmente para evaluar la calidad del software.

Las métricas que propone son preguntas que ponderan numéricamente un determinado atributo del producto de software. Después de obtener los valores para todas las métricas de un criterio específico, el promedio de todas ellas es el valor para ese factor.

Tabla 3.6
Tabla de evaluación Operación del Producto

Perspectiva	Factor o métrica	Criterio	Puntuación				
			1	2	3	4	5
OPERACION DEL PRODUCTO	Facilidad de uso	Facilidad de operación				x	
		Facilidad de comunicación					x
		Facilidad de aprendizaje				x	
		Formación					x
	Integridad	Control de accesos					x
		Facilidad de auditoría				x	
		Seguridad					x
	Corrección	Complejidad					x
		Consistencia				x	
		Trazabilidad o rastreabilidad				x	
	Fiabilidad	Precisión					x
		Tolerancia a fallos				x	
		Modularidad					x
		Simplicidad				x	
	Eficiencia	Exactitud				x	
		Eficiencia en ejecución					x
		Eficiencia en almacenamiento					x

Fuente: Elaboración propia

Para la ponderación porcentual de la perspectiva de la operación del producto se realizará la suma del valor de cada factor, y el promedio de los factores el cual definirá el valor.

Donde:

Operación del producto=OP; Factor=(F); Criterio=C; Cantidad de factores=n

$$F = \sum C / \text{número de C} \quad OP = (F_1 + \dots + F_n) / n$$

Remplazando los valores tenemos

$$OP = (0.86 + 0.93 + 0.86 + 0.88 + 0.93) / 5$$

$$OP = 0.89 \%$$

Sabiendo el resultado del valor de la perspectiva podemos interpretar que la operación del producto es de un valor del 86 %. El cual indica una buena operatividad del producto.

Tabla 3.7
Tabla de evaluación Operación del Producto

Perspectiva	Factor	Criterio	Puntuación				
			1	2	3	4	5
REVISIÓN DEL PRODUCTO	<i>Facilidad de mantenimiento</i>	Concisión				x	
		Auto descripción				x	
	<i>Facilidad de prueba</i>	Modularidad					x
		Simplicidad					x
		Auto descripción.				x	
		Instrumentación				x	

Fuente: Elaboración propia

Para la ponderación porcentual de la perspectiva de la revisión del producto se realizará la suma del valor de cada factor, y el promedio de los factores el cual definirá el valor.

Donde:

Revisión del producto=RP; Factor=(F); Criterio=C; Cantidad de factores=n

$$F = \sum C / \text{número de C} \quad RP = (F_1 + \dots + F_n) / n$$

Remplazando los valores tenemos

$$RP = (0.80 + 0.90) / 2$$

$$RP = 0.85 \%$$

Sabiendo el resultado del valor de la perspectiva podemos interpretar que la revisión del producto es de un valor del 85 %. El cual indica una buena revisión del producto.

Tabla 3.8
Tabla de evaluación Operación del Producto

Perspectiva	Factor	Criterio	Puntuación				
			1	2	3	4	5
TRANSICIÓN DEL PRODUCTO	<i>Reusabilidad</i>	Auto descripción				x	
		Modularidad					x
		Independencia entre sistema y software					x
	<i>Interoperabilidad</i>	Modularidad					x
		Compatibilidad de comunicaciones					x
		Compatibilidad de datos					x
		Estandarización en los datos				x	
	<i>Portabilidad</i>	Auto descripción.				x	
		Modularidad					x
		Independencia entre sistema y software					x
		Independencia del hardware					x

Fuente: Elaboración propia.

Para la ponderación porcentual de la perspectiva de la transición del producto se realizará la suma del valor de cada factor, y el promedio de los factores el cual definirá el valor.

Donde:

Revisión del producto=TP; Factor=(F); Criterio=C; Cantidad de factores=n

$$F = \sum C / \text{número de C} \quad TP = (F_1 + \dots + F_n) / n$$

Remplazando los valores tenemos

$$TP = (0.93 + 0.95 + 0.95) / 3$$

$$TP = 0.94 \%$$

Sabiendo el resultado del valor de la perspectiva podemos interpretar que la transición del producto es de un valor del 94 %. El cual indica una buena transición del producto.

Evaluaremos el sistema realizando un promedio total de las perspectivas de las cuales ya se tiene el resultado.

Dónde: OP: Operación de producto, RP: Revisión del producto

TP: Transición del producto

$$\text{Calidad} = OP + RP + TP / 3$$

Calidad=0.86+0.85+0.94/3 => **Calidad= 0.88 %**

3.8 ANÁLISIS DE COSTOS

En este punto se aplicarán métricas para calcular el esfuerzo invertido, con el modelo COCOMO II Constructive Cost Model (modelo de costos constructivo).

La planificación o estimación de costos es muy importante en todo proyecto, no solo de los requerimientos de hardware sino también de costos de tiempo y esfuerzo; COCOMO II, es un método de estimación de costos y esfuerzo de únicamente proyectos de software, que permite la estimación por medio de los módulos planificados en el software.

Actualmente existen varias herramientas que ayudan a obtener el análisis de la metodología COCOMO II.

En este proyecto, se utilizó la aplicación de la herramienta web COCOMO II la cual facilita el trabajo del cálculo, mismos que se describen en el capítulo II sección 2.6

The screenshot shows the COCOMO II - Constructive Cost Model web application. The interface includes several sections for inputting project data:

- Software Size:** Sizing Method (Function Points), Unadjusted Function Points (input field), Language (Source Lines of Code), and Language (Function Points).
- Software Scale Drivers:** Precedentedness, Development Flexibility, Architecture / Risk Resolution, Team Cohesion, Process Maturity, and Team Cohesion, all with dropdown menus.
- Software Cost Drivers:** Product (Required Software Reliability, Data Base Size, Product Complexity, Developed for Reusability, Documentation Match to Lifecycle Needs), Personnel (Analyst Capability, Programmer Capability, Personnel Continuity, Application Experience, Platform Experience, Language and Toolset Experience), Platform (Time Constraint, Storage Constraint, Platform Volatility), and Project (Use of Software Tools, Multisite Development, Required Development Schedule), all with dropdown menus.
- Maintenance:** Off (dropdown).
- Software Labor Rates:** Cost per Person-Month (Dollars) (input field).

Buttons for 'Calculate' and 'Auto Calculate' are visible at the bottom. The browser address bar shows 'softwarecost.org/tools/COCOMO/'.

Figura 3.52 Aplicación web COCOMO II
Fuente: <http://softwarecost.org/tools/COCOMO/>

El cálculo se base a los Puntos Fusión, explicado previamente en el capítulo II sección 2.6.6

COCOMO II - Modelo de costo constructivo

Método de dimensionamiento del tamaño del software: Puntos de funciones

Puntos de función no ajustados: 25 Idioma: Java

Controladores de escala de software

Precedencia: Alto Arquitectura / Resolución de riesgos: Alto Madurez del proceso: Nominal

Flexibilidad de desarrollo: Alto Cohesión del equipo: Alto

Controladores de costos de software

Producto

Fiabilidad de software requerida: Nominal Capacidad de analista: Alto Limitación de tiempo: Nominal

Tamaño de base de datos: Nominal Capacidad de programador: Alto Restricción de almacenamiento: Nominal

Complejidad del producto: Alto Continuidad de personal: Alto Volatilidad de plataforma: Nominal

Desarrollado para la reutilización: Alto Experiencia de aplicación: Alto

Documentación que coincide con las necesidades del ciclo de vida: Nominal Experiencia de plataforma: Nominal

Plataforma

Experiencia de lenguaje y conjunto de herramientas: Alto

Proyecto

Uso de herramientas de software: Alto

Desarrollo multitipo: Alto

Programa de desarrollo requerido: Bajo

Mantenimiento: Apagado

Costo de las tarifas laborales del software por persona-mes (dólares): 300

Calcular

Figura 3.53 Interfaz COCOMO II web- Selección de valores y ponderacion respectiva.

Fuente: <http://softwarecost.org/tools/COCOMO/>

Realiza la evaluación en base a selección de valores en cada valor, la descripción de los valores de cada criterio de evaluación se describe en el (cap II en la sección 2.6.5).

Datos introducidos.

Controladores de escala de software

Tabla 3.9 Ponderación de valores en escala de software

Campo	Descripción	Valor
PREC	Precedencia	2.48
FLEX	Flexibilidad en el Desarrollo	1.01
RESL	Arquitectura y Determinación del Riesgo	1.41
TEAM	Cohesión del Equipo	1.10
PMAT	Madurez del Proceso	1.56

Fuente: Propia

Producto a desarrollar.

Tabla 3.10
Ponderación de valores en producto a desarrollar

Campo	Descripcion	Valor
RELY	Confiabilidad requerida	1.10
DATA	Tamaño de la base de datos	1.00
CPLX	Complejidad del producto	1.17
RUSE	Requerimientos de reusabilidad	1.07
DOCU	Documentación acorde a las diferentes etapas del ciclo de vida	1.00

Fuente:Propia

Plataforma

Tabla 3.11
Ponderación de valores de plataforma

Campo	Descripcion	Valor
TIME	Limitaciones en el tiempo de ejecución	1.11
STOR	Limitaciones en el almacenamiento principal.	1.00
PVOL	Volatilidad de la plataforma.	0.87

Fuente:Propia

Personal

Tabla 3.12
Ponderación de valores en evaluacion de personal

Campo	Descripcion	Valor
ACAP	Capacidad de los analistas	0.85
PCAP	Capacidad del programador.	0.88
PCON	Continuidad del personal.	0.90
AEXP	Experiencia en las aplicaciones	0.88
PEXP	Experiencia en la plataforma	1.00
LTEX	Experiencia con el lenguaje y las herramientas	0.91

Fuente:Propia

Proyecto

Tabla 3.13
Ponderación de valores en proyecto

Campo	Descripcion	Valor
TOOL	Uso de herramientas de software.	0.78
SITE	Desarrollo en varios sitios	1.43
SCED	Cronograma requerido para el desarrollo.	1.00

Fuente:Propia

Costo de las **tarifas laborales del software** por persona-mes (dólares)=**300**

Resultados

Los resultados generados a partir de la la informacin ingresada a la herramienta web COCOMO II son:

Results

Software Development (Elaboration and Construction)

Effort = 2.5 Person-months
Schedule = 4.1 Months
Cost = \$763

Total Equivalent Size = 1325 SLOC

Figura 3.53 COCOMO II Web resultados
Fuente: Propia

Acquisition Phase Distribution

Phase	Effort (Person-months)	Schedule (Months)	Average Staff	Cost (Dollars)
Inception	0.2	0.5	0.3	\$46
Elaboration	0.6	1.5	0.4	\$183
Construction	1.9	2.5	0.8	\$580
Transition	0.3	0.5	0.6	\$92

Figura 3.54 Distribución de esfuerzo de software (persona-meses)

Fuente: <http://softwarecost.org/tools/COCOMO/>

Donde se describe que el desarrollo de software (Elaboración y construcción).

El esfuerzo= 2 personas/mes

Tiempo= 4.1 meses

Costo dólares = 763/mes - Costo Bolivianos=5,264.7/ mes

Tamaño equivalente total = 1325 SLOC (promedio de líneas de código por punto función)

El valor total de desarrollo y construcción seria de:

Costo Total= 5,264.7x 4.1

Costo Total=21,585.27 bolivianos - Costo en dólares= 3,128.3

3.9 SEGURIDAD DEL SISTEMA Y BASE DE DATOS

Las principales herramientas para la seguridad del sistema son: la Autenticación, control de acceso cifrado de contraseñas

Autenticación

Para la autenticación de usuario el sistema realiza la búsqueda en base de datos el cual si encuentra coincidencias pues le dará acceso al sistema caso contrario se restringirá el acceso al sistema.

Control de acceso

Luego de la autenticación viene el control de acceso, que se asegura de que solo las personas adecuadas o autorizadas puedan ver, modificar, agregar y/o eliminar la información.

Cifrado

Para el control de acceso de sistema en cuanto a las contraseñas de cada usuario. Las mismas son encriptadas para luego ser guardadas en la base de datos

Accesibilidad.

El sistema siempre está disponible ya que su alojamiento estará en servidores el cual funciona las 24 hrs con la aplicación de certificados SSL de encriptación de la información de extremo a extremo.

Base de datos

La base de datos al igual que el sistema cuenta con restricciones de acceso de usuario el cual es configurado para comunicación con el código fuente del sistema.

El acceso de la base de datos solo está autorizado para el administrador que tiene los privilegios de acceso a registrar modificar eliminar y búsquedas de información.

CAPITULO IV CONCLUSIONES Y RECOMENDACIONES

4.1 CONCLUSIONES

Las metodologías de análisis y diseño de sistemas, pueden ser mejor aprovechadas cuando son utilizadas en conjuntos, ya que cada una cuenta con herramientas que permiten identificar requerimientos, analizar problemas y diseñar la solución, de acuerdo a las preferencias y comodidad de cada desarrollador.

La utilización de software que facilite la programación de aplicaciones web, da la posibilidad de crear una interfaz que permite al usuario la facilidad de navegación, también facilitar la tarea del desarrollador.

El agente inteligente tiene el potencial para automatizar una variedad muy grande de actividades de la vida diaria, además de otras más específicas que permitirán a las personas usar más el tiempo en analizar los resultados y tomar decisiones basados en estos, que en hacer la investigación y recopilación de datos.

Con la implementación del sistema web para la gestión de almacenes y distribución se logró cumplir con los objetivos planteados por lo tanto se concluye lo siguiente:

1. Se logró automatizar los procesos de movimiento (Ingreso y salidas) de los productos en almacenes, brindando un control detallado, reduciendo perdida de productos, además brindando información actual, verídica, pudiendo tomar decisiones en tiempos oportunos, es decir que esté disponible la información cuando se la requiera.
2. Se aplicó un método de almacenamiento en el sistema web para que a partir de esta información pueda mejorar la accesibilidad, control y ubicación de cada producto dentro de los almacenes.
3. Se implementó el módulo de clientes lo cual facilitara el acceso al cliente brindando no solo información básica del cliente, si no también mostrando la ubicación geográfica en mapas de google maps, reduciendo tiempo en búsquedas de domicilio del cliente.
4. Se implementó el módulo de planificación el cual gestiona todos los pedidos para planificar la ruta de distribución de manera óptima de acuerdo a la fecha de entrega del cliente.

El modulo visualiza la ruta planificada de recorrido mostrando en mapas de google maps, reordena los puntos de parada (clientes) de acuerdo a la distancia que existe entre cada cliente.

5. Se implementó el módulo de distribución el cual contiene la información de cada pedido programado para su distribución, además el modulo cuenta con la visualización de la ruta planificada con la ubicación de los clientes y además con las instrucciones de conducción, reduciendo los tiempos de distribución, además permite controlar las entregas realizadas.

6. Con la optimización de registros en cuanto movimientos se logró facilitar el control de stock de productos disponibles, reduciendo perdidas, mermas etc. Además, permite que se puedan detectar errores de manera temprana.

7. Cada módulo implementado en el sistema genera sus reportes respectivos con datos actualizados, disponibles para su uso en cualquier momento, así facilitando la toma de decisiones de manera oportuna.

Entonces resumiendo todos los puntos mencionado el sistema web cumple con los objetivos planteados.

4.2 RECOMENDACIONES

A partir del presente trabajo se propone las siguientes recomendaciones, con el fin de buscar el mejoramiento del sistema:

- Cuando se requiera la ampliación y creación de nuevos módulos, se recomienda primero revisar la documentación para poder tomar una buena decisión, ya que el sistema presenta elementos reutilizables que podrían ser utilizados en los módulos nuevos.
- Se recomienda a la empresa, el cambio periódico de sus contraseñas para la seguridad de sus cuentas; sobre todo, verificar la fortaleza de la contraseña.
- La revisión periódica por cierto periodo de tiempo es recomendable para eficiencia y un funcionamiento adecuado del sistema.
- Se recomienda a la empresa, implementar, utilizar y administrar el sistema de acuerdo las instrucciones brindadas.

- Se recomienda realizar un sistema de facturación y control de ventas, ya que el sistema posee las características de interoperabilidad.
- Se recomienda el uso de tecnologías con radio frecuencia para inventarios las cuales puedan interactuar con el sistema.

BIBLIOGRAFIA

Martin Morales, 2004. Sistema de gestión de almacenes

Recuperado de: <http://www.grupoamsa.com/filamentos/index.html>

Camacho Lugones Mashiel M. 2017. Control de producción, pedidos, entregas e inventario caso: PHOVIDA. Universidad Mayor de San Andrés.

Gomez Jimenez Ginnola T. 2018 Sistema Web de control de inventarios, ventas de productos y mantenimiento de maquinaria basados en la arquitectura SCM. Caso:Cofigril Ltda. Universidad Mayor de San Andrés.

Wikipedia enciclopedia libre, 2019. KANBAN (DESARROLLO).

Recuperado de: <https://www.cursosgis.com/comparativa-de-los-principales-sistemas-gestores-de-bases-de-datos-sqbd/>

Pro Optim, 2019. Implementar kanban en la empresa.

Recuperado de: <https://blog.pro-optim.com/noticias/como-implementar-kanban-en-la-empresa-fases-y-reglas/>

Webempresas.com 2019. Gestion y control de inventarios.

Recuperado de:<https://www.webyempresas.com/gestion-y-control-de-inventarios-conceptos-y-metodos/>

Academia.edu, 2019. Cuadro comparativo entre metodologías de desarrollo de aplicaciones web

Recuperado de:

https://www.academia.edu/10851613/CUADRO_COMPARATIVO_ENTRE_METODOLÓGICAS_DE_DESARROLLO_DE_APLICACIONES_WEB

Cursosgis.com 2019. Comparativa de los principales sistemas de bases de datos.

Recuperado de:

<https://www.cursosgis.com/comparativa-de-los-principales-sistemas-gestores-de-bases-de-datos-sgbd/>

Jimdo. 2016 Modelo de McCall

Recuperado de: <http://vanevargas.jimdo.com/m%C3%B3dulos/modelos/modelo-de-mccall/>

If.deiby mayo (2012).Modelo McCall, ISO 9126, ISO2500.

Recuperado de: <http://mccallisos.blogspot.com/>

Wikipedia 2019 Cocomo

Recuperado de: <https://es.wikipedia.org/wiki/COCOMO>

ECURED, 2019. COCOMO II

Recuperado de: https://www.ecured.cu/COCOMO_II

Uniciencia vol. 32. 2018 Estimación de costo de software.

Recuperado de :

<https://pdfs.semanticscholar.org/e66c/f16c8607984bb4503f1ec78e73c0da9db5fe.pdf>

Maestros del Web by platzi, 2019. Los diferentes lenguajes de programación.

Recuperado de : <http://www.maestrosdelweb.com/los-diferentes-lenguajes-de-programacion-para-la-web/>

Monografias.com, 2019 Lenguajes de programación Web.

Recuperado de: <https://www.monografias.com/trabajos105/lenguajes-programacion-web/lenguajes-programacion-web.shtml>

DIARLU, 2019. Bases de datos más usados por desarrolladores Web.

Recuperado de: <https://www.diarlu.com/gestores-bases-datos/>

cr728.wordpress 2017. Manejadores de bases de datos.

Recuperado de: <https://cr728.wordpress.com/2017/11/13/ventajas-y-desventajas>

arcgis.com. Introducción a SIG

Recuperado de: <https://resources.arcgis.com/>

uwe.pst.ifi.lmu.de UWE- Ingeniería web basada en UML

Recuperado de : <https://uwe.pst.ifi.lmu.de/teachingTutorialRequirements.html>

Carolina 2013. Aplicación Metodología UWE

Recuperado de:

http://evangellyscarolinacabellorodriguez.blogspot.com/2013/10/metodologia-uwe_29.html

StatTec. 2000. Programación en N capas

Recuperado de: <https://starttecperu.com/tutoriales/visualstudio/programacion-en-n-capas>

Academia 2016. Arquitectura de N capas

Recuperado de: https://www.academia.edu/10102692/Arquitectura_de_n_capas

Web y empresas 2020 método de inventarios ABC y sus beneficios

Recuperado de: <https://www.webyempresas.com/metodo-abc-de-inventarios/>

Softdoit. Organización de almacenes con el método ABC

Recuperado de:

<https://www.softwaredoit.es/software-gestion-almacen-guias/metodo-abc-de-inventarios.html>

EcuRed. Arquitectura en capas

Recuperado de: https://www.ecured.cu/Arquitectura_en_Capas

Jacobson, Ivar, Grady Booch, and James Rumbaugh. El Proceso Unificado de Desarrollo de Software. México: Addison-Wesley, 1999.

Maria Eugenia Arevalo Lizaro 2010. Introducción al patrón de arquitectura por capas

Recuperado de: <https://arevalomaria.wordpress.com/2010/12/02/introduccion-al-patron-de-arquitectura-por-capas/>

Dolado, J. J., & Fernández, L. (2000). Medición para la gestión en la Ingeniería del Software.

McCall, J., Richards, P., & Walters, G. (1977). Factors in Software Quality. Volumes I, II and III: RADC Reports.

Jhon Perez y Wilson Martinez 2011. Modelo de calidad de software

Recuperado de:

<http://www.slideshare.net/rolmary/modelo-de-calidaddelsoftware1>

Wikipedia 2020 ISO/IEC 9126

Recuperado de: http://es.wikipedia.org/wiki/ISO/IEC_9126

COCOMO. Gómez, A. López, M. Migani, S. et al. (s.f.).

Recuperado de: <https://blogadmi1.files.wordpress.com/2010/11/cocomo01full.pdf>

El Modelo COCOMO.

Recuperado de: <http://csse.usc.edu/TECHRPTS/1984/usccse84-500/usccse84-500s.pdf>.

[Boehm 1995/1] Boehm B.W., Clark B., Horowitz E., Westland C., Madachy R., Selby R., Cost Models for Future Software Life Cycle Processes: COCOMO II , Annals of Software Engineering Special Volume on Software Process and Product

Measurement, 1995, Vol 1, pp. 45-60. Primer paper que trata COCOMO II.
Recuperado de: <http://sunset.usc.edu/COCOMOII/cocomo.html>

[COCOMO II.0] Documentos de la ayuda del software COCOMO II.0.

[Boehm 1995/2] Boehm B.W., Clark B., Horowitz E., Westland C., Madachy R., Selby R., The COCOMO 2.0 Software Cost Estimation Model.

Recuperado de: <http://sunset.usc.edu/COCOMOII/cocomo.html>

Cocomo - Un modelo de estimación de proyectos de software Adriana Gómez,
María del C. López, Silvina Migani, Alejandra Otazú 2010

Bosque Sendra, J.; 2000: Sistemas de Información Geográfica

ANEXOS

Análisis y aplicación de la metodología ABC en inventarios.

Realizando el Análisis ABC

No hay un criterio para determinar qué productos entran en qué categoría. Los criterios de categoría deben definirse específicamente para cada empresa ya que el objetivo es poder interpretar esos datos de una manera que sea significativa para la empresa, y pueda verse reflejado en el sistema web.

Unos ejemplos de la categorización más comunes para estas categorías son:

- **A:** 20% de los productos que representan el 70% de consumo anual
- **B:** 30% de los productos que representan el 25% de consumo anual
- **C:** 50% de los productos que representan el 5% de consumo anual

Una forma útil de dividir los productos en categorías:

- Multiplica la cantidad (anual o mensual) de artículos por el costo por artículo para calcular el valor de uso anual/ mensual para cada producto.
- Haz una lista de cada producto en orden descendente según el valor de uso del producto.
- Suma el valor de uso y la cantidad de elementos.
- Traduce cada elemento y su valor de uso en la fracción o porcentaje del total acumulado.
- Crea un gráfico para conectar la cantidad de elementos y el valor de uso. Divide el cuadro en A, B y C.

La tabla refleja el análisis y su categorización.

Nº	Producto	Cantidad vendida / mes	Costo unitario o bs	Total ventas	%participación	Participación acumulada	Clasificación
1	Monello cat 15 kg	950	225	213750	22%	22%	A
5	Monello dog R/P	600	350	210000	21%	43%	A
2	Monello dog tradicional	400	300	120000	12%	55%	B
4	Monelloo dog Filothes	300	350	105000	11%	66%	B
9	Ban cat 25 k	465	220	102300	10%	77%	B

3	Monello cat filothés	500	180	90000	9%	86%	C
6	Birbo Cat 20 kg	300	225	67500	7%	93%	C
7	Birbo dog 25 kg	200	225	45000	5%	97%	C
8	Bantid 25 kg	148	190	28120	3%	100%	C

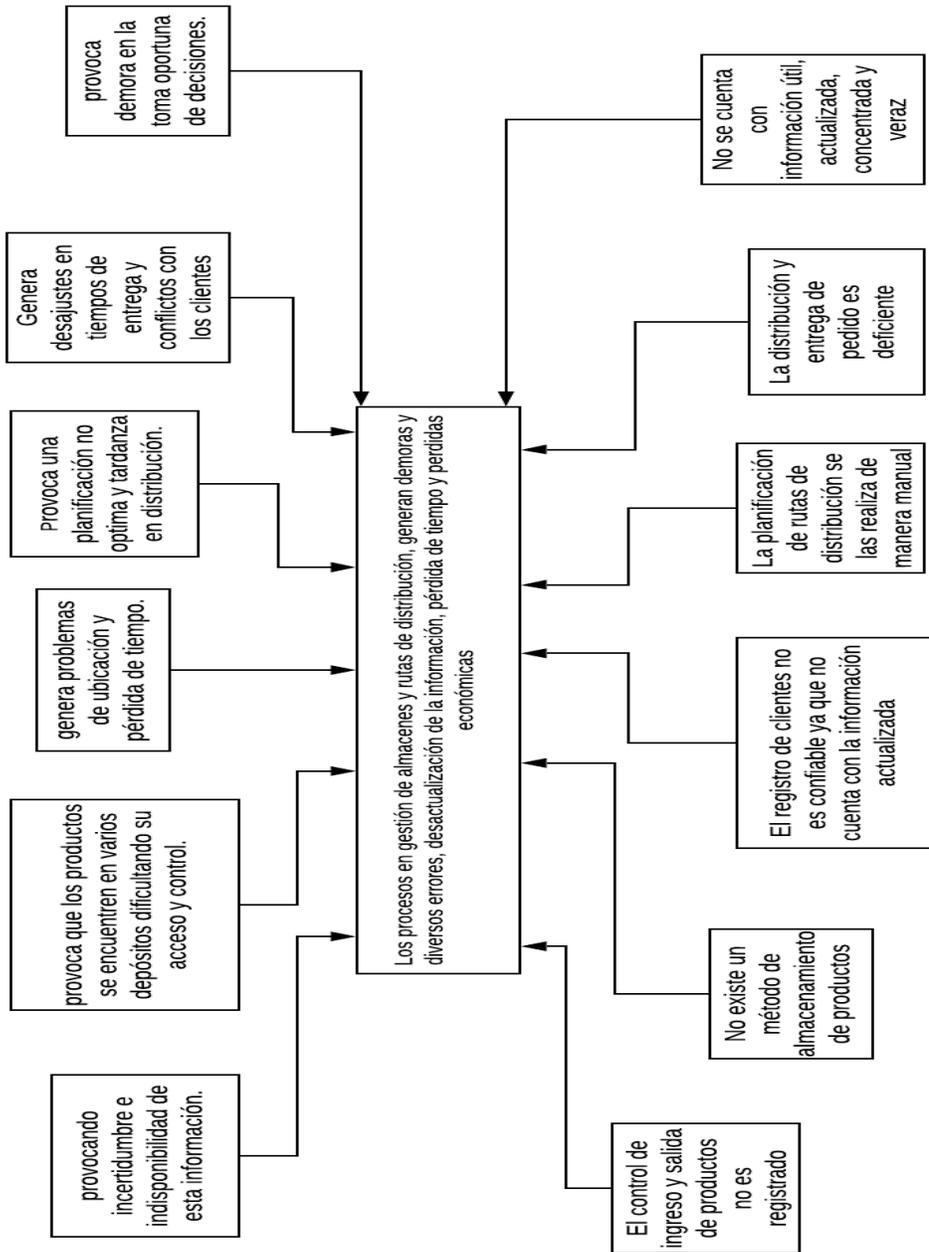
Hay que tener presente cómo se controla los productos en cada categoría. Verificar los costos asociados con mantener estos productos en inventario.

Si actualmente se comprara cantidades uniformes, probablemente se esté haciendo un pedido excesivo o insuficiente de la gran mayoría de productos. Esto significa que los costes de almacenamiento, entrega y administración son más altos de lo necesario.

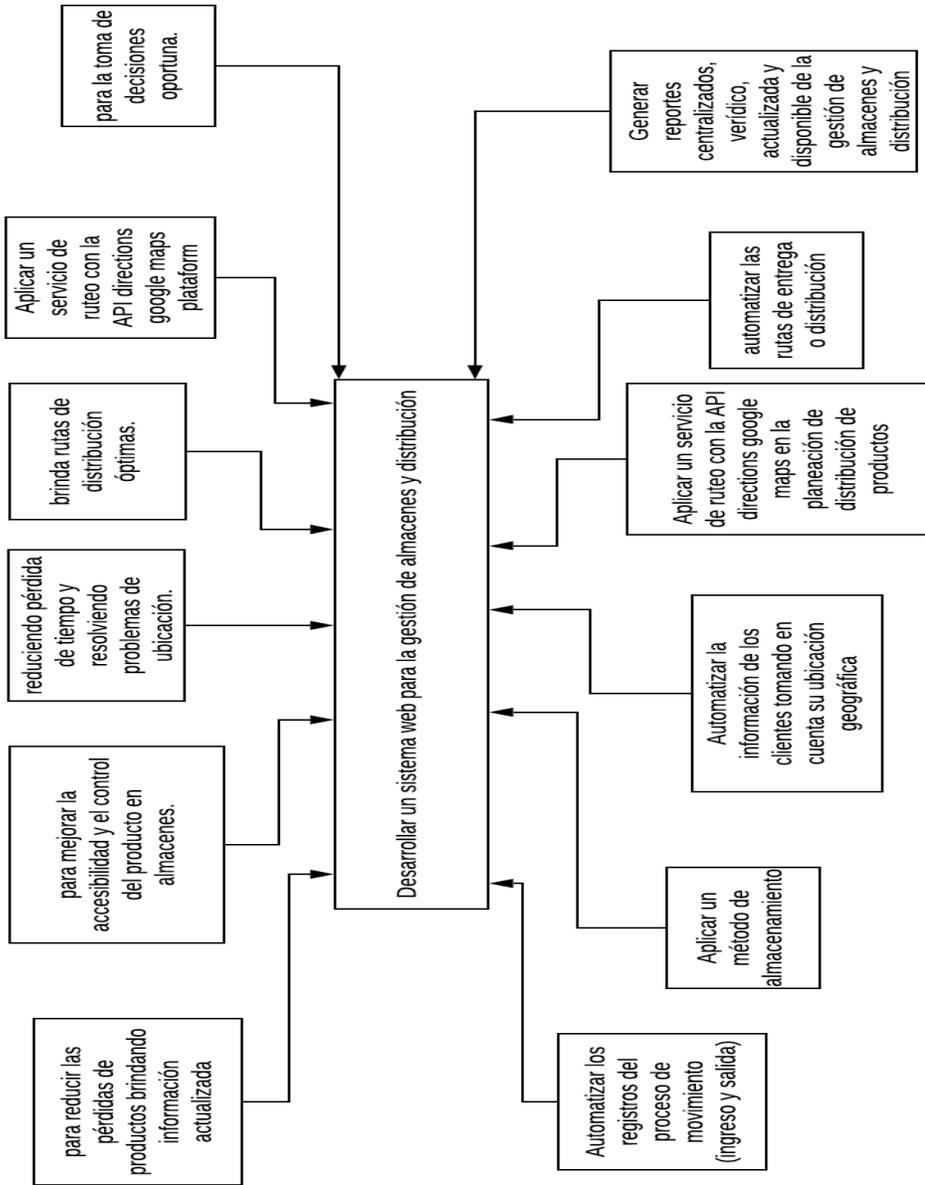
Entonces sabiendo los resultados del análisis de método ABC se puede cambiar el tipo de almacenamiento, es decir reubicar los lugares de acuerdo a la categoría asignada a cada producto.

Ahora que ya se sabe que en que producto vale la pena más invertir, también emplear más horas de trabajo porque estos son los productos que valen más la pena.

ARBOL DE PROBLEMAS



ARBOL DE OBJETIVOS



DOCUMENTOS USADOS POR LA EMPRESA ADIPHARM S.R.L.

HOJA DE RUTA

FECHA	M CAT FILOTHES 1 KG	M CAT 15 KG	M CACHORRO 25KG	M TRADICIONAL 25 KG	M R/P 25 KG	BIRBO 20 KG	BANCAT 25KG	BANDIT 25 KG	BIRBO CAT 25 KG	BIRBO CAT 15KG	BIRBO DOG 15KG
25/03/2020											
3) Vet. Debezonan / Calle J. V. /		20	5	3	2						
4) Vet. S.O.S. / Hda. Julia	20	5	2	1							
5) Dra. Roxana / Hda. Chirera			5			4		2			
6) Vet. G. / Rio Saca Pl.		2	2	2	2		4			5	
7) Complejo Vet. Gue.											
8) Del Sur / Caja		7	2	1							
9) Vet. Copacabana / Esp.	40	3			4	4					2
10) Rosam / Uruch. Villa Alta		10	4	2	10						
11) Huanchoyama / Cruzal				3		5					
total	80	42	20	12	18	14	4	2	10	5	2

Hola de planificación de ruta para el distribuidor

14/05/20

Clusquis

- 30 Meat
- 5 Tradicional
- 3 Bancat
- 10 Meat Fildres ✓
- Vinos

La Granja Andina

- 100 Meat ✓
- 50 Cachorros
- 30 Tradicionales
- Zones

S.O.S.

- 10 Babudos
- 3 Cachorros
- 2 R/P Lanes
- Ganegro
- 5 Pomas
- 5 Meat Lane
- 10 A. Castados

Recepción de pedidos del cliente

IMPORTADORA DE PRODUCTOS VETERINARIOS
LA PAZ - BOLIVIA

Fone: 79518035 - 79618740 - 79548863

ORDEN DE PEDIDO Nº 02398

Cliete: *F. L.* Tel: DIA: *27* MES: *9* AÑO: *19*

Dirección: FECHA DE VENCIMIENTO:

FORMA DE PAGO: CREDITO CONTADO

CANT.	DESCRIPCION		PRECIO UNIT.	TOTAL
2	MONELLO BATATUN SALMON Y POLLO	15 Kg.	223	669
	MONELLO CAT ATUN SALMON Y POLLO	1 Kg.		
	MONELLO CAT FLOTHER	1 Kg.		
	MONELLO DOG FLOTHER	25 Kg.		
	MONELLO DOG FLOTHER	1 Kg.		
	MONELLO DOG TRADICIONAL	25 Kg.		
	MONELLO DOG TRADICIONAL	1 Kg.		
	MONELLO DOG RP	25 Kg.		
	MONELLO DOG RP	1 Kg.		
	BIRBO CAT	25 Kg.		
	BAVOD DOG	25 Kg.		

TOTAL: *669*

OBSERVACIONES:

Recibi Conforme *Entregue Conforme*

Una vez retirado el producto, no se aceptan devoluciones

Hoja de entrega de pedido

MANUAL DE USUARIO
SISTEMA DE GESTION DE ALMACENES Y DISTRIBUCION
ADIPHARM S.R.L.

Ítem	Detalle
Título del Documento	Manual de Usuario –SISTEMA DE GESTION DE ALMACENES Y DISTRIBUCION (ADIPHARM S.R.L.)
Nombre Archivo Físico	MU – Version 1_0.docx
Propiedad	ADIPHARM S.R.L
Fecha Creación	13/07/2020
Última Modificación	

Contenido

INTRODUCCIÓN	133
OBJETIVO DEL SISTEMA	133
GUIA DE USO	133
Requisitos del sistema.....	133
Ingreso al sistema.....	133
MODULOS DEL SISTEMA.....	134
MANIPULACION DEL SISTEMA.....	136
Registro de datos:	136
Actualización de datos:	137
Eliminar registro.....	138
Busquedas	138
Reportes	138
Módulo de clientes	139
Módulo de Proveedores	139
Módulo de pedidos.....	140
Módulo de productos	140
Módulo de almacén.....	141
Módulo de movimientos	141
Módulo de planificación.....	141
Módulo de distribución	142
Cerrar session	143

INTRODUCCIÓN

El presente documento pretende servir de guía para el uso adecuado de los diferentes módulos del “Sistema web para la gestión de almacenes y distribución”, considerando como puntos importantes la elaboración de registros, administración de inventarios, planeación de rutas y distribución de productos y las distintas ventajas que este Sistema le ofrece.

Se recomienda que durante la lectura de este documento vaya practicando y contrastando con el aplicativo en sí.

OBJETIVO DEL SISTEMA

El objetivo del sistema, es permitir al personal mejorar los procesos en gestión de almacenes y distribución mejorando la efectividad de su administración.

GUIA DE USO

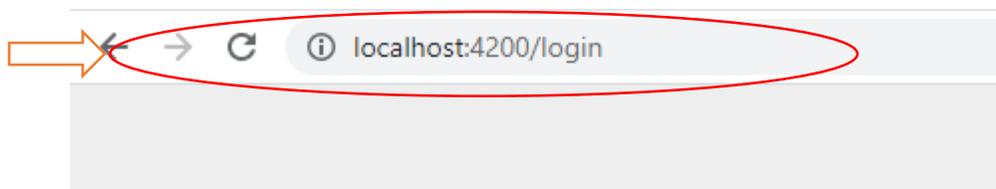
Requisitos del sistema

Para una adecuada funcionalidad del sistema, es necesario contar con las siguientes características:

- Una computadora o equipo móvil configurada con acceso a INTERNET o estar configurada para conectarse con una red local.
- Debe tener instalado Adobe Reader PDF, cualquier versión, para el Reporte
- Debe tener instalado navegadores de internet Mozilla o Google Chrome.

Ingreso al sistema

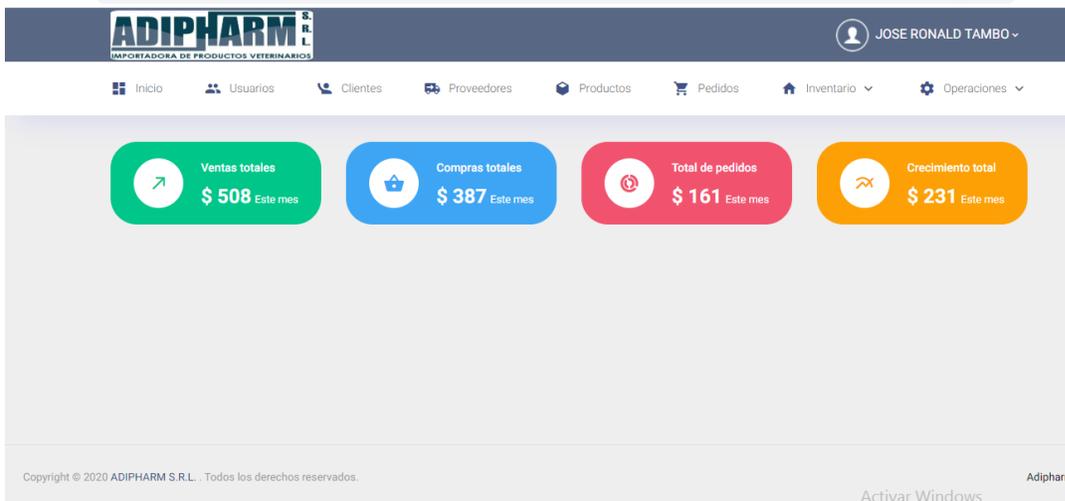
Como primer paso se debe ingresar a la dirección de página, que nos llevara al acceso del sistema. La dirección del sistema puede variar dependiendo del dominio de alojamiento del sistema (localhost).



Para el ingreso al sistema se necesita contar con un NOMBRE DE USUARIO y su CONTRASEÑA DE ACCESO, si no cuenta con esto debe consultar al administrador del sistema para que pueda darle los accesos correspondientes.



Una vez ingresado el nombre de usuario y contraseña pulsar la opción de LOGIN entonces se accederá al sistema mostrando el siguiente interfaz.



MODULOS DEL SISTEMA

El acceso a cada módulo está limitada a los privilegios que es asignado a cada usuario. Para ingresar al módulo clic en la descripción de cada módulo.

El sistema contempla los siguientes módulos:

1 Usuarios

El modulo presenta una interfaz con una lista de todos los usuarios en la que puede gestionar toda la información del usuario como ser: registros actualizaciones, eliminación y reportes.

2 Clientes.

El módulo de clientes muestra dos secciones una en la que se gestiona toda la información del cliente y gestión del cliente como ser: registro, actualización, eliminación y reportes.

En la segunda sección se muestra la ubicación geográfica del cliente que se le asigna al momento del registro.

3 Proveedores

El modulo presenta una interfaz con una lista de todos los Proveedores en la que puede gestionar toda la información del proveedor como ser: registros actualizaciones, eliminación y reportes.

4 Productos.

El modulo presenta una interfaz con una lista de todos los productos en la que puede gestionar toda la información del producto como ser: registros actualizaciones, eliminación y reportes.

5 Pedidos.

El modulo presenta una interfaz con una lista de todos los pedidos en la que puede gestionar toda la información del pedido como ser: registros actualizaciones, eliminación y reportes.

6 Inventarios.

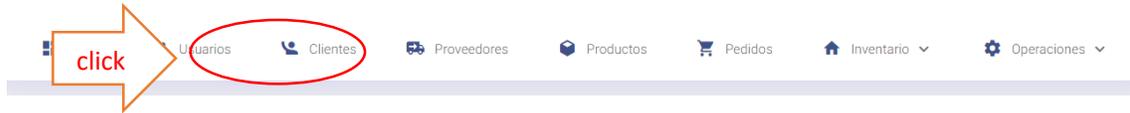
Cuenta con dos sub-módulos:

- Almacén: El submodulo despliega la lista de todos los almacenes, el modulo presenta opciones para gestionar toda la información de registro, actualizaciones, eliminación y reportes
- Movimientos: El submodulo despliega la lista de todos los movimientos, el modulo presenta opciones para gestionar toda la información de registro, actualizaciones, eliminación y reportes

7 Operaciones

- Planificación: El modulo despliega toda la lista de pedidos para poder ser planificados.

- Distribución: El modulo despliega todos los pedidos a distribuir mostrando la suma total de productos, el mapa con la ruta desplegada junto a las instrucciones de conducción.



MANIPULACION DEL SISTEMA

La siguiente instrucción de uso aplica para los módulos de: Usuarios,

Proveedores, Productos, Pedidos, Almacenes y Movimientos.

Cada módulo mencionado varía en los campos de registro, listado de registros, y presentación de reporte.

Registro de datos:

- (1) Click en el boton de nuevo registro, desplegara una ventana modal el cual contiene el formulario de registro.
- (2) Click en el bolton enviar para confirmar el registro.



REGISTRO DE NUEVO USUARIO

Registre cuidadosamente los datos requeridos.
Los campos marcados de color rojo son obligatorios.

<p>Nombres</p> <input style="width: 95%;" type="text" value="Ejemplo: Juan Jose"/> <p>Apellido Paterno</p> <input style="width: 95%;" type="text" value="Ap. paterno"/> <p>Apellido Materno</p> <input style="width: 95%;" type="text" value="Ap. Materno"/> <p>CI</p> <input style="width: 95%;" type="text" value="CI ejemplo:9586575"/> <p>Telf. Contacto</p> <input style="width: 95%;" type="text" value="Ej. 71586950"/> <p>Dirección</p> <input style="width: 95%;" type="text"/>	<p>Fecha Nacimiento</p> <input style="width: 95%;" type="text" value="dd/mm/aaaa"/> <p>Cargo</p> <input style="width: 95%;" type="text"/>
<p>Seleccione Roi de usuario:</p> <input style="width: 95%;" type="text"/>	<p>Salario</p> <input style="width: 95%;" type="text"/>
<p>Ciudad de operación</p> <input style="width: 95%;" type="text"/>	

(3)click en cancelar para anular el registro.

Actualización de datos: Para poder realizar la actualización de registros dar click en el icono  desplegara un formulario con los datos respectivos cargados en los campos.

Para confirmar el proceso dar click en  o en su defecto para descartar los cambios en 

ID	AP. MATERNO	TELF/CEL	DIRECCION	ROL USUARIO	CARGO	
	Arce	588875566	asdasd	VENDEDOR		 
	Mamani	78569832	Zona de los palotes	DISTRIBUIDOR	Distribuidor	 

ACTUALIZA DATOS DE EMPLEADO .

Editar cuidadosamente los datos requeridos.

Si el campo de cambia a color rojo es obligatorio.

Nombres	Fecha Nacimiento
<input type="text" value="Esteban"/>	<input type="text" value="17/06/1999"/>
Apellido paterno	Cargo
<input type="text" value="Catacora"/>	<input type="text" value="Vendedor"/>
Apellido materno	Salario
<input type="text" value="Arce"/>	<input type="text" value="2100"/>
CI	Comision
<input type="text" value="1111111"/>	<input type="text" value="1%"/>
Seleccione estado civil:	Seleccione área de trabajo departamental:
<input type="text" value="Soltero"/>	<input type="text" value="La Paz"/>
Nacionalidad	Ciudad de operación
<input type="text" value="Boliviana"/>	<input type="text" value="El Alto"/>
Telf. Contacto	
<input type="text" value="588875566"/>	
Dirección	
<input type="text" value="asdasd"/>	

Activar Windows

Eliminar registro

Para realizar la eliminación respectiva del registro se debe dar click en el icono  , elimina el registro de la fila seleccionada, mostrara una alerta de seguridad. Para confirmar la eliminación de registro.

IO	AP. MATERNO	TELF/CEL	DIRECCION	ROL USUARIO	CARGO
Arce	588875566	asdasd		VENDEDOR	Vendedor
Mamani	78569832	Zona de los palotes		DISTRIBUIDOR	Distribuidor



Busquedas

Las búsquedas pueden ser realizadas introduciendo el texto en el panel, y mostrada los resultados encontrados en la tabla de registros.

JOSE|

Nº	NOMBRES	AP. PATERNO	AP. MATERNO	TELF/CEL	DIRECCION	ROL USUARIO	CARGO
1	Juan Jose	Mamani	Mamani	78569832	Zona de los palotes	DISTRIBUIDOR	Distribuidor
2	JOSE ALEX	BAPTISTA	QUISPE	65987542	ZONA VILLA ADELA PLAZA SANTIDAD N 258	VENDEDOR	AYUDANTE
3	JOSE RONALD	TAMBO	ARCANI	61216486	ZONA SANTA ROSA CALLE F N 24A	ADMIN	JEFE DE ALMACEN

« Previous **1** Next »

Reportes

Para poder generar los reportes, dar click  mostrada una nueva ventana, con los datos requeridos.



**SISTEMA DE CONTROL
ALMACENES Y DISTRIBUCION**

Fecha:
13/7/2020

REPORTE DE USUARIOS

Lista de usuario activos

Nº	NOMBRES	AP. PATERNO	AP. MATERNO	Telf/Cel	DIRECCION	ROL USUARIO	CARGO
1	Esteban	Catacora	Arce	588875566	asdasd	VENDEDOR	Vendedor
2	Juan Jose	Mamani	Mamani	78569832	Zona de los palotes	DISTRIBUIDOR	Distribuidor
3	aaa	aaa	aaa	123456888	asdasdasd	DISTRIBUIDOR	Distribuidor
4	JOSE ALEX	BAPTISTA	QUISPE	65987542	ZONA VILLA ADELA PLAZA SANTIDAD N 258	VENDEDOR	AYUDANTE
5	JOSE	TAMBO	ARCANI	61216486	ZONA SANTA ROSA	ADMIN	JEFE DE

Activ
Ve a C

Módulo de clientes

El módulo de clientes realiza la gestión de todos los clientes, tales como los (1) registros, (2) actualizaciones, (3) reportes (4) búsquedas y (5) eliminaciones. Cabe mencionar que también realiza el registro de la ubicación de geográfica del cliente.

Módulo de Clientes

Lista de clientes registrados en el sistema

1 NUEVO CLIENTE

Reporte PDF

4

3

2

5

UBICACION GEOGRAFICA DE CLIENTES

RAZON SOCIAL	DEPARTAMENTO	CIUDAD	DIRECCION	TELEFONO/CELUL
Pasta sucia	La Paz	El Alto	Ceja calle 3 N100	775807191
Mundo Animal	La Paz	El Alto	zona alto lima plaza triangular N 1500	76985236
El Gato Pastor	La Paz	La Paz	villa armonia	2147483647
La Granja Andina	La Paz	El Alto	Zona 16 de julio , calle pascoe N 259	26865352

Módulo de Proveedores

El módulo de Proveedores realiza la gestión de todos los proveedores, tales como los (1) registros, (2) actualizaciones, (3) reportes (4) búsquedas y (5) eliminaciones.

Módulo de Proveedores

Lista de proveedores registrados en el sistema

1 NUEVO PROVEEDOR

Reporte PDF

4

3

2

5

Inicio Usuarios Clientes Proveedores Productos Pedidos Inventario Operaciones

Nº	NOMBRE PROVEEDOR	TELEFONO	DIRECCION	PAIS	EMAIL
1	PRO-ANI S.R.L	59226522	Santa cruz, carretera warnes kilometro 5 N585	Bolivia	pro-ani@gmail.co

Módulo de pedidos

El módulo de pedidos realiza la gestión de todos los pedidos, tales como los (1) registros, (2) actualizaciones, (3) reportes (4) búsquedas y (5) eliminaciones.

ORDEN #	FECHA ENTREGA	CLIENTE	CIUDAD	ESTADO ENTREGA	DETALLE ENTREGA
1	2020-07-09	Patita sucia	El Alto	Despachado	Ver detalle entrega
2	2020-07-09	Mundo Animal	El Alto	Planificado	Ver detalle entrega
3	2020-07-08	El Gato Pastor	La Paz	Pendiente	Ver detalle entrega
4	2020-07-08	El Gato Pastor	La Paz	Despachado	Ver detalle entrega
5	2020-07-08	Mundo Animal	El Alto	Despachado	Ver detalle entrega
6	2020-07-09	El Gato Pastor	La Paz	Pendiente	Ver detalle entrega

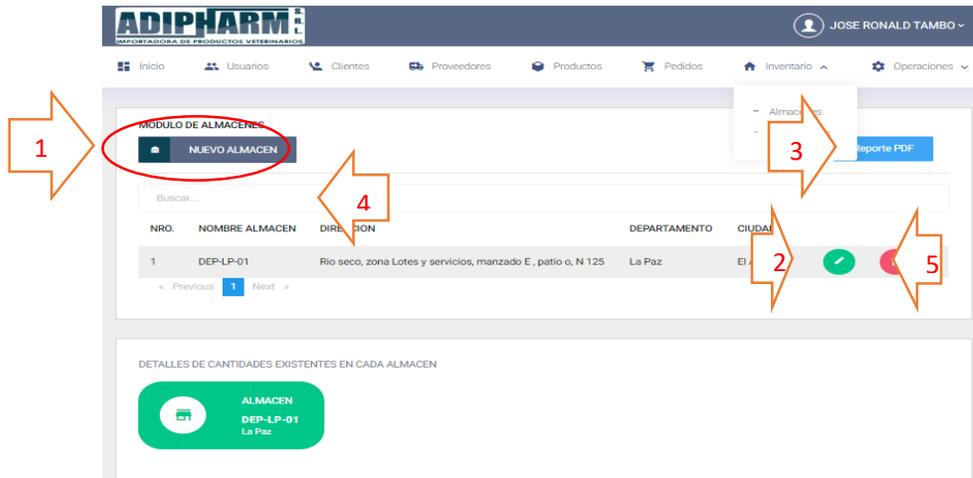
Módulo de productos

El módulo de productos realiza la gestión de todos los productos, tales como los (1) registros, (2) actualizaciones, (3) reportes (4) búsquedas y (5) eliminaciones.

Nº	PRODUCTO	PESO	LINEA PRODUCTO	PRECIO	PROVEEDOR
1	Monello Dog Tradicional	25 kg	Monello premium especial	300	Nutrire S.R.L.
2	Monello cat	15 kg	Monello premium especial	225	Nutrire S.R.L.

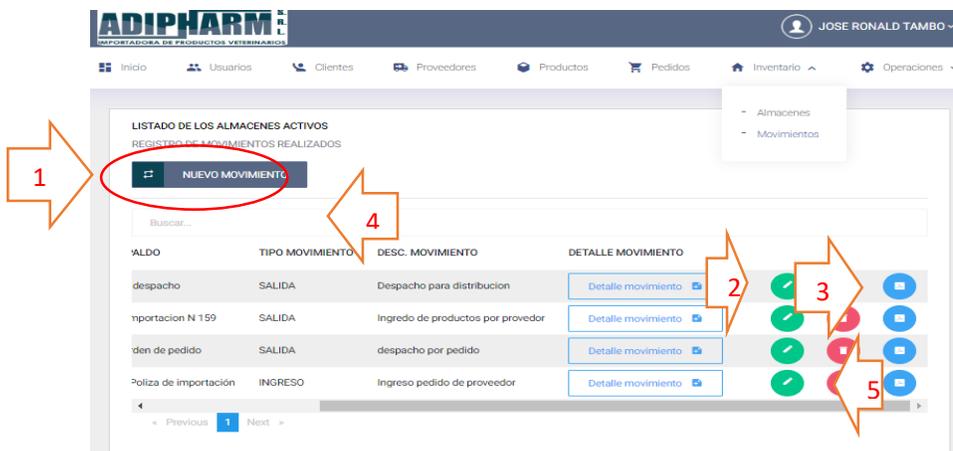
Módulo de almacén

El módulo de almacén realiza la gestión de todos los almacenes, tales como los (1) registros, (2) actualizaciones, (3) reportes (4) búsquedas y (5) eliminaciones.



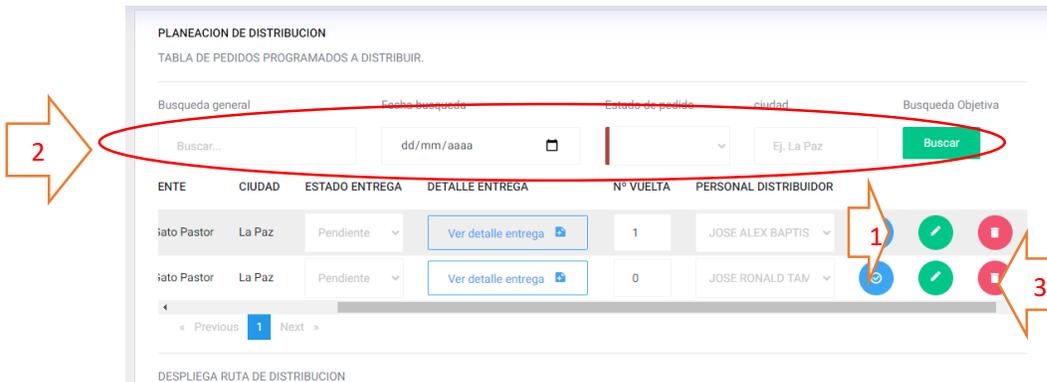
Módulo de movimientos

El módulo de movimientos realiza la gestión de todos los movimientos, tales como los (1) registros, (2) actualizaciones, (3) reportes (4) búsquedas y (5) eliminaciones.



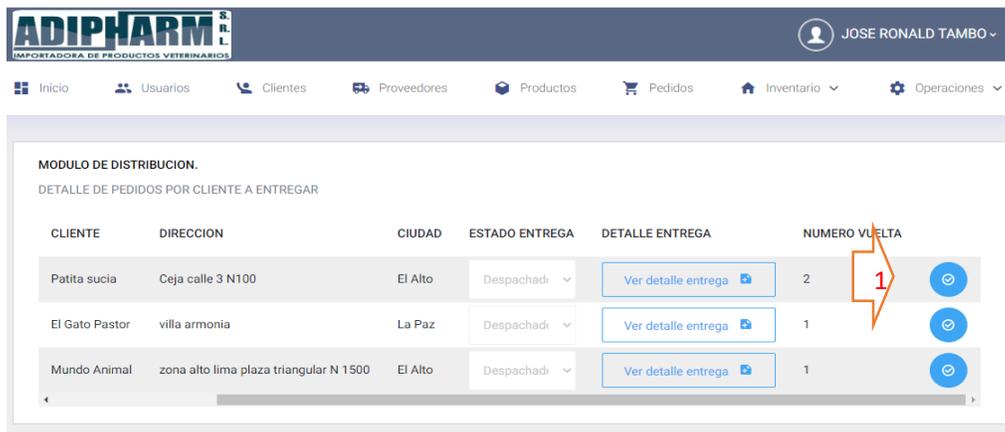
Módulo de planificación

El módulo de planificación realiza la gestión de toda la planificación, tales como los (1) actualizaciones, (2) búsquedas y (3) eliminaciones.



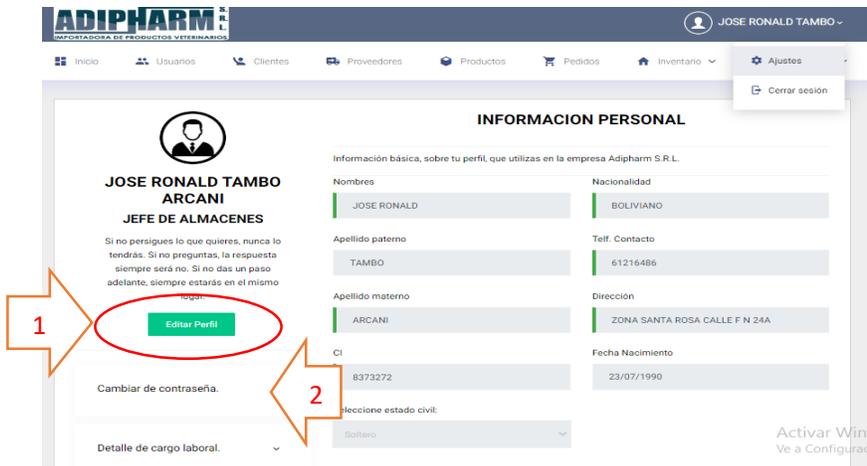
Módulo de distribución

El módulo de planificación realiza la gestión de toda la planificación, tales como los (1) actualizaciones, (2) búsquedas



Módulo de perfil de usuario

El módulo de perfil de usuario realiza la gestión de información del mismo usuario, tales como los (1) actualizaciones, (2) actualización de contraseña



Cerrar session

Click en la opción cerrar session, sale del sistema.



MANUAL TECNICO
SISTEMA DE GESTION DE ALMACENES Y DISTRIBUCION

Ítem	Detalle
Título del Documento	Manual de Tecnico –SISTEMA DE GESTION DE ALMACENES Y DISTRIBUCION (ADIPHARM S.R.L.)
Nombre Archivo Físico	MT – Version 1_0.docx
Ubicación Archivo Físico	
Propiedad	ADIPHARM S.R.L
Fecha Creación	14/07/2020
Última Modificación	

INTRODUCCIÓN

El presente documento pretende servir de guía para el uso adecuado de la instalación del sistema.

El manual contiene información detallada de los pasos y los requisitos mínimos para el uso de futuros desarrolladores puedan realizar testeos pruebas y modificaciones sobre el código fuente.

Se recomienda que durante la lectura de este documento vaya practicando y contrastando con el aplicativo en sí.

REQUERIMIENTOS TÉCNICOS

Requerimientos de Software

- Sistema operativo Windows 7 en adelante.
- Navegador web (Chrome, Mozilla, Opera, etc.) con conexión a internet.
- Software editor de código fuente (Visual Code, ATOM, Sublime, etc).
- Gestor de base de datos (Maria DB), para el almacenamiento y la administración de datos.
- Emulador de servidor (Xampp 7.3.8), (apache) para la administración de la base de datos.
- Lenguaje PHP 7.3.8 en el emulador Xampp para correr código fuente desarrollado con framework Codeingniter 3.1.9
- Entorno de ejecución NodeJS server para la ejecución de código JavaScript del lado del servidor.
- Entorno de desarrollo para fron-end Angular cli 8.
-

Requerimientos mínimos de Hardware

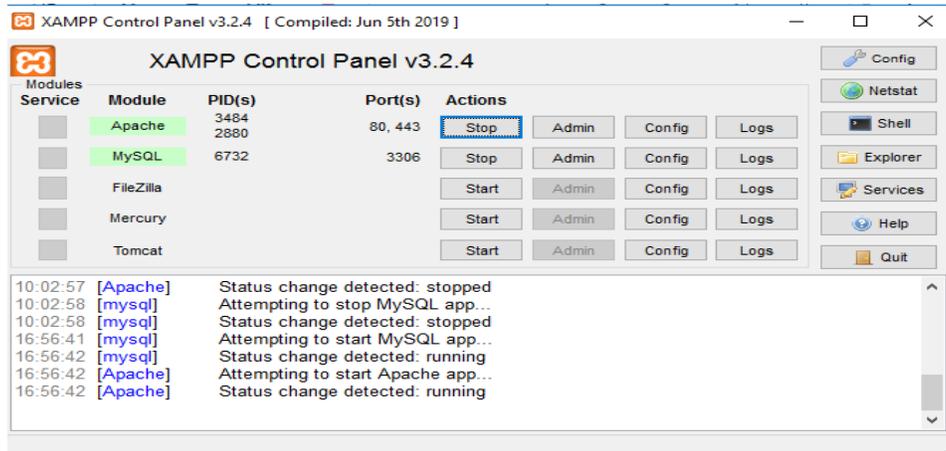
- SO: Windows® 7 o superior.
- Procesador: 1.4 GHz.
- Memoria: 1 Gb de RAM.

Instalación

1) Primeramente realizar la instalación de Xampp 7.3.8 ejecutando su respectivo instalador.

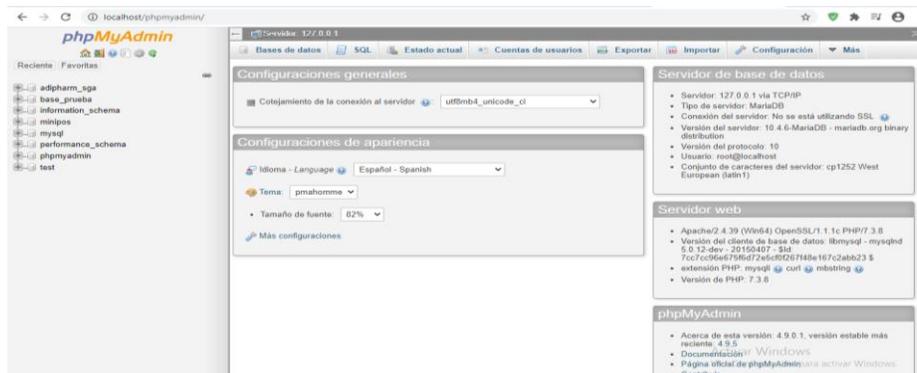
Una vez instalado Xampp tendremos instalado en emulador de servidor apache, el lenguaje de desarrollo PHP y el gestor de base de datos MariaDB.

Una vez instalado el Xampp mostrará una interfaz con las características que se ve en la imagen.



Arrancar los servicios de apache y mysql como se muestra en la imagen.

Una vez levantado los servicios de Xampp nos dirigimos a la interfaz de phpMyAdmin, para verificar interfaz del gestor de base de datos Maria DB.



2) Crear la base de datos: Crear con el nombre adipharm_sga posteriormente ejecutar el script adipharm_sga para poder cargar la base de datos del sistema.

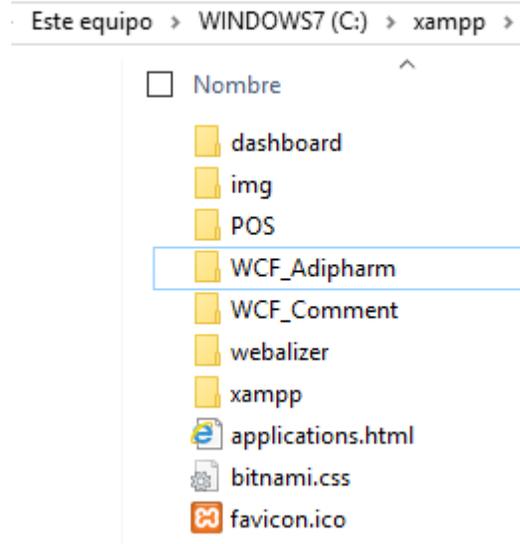
 adipharm_sga (2).sql

La ejecución del script debe ser dentro de la nueva base de datos creada.

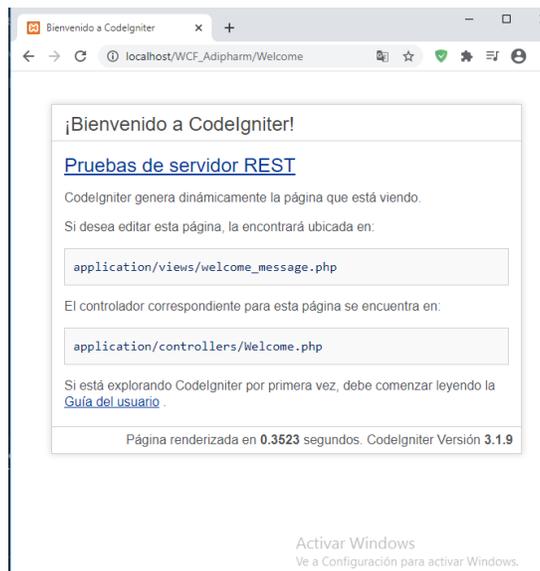
3) Instalación del código fuente de Codeigniter: Para cargar el core del sistema nos dirigimos al directorio de instalación de Xampp a la carpeta **htdocs**

 > Este equipo > WINDOWS7 (C:) > xampp > htdocs

Pegar la carpeta WCF_Adipharm en la ubicación mencionada.



Una vez realizado el procedimiento abrir el navegador (Chrome, mozilla, etc) dirigirse a la dirección “Localhost/WCF_Adipharm/Welcome”



Si muestra la siguiente pantalla, se realizó con éxito la instalación del código de lado de servidor (back-end). Caso contrario volver a repetir los procesos.

4) instalación de Node.js: Para la instalación de Node.js descarga la aplicación, una vez descargada la aplicación ejecutar el archivo de instalación.



Primero verificar que Node.js se haya instalado correctamente para eso abrimos la consola cmd y ejecutar el siguiente comando:

```
1 node -v
```



La instalación de Node.js nos ayudara a ejecutar el código del front-end.

5) Instalación de Angular CLI: Angular cli es una herramienta de interfaz de línea de comandos que se usa para desarrollar aplicaciones angulares es decir para interfaz interactiva de lado del front-end.

Debe abrir la consola de comando y ejecutar el siguiente código:

```
npm install -g @angular/cli
```

Una vez instalada el paquete de angular dirigirnos a la carpeta que contiene el proyecto SIS-ADI

```
PS C:\Users\JOSSO\Desktop\proyectosAngular\SIS-ADI>
```

Una vez ubicados en la dirección de la carpeta ejecutar el siguiente código “npm Install”.

```
PS C:\Users\JOSSO\Desktop\proyectosAngular\SIS-ADI> npm install
```

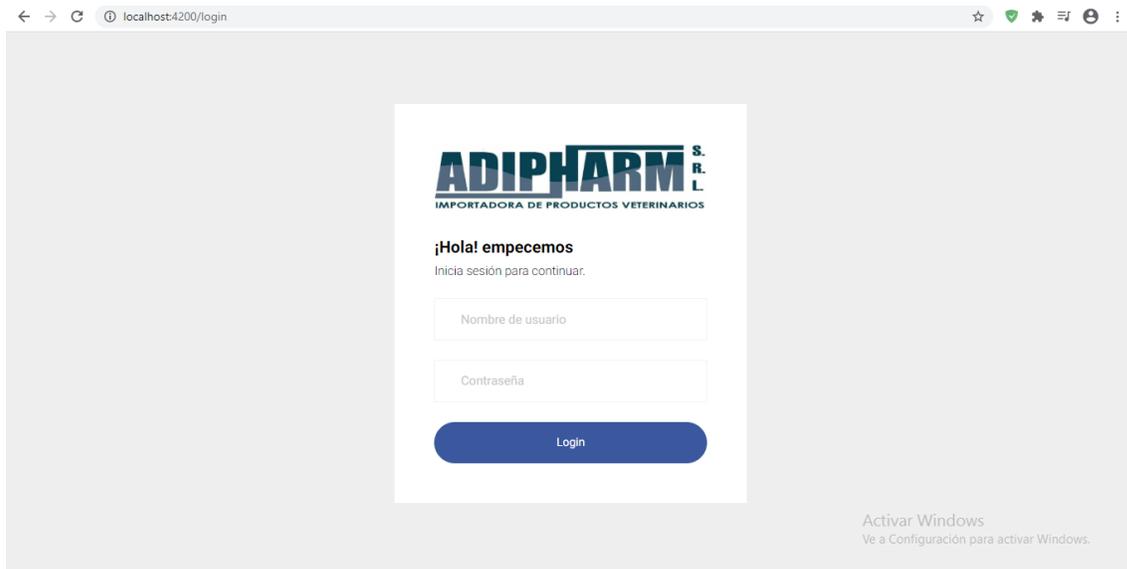
Descargara todas las dependencias para que pueda ejecutarse el sistema web.

Una vez termine el proceso ejecutar el comando “ng serve” que ejecutara y levantara el sistema.

```
PS C:\Users\JOSSO\Desktop\proyectosAngular\SIS-ADI> ng serve
```

Posteriormente una vez compilado el proyecto ir al navegador, dirigirse a la siguiente dirección “localhost:4200” en la que el sistema estará cargado.

Si realizo los pasos adecuadamente le mostrara la interfaz de inicio de sesión.



Caso contrario debe volver a realizar de nuevo el proceso (4).

El Alto, julio 2020

Señor:
**HONORABLE CONSEJO DE CARRERA
INGENIERÍA DE SISTEMAS
UNIVERSIDAD PUBLICA DE EL ALTO**
Presente.-

Ref.: AVAL DE CONFORMIDAD

De mi mayor consideración Honorable Consejo:

Por intermedio de la presente hago llegar un saludo cordial y deseándole éxitos en sus labores cotidianas.

Mediante la presente tengo a bien comunicarle mi conformidad del proyecto de grado "**Sistema web para la gestión de almacenes y distribución. Caso: ADIPHARM S.R.L.**" que propone el postulante: Univ. Jose Ronald Tambo Arcani con cédula de identidad 8373272 L.P. para su defensa pública, evaluación correspondiente a la materia Taller de Licenciatura II, de acuerdo a reglamento vigente de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular y a la espera de una respuesta positiva, me despido muy cordialmente con las consideraciones del caso.

Atentamente:


.....
Lic. Rosa Patricia Nina Chura
TUTOR: REVISOR

El Alto, 10 de julio de 2020

Señores:

**HONORABLE CONSEJO DE CARRERA
INGENIERIA DE SISTEMAS**
Presente. –

REF: AVAL DE CONFORMIDAD

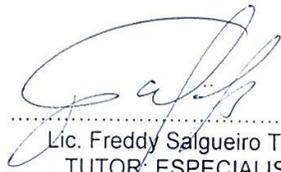
De mi mayor consideración Honorable consejo:

Por intermedio de la presente hago llegar un saludo cordial y deseándole éxitos en sus labores cotidianas.

Mediante la presente tengo a bien comunicarles mi conformidad en la realización del proyecto de grado "Sistema web para la gestión de almacenes y distribución" caso ADIPHARM S.R.L. que propone el postulante: Univ. José Ronald Tambo Arcani, con cedula de identidad 8373272 LP. Para su evaluación correspondiente a la materia Taller de licenciatura II de acuerdo al reglamento vigente de la carrera de Ingeniería de Sistemas de la casa superior de estudios Universidad Pública de EL Alto.

Sin otro particular, reciba saludos cordiales.

Atentamente:


.....
Lic. Freddy Salgueiro Trujillo
TUTOR: ESPECIALISTA



El Alto, julio de 2020

El suscrito **Adolfo Llanque**, gerente general de la empresa ADIPHARM S.R.L. a petición escrita del interesado.

CERTIFICA

La empresa ADIPHARM S.R.L. se complace en extender su aval de conformidad de desarrollo del proyecto titulado: "SISTEMA WEB PARA LA GESTION DE ALMACENES Y DISTRIBUCION", a solicitud del Sr. José Ronald Tambo Arcani con CI: 8373272 LP. Cumpliendo con el proceso de elaboración y construcción del mismo.

Brindamos ese certificado a pedido del interesado, acreditando su participación y el desarrollo del proyecto en nuestra empresa, además de brindar nuestro apoyo necesario a nuestra sociedad brindando nuevas tecnologías y herramientas las cuales están enmarcadas en nuestro objetivo como empresa.

ADIPHARM S.R.L.
Adolfo Llanque
Gerente General



ZONA RIO SECO URB. LOTES Y SERVICIOS MANZANO "P" PATIO "10" N° 1040
 Telf.: 2104175 - 2104173 Cel: 79107986 - 70548863
 mljaneth@hotmail.com

El Alto, 09 de julio de 2020

Señores:

**HONORABLE CONSEJO DE CARRERA
INGENIERIA DE SISTEMAS**
Presente. –

REF: AVAL DE CONFORMIDAD

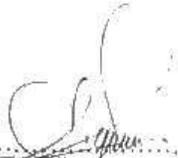
De mi mayor consideración Honorable consejo:

Por intermedio de la presente hago llegar un saludo cordial y deseándole éxitos en sus labores cotidianas.

Mediante la presente tengo a bien comunicarles mi conformidad en la realización del proyecto de grado "**Sistema web para la gestión de almacenes y distribución**" caso ADIPHARM S.R.L. que propone el postulante: Univ. José Ronald Tambo Arcani, con cedula de identidad 8373272 LP. Para su evaluación correspondiente a la materia Taller de licenciatura II de acuerdo al reglamento vigente de la carrera de Ingeniería de Sistemas de la casa superior de estudios Universidad Pública de EL Alto.

Sin otro particular, reciba saludos cordiales.

Atentamente:


.....
Ing. Marisol Arguedas Balladares
TUTOR: METODOLOGICO