

UNIVERSIDAD PÚBLICA DE EL ALTO

CARRERA INGENIERÍA DE SISTEMAS



PROYECTO DE GRADO

“PORTAL WEB PARA EL CONTROL DE ALMACENES Y ACTIVOS FIJOS”

CASO: LAB. ESFASA

Para Optar al Título de Licenciatura en Ingeniería de Sistemas

MENCIÓN: INFORMATICA Y COMUNICACIONES

Postulante: Univ. Javier Calle Gutiérrez

Tutor Metodológico: Ing. Marisol Arguedas Balladares

Tutor Revisor: Ing. Milton Osvaldo Zurita Benito

Tutor Especialista: Lic. María Isabel Conde Altamirano

EL ALTO-BOLIVIA

2020

(TALLER DE GRADO II)

DEDICATORIA

Dedico el presente proyecto al ser que me dio

La vida mi querida madre, a la que entrego su

Vida para que yo siga adelante.

A mi esposa, mis hijos y toda mi familia, que han

Hecho todo lo posible para ayudarme en esta

Parte importante de mi vida.

A los profesionales Tutores por guiarme por

La senda correcta para llegar a la meta.

AGRADECIMIENTO

A dios por darme un día más de vida, al el que guía mis pasos por senderos seguros, dotándome de paciencia y paz mi corazón, y haberme permitido cumplir con un deseo de terminar mi carrera profesional.

A mis tutores Ing. Marisol Arguedas, Ing. Milton Osvaldo Zurita. Lic. María Isabel Conde, Ing. Enrique Flores Baltazar por la paciencia y la vocación de docente.

A mis queridos profesionales de Laboratorios ESFASA, por la confianza dada y por la amistad que recibí dentro de sus instalaciones.

A mi querida Universidad Pública de el Alto, por acogerme en sus aulas, como también a mi querida Carrera Ingeniería de Sistemas.

RESUMEN

El presente proyecto de Grado titulado “Portal Web para el Control de Almacenes y Activos Fijos Caso: Laboratorios ESFASA” ha sido desarrollado para la institución que me abrió las puertas y me apoyo, con el objetivo de automatizar los procesos y mejorar los tiempos de respuesta hacia aquellos obstáculos que se presentan en la cadena de fabricación de medicamentos.

Para el desarrollo del proyecto se utilizó la metodología Ágil Scrum, que propone un modelo incremental, basado en interacciones y revisiones continuas. También se utilizó en cada una de las iteraciones la metodología UWE, que se especializa en el diseño de aplicaciones WEB.

Para la conclusión del desarrollo del portal web se utilizó una herramienta primordial el lenguaje de programación PHP, y el gestor de datos MYSQL.

A fin de asegurar la calidad del software, se ha basado en el estándar ISO 9126, donde indica que se puede medir la calidad del software dependiendo del tipo del producto, en este caso por tratarse de un sistema web de registro y control se midió la usabilidad, funcionalidad, mantenimiento y portabilidad.

INDICE GENERAL

CAPITULO I	1
1. MARCO PRELIMINAR	2
1.1 Introducción.	2
1.2.1 Misión	3
1.2.1 Visión	3
1.2.3 Antecedentes internacionales.	4
1.2.4 Antecedentes nacionales.	4
1.3 Planteamiento del problema	6
1.3.1 Problema principal	6
1.3.2 Problemas secundarios	6
1.4.- Objetivos	7
1.4.1 Objetivo general	7
1.4.2 Objetivos específicos	7
1.5.- Justificación.	7
1.5.1. Justificación técnica.	7
1.5.2. Justificación económica.	8
1.5.3. Justificación social	9
1.6 Metodología	9
1.6.1 UWE (Plataforma Web)	9
1.7. Métrica de Calidad WEB	12
1.8 modelo de costes COCOMO	13
1.8.1 prueba de la caja blanca	14
1.9. Herramientas	15
1.9.1 PHP	16
1.9.2 MYSQL	17
1.9.4 AJAX	17
1.9.5 JAVA SCRIPT	20
1.9.6 BOOTSTRAP	20
1.9.7 MVC (Modelo Vista Controlador).	22
1.9.8 Linux Ubuntu	22

1.8 Límites y alcances	23
1.8.1 Límites:	23
1.8.2 Alcances:	23
1.9 Aportes:	24
2. MARCO TEORICO	26
2.1 Generalidades	26
2.2 Portal web.	27
2.2.1 Servidor Web.	29
2.2.2 Apache.	31
2.3.- Descripción del sistema.	32
2.3.2.- FUNCIONAMIENTO Y TIPOS. -	33
2.4 Metodología de desarrollo	36
2.4.1 Ingeniería de requerimiento	36
2.4.2 El límite de la ingeniería de requerimientos	40
2.4.3 Especificación de los requerimientos	43
2.4.4 Lista consolidada de requerimientos. –	44
2.5 Metodología ágil	44
2.5.1.- SCRUM	45
2.5.2.-Los procesos de scrum	48
2.5.3.- Inicio del ciclo	48
2.5.4.-Sprint	49
2.5.5.-Planeamiento del sprint/ sprint planning	49
2.5.6. - Reunion de equipo scrum /scrum team meeting	49
2.5.7.- Refinamiento del backlog/baklog refinement	50
2.5.8.- Revision del sprint /sprint review	50
2.5.9.- Retrospectiva del sprint/restrospective	50
2.5.10.-Herramientas scrum	51
2.6. UWE (UML BASE WEB ENGINEERING)	54
2.6.1 Modelos de UWE	54
2.7 Metodología de proceso.	58
2.7.1 Principios del modelo	58
2.7.2 Característica del modelo Scrum	59

2.7.3 Fases de la metodología scrum.....	60
2.8.1 Características de la ingeniería de software	64
2.8.2 Ciclo de vida.....	64
2.10 Pruebas.....	69
2.11 métricas de calidad de software.....	69
2.11.1 Métricas para la Calidad de la especificación	70
2.11.2 Métricas para Código fuente.....	70
2.11.3 Métricas para Mantenimiento.....	71
2.12 Seguridad.....	73
2.12.1 Definición de Seguridad Informática.....	73
2.12.2 Fiabilidad, Confidencialidad, Integridad y Disponibilidad.....	73
2.13 Estimación de costos en proyectos de software	75
2.13.1 modelo cócono	75
2.12 Seguridad.....	77
CAP III	80
3. MARCO APLICATIVO	81
3.1 Introducción	81
3.2 análisis situación actual.....	82
3.3 Análisis de requerimientos. -	84
3.3.1 Funciones del sistema	88
3.3.2 Requerimientos funcionales.....	91
3.3 Diseño	104
3.3.1 Evaluación de la fase de inicio	104
3.3.2 Planificación fase de elaboración.	104
3.4 Desarrollo del modelo	105
3.4.1 Modelo Conceptual.....	105
3.4.1.1 Diagrama de Secuencias.....	105
3.5 Fase de Construcción.....	110
3.5.1 Base de datos.....	111
3.5.2.1 Modelo Entidad Relación: Control de Almacenes	111
3.5.2 Patrón Modelo Vista Controlador	112
3.5.3 Diseño de Interfaces.....	113

3.6 FASE DE TRANSICION	116
3.6.1 Pruebas de la caja blanca.....	116
3.6.1 Pruebas de Caja Blanca Login Al sistema.....	117
CAPITULO IV	122
4 Calidad y seguridad	123
4.1 Evaluación de calidad.....	123
4.1.2 Definiendo metas de evaluación	123
4.3 Análisis de costo.....	128
4.3.1 Método de estimación de costos COCOMO II	128
CAP V	134
5. Conclusiones y Recomendaciones	135
5.1 Conclusiones.	135
5.2 Recomendaciones	136
Bibliografía	137
Anexo A	140
A.1. árbol de problemas	140
A2 Árbol de Objetivos	141
Anexo B1	143
CUESTIONARIO DE ENTREVISTA.....	143
ANEXO C1	146
MANUAL DEL USUARIO Y DEL ADMINISTRADOR.....	146
DICCIONARIO DE DATOS.....	157

CAPITULO II

INDICE DE FIGURAS

Fig. 2 1 Movimiento de la informacion_____	26
Fig. 2. 2 Arquitectura Web (Web)_____	27
Fig. 2 3 funcionalidad del portal _____	29
Fig. 2 4 Estadística de servidores en el Mundo _____	30

Fig. 2 5 Iteración en cascada _____	41
Fig. 2 6 equipo central de Scrum _____	47
Fig. 2 7 Proceso de Scrum _____	48
Fig. 2 8 fase de la Metodología Scrum_____	60
Fig. 2 9 etapas de la ingeniería de software _____	64
Fig. 2 10 Vista de código HTML_____	68
Fig. 2 11 herramientas utilizadas en el proyecto_____	69

CAPITULO III

INDICE DE FIGURAS

fig. 3 1 Modelo del negocio del sistema de control de almacenes e inventarios _	88
fig. 3 2 Casos de uso del Modelo del Negocio del Sistema Actual: Gestión de Control de Almacenes y activos fijos _____	83
Fig. 3 3 Casos de uso del Modelo del Negocio del Sistema Actual: Gestión de Activos Fijos_____	84
fig. 3 4 Casos de uso del Modelo del Negocio del Sistema Actual: Gestión de Almacenes _____	91
fig. 3 5 Casos de uso del Modelo del Negocio del Sistema Actual: Gestión de Salidas y Ventas de Almacenes. _____	95
fig. 3 6 Diagrama de Uso: Inventario de Activos Fijos _____	97
fig. 3 7 Diagrama de caso de Uso: Renta de activos Fijos._____	97
fig. 3 8 Caso de Uso Expandido Inventario de Activos Fijos _____	99
fig. 3 9 Caso Expandido: Registro de Activos Fijos_____	101
fig. 3 10 diagrama de caso de uso registro activos fijos _____	103
fig. 3 11 Diagrama de Secuencia: Gestionar Ingresos a Almacén _____	105

fig. 3 12 Diagrama de Colaboración: Gestionar Ingresos a Almacén _____	106
fig. 3 13 Diagrama de Secuencia: Registro de Materiales _____	106
fig. 3 14 Diagrama de Secuencia: Actualizar kardex _____	107
fig. 3 15 Diagrama de Colaboración: Actualizar Kardex _____	107
fig. 3 16 Diagrama de Secuencias: Órdenes de Compra _____	108
fig. 3 17 Diagrama de Colaboración: Órdenes de Compra _____	108
fig. 3 18 Diagrama de Secuencia: Órdenes de Venta _____	109
fig. 3 19 Diagrama de Colaboración: Órdenes de Compra _____	109
fig. 3 20 Diagrama de secuencia: Renta de Activos fijos _____	110
fig. 3 21 Diagrama de Clases: Control de Almacenes e Inventarios _____	110
fig. 3 22 Diagrama de Componentes: Control de Almacenes e Inventarios ____	111
fig. 3 23 modelo entidad relación: diagrama de clases del sistema _____	112
fig. 3 24 Movimiento de (MVC) _____	113
fig. 3 25 Función ingresar código fuente _____	119
fig. 3 26 condigo fuente función ingreso con Ajax _____	120
fig. 3 27 código fuente función registrar _____	121

INDICE DE TABLAS

CAP II

Tabla 2 1 análisis de requerimientos _____	42
Tabla 2 2 Constante de Complejidad _____	76

INDICE DE TABLAS

CAPITULO III

Tabla3 1 requerimientos de Control de Almacenes e Inventarios _____	85
Tabla3 2 requerimientos de Control de Activos Fijos _____	86
Tabla3 3 caso de uso almacenero _____	92
Tabla3 4 Caso de Uso encargado almacén _____	93
Tabla3 5 caso de uso laboratorista _____	93
Tabla3 6 caso de uso de uso Gerencia _____	94
Tabla3 7 Salida de Materiales de Almacén _____	96
Tabla3 8 caso de uso renta de activos fijos _____	9
Tabla3 9 Caso de Uso Inventario de Activos Fijos _____	100
Tabla3 10 caso de uso registro de activos fijos _____	102
Tabla3 11 Tiempo de desarrollo de Software _____	104

INDICE DE TABLAS

CAPITULO IV

Tabla 4 1 Usabilidad _____	125
Tabla 4 2 Funcionalidad _____	126
Tabla 4 3 Confidencialidad _____	127
Tabla 4 4 Eficiencia _____	127
Tabla 4 5 Ecuaciones de COCOMO _____	129
Tabla 4 6 Constantes de Complejidad _____	129
Tabla 4 7 Factores de Trabajo _____	130
Tabla 4 8 Resultados del Sistema _____	133

INDICE DE TABLAS

CAPITULO 5

Tabla 5 1 Objetivos y Conclusiones_____	135
---	-----

INDICE DE TABLAS

DICCIONARIO DE DATOS

TABLA 1 Sucursal.....	157
TABLA 1 2 Usuario	157
TABLA 1 3 Ingreso.....	158
TABLA 1 4 Persona	159
TABLA 1 5 Empleado.....	159
TABLA 1 6 Categoría	160
TABLA 1 7 Artículo.....	160
TABLA 1 8 Detalle Ingreso.....	161
TABLA 1 9 Detalle Pedido	161
TABLA 1 10 Pedido.....	162
TABLA 1. 11 Venta	162
TABLA 1.12 Unidad de Medida.....	163
TABLA 1 13 Crédito	163
TABLA 1 14 Detalle Documento Sucursal	163
TABLA 1.15 Tipo de Documento.....	164
TABLA 1.16 Global	164

TABLA 1.17 Activos.....	170
TABLA 1.18 Asignación.....	171
TABLA 1.19 Bajas.....	171
TABLA 1.20 Estado.....	172

CAPITULO I

1. MARCO PRELIMINAR

1.1 Introducción.

En la actualidad los Sistemas basados en tecnología WEB se constituyen como las herramientas principales para satisfacer las constantes necesidades en las organizaciones. Debido a ello, surgen las necesidades de usar entorno web, con el fin de mejorar la productividad y el logro de los objetivos.

Laboratorios ESFASA S.R.L. cuenta con dos sectores que generan información preponderante en sus actividades diarias. Una Almacenes, donde se registra el ingreso de las compras de reactivos, cajas de cartón, tapas de botella, etiquetas, ropa de trabajo etc. Y la otra los Activos Fijos donde se asienta los registros de: (Maquinaria, Equipos de Computación, muebles, etc.).

El portal web caso: Lab. ESFASA S.R.L, dará un nuevo servicio a la administración en el sector de almacenes controlando las entradas de (reactivos, embalajes, material de vidrio, plástico, cartón, etiquetas, para producto terminado.

El presente proyecto se apoya en las metodologías, SCRUM, UWE para iniciar la primera fase de desarrollo, y para luego hacer el uso de las herramientas PHP, MYSQL, BOOSTSTRAP, JAVA SCRIPT, HTML, AJAX.

También haremos uso de la arquitectura de software modelo vista controlador (MVC), la cual es muy utilizada en las aplicaciones web.

Esta institución actualmente utiliza sistema operativo (Linux Ubuntu versión 16.4) en sus equipos de computación.

1.2 antecedentes Institucionales.

Laboratorios ESFASA S.R.L. Se funda en Bolivia en el año de 1968, durante la presidencia de Rene Barrientos Ortuño, bajo la presidencia de Don Roberto Sabauste, inicialmente con licencia de “LABORATORIOS ANDREU”. Laboratorios ESFASA S.R.L.se dedica a la elaboración de productos farmacéuticos, actualmente ubicada la zona de Sopocachi, calle Méndez Arcos # 957.

1.2.1 Misión

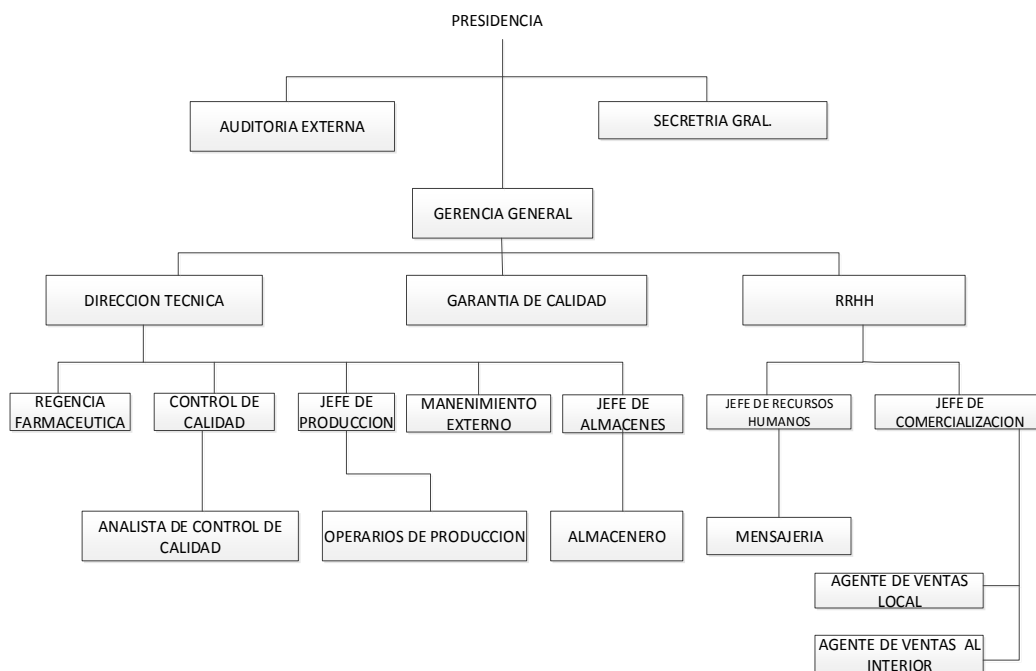
Laboratorios ESFASA S.R.L. es un grupo humano comprometido y dedicado a proteger la salud elaborando especialidades farmacéuticas de calidades comprobadas y accesibles a la sociedad.

1.2.1 Visión

Ser líderes en la industria farmacéutica nacional, desarrollado productos con la más alta e innovadora tecnología.

Dirección: zona Sopocachi plaza Andreu calle Méndez arcos nro. 957-b

Organigrama Laboratorios ESFASA S.R.L.



1.2.3 Antecedentes internacionales.

Desarrollo de un sistema de información para la gestión de los proyectos de responsabilidad social del programa de Ingeniería de Sistemas de la Universidad Católica de Colombia

Universidad Católica de Colombia facultad de ingeniería programa de ingeniería de sistemas

Universitario: Juan Manuel Pascagaza Gitierrez

Bogotá D.C. 2018

Trabajo de grado para optar al Título de Ingeniero de Sistemas

Director ING. Nixon Alonso Duarte Acosta

El presente proyecto se utilizó:

ASP.NET

Marco de servicios web XML

PHP, Java 2 Enterprise Edition (J2EE), Python.

1.2.4 Antecedentes nacionales.

Habiendo realizado una revisión de trabajos de grado de Ingeniería Sistema e Informática que tienen alguna Similitud con el tema tratado que muestran diferentes enfoques, he encontrado diferentes proyectos, pero con la misma intención de poder organizar y controlar la cadena de producción paso a paso y sin violar los métodos científicos ya estandarizados.

Así podemos hacer notar los proyectos de grado que son similares al nuestro, para conocer cómo se realizaron que metodologías se tomaron en cuenta para llegar a un objetivo satisfactorio.

Proyecto de Grado UMSA Carrera Informática

“sistema de Información Vía web para control de almacenes Fabrica de fideos santa rosa”

Postulante. Julia Mamani Mamani

Tutor: Lic. Efrain Silva Sanchez

Revisor: Lic. msc. Carlos Mullisaca Choque para el desarrollo del sistema se empleó la metodología de procesos unificado racional (rup) como herramienta, y se utiliza el lenguaje unificado de modelado (UML), para realizar la técnica de inventario de bienes el modelo primero entrar primero en salir (PEPS) año 2009

Proyecto de Grado UMSA Carrera Informática

“control y seguimiento de activos fijos del servicio exterior via web ministerio de relaciones exteriores y culto”

Postulante: Mario Candillo zanga

Tutor: Lic. Franz cuevas Quiroz m.sc.

Revisor: Lic. Rubén Alcon López

En el desarrollo del software se ha utilizado la metodología orientada a objetos (POO) el proceso unificado de racional (RUP), uso de UML para el modelado de sistemas. Año 2009

Proyecto de grado UMSA carrera de Informática

“sistema web para el control de activos fijos y almacenes caso Fundación cuerpo de cristo- el alto”

Postulante. Freddy Quispe Yujra

Tutor Metodologico: m.sc. aldo ramiro valdez alvarado

Asesor: m.sc Carlos Mullisaca Choque

2009

Para la realización de este proyecto de grado se utilizó la metodología aup (agile unified process), también se utilizó la metodología uwe (uml-bose web engineering) ya que esta metodología se especializa en el diseño de aplicaciones web. Además, el sistema web fue desarrollado con asp.net 2008, como lenguaje de programación se utilizó vb.net y como gestor de base de datos se utilizó sql server 2008. La calidad fue evaluada mediante la metodología web-site qem (quality evaluación methology) Mediante las características de calidad descritas en el estándar iso 9126 para estimar el coste, esfuerzo y tiempo se utilizó cócono ii y los indicadores van (valor actual neto) tir (tasa interna de retorno).

1.3 Planteamiento del problema

1.3.1 Problema principal

¿Qué efecto ocasiona la mala administración de ESFASA S.R.L. considerando de no contar con un sistema confiable generando duplicidad en la información y la demora de entrega de productos ocasionando alta ineficiencia?

1.3.2 Problemas secundarios

- No contar con una base de datos segura y confiable, para registrar las compras y las ventas
- La disponibilidad y fiabilidad de la información a tiempo real no es certera y eso repercute en la administración de la empresa.
- El procedimiento, de la información manual genera, duplicidad en los informes mensuales.
- El Descontrol en la ubicación de los productos, provoca pérdida de tiempo.
- Demora en la entrega de reportes actualizados de stock de productos.
- Información insuficiente del material requerido para la elaboración de medicamentos.

- control inadecuado de los activos fijos en el procedimiento de asignación de usuarios.

1.4.- Objetivos

1.4.1 Objetivo general

Desarrollar un Portal Web para el control de almacenes y activos fijos para mejorar la administración y manejar información de forma rápida y eficiente evitando duplicidad de información

1.4.2 Objetivos específicos

- Diseñar una base de datos que contenga la información completa sobre las compras, ventas y activos fijos seguro y confiable.
- Diseñar un sistema de reportes sobre compras y ventas, que estará al alcance del usuario con soporte web en cualquier momento y lugar.
- Diseñar un sistema de fácil manejo con una interfaz amigable.
- Diseñar un sistema de datos de activos fijos para mejorar prestaciones de los muebles e inmuebles.
- Diseñar la red y configurar el servidor web de LAB. ESFASA S.R.L.

1.5.- Justificación.

1.5.1. Justificación técnica.

La creciente complejidad de las organizaciones ha aumentado en gran medida la necesidad de tener información más conveniente y oportuna. Actualmente laboratorios ESFASA fijo su mirada en tecnologías Web, para un mejor desenvolvimiento de sus procesos de información, para mejorar el servicio de manera más rápida e eficaz. En un entorno web.

El sistema a desarrollar es de suma importancia, porque con él se busca mejorar los procesos actuales permitiendo conocer en tiempo real la información de cada uno de ellos, y dentro el almacén saber cuáles están en uso o han sido desincorporados, a su vez permitirá controlar el inventario del almacén, que materiales se han entregado a los diferentes departamentos, con que material se dispone y que departamento tiene pedidos realizados y faltan por despachar, toda esta información es de suma importancia y la misma con la sistematización de los procesos podrá ser suministrada de manera oportuna, confiable y veraz.

Por otro lado, el hardware que ESFASA tiene es solo utilizado para realizar cartas y correos electrónicos. Con la implementación del portal web las reparticiones inmiscuidas con el almacén recibirán una mejor atención sobre la información requerida.

1.5.2. Justificación económica.

Los problemas mencionados sobre el manejo de información que este caso es un recurso económico principal de Lab. ESFASA S.R.L., no será obstáculo porque el desarrollo del software tiene un costo muy bajo en relación a los beneficios que obtendrá la industria al implantarlo.

Además, Lab. ESFASA S.R.L., cuenta con el hardware necesario para implantar este Portal. Por lo cual no significara un gasto adicional.

La implementación del Portal web se realizará con software libre (sistema operativo Linux), PHP, MySQL lo que reducirá el costo del desarrollo notablemente, al no comprar licencias.

1.5.3. Justificación social

Con el desarrollo de este proyecto, los empleados del Laboratorio podrán realizar sus tareas con mayor comodidad y facilidad, posibilitando una mayor eficiencia de los mismos. Se pretende un crecimiento y consolidación de LABORATORIOS ESFASA S.R.L., su interfaz amigable hará que haya una mejor comunicación entre las áreas inmiscuidas, el personal operativo del área de ventas no pasará mucho tiempo en trabajos tediosos y repetitivos. La Gerencia tendrá el beneficio de estar bien informado en el momento que desee para así tomar decisiones oportunas.

La disponibilidad de consulta demandada de componentes beneficiara tanto al personal operativo como ejecutivo, el mismo permitirá búsqueda de información rápida y oportuna en el momento que se la requiera.

La propuesta de solucionar permitirá mejorar la confidencialidad de los procesos de compra y venta, mejorar los tiempos de respuesta, minimizar procesos operativos y administrativos con el personal del, Laboratorio como también con los proveedores y clientes finales.

1.6 Metodología

Portal web para el control de almacenes y activos fijos

Utiliza la siguiente metodología:

1.6.1 UWE (Plataforma Web)

Es un Método de Ingeniería de Software para el desarrollo Web basado en UML, es decir es una herramienta basada en UML, pero para aplicaciones Web, esto con lleva a que cualquier diagrama UML puede ser usado, debido a que es una extensión de UML.

El modelo que propone UWE está compuesto por 6 etapas o submodelos:

- Modelo de Casos de Uso. - los casos de uso bajo la forma de acciones y reacciones el comportamiento del sistema desde el punto de vista del usuario, permiten definir los límites del sistema y las relaciones entre el sistema y el entorno. Los casos de uso son descripciones funcionales del sistema; describen como los actores pueden usar un sistema.
- Modelo de Contenido. - contiene los elementos estructurales que proporcionan una importante visión de los requisitos de contenido para una web App.
- Modelo de Usuario. - Las declaraciones y los deseos de un usuario se obtienen por medio del modelado de un usuario; a través de éste podremos saber el perfil que tiene un usuario, los intereses sobre ciertos recursos y que recursos derivan otros intereses para un usuario. Un modelo de usuario es la representación de las características que tiene un usuario, y en base a éstas se podrán tomar decisiones en la interacción [Fischer, 2000]; ésta es la parte inicial del sistema de agentes, a través de los intereses generados por el servidor de modelos de usuarios se podrá tener una serie de valores para la navegación y recomendación de recursos a través de las creencias y los deseos del mismo.
- Modelo de estructura. - El modelo o estructura organizacional se define como la forma en la que se distribuyen los departamentos, actividades, grupos y profesionales en una empresa. ... La cadena de mando, que se

rige bajo un sistema de jerarquía en beneficio de las comunicaciones y coordinaciones en la organización.

- **Modelo Abstracto.** - es un concepto fundamental en la informática y el desarrollo de software. El proceso de abstracción también puede denominarse modelado y está estrechamente relacionado con los conceptos de teoría y diseño. Los modelos también pueden considerarse tipos de abstracciones por su generalización de aspectos de la realidad.

Abstracción (informática) -

[https://es.qaz.wiki/wiki/Abstraction_\(computer_science\)](https://es.qaz.wiki/wiki/Abstraction_(computer_science))

Modelo de Adaptación. - adaptación se basa en el concepto que postula que las poblaciones de organismos cambian con el tiempo como resultado de la selección natural. La evolución adaptiva está guiada por un aumento de sobrevivencia y/o un aumento del éxito reproductivo.

Fases:

- **Fase de requisitos:** 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.
- **Fase de análisis y diseño:** UWE distingue entre diseño conceptual, de modelo de usuario, de navegación, de presentación, de adaptación, de la arquitectura, en el diseño detallado de las clases y en la definición de los subsistemas e interfaces.

- **Fase de implementación:** UWE incluye 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. (Hernández Eduard; 2012).

1.7. Métrica de Calidad WEB

(<https://grupodigital360.com/category/seo-posicionamiento-web/>)

El ISO 9126 es el estándar internacional establecido como método de evaluación de la calidad de evaluación de la calidad de las aplicaciones.

Esta norma se interesa por la creación de las webs, esta una herramienta útil que ayuda al creador, por una serie de mediciones, mejorar su trabajo.

Estas directrices se crearon en 1991. Su procedimiento establece las características que propician la calidad del software, referidas a la calidad de visión Interna y externa. Y así mismo, define los para metros a medir en el desarrollo web como: funcionalidad, confiabilidad, usabilidad, eficacia, flexibilidad de mantenimiento y portabilidad.

- **Funcionalidad.** - es la capacidad del programa de cubrir los requisitos utilitarios establecidos implícita y explícitamente
- **Confiabilidad.** - en este caso se estudia si la estructura puede mantener las prestaciones por un tiempo y bajo ciertas condiciones definidas.
 - **Usabilidad.** - Mide el esfuerzo necesitado por el usuario para usar el modelo a satisfacción.

- **Eficacia.** - considera la dependencia entre las prestaciones del sistema y las condiciones de servicio.
- **Mantenibilidad.** - Atributos que califican el empeño de ampliar, modificar o resolver los problemas.
- **Portabilidad.** - Determina si el modelo es capaz de ser transferido a otra plataforma.

1.8 modelo de costes COCOMO

(Constructive cost Model)

Es utilizado en proyectos de desarrollo de software para estimar los costes del mismo en función de tres sub modelos: COCOMO I, COCOMO II y COCOMO III. El modelo COCOMO es uno de los sistemas de estimación de costes más utilizados en proyectos de desarrollo de software. La estandarización de su uso y la facilidad de la aplicación del mismo junto con la aproximación al coste real, han convertido a este modelo en uno de los referentes en este tipo de proyectos (Aparicio C, 2012).

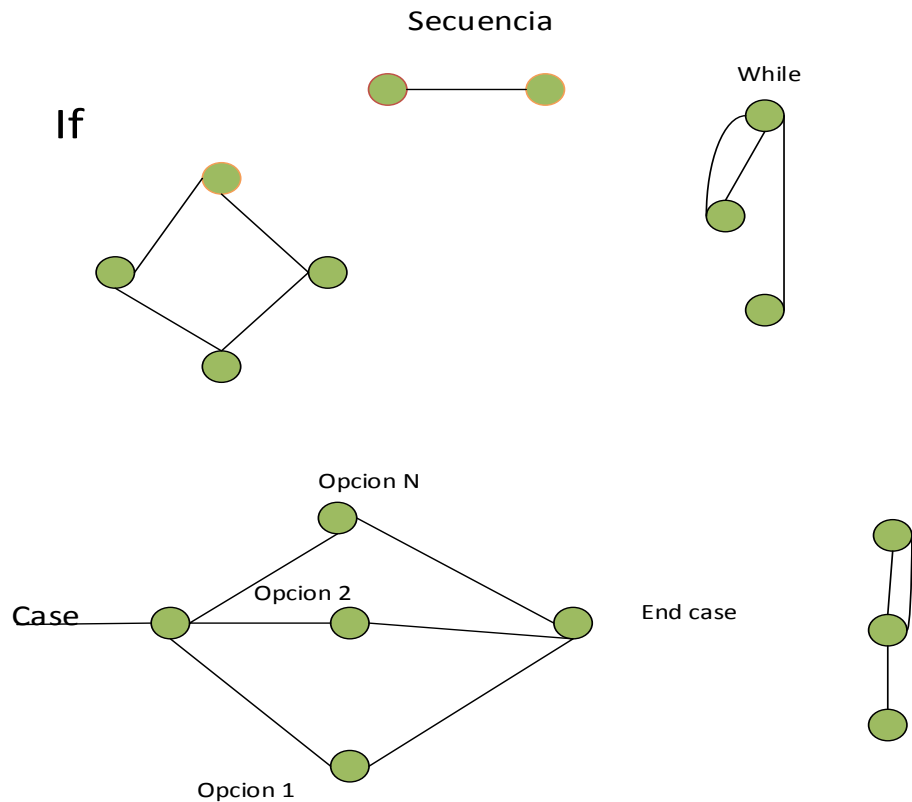
El modelo COCOMO I estima el coste del proyecto pequeño o mediano en función de número de líneas de código estimadas. Es un modelo que permite estimar el costo, el esfuerzo, y programar la hora de planificar una nueva actividad de desarrollo de software. En el modelo COCOMO I uno de los factores más importantes que influye en la duración y el costo de un proyecto de software es el Modo de Desarrollo. (Gonzales E, 2013).

En COCOMO 1, el tamaño del software se expresa mediante líneas de código. Por el contrario, COCOMO 2 proporciona más factores para expresar el tamaño del software, como puntos de objeto, línea de código y también puntos de función.

1.8.1 prueba de la caja blanca

– Técnicas de caja blanca o estructural, que se basan en un minucioso examen de los detalles procedimentales del código a evaluar, por lo que es necesario conocer la lógica del programa. – Técnicas de cajas negras o funcionales, que realizan pruebas sobre la interfaz del software a probar La Figura 1.1 representa gráficamente la filosofía de las pruebas de caja blanca. Como se puede observar las pruebas de caja blanca necesitan conocer los detalles procedimentales del código, mientras que las de caja negra únicamente necesitan saber el objetivo o funcionalidad que el código ha de proporcionar.

Fig. 1 1 Secuencia de la prueba de la caja Blanca



A primera vista parecería que una prueba de caja blanca completa nos llevaría a disponer de un código perfectamente correcto. De hecho, esto ocurriría si se han probado todos los posibles caminos por los que puede pasar el flujo de control de un programa. Sin embargo, para programas de cierta envergadura, el número de casos de prueba que habría que generar sería excesivo, nótese que el número de caminos incrementa exponencialmente a medida que el número de sentencias condicionales y bucles aumenta. Sin embargo, este tipo de prueba no se desecha como impracticable. Se pueden elegir y ejercitar ciertos caminos representativos de un programa.

A este tipo de técnicas se le conoce también como Técnicas de Caja Transparente o de Cristal. Este método se centra en cómo diseñar los casos de prueba atendiendo al comportamiento interno y la estructura del programa. Se examina así la lógica interna del programa sin considerar los aspectos de rendimiento. El objetivo de la técnica es diseñar casos de prueba para que se ejecuten, al menos una vez, todas las sentencias del programa, y todas las condiciones tanto en su vertiente verdadera como falsa. Como se ha indicado ya, puede ser impracticable realizar una prueba exhaustiva de todos los caminos de un programa. Por ello se han definido distintos criterios de cobertura lógica, que permiten decidir qué sentencias o caminos se deben examinar con los casos de prueba

1.9. Herramientas

Las herramientas a utilizarse será la UML (unified Modeling Lenguaje), Lenguaje de programación PHP, gestor de base de Datos MySQL. Además de herramientas de programación como son los frameworks .

UML.- Es un lenguaje de modelado visual de sistema. Aunque UML está más asociado con modelar sistemas de software orientados a objetos, tiene una aplicación más amplia debido a sus mecanismos de extensibilidad. Los diagramas UML son legibles por las personas y los ordenadores pueden mostrarlos fácilmente, pero es importante apreciar que UML no nos proporciona ningún tipo de metodología de modelado, simplemente proporciona una sintaxis visual que podemos utilizar para construir modelos. Esto es un punto importante a mencionar, ya que cualquier persona que empieza con el desarrollo de software puede pensar que UML puede desarrollarse un proyecto de un sistema de información solamente en base a sintaxis, como si se tratase de una metodología.

Y para eso utilizamos los diagramas que nos hacen ver como realmente se visualiza el proceso de cualquier proyecto de sistemas y así tenemos los diagramas:

Diagramas de caso de uso. -

Diagramas de clases

Diagramas de secuencia

Para luego desarrollar el código en cualquier lenguaje java, c++, Python visual.net

1.9.1 PHP

(<https://desarrolloweb.com>)

Es el lenguaje de programación más usado del lado del servidor en la web.

Utilizando php puedes realizar el backend¹d de aplicaciones web de una manera sencilla y accesible.

¹ El **Back-End** es la parte o rama del desarrollo web encargada de que toda la lógica de una página funcione. Consiste en el conjunto de acciones que pasan dentro de una web, pero que no podemos ver. Un ejemplo de esto es la comunicación con el servidor.

1.9.2 MYSQL

MySQL es un sistema de gestión de datos relacionales utilizando de sus diferentes capacidades con tablas, vistas, procedimientos almacenados, funciones. Etc. Una compañía sueca llamada MySQL AB originalmente desarrolló MySQL en 1994.

En cuanto a la definición general, MySQL es un sistema de gestión de bases de datos relacionales de código abierto (RDBMS, por sus siglas en inglés) con un modelo cliente-servidor. Es un software o servicio utilizado para crear y administrar bases de datos basadas en un modelo relacional.

1.9.4 AJAX

(<https://www.aprende-web.net/progra/ajax/ajax>)

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

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

Con Ajax podemos crear páginas interactivas. En éstas solicitamos datos al servidor, los cuales podemos tener guardados en otras páginas o en bases de datos. El servidor devuelve los datos, los cuales se cargan en la misma página y en segundo plano. Lo de "segundo plano" significa que mientras esperamos que se

reciban los datos la página no se queda parada, y el usuario o la programación de la página pueden seguir haciendo otras cosas.

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

Componentes de AJAX

Ajax es una combinación de los siguientes lenguajes de programación y elementos:

HTML (o XHTML) y CSS: Base para el diseño de la página.

DOM y Javascript: Forma de acceder dinámicamente a las partes de la página.

Objeto XMLHttpRequest: Es el que permite la comunicación asíncrona (en segundo plano) con el servidor.

XML: Formato en el que están los datos que se solicitan al servidor; aunque otros formatos también pueden funcionar, como son HTML, texto plano (txt), json... etc.

PHP: En este manual trataremos también cómo mandar datos al servidor. Este los recoge mediante PHP. Una vez enviados pueden guardarse en una base de datos o procesarlos para enviar alguna información.

Cómo funciona Ajax

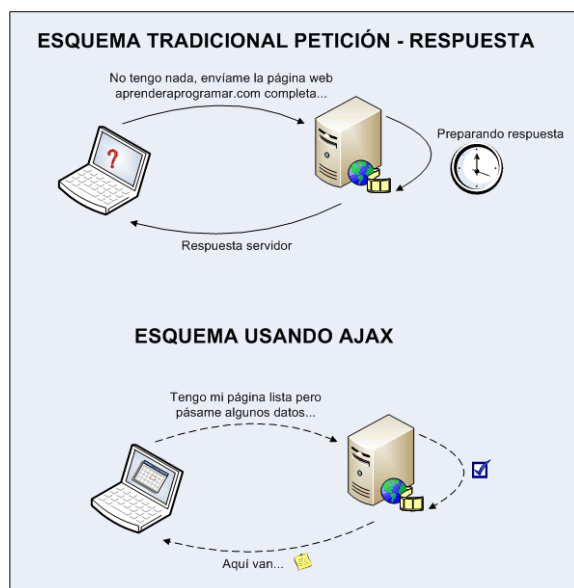
Usando sólo PHP u otros lenguajes de servidor, al hacer una petición, el servidor realiza una serie de tareas y después nos devuelve los datos. Mientras se realiza este proceso la página permanece en espera, es decir está parada. Esto puede que no tenga importancia si se manejan pocos datos y el servidor tiene potencia para responder rápidamente. Sin embargo, si se manejan muchos datos o hay muchas

peticiones a la vez (páginas muy visitadas), el tiempo de respuesta puede ser más largo. Mientras se espera la respuesta la página permanece parada.

Con Ajax al trabajar de forma asíncrona, permite que el usuario pueda seguir haciendo otras cosas o la página pueda mostrar otras cosas mientras se produce la respuesta. El siguiente gráfico muestra la forma de trabajar pidiendo datos al servidor de forma síncrona (sin Ajax) o asíncrona (con Ajax):

Forma clásica de trabajar al interactuar con el servidor

Fig. 1 2 Esquema de petición y respuesta



1.9.5 JAVA SCRIPT

JavaScript es el lenguaje de programación encargado de dotar de mayor interactividad y dinamismo a las páginas web. Cuando JavaScript se ejecuta en el navegador, no necesita de un compilador. El navegador lee directamente el código, sin necesidad de terceros. Por tanto, se le reconoce como uno de los tres lenguajes nativos de la web junto a HTML (contenido y su estructura) y a CSS (diseño del contenido y su estructura).

Con este lenguaje de programación del lado del cliente (no en el servidor) podemos crear efectos y animaciones sin ninguna interacción, o respondiendo a eventos causados por el propio usuario tales como botones pulsados y modificaciones del DOM (document object model). Por tanto, nada tiene que ver con el lenguaje de programación Java, ya que su principal función es ayudar a crear páginas webs dinámicas.

El código de programación de JavaScript se ejecuta en los navegadores, ya sean de escritorio o móviles, ya sean Android o iPhone. Sirve para exactamente lo mismo, da igual en el tipo de dispositivo que se ejecute el navegador.

JavaScript es capaz de detectar errores en formularios, de crear bonitos sliders que se adapten a cualquier pantalla, de hacer cálculos matemáticos de forma eficiente, de modificar elementos de una página web de forma sencilla. Pero también JavaScript es el encargado de que existan herramientas como Facebook Pixel y tantas otras, que son claros ejemplos de JavaScript.

1.9.6 BOOTSTRAP

(<https://blog.baehost.com/bootstrap>)

Fig. 1 3 Bootstrap es Responsive



Bootstrap es un framework CSS de código abierto que favorece el desarrollo web de un modo más sencillo y rápido. Incluye plantillas de diseño basadas en HTML y CSS con la que es posible modificar tipografías, formularios, botones, tablas, navegaciones, menús desplegables, etc. También existe la posibilidad de utilizar extensiones de JavaScript adicionales.

Fue desarrollado inicialmente por Twitter en 2011 y permite crear interfaces de usuario limpias y compatibles con todo tipo de dispositivos. Entre las ventajas que tiene Bootstrap es que favorece el cuál se utiliza para mejorar la experiencia de los usuarios en el sitio web y en consecuencia el beneficio de usar responsive design en un sitio web, es principalmente que el sitio web se adapta automáticamente al dispositivo desde donde se acceda.

Es un marco de trabajo que ayuda, a mejorar la apariencia de las páginas webs y adaptarlos al tamaño de cualquier display (pantalla).

1.9.7 MVC (Modelo Vista Controlador).

MVC es un patrón de diseño arquitectónico de software, que sirve para clasificar la información, la lógica del sistema y la interfaz que se le presenta al Usuario. En este tipo de arquitectura existe un sistema central o controlador que gestiona las entradas y salidas del sistema, uno o varios modelos que se encargan de buscar datos e información necesaria y una interfaz que muestra los resultados al usuario final. Es muy usado en el desarrollo web porque al tener que interactuar varios lenguajes, para crear un sitio es muy fácil generar, confusión entre cada componente si estos no son separados de la forma adecuada. Este patrón permite modificar cada uno de sus componentes sin necesidad de afectar a los demás.

Modelo. - el modelo se encarga de los datos generalmente (pero no obligatoriamente) consultando la base de datos.

Controlador. - se encarga de controlar, recibe las órdenes del usuario y se encarga de solicitar los datos del modelo y comunicárselos a la vista.

Vistas. - son la representación de lo visual de los datos, todo lo que tenga que ver con la interfaz gráfica va aquí. Ni el modelo ni el controlador se preocupan de cómo se verán los datos, esa responsabilidad es de la vista.

1.9.8 Linux Ubuntu

Ubuntu es un sistema operativo de software libre, la distribución de GNU/ Linux (también abreviado como distro) consiste en una variada recopilación de aplicaciones y herramientas a lado del núcleo de Linux. Ubuntu es una distribución fácil de usar orientada a: usuario de escritorio como al servidor, se encuentra

mantenida por una comunidad de desarrolladores que reciben el apoyo de la empresa canonical, la cual vende servicios relacionadas con la distribución.

Actualmente Ubuntu soporta las arquitecturas Intel x86 (IBM compatible PC)

AMD64 (Hammer) y power Pc (Apple iBook y power book, G4, G5)

Ubuntu significa o proviene de la palabra de idioma africano y significa “Humanity to Others” ampliando la palabra de software libre.

La filosofía de Ubuntu es el manifiesto de que el software debe estar disponible sin coste alguno y también de tener la posibilidad de adaptarlo a las necesidades de cada usuario.

De esta forma el proyecto actual se desarrollará para el sistema operativo Linux/Ubuntu. Como ya enunciamos es software libre y se acomoda a los requerimientos de nuestro proyecto.

1.8 Límites y alcances

1.8.1 Límites:

El presente proyecto será una herramienta en un entorno web con una base de datos segura y rápida, facilitará la interoperabilidad de los Laboratorios ESFASA a nivel país en temas de compras y ventas del almacén principal. El límite del proyecto comprende todo lo referente al Almacén y activos fijos de laboratorios ESFASA S.R.L. situada en la ciudad de la paz.

1.8.2 Alcances:

El presente proyecto nos permitirá automatizar los siguientes procesos.

y sus módulos son:

a) Se realizará un módulo de sistema web multiusuario.

- b) Realizar un módulo para gestionar los roles de usuarios.
- c) realizar un módulo de registro de activos fijos.
- d) realizar el modulo donde se calcula la vida útil de los activos fijos.
- e) realizar el módulo de entradas de almacenes.
- f) realizar el módulo de salidas de almacén.
- g) Realizar modulo para generar reportes de compras.
- h) realizar modulo para generar reporte de ventas
- i) Realizar módulo de informes sobre entradas y salidas de almacenes.
- j) realizar un módulo donde se registre los clientes.
- k) realizar módulo de registro de proveedores.

Estos son los alcances y módulos del portal web con ello se garantizan una gestión eficaz y segura.

1.9 Aportes:

El aporte que realizará el “portal web para control de almacenes y activos fijos caso (laboratorios ESFASA), permitirá controlar los puntos críticos en la cadena de suministro de materia prima desde la recepción y despacho al área de producción. Así como la gestión de inventario todo este estudio se basa en hechos reales, donde se trabaja con una gran variedad de insumos químicos y de empaque, que lleve a establecer mejoras de operación en el sistema de gestión inventario. Por ende, el mejoramiento de toda la cadena productiva para beneficio tanto del empleador como empleados de “LAB. ESFASA S.R.L.”

CAP II

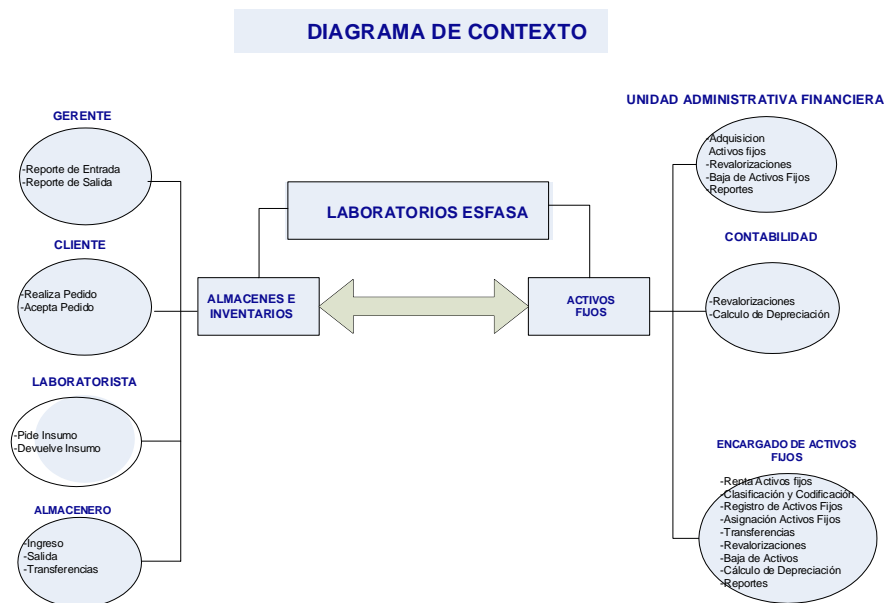
2. MARCO TEORICO

2.1 Generalidades

Dentro de una empresa es de vital importancia la compra y venta de bienes o servicios; de aquí la importancia del manejo de los Almacenes e inventarios, tanto en empresas como en dependencias del gobierno central, instituciones educativas etc. Cada vez más son las empresas, así como diversas instituciones que dedican su esfuerzo a conseguir un buen sistema de información de control de almacenes e inventarios para la cadena de suministro.

Por lo tanto, para lograr un control efectivo de los inventarios es necesario una buena coordinación y una cooperación entre los elementos del sistema. De forma particular, el presente proyecto analiza la situación del almacén de "LABORATORIOS ESFASA S.R.L." que fabrica y distribuye productos farmacéuticos a las diferentes instituciones de salud y también venta en las farmacias de los 9 departamentos de nuestro país.

Fig. 2 1 El Movimiento de la Información



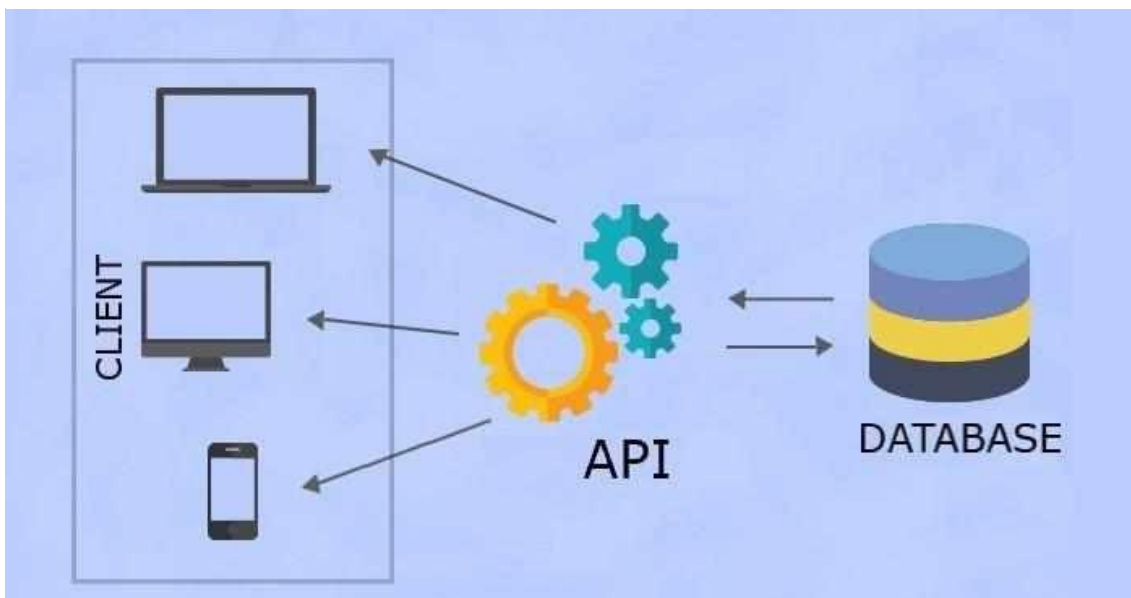
2.2 Portal web.

En los últimos años, se ha visto como la población ha incrementado su uso de Internet para realizar cualquier tipo de acción (búsqueda de datos, compras y transacciones, entre otras.) así viendo el presente, el mundo entero se contagió del covid -19 a causa de ello todos nos pusimos en cuarentena y la tecnología nos ayudó a seguir relacionando con nuestro familiares, para hacer compras y lo más importante para conocer todo lo que ocurre en el mundo en tiempo real, haciendo que las tecnologías web sean muy utilizadas y se tenga en el mercado una alta demanda del desarrollo de las mismas.

Una aplicación web es una herramienta a través de la que un usuario (cliente) realiza una serie de peticiones, de las cuales recibe una respuesta.

Una arquitectura web, basada en API Rest, es de la siguiente manera:

- Fig. 2. 1 Arquitectura Web



Se separa en tres partes importantes:

1.Data base: Donde se almacenan los datos a los que hace referencia una aplicación para su funcionamiento.

2.API: Interfaz para el intercambio de mensajes o datos. Cuenta con una serie de funciones y procedimientos que permite la interacción más sencilla con el cliente abstrayendo la lógica de una aplicación más robusta, brindando respuestas a los clientes.

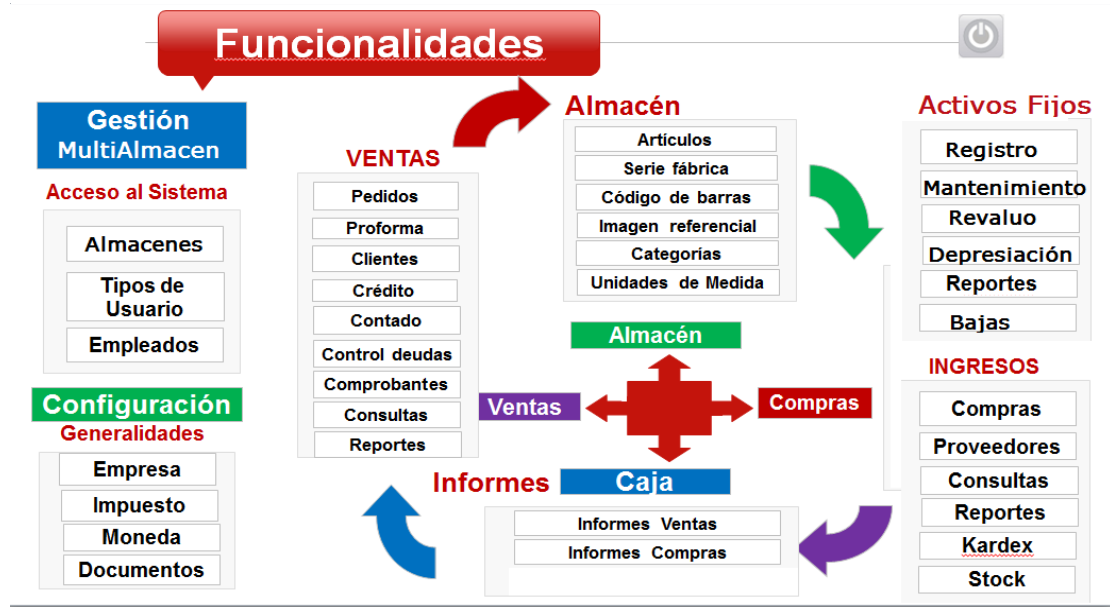
3.Cliente: Navegadores de distintos dispositivos que acceden a la API para hacer peticiones.

El portal web facilitara la búsqueda y localización de información sin salir del portal, y además incentivara el uso recurrente de los servicios.

El portal WEB proporcionara:

- INFORMACION. - buscadores, directorios, noticias y Servicios.
- PARTICIPACION. - Manejan aplicaciones
- COMODIDAD. - Brindan un acceso a la mayor cantidad de información desde un único sitio.

Fig. 2.2 funcionalidad del portal



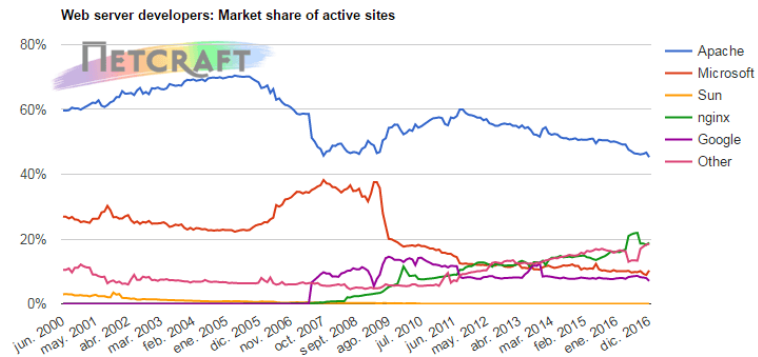
2.2.1 Servidor Web.

Generalmente usamos la palabra servidor para referirnos a las computadoras especialmente preparadas para almacenar sitios web. Pero estrictamente un servidor es el software que se encuentra instalado en dichas computadoras con la cual es posible el funcionamiento de internet.

Específicamente es un programa que se ejecuta continuamente en un computador, manteniéndose a la espera de peticiones de ejecución que le hará un cliente o un usuario de internet. El servidor web se encarga de contestar a estas peticiones de forma adecuada, entregando como resultado una página web o información de todo tipo de acuerdo a los comandos solicitados.

Los servidores web más utilizados según referencia estadística de la compañía NETCRAFT. Ellas salen de comprobar la cantidad de dominios en los que se encuentra presente cada uno de los servidores. Veamos el siguiente gráfico.

Fig. 2 3 Estadística de servidores en el Mundo



Developer	November 2016	Percent	December 2016	Percent	Change
Apache	80,012,251	46.67%	77,011,462	45.27%	-1.41
nginx	31,239,615	18.22%	32,113,723	18.88%	0.65
Microsoft	15,257,724	8.90%	17,554,286	10.32%	1.42
Google	13,607,864	7.94%	12,002,411	7.05%	-0.88

Diferencias entre portal web, sitio, y pagina web.

Antes de entrar en la definición de portal web, hay que aclarar un tema que resulta confuso: los portales, sitios, y páginas web si tiene relación entre ellos, pero no necesariamente son lo mismo.

- Una página web es una fuente de información dentro de la WORLD WIDE WEB elaborada en lenguaje HTML, a la cual se puede acceder por medio de un navegador de internet. Las páginas web puede contener hiperenlaces a otras páginas web, constituyendo así la red de WWW.
- Un sitio web es un espacio en la WWW que contiene un conjunto de páginas, normalmente relacionados con un dominio o subdominio de internet.

- **Un portal web** es un sitio web que funciona con acceso a una amplia gama de recursos y servicios, entre lo que pueden encontrarse foros, buscadores, aplicaciones documentos etc.
- **Tipos de portales web.**- podemos proceder a la clasificación de distintos portales web, entre ellos encontramos:
 - **Portales horizontales.**- También conocidos como portales masivos, están dirigidos a una audiencia amplia, y brindan una gran gama de información y servicios sin necesidad de navegar por varios sitios para encontrarlos.
 - Entre los sitios que clasifican en esta categoría se encuentran Terra, AOL, Alta Vista, UOL, Yahoo, MSN.
 - **Portales verticales.**- Estos sitios provee información y servicios a un público segmentado, ofreciendo contenidos concretos con relación a temas específicos.
 - Entre estos portales podemos mencionar sitios como: Forbes, InfoEmpleo, Portal Música, entre otros.

2.2.2 Apache.

El servidor apache es un servidor web HTTP de código abierto, para plataformas UNIX (BSD, GNU/Linux etc.) Microsoft Windows, Macintosh y otras. Su nombre se debe a que alguien quería que tuviese la connotación de algo que es firme y enérgico, pero o agresivo, por ello se tomó el nombre de la última tribu apache.

El servidor apache es desarrollado y mantenido por una comunidad de usuarios bajo la supervisión del apache software foundation dentro del proyecto HTTP server (httpd).

Presenta entre otras características altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de interfaz gráfica que ayude en su configuración.

Desde 1996, es el servidor HTTP más usado. Jugó un papel fundamental en el desarrollo fundamental de la World Wide Web y alcanzó su máxima cuota de mercado en 2005 siendo el servidor empleado en el 70% de los sitios web en el mundo, sin embargo, ha sufrido un descenso en su cuota de mercado en los últimos años. En 2009 se convirtió en el primer servidor web que alojó más de 100 millones de sitios web.

La mayoría de las vulnerabilidades de la seguridad descubiertas y resueltas tan sólo pueden ser aprovechadas por usuarios locales y no remotamente. Sin embargo, algunas se pueden accionar remotamente en ciertas situaciones, o explotar por los usuarios locales malévolos en las disposiciones de recibimiento compartidas que utilizan PHP como módulo de Apache.

Para la implementación del proyecto, utilizaremos este servidor por ser de código abierto. Ya que la mayoría de las empresas prefieren de este servidor por ser liviano y fácil de configurar.

2.3.- Descripción del sistema.

2.3.1. Almacenes e inventarios.

La principal función comercial e industrial es la salvaguarda y abastecimiento de materiales y productos y del adecuado suministro de los equipos, materiales o artículos de la calidad requerida, en las cantidades solicitadas y el tiempo oportuno.

La administración moderna de cualquier actividad, el buen manejo de un almacén es un medio para lograr eficiencia en los procesos de recepción, control, consolidación y envío de productos disminuyendo mermas y deterioro generando un valor agregado en los insumos, este concepto ahuyenta la idea de que un almacén es un mal necesario cuya función principal es la de agregar gastos y disminuir utilidades.

2.3.2.- FUNCIONAMIENTO Y TIPOS. -

La administración de los almacenes es una de las operaciones muy importantes para una empresa y una compañía, ya que el resultado se refleja en los estados financieros (causados por los inventarios) además, es una función, en el plan general de la operación de la empresa, donde cada actividad se relaciona con un patrón calculado para producir una acción conjunta y dirigida a una meta.

Es necesario conocer los objetivos de la empresa para planear los almacenes y dirigir sus actividades. El responsable de los almacenes debe recibir la información precisa de tales objetivos para que el, y su personal orienten sus esfuerzos hacia ellos y para que puedan ser delineadas todas las funciones dentro el almacén.

La manera de administrar y de organizar el almacén depende de varios factores, tales como el tamaño y plan de organización de la compañía, el grado de centralización deseado, la variedad de productos almacenados y la flexibilidad de los equipos.

Las siguientes funciones son comunes en todo tipo de almacenes:

- 1.- recepción de materiales o productos en el almacén
- 2.- registro de salidas en el almacén

3.- almacenamiento de materiales

4.- mantenimiento del equipo y del almacén

5.- coordinación del almacén con otros departamentos

Todo almacén puede considerarse redituable para un negocio según el apoyo que preste las funciones productoras de utilidad. Producción y ventas, es importante hacer hincapié en que lo almacenado debe tener un movimiento rápido de entradas y salidas manteniendo organizado un esquema PEPS (primeras entradas primeras salidas).

Los costos de almacén pueden desglosarse como siguen:

- Interés sobre el capital inmovilizado representando por el valor de los inventarios
- Gastos del seguro
- Espacio ocupado al precio de la localidad por metro cuadrado
- Amortización del edificio y del equipo de almacenamiento y manejo de materiales
- Devaluación de la mercancía
- Costos del personal
- Mantenimiento de instalaciones y equipo.

La mercancía que resguarda, custodia, controla y abastece un almacén puede ser de los siguientes tipos. Materias primas y partes componentes

- Materiales auxiliares producto en proceso
- Productos terminados herramientas
- Refacciones
- Material de desperdicio
- Materiales obsoletos
- Devoluciones.

El negocio puede ser una empresa manufacturera, distribuidora, almacenadora o una tienda de productos masivos.

2.3.3.- ORGANIZACIÓN

En términos generales todo almacén debe estar integrado por una gerencia de almacén y las siguientes subgerencias administrativas, obviamente dependiendo del tamaño del mismo puede organizarse jerárquicamente en áreas de supervisiones o direcciones. Esto determinara la empresa a su propio uso y criterio las siguientes áreas administrativas son:

- 1.- Recepción
- 2.- Manejo y control
- 3.- Despacho o envío
- 4.- Registro
- 5.- Servicio de transporte
- 6.- Servicios complementarios

El objetivo de la gerencia es planear, organizar, y dirigir y controlar las operaciones de almacenamiento, aprovechando la capacidad del personal, buscando mejores resultados con un mínimo de esfuerzo, tiempo, espacio y materiales. Los objetivos de las otras áreas son:

a.- RECEPCION

Planear, dirigir, y controlar la entrada física de los bienes adquiridos por la empresa. Buscar la mayor eficacia de los métodos de descarga, inspección y verificación. Un proceso de recepción efectivo y rápido solo puede lograrse si existe un sistema racional de trabajo.

La revisión, modificación o ajuste de sistemas de recepción son responsabilidad de cada empresa, ya que deben de ajustarse a la naturaleza de cada almacén.

b.- MANEJO Y CONTROL Controlar, organizar y dirigir operaciones para tener el mejor manejo y control de la mercancía dentro del almacén, para lograr la mayor protección y conservación de los bienes bajo su cuidado. Asegurar su fácil rápida identificación, así como el optimizar el espacio.

Los métodos y procedimientos varían según la cantidad y características de los bienes, el espacio disponible, el tipo del equipo, y el número y calificación del personal.

2.4 Metodología de desarrollo

2.4.1 Ingeniería de requerimiento

La ingeniería de requerimiento es el proceso de recopilar analizar y verificar las necesidades del cliente. La meta de la ingeniería de requerimientos es entregar una descripción (especificación) de requerimientos de software correctos y completos. La ingeniería de requerimientos apunta a mejorar la forma en que comprendemos y definimos los sistemas de software complejos. Existen varias definiciones de requerimientos, entre las cuales podemos citar las siguientes:

Los requerimientos fueron definidos por IEEE como (IEEE90):

- 1.- Condición o capacidad requerida por el usuario para resolver un problema o alcanzar un objetivo.
- 2.- Condición o capacidad que debe satisfacer o poseer un sistema o una componente de un sistema para satisfacer un contrato, un estándar, o una especificación u otro documento formalmente impuesto.

3.- representación documentada de una condición o capacidad de un proyecto.

SEGÚN ZAVE.

Rama de la ingeniería del software que trata con el establecimiento de los objetivos, funciones y restricciones de los sistemas de software.

Asimismo, se ocupa de la relación entre estos factores con el objeto de establecer especificaciones precisas.

SEGÚN BOEHM

Ingeniería de requerimientos es la disciplina para desarrollar una especificación completa consistente y no ambigua, la cual servirá como base para acuerdos comunes entre todas las partes involucradas y donde se describen las funciones que realizara el sistema.

SEGÚN LOUCOPOULOS

Trabajo sistemático de desarrollo de requisitos, a través de un proceso iterativo y cooperativo de análisis del problema, documentando los resultados en una variedad de formatos y probando la exactitud del conocimiento adquirido.

SEGÚN LEITE

Como resultado del análisis se desarrolla la especificación de requerimientos de software. La revisión esencial para asegurar que el realizador del software y el cliente tengan la misma percepción del sistema.

Una vez finalizado el proyecto de desarrollo de sistema, y contado con un buen análisis de requerimientos, podremos evaluar la calidad del producto terminado.

Definición1.- la (Ingeniería de Requerimientos) es el proceso para establecer los servicios que el sistema debería de proveer y las restricciones bajo las cuales debería operar y ser desarrollado, se llama ingeniería de requerimientos [Som95].

La ingeniería de requerimientos es por tanto una actividad esencialmente de interacción con los interesados² en el sistema. Es incorrecto y extremadamente riesgoso que los ingenieros de software, establezcan los requerimientos de sistema (a menos que la estrategia de la empresa sea forzar a los potenciales clientes adecuarse al sistema tal cual esta, como es el caso de los programas de uso masivo) sin haber mantenido numerosas reuniones con diferentes representantes del cliente, sin haberles mostrado prototipos del sistema, sin haberles hecho una y otra vez las mismas preguntas, sin haber comprendido el negocio del cliente.

Pero ¿qué es un requerimiento? Si bien no es vital dar una definición formal este concepto, creemos que las siguientes pueden ayudar el tema.

Definicion2.- El (requerimiento o Requisito). Un requerimiento es una característica del sistema o una descripción de algo que el sistema es capaz de hacer con el objeto de satisfacer el propósito del sistema, lo que ha sido apropiadamente documentado y validado por el solicitante.

Los requerimientos tratan exclusivamente sobre los fenómenos del dominio de aplicación y no sobre la máquina que los implementa.

Además, es importante resaltar algunos puntos de las definiciones anteriores.

² Interesados es la traducción del término inglés stakcholders.

No es lo mismo un pedido o deseo de un usuario o cliente que un requerimiento. No todos los pedidos o deseos de un usuario o cliente se convierten necesariamente en requerimientos, pero si todos los requerimientos se originan en un pedido o deseo de un usuario o cliente. Para que un pedido o deseo de un usuario o cliente se convierta en requerimiento, este debe ser documentado apropiadamente y el solicitante debe validarlo.

Los ingenieros del software no originan requerimientos; su función es convertir pedidos de los usuarios o clientes en requerimientos. Luego deben proveer un sistema que los implemente.

Los requerimientos siempre están en el entorno del sistema que ese va desarrollar, nunca dentro de él. Por consiguiente, palabras tales como tabla, XML, clase, subrutina, etc. No pueden formar parte de un requerimiento.

A continuación, enunciaremos 2 grandes categorías de requerimientos:

Definición3.- (Requerimientos Funcionales). Un requerimiento es funcional cuando describe una interacción entre el sistema y su medio ambiente. Los requerimientos funcionales describen como debe comportarse el sistema ante un estímulo.

Si bien los requerimientos funcionales. O la función o funcionalidad del sistema, son esenciales para poder construir el sistema correcto, existen ciertas cualidades o atributos que los usuarios esperan del sistema que no tiene una relación simple con la funcionalidad que desean. A estas cualidades o atributos se los llama requerimientos funcionales.

Definición 4.- (Requerimientos no funcionales o atributos de calidad o cualidades del sistema). Un requerimiento no funcional es una restricción sobre el sistema o su proceso de producción.

Un ejemplo de requerimiento no funcional:

El sistema debe ejecutarse sobre Linux Ubuntu 16.4 LTS

Lamentablemente, es un problema recurrente en el desarrollo de sistemas de software que los requerimientos funcionales no solo ocupen la primera línea sino la única en el proceso de desarrollo. Esto es la causa de innumerables modificaciones extensivas puesto que la mayor parte de las veces los sistemas de software no se modifican porque sean funcionalmente incorrectos, sino porque son difíciles de mantener, portar, escalar, o son muy lentos o vulnerables. Claramente es necesario introducir muchos cambios más o menos importantes en diferentes partes del sistema. Conviene mencionar que no siempre es simple determinar si un requerimiento es funcional o no funcional.

2.4.2 El límite de la ingeniería de requerimientos

Existe un límite fundamental, y sobre todo inherente a la construcción del software, que afecta a la Ingeniería De requerimientos. Este límite se puede expresar de la siguiente forma: es virtualmente imposible iniciar la construcción de un sistema de software teniendo la lista completa y consistente de todos los requerimientos, en un tiempo razonable.

Entre las razones que justifican la hipótesis tenemos las siguientes:

Usualmente se requiere que los grandes sistemas de software se constituyan una superación del estado de arte en cierto dominio de la aplicación.

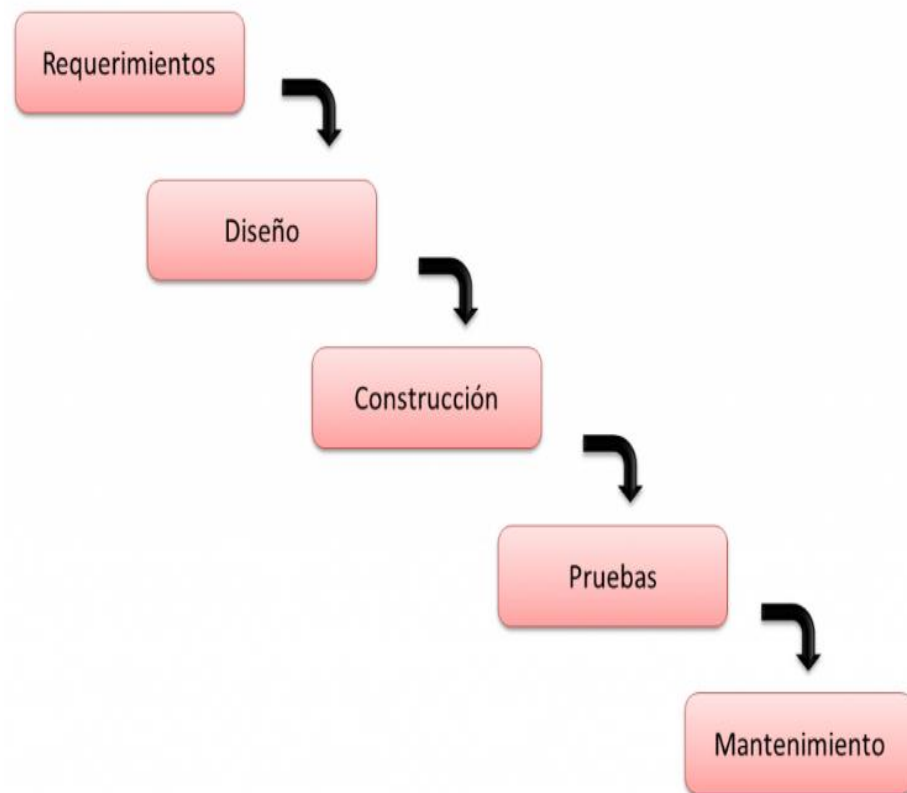
En general la gente paga por el sistema y quienes lo usan no son los mismos, lo que genera conflictos entre lo que se puede pagar y lo que se necesita. La introducción de un nuevo sistema de software importante en una organización es

tan disruptiva³ que suele producir cambios igualmente importantes en la organización lo que, a su vez genera nuevos requerimientos sobre el sistema.

En opinión este límite real y propio de la naturaleza del software y de la tecnología informática en general. Es tan irreal pensar que existe alguna forma.

De tener todos los requerimientos en un tiempo razonable, como peligroso no pensar en los requerimientos futuros. Esta es una regla de juego que juegan los ingenieros del software y por lo tanto hay que crear y aplicar, principios, metodologías, técnicas y herramientas para ganar el juego sin ser penalizados por esa regla.

Fig. 2 4 desarrollo en cascada



³ Disruptivo es un término que procede del inglés disruptive y que se utiliza para nombrar a aquello que produce una ruptura brusca. Por lo general el término se utiliza en un sentido simbólico, en referencia a algo que genera un cambio muy importante o determinante (sin importar si dicho cambio tiene un correlato físico).

Tabla 2 1 análisis de requerimientos

COHESIVO	El requerimiento habla una única cosa
COMPLETO	El requerimiento este enunciado en único lugar y no falta información
CONSISTENTE	El requerimiento no contradice ningún otro requerimiento y es consistente con cualquier otra información relevante
CORRECTO	El requerimiento apunta soportar el negocio detrás del sistema como fue enunciado por el cliente.
ACTUALIZADO	El requerimiento no se volvió obsoleto por el paso del tiempo.
OBSERVABLE EXTERNAMENTE	El requerimiento habla de una característica del sistema que es observable o experimentada por el usuario. Los requerimientos que hablan de arquitectura, diseño o implementación no son, en realidad requerimientos.
	El requerimiento puede implementarse dentro de los límites del proyecto.
NO AMBIGUO	El requerimiento este enunciado de forma clara y concisa utilizando el vocabulario del dominio de aplicación, habiéndose definido con precisión todos los términos técnicos y siglas. El enunciado tiene una única interpretación. Sustantivos, adjetivos preposiciones, verbos y frases subjetivas que resultan vagas o poco claras deben ser evitados. Se prohíbe el uso de oraciones negativas o compuestas.
VALIDABLE	La implementación del requerimiento puede validarse a través de inspección, análisis, demostración o prueba.

- 1 El requerimiento no necesita validación
- 2 El requerimiento debe ser validado por al menos un interesado más: y
- 3 El requerimiento debe ser validado por todos los interesados

2.4.3 Especificación de los requerimientos

En pocas palabras en esta etapa la lista consolidada de requerimientos se escribe desde la perspectiva del desarrollador. Esta etapa puede incluirse en las fases denominadas MODELO ESTRUCTURAL o MODELO FUNCIONAL en el modelo del ciclo de la cascada (fig. 1) como veremos posteriormente.

Los productos de la Ingeniería de requerimientos son:

Prototipos. Nunca se enfatizará lo suficiente la importancia crucial de usar inteligentemente prototipos para obtener requerimientos funcionales. En opinión debe ser de la forma de interacción casi excluyente entre los ingenieros de requerimientos y el cliente, durante esta fase.

Nos concretamente a los requerimientos desechables un prototipo desechable es un programa que imita al menos algunos de los requerimientos funcionales pero que luego de que los requerimientos se han validado, el programa se descarta.

Escenarios. Es muy complejo y costoso, y muchas veces imposible, prototipar los atributos de calidad de un sistema (disponibilidad, seguridad, testeabilidad, etc.). En consecuencia, en lugar de prototipos, se utilizan escenarios de atributos

De calidad como técnica efectiva para determinar con suficiente precisión cual es el significado que el usuario que le asigna a las cualidades del sistema.

2.4.4 Lista consolidada de requerimientos. –

Es un documento que consolida, organiza y estructura la información recolectada durante la etapa de obtención de requerimientos.

El vocabulario y las notaciones que se utilicen para escribir esta lista deben ser estrictamente las del cliente o las del dominio de aplicación. Esencialmente es un documento para el cliente. Es decir, en general, este documento no debería incluir términos de lado técnico del desarrollador, si no en forma natural que pueda ser entendido por el cliente. El documento se genera fundamentalmente a partir de los prototipos, escenarios de atributos de calidad entrevistas y observaciones sobre el entorno del sistema que han efectuado los ingenieros de requerimientos.

Este documento debe incluir al menos los siguientes contenidos

2.5 Metodología ágil

(Rina Elizabeth López Menéndez de Jiménez 8/01/2015) Hace varios años las empresas que desarrollan software creían que con un modelado eficiente y con las herramientas case que había en el mercado eran suficientes para dicho cometido, pero se dieron cuenta que no es la verdadera realidad, ya que para mejor desarrollo de un software de calidad que pueda dejar al cliente satisfecho en sus requerimientos. Debido a esto hoy en día creció un gran interés por las metodologías ágiles para agilizar el tiempo de desarrollo y garantizar el uso eficiente de los recursos.

Las metodologías ágiles surgen de una iniciativa en el año 2001, tras una reunión celebrada en Utah, EEUU con la participación de un grupo de 17 expertos en el área de desarrollo de software, los cuales manifestaron la importancia que el equipo

desarrollador respondiera de forma oportuna a los cambios que puedan surgir a lo largo de la ejecución del proyecto.

2.5.1.- SCRUM

Dentro de los frameworks para desarrollar software, scrum es en los últimos tiempos el standart que muchas empresas de tecnología están utilizando para acortar tiempos de desarrollo, y entregar un producto de calidad.

Aproximadamente en 1986 Hirotaka Takeuchi e Ikujiro Nonaka describieron una nueva forma para el desarrollo de productos comerciales, que incrementan la rapidez y flexibilidad en el proceso. Ellos comparan este nuevo método, en el cual las fases se traslapan de manera intensa y el proceso completo es realizado por un equipo con funciones diversas, como en el rugby, donde el equipo entero actúa como un solo hombre para intentar llegar al otro campo, pasando el balón de uno a otro. Estos casos de estudio se originan de las industrias automovilísticas, así como la fabricación de máquinas fotográficas, computadoras e impresoras.

En 1991 Peter Degrace y Leslie Stahl en su libro *Wicked problems, Righteous Solutions*, se refirieron a esta aproximación como Scrum, un término propio del rugby mencionado en el artículo por Takeuchi y Nonaka.

Scrum es un método ágil de desarrollo de software más utilizado del mundo. Se presentó en 1995 y hoy en día.

Más del 70% de los equipos de desarrollo de software en el mundo lo usan.

Scrum es una metodología ágil y flexible para gestionar el desarrollo de software cuyo principal objetivo es maximizar el retorno de la inversión para la empresa

(ROI). Se basa en construir primero la funcionalidad de mayor valor para el cliente y en los principios de inspección continua, adaptación autogestión e innovación.

Con la metodología Scrum el cliente se entusiasma y se compromete con el proyecto dado que lo ve crecer iteración a iteración. Así mismo le permite en cualquier momento realinear el software con los objetivos de negocio de su empresa, ya puede introducir cambios funcionales o de prioridad en el inicio de cada nueva iteración sin ningún problema.

Esta metodología de trabajo promueve la innovación, motivación y compromiso del equipo que forma parte del proyecto, por lo que los profesionales encuentran un ámbito propicio para desarrollar sus capacidades.

En la actualidad, todo proyecto debe entregarse lo más rápido posible y con una calidad impecable y el desarrollo de software no se queda atrás. Es común que los clientes (internos o externos) soliciten aplicaciones cada vez más complejas, tanto en su desarrollo como en su análisis. Como resultado, son muchas las ocasiones en las que no se llegan a cumplir las necesidades de los clientes. Pero para solucionar este tipo de problemas, existen soluciones con scrum.

Scrum es un framework adaptable, iterativo, rápido, flexible y eficaz. Que está *diseñado* para entregar valor al cliente durante todo el desarrollo del proyecto. El objetivo primordial es satisfacer las necesidades del cliente a través de un entorno de transparencia en la comunicación, responsabilidad colectiva y progreso continuo.

Los roles con Scrum, que son dos roles:

Roles centrales estos roles son aquellos que su participación es indispensable para la realización del proyecto, están comprometidos con el proyecto y son responsables del éxito de cada sprint y del proyecto en general. Y son:

Product owner

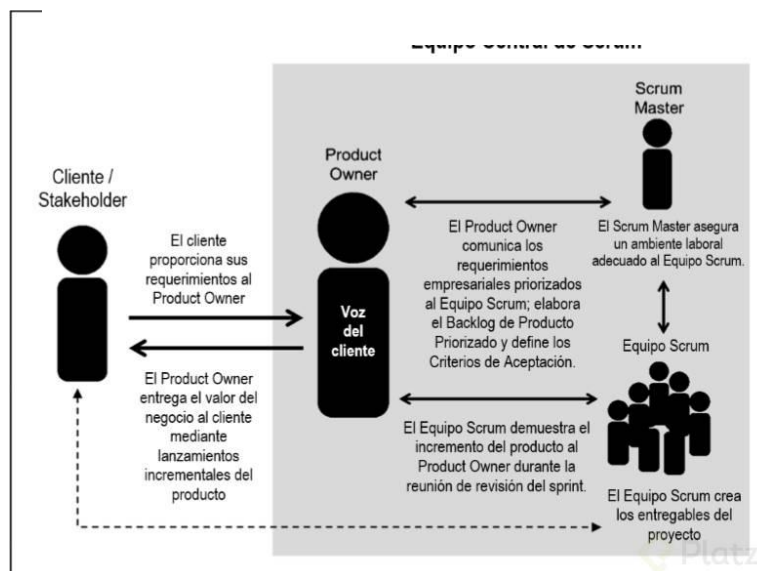
Scrum master

Equipo Scrum

Los roles no centrales son aquellos cuya participación en el proyecto es importante siempre identificar los individuos de esta categoría y mantenerlos siempre presentes, en cualquier momento su rol puede ser decisivo para el proyecto (por ejemplo, si es un sponsor) y los cuales son:

- Stakeholders
- Cliente
- Usuario
- Patrocinador (sponsor)
- Vendedores
- Scrum guidance body

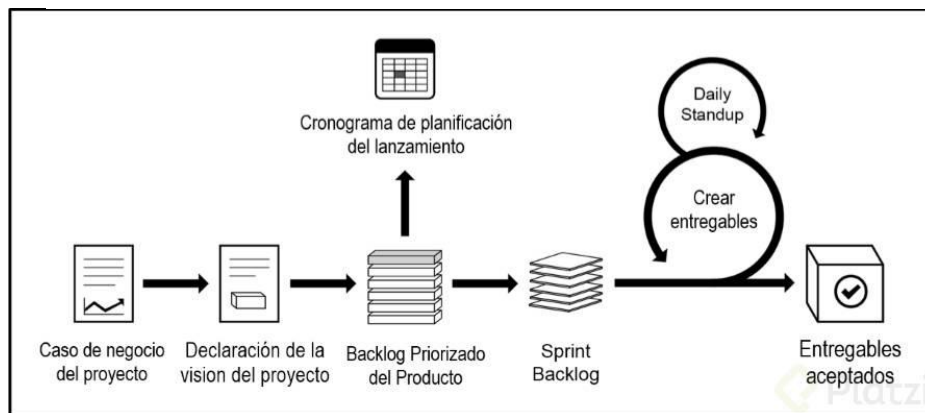
Fig. 2.5 equipo central de Scrum



2.5.2.-Los procesos de scrum

Los procesos de Scrum están enmarcados en cajas de tiempo que son uno de los principios del marco del trabajo y es lo que nos permite manejar eficazmente la planeación y ejecución del proyecto.

Fig. 2 6 Proceso de Scrum



2.5.3.- Inicio del ciclo

El ciclo se inicia con la reunión de los interesados (stakeholders) en la que se crea la descripción de la visión del proyecto. Luego el product owner crea la lista priorizada del producto (prioritized product backlog) que contiene la lista de los requerimientos en orden de prioridad para el negocio y el proyecto en formas de historia de usuario.

2.5.4.-Sprint

Es la unidad básica de trabajo en equipo Scrum. Esta es la característica principal que marca la diferencia entre Scrum y otros modelos para el desarrollo ágil. Es una simple iteración llevada a cabo por los miembros del equipo. Un equipo puede completar varios sprints durante el desarrollo del proyecto. Un sprint inicia con un equipo que se compromete a realizar el trabajo y finaliza con la demostración del entregable. El tiempo mínimo para un sprint es de una semana y el máximo de 4 semanas. Dentro del desarrollo de un sprint se llevan a cabo ciertos eventos, que reciben el nombre de **** Scrum events o eventos Scrum****

2.5.5.-Planeamiento del sprint/ sprint planning

Todos los involucrados en el equipo se reúnen para planificar el sprint. Durante este evento se decide que requerimiento o tareas se asignara a cada uno de los elementos del equipo. Cada integrante deberá asignar el tiempo que crea prudente llevar a cabo sus requerimientos. De esta manera se define el tiempo de duración del sprint.

2.5.6. - Reunion de equipo scrum /scrum team meeting

Estas reuniones se deben realizar diariamente con un máximo de 15 minutos. Siempre en el mismo horario y lugar. Ellas, cada miembro del equipo deberá responder tres simples preguntas:

¿Qué hiciste ayer?

¿Qué tienes planeado para hoy?

¿Qué obstáculos encontraste en el camino?

Estas reuniones sirven para que todos los miembros del equipo se apoyen entre ellos. Si alguno de ellos tiene inconveniente que tome más tiempo del asignado en resolverse; este debe tratarse a fondo en una reunión enfocada en buscar la mejor solución para ello.

2.5.7.- Refinamiento del backlog/baklog refinement

El product owner revisa cada uno de los elementos dentro del product backlog con el fin de esclarecer cualquier duda que pueda surgir por parte del equipo de desarrolladores. También sirve para volver a estimar el tiempo y esfuerzo dedicado a cada uno de los requerimientos.

2.5.8.- Revision del sprint /sprint review

Los miembros del equipo y los clientes se reúnen para mostrar el trabajo de desarrollo de software que se ha completado. Se hace una demostración de todos los requerimientos finalizados dentro del sprint. En este punto no es necesario que todos los miembros del equipo hablen. Pueden estar presentes pero la presentación estará a cargo del Scrum Master y el Product owner

2.5.9.- Retrospectiva del sprint/restrospective

En esta reunión o evento, el product owner se reúne con todo el equipo de trabajo y su Scrum Master para hablar sobre lo ocurrido durante el sprint. Los puntos principales a tratar en este evento o reunión son:

Que hizo mal durante el sprint para poder mejorar al próximo

Que hizo bien para seguir en la misma actitud del éxito

Que inconvenientes se encontraron y no permitieron poder avanzar como se tenía planificado.

2.5.10.-Herramientas scrum

Para poder definir las respuestas a estas preguntas, se hace uso de ciertas herramientas que Scrum nos provee. Y estas son:

Backlog de producto / product backlog

Esto puede referirse a todo el elemento que sea parte del proyecto: puede ser bug, o una referencia o parte o requerimiento. Brindan información muy general del proyecto y muchas veces no son tomados como requerimientos oficiales.

Historias del usuario/ users stories

Es un elemento especial del producto backlog. Son llamados historias porque en ellos se proporciona información sobre cómo debe ser el comportamiento del requerimiento que se está trabajando, de igual manera, proporciona información directa del cliente en caso de existir algún cambio. Generalmente estos si son tomados como requerimientos oficiales.

Backlog del sprint / sprint backlog

Es el conjunto de elementos tomados del producto backlog que fueron priorizados, medidos y aceptados en las reuniones de sprint planning. Estos en conjunto con sus respectivos user stories, forman oficialmente los requerimientos a elaborar en cada uno del sprint que tendrá el proyecto.

Panel de tareas /the taskboard

El panel de tareas muestra todas y cada una de las tareas que tiene asignadas a cada uno de los miembros del equipo. Esta tabla se divide en tres columnas que representan el estado de la tarea:

1.- Por hacer

2.- Haciendo

3.- Terminado

Al inicio del sprint todas en la primera columna. Al momento de pasar una tarea a la columna número dos, indicara el Scrum Master y al product ownwer que está haciendo cada miembro del equipo y cuánto tiempo lleva trabajando en dicha tarea. Al finalizar la tarea, esta debe cambiarse a la última columna. Esto quiere decir que está listo para que haga las pruebas necesarias.

Definicion de listo/ definition of done

Todo equipo eficaz y ágil tiene ciertos acuerdos que deben de cumplirse antes de dar por finalizado un proyecto. Y son estos:

- Todas las tareas están completas
- Revisión de código / code reviewed
- Pruebas realizadas a cada elemento desarrollado
- Revisión por parte de los clientes (que cumpla necesidades)
- La revisión de las condiciones de aceptación por parte del product ownwer

Estas herramientas son útiles no solo durante un Sprint; sino que ayudan a lo largo del proyecto.

Ya que ayudan al equipo a entender porque lo que están haciendo. Son visibles para cada uno de los miembros del equipo y para las personas que están fuera también. Scrum no es más que una metodología que puede ser aplicable a cualquier tipo de proyecto. Aplicarlo requiere de un cambio de cultura laboral por cada uno de los miembros que componen dicho equipo. Pero cuando el resultado sea hacer bien

los proyectos en el menor tiempo posible y al menor costo, todo el sacrificio habrá valido la pena.

Un sprint es una de las cajas de tiempo de Scrum, tiene una duración de 1 a 6 semanas en las que el equipo de Scrum trabaja en la creación de los entregables, el sprint inicia con la reunión de planeación de Sprint (Spint Planning meeting) esta reunión tiene una duración de 8 horas para un sprint de 4 semanas, tiempo en el que analizan las historias de usuario y de acuerdo a la prioridad se incluyen en el sprint Backlog que es el listado de tareas que se van a implementar durante el sprint que inicia.

Daily standup

Durante el sprint se realizan reuniones diarias llamadas daily Standup, durante 15 minutos máximos el equipo scrum se reúne para discutir el progreso diario y si hay impedimentos. Al final sprint se realiza una reunión de revisión de sprint (sprint review Meeting) en la que se hace una demostración de los entregables desarrollados al product own⁴wers y a los stakeholders relevantes.

Entregables

Los entregables cumplen con los criterios de aceptación definidos el product ownwers los acepta y los reinicia el ciclo, una de las ventajas de usar Scrum es que siempre se está pensando en mejorar por eso es importante siempre al final de cada sprint realizar una reunión de retrospectiva del sprint en la que se analizan los problemas presentados y las lecciones aprendidas.

⁴ Product ownwer = dueño del producto es la voz del cliente y el responsable de desarrollar, mantener y priorizar las tareas del backlog

2.6. UWE (UML BASE WEB ENGINEERING)

(G. Maigua, E. López, Buenas prácticas en la dirección y gestión de proyectos informáticos, Argentina: Editorial de la Universidad Tecnológica Nacional – edUTecNe, 2012, p.13.)

UWE está basado en estándares de la OMG como UML, Model Driven Architecture de OMG (MDA), Object Constraint Language (OCL) y eXtensible Markup Language (XML), asegurando su seguimiento mediante guías y especificaciones para el uso de tecnologías orientadas a objetos.

El principal objetivo del enfoque UWE es proporcionar: un lenguaje de modelado específico del dominio basado en UML; una metodología dirigida por modelos; herramientas de soporte para el diseño sistemático; y herramientas de soporte para la generación semiautomática de Aplicaciones Web.

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

2.6.1 Modelos de UWE

El método UWE consiste en la construcción de seis modelos de análisis y diseño. Dicha construcción se realiza dentro del marco de un proceso de diseño iterativo e incremental. Las actividades de modelado abarcan: el análisis de requerimientos, diseño conceptual, modelo de usuario, diseño de la navegación, de la presentación y diseño de la adaptación.

Los principales artefactos que produce el método de diseño de UWE son los siguientes:

Un Modelo de Requerimientos que captura los requerimientos del sistema.

Un Modelo Conceptual para el contenido (modelo de contenido).

Un Modelo de Usuario.

UN Modelo de Navegación que comprende la estructura de la navegación.

Un Modelo de Presentación que abarca modelos estáticos y dinámicos (modelo de estructura de la presentación, modelo del flujo de la presentación, modelo de interface abstracta de usuario, y modelo de ciclo de vida del objeto).

Un modelo de adaptación.

El análisis proporciona una visión general del sistema, el objetivo es conseguir una comprensión más precisa de los requisitos refinándolos y estructurándolos para lograr una descripción de los mismos, que sea fácil de mantener y que ayude a estructurar el sistema entero. El estudio del análisis es de mayor importancia durante la fase de elaboración, comprendiendo las siguientes tareas: Diagramas de caso de uso, diagrama de secuencias y diagrama de comunicación.

Modelado por casos de uso

Representa la funcionalidad que ofrece el sistema en lo que refiere a su interacción externa.

Un Diagrama de casos de Uso se emplea para visualizar el comportamiento del sistema una parte del o de una sola clase y como se relaciona con su entorno, en este nivel no se distingue los procesos manuales de los automatizados.

Descripción de los actores del negocio

Representa terceros fuera del sistema desarrollado, identificando de esta manera el entorno externo del sistema. Una clase o una persona externa.

Análisis de requerimientos del sistema

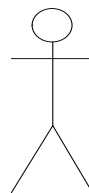
En el Proceso de desarrollo de un sistema hay que identificar los requerimientos que el sistema debe cumplir para satisfacer las necesidades de los usuarios finales y lograr un acuerdo entre el equipo de desarrollo y el cliente,

Los requerimientos son todas aquellas funciones que deberá realizar el producto software, las mismas deben estar diferenciadas entre aquellas que son funciones evidentes para el usuario de aquellas que son ocultas, se estructuran de forma natural mediante: diagramas de casos de uso de alto nivel y casos de uso expandido.

Diagramas de caso de alto nivel

El diagrama de casos de uso de alto nivel se representa como una caja rectangular con el nombre en su interior. Los casos de uso están en el interior de la caja del sistema y los actores fuera y cada actor está unida a los casos de uso en los que participa mediante una relación

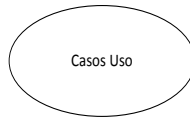
Actor. - Es un rol que tiene un usuario o labor que representa una persona con respecto al sistema, un actor puede tener varios roles y un usuario concreto del sistema es una instancia de un actor.



Representación Gráfica de un Actor

Fuente:[LARMAN 1999]

Casos de uso. - Fragmentos de funcionalidad que el sistema ofrece para aportar un resultado de valor para sus actores especifica una secuencia de acciones que el sistema puede llevar a cabo interactuando con sus actores.

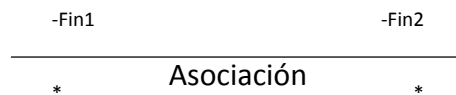


Representación Gráfica de un Caso de Uso

Fuente:[LARMAN 1999]

Relaciones entre casos de usos. -

Asociación, es un tipo de relación más básica que indica la invocación desde un actor o caso de uso a otra función (caso de uso). Dicha relación se denota con una flecha simple.



Generalización, cumple dos funciones, la de inclusión cuando un actor utiliza el caso de uso y la de herencia cuando un caso de uso es similar a otros en sus características.



Casos de uso expandidos

Describe los casos de uso de mayor detalle enumerando paso a paso los eventos que se presenta durante una ocurrencia típica del caso de uso. El formato de la ficha de la descripción del caso de uso expandido es el siguiente:

Tabla 1 tabla de entidades

Casos de Uso	(Nombre del caso de uso)
Actores	(Usuarios involucrados con el caso de uso)
Propósito	(Función del caso de uso)
Resumen	(Narración breve de la funcionalidad del caso de uso y su interacción con los actores involucrados)
Tipo	(Tipo de caso de uso: Ej. Primario, secundario, abstracto)
Referencias cruzadas	(Casos de uso o funciones del sistema relacionados con el caso de uso en estudio)

2.7 Metodología de proceso.

2.7.1 Principios del modelo

SCRUM como uno de los marcos de trabajo más populares, tiene como base 6 principios que orientan la gestión de un proyecto SCRUM a lo largo de todas sus fases. Los principios de SCRUM no son negociables porque su propósito es asegurar la implementación efectiva del marco de trabajo. Lo cual se traduce en confianza y transparencia en la gestión de proyecto.

Son 6 principios básicos que rigen las actividades de cada proceso:

1. Control empírico de procesos
2. Auto-Organización
3. Colaboración
4. Priorización basada en valor
5. Bloque de tiempo asignado

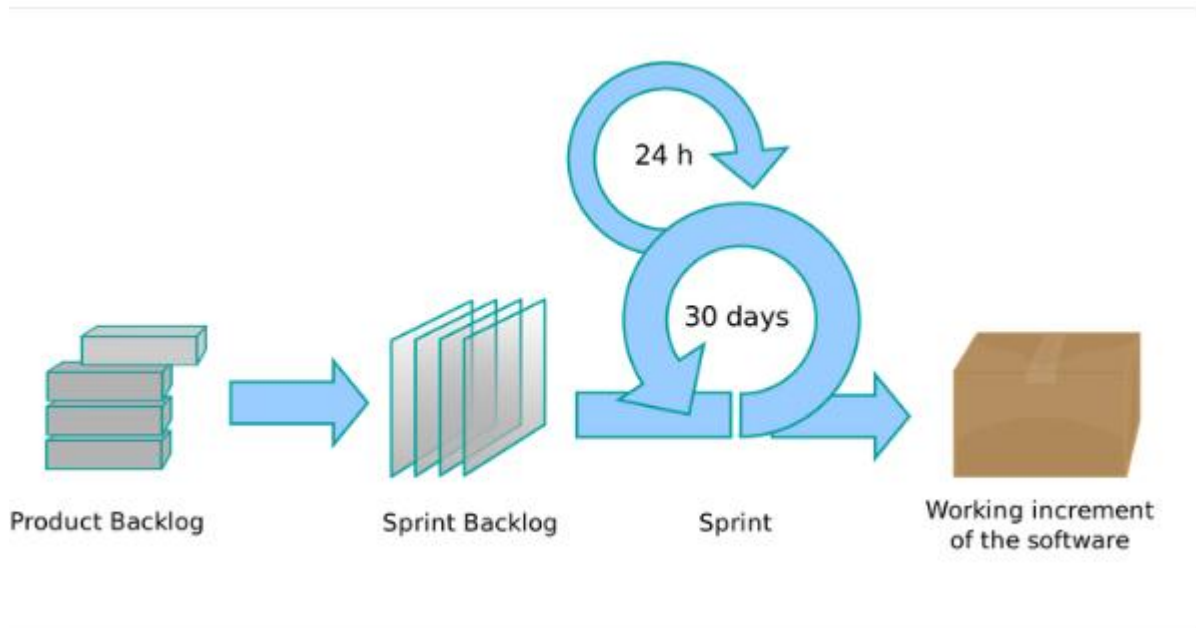
6. Desarrollo Interactivo

2.7.2 Característica del modelo Scrum

La metodología se basa en:

- El desarrollo incremental de los requisitos del proyecto en bloque temporales cortos y fijos
- Se da prioridad a lo que tiene más valor para el cliente
- El equipo se sincroniza diariamente y se realizan las adaptaciones necesarias
- Tras cada iteración (un mes o menos entre cada una) se muestra al cliente el resultado real obtenido, para que este tome las decisiones necesarias en relación a lo observado
- Se le da la autoridad necesaria al equipo para poder cumplir los requisitos
- Fijar tiempos máximos para lograr objetivos
- Equipos pequeños (de 5, 9 personas cada uno)

Fig. 2 7 fase de la Metodología Scrum



2.7.3 Fases de la metodología scrum

Conocer las fases de la metodología SCRUM es tan simple como saber que todo proceso funciona a partir de insumos. Luego, viene un proceso de desarrollo que conduce a un producto final.

Cada una de las etapas de SCRUM forma parte de una meta en común que busca satisfacer las exigencias y necesidades planteadas por los Project Manager, y al mismo tiempo, cumplir con los plazos de entrega de un proyecto.

Las 5 Fases De SCRUM

Las fases de la metodología SCRUM se reparten en 16 procesos o tareas, que a su vez se resumen en 5 pasos o etapas de implementación:

1. Inicio
2. Planificación y estimación
3. Implementación
4. Revisión y retrospectiva

5. Lanzamiento

Inicio

La primera fase se encarga de estudiar y analizar el proyecto identificando las necesidades básicas del sprint.

En el contexto de las metodologías ágiles, un sprint es un mini-proyecto con una duración no mayor a un mes que se interconecta con otros mini-proyectos para dirigirnos a los objetivos generales y específicos del proyecto general.

Las preguntas a hacer en la fase de inicio son:

- *¿Qué quiero?*
- *¿Cómo lo quiero?*
- *¿Cuándo lo quiero?*

La metodología SCRUM da preferencia a la formación de equipos pequeños de mínimo 3 y máximo 5 personas, pues se facilita la fluidez de las ideas y se aporta creatividad al grupo.

Entre los primeros pasos de SCRUM, tenemos 6 procesos:

1. Crear la visión del proyecto
2. Identificar a los Master SCRUM o SCRUM Master y a los *stakeholders*.
3. Formar equipos SCRUM
4. Desarrollar épicas
5. Crear *backlogs* o listas de requerimientos priorizando el producto
6. Planificar el lanzamiento

Planificación Y Estimación

La segunda fase de SCRUM incluye normalmente los siguientes pasos:

1. Crear, estimar y comprometer historias de usuario.
2. Identificar y estimar tareas.
3. Crear el *sprint backlog* o iteración de tareas.

La clave para llevar una buena administración de los proyectos es hacer una planificación y estimación del sprint, lo que te ayudará a establecer metas fijas y a cumplir con los plazos.

Tal vez esta sea la fase más importante del proyecto, pues si eres el Master Scrum tendrás que delegar las tareas correspondientes a cada grupo y hacer las estimaciones de tiempos de entrega, así como crear una lista ordenada para clasificar el trabajo según su prioridad.

Por ello, es necesario que utilices un Software Scrum que facilite la asignación de tareas y “haga rodar” el trabajo en equipo.

Implementación

Al llegar a la tercera de las 5 fases de SCRUM, nos topamos con la implementación del proyecto.

Es decir, la sala de reuniones donde se discute el *sprint* y se explora cómo optimizar el trabajo de cada grupo SCRUM para darle forma definitiva al proyecto.

En la implementación se cumple con los siguientes procesos:

1. Crear entregables.
2. Realizar *daily stand-up*.
3. Refinanciamiento del backlog priorizado del producto.

En la fase de implementación o desarrollo no deberían hacerse cambios innecesarios de última hora (se supone que para evitarlo existe una fase de planificación).

Aun así, si necesitas hacer un movimiento que será clave para el éxito del *sprint*, no dudes en proceder. Eso sí: discútelo primero con el grupo y los *stakeholders*, para que no haya confusiones o malos entendidos.

Revisión Y Retrospectiva

Una vez que ya todo está maquetado e implementado, deberás hacer la revisión del proceso, que no es más que la autocrítica o evaluación interna del grupo respecto a su propio trabajo.

Es importante sumar opiniones constructivas y aportar soluciones viables.

Entre los pasos más importantes para realizar en esta fase tenemos:

1. Demostrar y validar el *sprint*.
2. Retrospectiva del *sprint*.

Lanzamiento

La última de las fases del método SCRUM es el lanzamiento.

Con esto nos referimos al desenlace del proyecto y entrega del producto, donde deberías cumplir con 2 únicas tareas que son:

1. Enviar entregables.
2. Enviar retrospectiva del proyecto.

Conclusiones Las fases de la Metodología SCRUM son especialmente útiles durante el desarrollo de software, pero el método también es aplicable a cualquier tipo de empresa y proyecto donde el trabajo en equipo sea primordial.

2.8.- Ingeniería de software

La Ingeniería de Software es la especialidad de la ingeniería que concibe y mantiene las aplicaciones de software aplicando técnicas y pruebas de las ciencias computacionales, conducción de proyectos y otros campos

2.8.1 Características de la ingeniería de software

Es una especialidad de la ingeniería que tiene como objetivo principal, el desarrollo costeable de sistemas de software confiables que funcionen de modo eficiente y comprende todos los aspectos de la producción del software.

El software presenta 3 elementos que lo caracterizan:

1. Los programas y/o algoritmos.
2. Las estructuras de datos.
3. Los documentos.

2.8.2 Ciclo de vida

La ingeniería de Software consta de siete etapas. A continuación, cada una de ellas brevemente definidas:

Fig. 2 8 etapas de la ingeniería de software



- **Etapa de análisis:**

Es el procedimiento de investigación de un problema al que se desea encontrar la solución. Se define con claridad el Problema que hay que resolver o el programa que se desea inventar, identificando los elementos principales que conformarán el producto.

- **Etapa de Diseño:**

Es el procedimiento que emplea la información acumulada en la etapa de análisis al diseño del producto. La labor principal de la etapa de diseño es crear un modelo o las características precisas para el producto o Componentes del Sistema.

- **Etapa de Desarrollo:**

Consiste en el empleo de los diseños creados durante la etapa de diseño para elaborar los elementos a utilizarse en el sistema.

- **Etapa de Pruebas o Verificación Prueba:**

- Etapas de la Ingeniería de Software

Consiste en garantizar que los elementos individuales que componen el sistema o producto, presentan las características requeridas en la especificación creada durante la etapa de diseño.

- **Etapa de Implementación o Entrega Implantación:**

Consiste en la distribución del producto y hacerlo llegar a manos del cliente.

- **Etapa de Mantenimiento:**

Consiste en aplicar las soluciones apropiadas a cualquier problema del producto y re- liberar el producto mejorado, dándole una nueva versión.

- **Etapas finales EOL (End-of-Life)**

Consiste en ejecutar todas las labores que garanticen que tanto los clientes como los empleados tienen la certeza de que el producto ya no estará más a la disposición, por lo que no se venderá más. (Ver

2.9 Herramientas

Cada día que pasa, el uso del software es la mejor alternativa para aligerar y sistematizar labores que están en el desempeño de los procesos

En la evolución del software no es la excepción; en este caso los instrumentos se han llamado CASE (Ingeniería De Software Asistida Por Computador).

CASE (Ingeniería De Software Asistida Por Computador)

Incorporan un conjunto de programas que permiten mejorar la calidad de un producto, ofreciendo apoyo constantemente a los analistas, ingenieros de software y desarrolladores.

CASE es la adaptación de procesos y tecnología que dan rendimiento a los programas, a través de otros, procesos y su correspondiente documentación.

Sublime Text es un editor de texto y editor de código fuente está escrito en C++ y Python para los plugins. Desarrollado originalmente como una extensión de Vim, con el tiempo fue creando una identidad propia, por esto aún conserva un modo de edición tipo llamado *Vintage*

Características

- **Minimapa:** consiste en una previsualización de la estructura del código, es muy útil para desplazarse por el archivo cuando se conoce bien la estructura de este.

- **Multi Selección:** Hace una selección múltiple de un término por diferentes partes del archivo.
- **Multi Cursor:** Crea cursores con los que podemos escribir texto de forma arbitraria en diferentes posiciones del archivo.
- **Multi Layout:** Trae siete configuraciones de plantilla podemos elegir editar en una sola ventana o hacer una división de hasta cuatro ventanas verticales o cuatro ventanas en cuadrícula.
 - **Soporte nativo para infinidad de lenguajes:** Soporta de forma nativa 43 lenguajes de programación y texto plano.
 - **Syntax Highlight configurable:** El remarcado de sintaxis es completamente configurable a través de archivos de configuración del usuario.
 - **Búsqueda Dinámica:** Se puede hacer búsqueda de expresiones regulares o por archivos, proyectos, directorios, una conjunción de ellos o todo a la vez.
 - **Auto completado y marcado de llaves:** Se puede ir a la llave que cierra o abre un bloque de una forma sencilla.
 - **Soporte de Snippets y Plugins:** Los snippets son similares a las macros o los bundles además de la existencia de multitud de plugins.
 - **Configuración total de Keybindings:** Todas las teclas pueden ser sobrescritas a nuestro gusto.
 - **Acceso rápido a línea o archivo:** Se puede abrir un archivo utilizando el conjunto de teclas Cmd+P en Mac OS X o Ctrl+P en Windows y Linux y escribiendo el nombre del mismo o navegando por una lista. También se puede ir a una línea utilizando los dos puntos ":" y el número de línea.

- **Paleta de Comandos:** Un intérprete de Python diseñado solo para el programa con el cual se puede realizar infinidad de tareas.
- **Coloreado y envoltura de sintaxis:** Si se escribe en un lenguaje de programación o marcado, resalta las expresiones propias de la sintaxis de ese lenguaje para facilitar su lectura.
- **Pestañas:** Se pueden abrir varios documentos y organizarlos en pestañas.
- **Resaltado de paréntesis e indentación:** Cuando el usuario coloca el cursor en un paréntesis, corchete o llave, resalta esta y el paréntesis, corchete o llave de cierre o apertura correspondiente.
- **Sin impresión**

Fig. 2 9 Vista de código HTML

```

1 <div class="panel panel-default">
2   <div class="panel-heading">
3     <div class="box-header with-border">
4       <h3 class="box-title">Catálogo de Artículos</h3>
5       <div class="box-tools pull-right">
6         <button class="btn btn-box-tool" data-widget="collapse"><i class="fa fa-minus"></i></button>
7
8         <button class="btn btn-box-tool" data-widget="remove"><i class="fa fa-times"></i></button>
9       </div>
10    </div>
11  </div>
12  <!-- /.box-header -->
13  <div class="box-body">
14    <div class="row">
15      <div class="col-md-12">
16        <!-- Contenido -->
17
18        <div class="col-sm-14" id="VerForm">
19
20      <form role="form" name="frmArticulos" id="frmArticulos" enctype="multipart/form-data">
21        <div class="row">
22          <div class="col-lg-12 col-sm-12 col-md-12 col-xs-12">
23            <label>los campos con (*) son obligatorios</label>
24          </div>
25          <div class="col-lg-4 col-md-4 col-sm-6 col-xs-12 left">
26            <label>Categoria (*):</label>
27            <div class="form-group has-success">
28              <input id="txtSucursal" type="hidden" maxlength="50" class="form-control"
29                names="txtSucursal" required="" placeholder="" autofocus="" />
30
31              <select id="cboCategoria" name="cboCategoria" class="form-control">
32
33            </select>
34          </div>
35          <div class="col-lg-6 col-md-6 col-sm-6 col-xs-12 left">
36            <label>Codigo de Ubicacion (*):</label>
37            <div class="form-group has-success">
38              <input id="txtCodubi" type="text" maxlength="25" class="form-control" name="

```

Fig. 2 10 herramientas utilizadas en el proyecto

HERRAMIENTAS DE DESARROLLO DE SOFTWARE

▶ Lenguaje de Programación

PHP



▶ Gestor de base de Datos

MySQL



▶ Librería JQuery

Tecnología Ajax



▶ Diseño Web Responsive

Bootstrap



2.10 Pruebas

Las pruebas suministran una información selecta sobre los requerimientos implícitos y explícitos de los clientes. En el actual proyecto se usará la norma ISO 9126, la cual es una norma internacional que utilizan la mayoría de los desarrolladores de aplicaciones web. Esta norma se basa en 5 pilares que son: la funcionalidad, confiabilidad, Usabilidad, mantenibilidad, y la portabilidad del sistema.

2.11 métricas de calidad de software

La calidad de software es: la concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente.

2.11.1 Métricas para la Calidad de la especificación

El estándar ISO 9621 ha sido desarrollado en un intento de identificar los atributos clave de calidad: funcionalidad, usabilidad, facilidad de mantenimiento, portabilidad, confiabilidad y eficiencia

Funcionalidad. - Se trata de identificar la capacidad que el producto software tiene para proporcionar funcionalidades que satisfagan necesidades específicas.

Como no puede ser medida directamente, debe ser medido indirectamente de otras medidas directas como el punto de fusión propuesto por Albretch.

Para el cálculo de esta métrica se determina los valores de las cinco características de dominio de información: número de entradas, salidas, peticiones de usuario, archivo o interfaces de usuario. Asociando a estos dominios un valor de complejidad y calculando el punto de fusión con la siguiente relación.

$$PF = \text{Cuenta_total} * [0.65 + 0.01 * \text{Sum} (Fi)]$$

Cuenta_total = Suma de todas las entradas

0,65 = Valor de ajuste de complejidad mínimo

0.01 = Factor de conversión con un error del 1%

2.11.2 Métricas para Código fuente

Confiabilidad. - Es lo que se puede esperar que un programa lleve a cabo su función con exactitud requerida. Estadísticamente como la probabilidad de operación libre de fallos de un programa de computadora, en un entorno determinado y un tiempo específico.

Para el cálculo de la confiabilidad del sistema utilizamos la siguiente ecuación:

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

F(t): probabilidad de Ejecución con fallas

T: Representa el periodo de tiempo en el que se pone a prueba el sistema

1-F(t): Probabilidad de ejecución sin fallas

Para hallar la probabilidad de falla del sistema F(t) en el periodo t, se utiliza la función exponencial dado por:

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

f: funcionalidad del sistema.

λ : probabilidad de fallo en un periodo de tiempo.

t: tiempo total en el que se hace el cálculo de fallo.

2.11.3 Métricas para Mantenimiento

Mantenimiento

La facilidad con que una modificación puede ser realizada. Está indicada por los siguientes sub atributos; facilidad de análisis, facilidad de cambio de establecido y facilidad de prueba.

Este factor de calidad viene dado según el estándar IEEE-982-1998 por la métrica del índice de madurez del software (IMS) que proporciona una indicación de estabilidad del producto software (basada en cambio que ocurre en cada versión de producto). Se determina la siguiente formula

$$IMS = (Mt - (Fa + Fc + Fd)) / Mt$$

Mt = Numero de módulos en la versión actual

Fa = Numero de módulos en la versión actual que se añadido

Fc = Numero de módulos en la versión actual que se ha cambiado

Fd = Numero de módulos de la versión anterior, que se han borrado en la versión actual

A medida que el IMS se aproxima a 1, el producto empieza a estabilizar

Usabilidad

La usabilidad se determina calculando el porcentaje de una encuesta elaborado a un cierto número de usuarios finales.

Es el grado en que el software es fácil usar, se mide a través del esfuerzo necesario para aprender a usar el sistema, ya sea según su interfaz o manual de usuario.

Portabilidad

La facilidad con que el software puede ser llevado de un entorno a otro, o cambiar a un ambiente determinado. El grado de portabilidad del sistema está dado por la siguiente ecuación:

$$GP = 1 - (\text{Costo de Transportar} / \text{Costo de Re-Desarrollar})$$

GP = Grado de Portabilidad

GP > 0, la portabilidad del sistema es más rentable que el re-desarrollo

GP = 1, la portabilidad es perfecta.

GP < 1, Re-desarrollar es más rentable que transportarlo.

2.12 Seguridad

2.12.1 Definición de Seguridad Informática

La seguridad informática consiste en asegurar en que los recursos del sistema de información de una organización se utilizan de la manera que se decidió y que el acceso a la información allí contenida, así como su modificación solo sea posible a las personas que se encuentren acreditadas y dentro de los límites de su autorización.

2.12.2 Fiabilidad, Confidencialidad, Integridad y Disponibilidad

Si bien es cierto que todos los componentes de un sistema informático están expuestos a un ataque, son los datos y la información los sujetos **principales** de protección de las técnicas de seguridad. La seguridad informática se dedica principalmente a proteger la confidencialidad, la integridad y la disponibilidad de la información, por tanto, actualmente se considera que la seguridad de los datos y la información comprenden 3 aspectos fundamentales:

Confidencialidad

Integridad (seguridad de la información)

Disponibilidad

Hay que tener en cuenta que tanto las amenazas como los mecanismos para contrarrestarla suelen afectar a estas 3 características de forma conjunta por tanto un fallo del sistema que haga que la información no sea accesible puede llevar consigo una pérdida de integridad. Generalmente tiene que existir los 3 aspectos descritos para que haya seguridad. Dependiendo del entorno en el que trabaje un sistema, a sus responsables les interesara dar prioridad a un cierto aspecto de la

seguridad. Junto a estos 3 conceptos fundamentales se suele estudiar conjuntamente la *autenticación* y el *no repudio*. Suele referirse al grupo de estas características como **CIDAN**, nombre sacado de la inicial de cada Característica. Los diferentes servicios de seguridad dependen unos de otros jerárquicamente, A sí si no existe el primero no puede aplicarse el siguiente.

Disponibilidad: Se trata de la capacidad de un servicio, de unos datos o de un Sistema a ser accesible y utilizable por los usuarios o procesos autorizados cuando lo requieran. También se refiere a la capacidad de que la información pueda ser recuperada en el momento que se necesite.

Confidencialidad: Se trata de la cualidad que debe poseer un documento o Archivo para que éste solo se entienda de manera comprensible o sea leído por la persona o sistema que esté autorizado.

Un ejemplo de control de la confidencialidad sería el uso cifrado de clave Simétrica en el intercambio de mensajes.

Integridad: Es la cualidad que posee un documento o archivo que no ha sido alterado y que además permite comprobar que no se ha producido manipulación alguna en el documento original.

Alta disponibilidad (High Availability): son sistemas que están disponibles las 24 horas al día, 7 días a la semana, 365 días al año.

La disponibilidad se presenta en niveles:

Base: Se produce paradas previstas y imprevistas.

Alta: Incluyen tecnologías para disminuir el número y la duración de interrupciones imprevistas, aunque siempre existe alguna interrupción imprevista.

Operaciones continuas: Utilizan tecnologías para segura que no hay interrupciones planificadas

Sistemas de disponibilidad continua: Se incluyen tecnologías para asegurarse que no habrá paradas imprevistas ni previstas.

Sistemas de tolerancia al desastre: requieren de sistemas alejados entre sí para asumir el control en una interrupción provocada por un desastre.

Autenticación: Es la situación en la cual se puede verificar que un documento ha sido elaborado o pertenece a quien el documento dice. Las autenticaciones de los sistemas informáticos se realizan habitualmente mediante nombre y contraseña.

No repudio: El no repudio o irrenunciabilidad es un servicio de seguridad estrechamente relacionado con la autenticación y que permite probar la participación de las partes en una comunicación.

Existen 2 posibilidades:

No repudio en origen: el emisor no puede negar el envío porque el destinatario tiene pruebas del mismo el receptor recibe una prueba infalsificable del envío.

No repudio de destino: el receptor no puede negar que recibió.

2.13 Estimación de costos en proyectos de software

2.13.1 modelo cócono

Una metodología que se encarga de medir proyectos de software es COCOMO. (COConstructive COSt MOdel) se debe a Barry Bohem, y está orientada a las líneas de código.

Hay una jerarquía de modelos COCOMO: básico, intermedio y avanzado, la cual se aplica a tres tipos diferentes de software:

Orgánico: proyectos relativamente sencillos, menores de 50,000 líneas de código.

Se tiene experiencia en proyectos similares y en un entorno estable

Semiacoplado: proyectos intermedios en complejidad y tamaño. La experiencia en este tipo de proyectos es variable, y las restricciones intermedias.

Empotrado: proyectos bastantes complejos, en los que apenas se tiene experiencia y en un entorno de gran innovación técnica, Se trabajan con unos requisitos muy restrictivos y de gran volatilidad.

Dado que se va a emplear una sola variable (línea de código), se empleará COCOMO básico, ya que es un modelo invariable estático, con la que se obtiene una valoración objetiva del esfuerzo realizado. Este proyecto será considerado como software orgánico, ya que tiene menos de 50.000 líneas de código. La ecuación del esfuerzo básico de COCOMO tiene la siguiente formula:

$$E = ESFUERZO = a(KLDC)^b = PERSONA * MES$$

Donde KLDC es el número de líneas de código, distribuido en millares, para el proyecto. La ecuación de tiempo de desarrollo es:

$$T = \text{Tiempo de Duracion del Desarrollo} = C \text{ ESFUERZO (meses)}$$

Por su parte los coeficientes a, b, c y d se obtienen empíricamente del estudio de una serie de proyectos, y sus valores son:

Tabla 2 2 Constante de Complejidad

Proyecto de software	a	b	c	d
-----------------------------	----------	----------	----------	----------

Organico	2.4	1.05	2.5	0.38
Semiacoplado	3.0	1.12	2.5	0.35
empotrado	3.6	1.20	2.5	0.32

2.12 Seguridad

Definición de Seguridad Informática

La seguridad informática consiste en asegurar en que los recursos del sistema de información de una organización se utilizan de la manera que se decidió y que el acceso a la información allí contenida, así como su modificación solo sea posible a las personas que se encuentren acreditadas y dentro de los límites de su autorización.

Fiabilidad, Confidencialidad, Integridad y Disponibilidad

Si bien es cierto que todos los componentes de un sistema informático están expuestos a un ataque, son los datos y la información los sujetos principales de protección de las técnicas de seguridad. La seguridad informática se dedica principalmente a proteger la confidencialidad, la integridad y la disponibilidad de la información, por tanto, actualmente se considera que la seguridad de los datos y la información comprenden 3 aspectos fundamentales:

- *Confidencialidad*
- *Integridad (seguridad de la información)*
- *Disponibilidad*

Hay que tener en cuenta que tanto las amenazas como los mecanismos para contrarrestarla suelen afectar a estas 3 características de forma conjunta por tanto un fallo del sistema que haga que la información no sea accesible puede llevar

consigo una pérdida de integridad. Generalmente tiene que existir los 3 aspectos descritos para que haya seguridad. Dependiendo del entorno en el que trabaje un sistema, a sus responsables les interesara dar prioridad a un cierto aspecto de la seguridad. Junto a estos 3 conceptos fundamentales se suele estudiar conjuntamente la *autenticación* y el *no repudio*. Suele referirse al grupo de estas características como **CIDAN**, nombre sacado de la inicial de cada Característica.

Los diferentes servicios de seguridad dependen unos de otros jerárquicamente, así si no existe el primero no puede aplicarse el siguiente.

Disponibilidad: Se trata de la capacidad de un servicio, de unos datos o de un Sistema a ser accesible y utilizable por los usuarios o procesos autorizados cuando lo requieran. También se refiere a la capacidad de que la información pueda ser recuperada en el momento que se necesite.

Confidencialidad: Se trata de la cualidad que debe poseer un documento o archivo para que éste solo se entienda de manera comprensible o sea leído por la persona o sistema que esté autorizado.

Un ejemplo de control de la confidencialidad sería el uso cifrado de clave simétrica en el intercambio de mensajes.

Integridad: Es la cualidad que posee un documento o archivo que no ha sido alterado y que además permite comprobar que no se ha producido manipulación alguna en el documento original.

Alta disponibilidad (High Availability): son sistemas que están disponibles las

24 horas al día, 7 días a la semana, 365 días al año.

La disponibilidad se presenta en niveles:

Base: Se produce paradas previstas e imprevistas.

Alta: Incluyen tecnologías para disminuir el número y la duración de interrupciones imprevistas, aunque siempre existe alguna interrupción imprevista.

Operaciones continuas: Utilizan tecnologías para segura que no hay interrupciones planificadas *sistemas de disponibilidad continua:* Se incluyen tecnologías para

Asegurarse que no habrá paradas imprevistas ni previstas.

Sistemas de tolerancia al desastre: requieren de sistemas alejados entre sí para asumir el control en una interrupción provocada por un desastre.

Autenticación: Es la situación en la cual se puede verificar que un documento ha sido elaborado o pertenece a quien el documento dice. Las autenticaciones de los sistemas informáticos se realizan habitualmente mediante nombre y contraseña.

No repudio: el no repudio o irrenunciabilidad es un servicio de seguridad estrechamente relacionado con la autenticación y que permite probar la participación de las partes en una comunicación.

Existen 2 posibilidades: no repudio en el origen o sea el emisor no puede negar el envío porque el destinatario, tiene pruebas del mismo el receptor recibe una prueba infalsificable del envío. No repudio de destino: el receptor no puede negar que recibió.

CAP III

3. MARCO APLICATIVO

3.1 Introducción

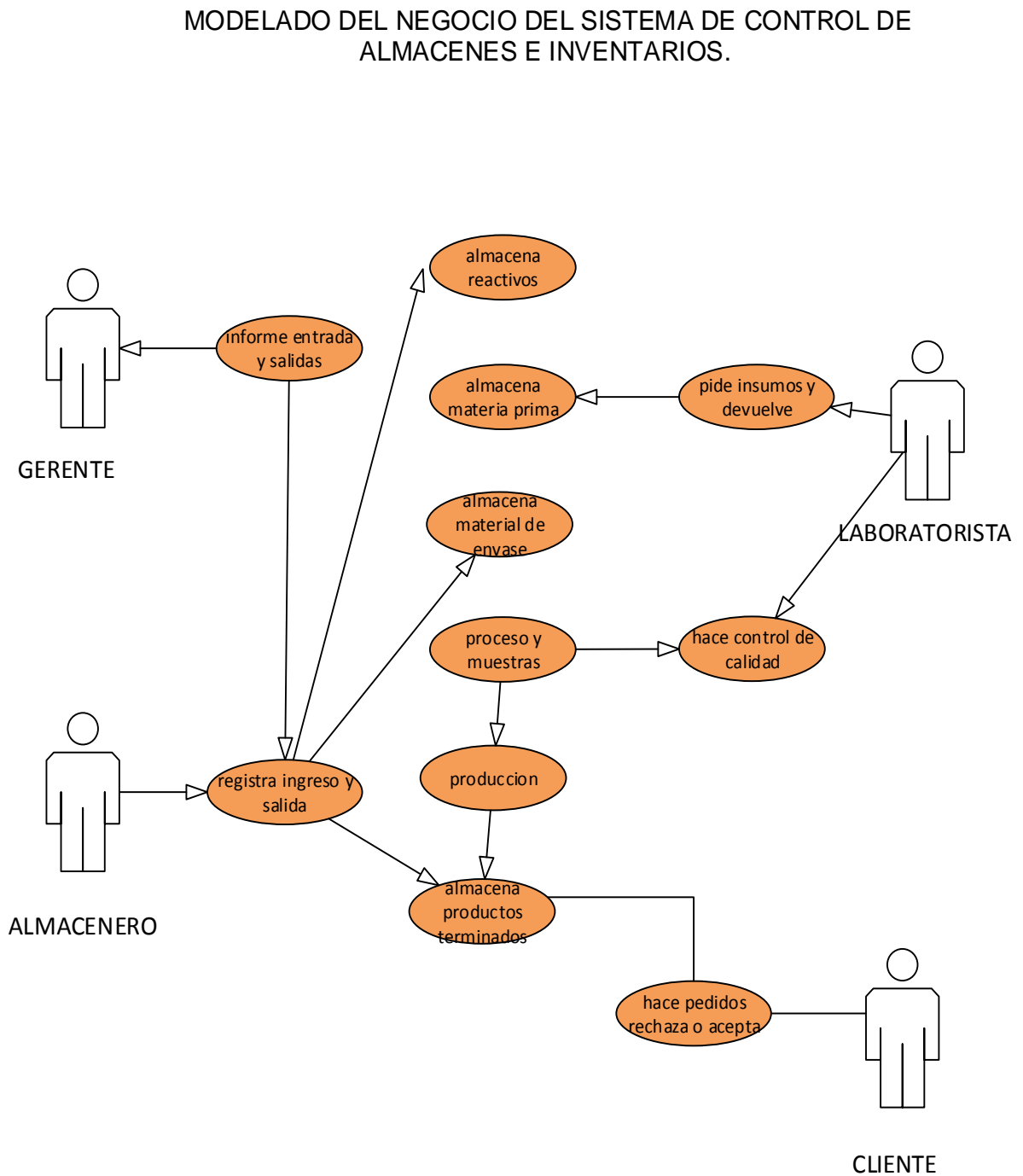
En este capítulo se define el marco de trabajo y las tareas que se requieren para desarrollar, construir e implementar el sistema, se tiene como finalidad poner en práctica todo lo mencionado en los capítulos anteriores,

Para diseñar el ciclo de vida de desarrollo del sistema se empleará scrum, que implementa el desarrollo iterativo e incremental como estrategia o paradigma de ingeniería de Software. Buscando obtener un producto de calidad minimizando los riesgos y evolucionando el producto en cada uno de las fases y flujos de trabajo sobre el cual itera.

En los siguientes puntos ampliaremos cada una de las fases de la metodología scrum, desarrollando todos los flujos de trabajo que se iteran sobre estos.

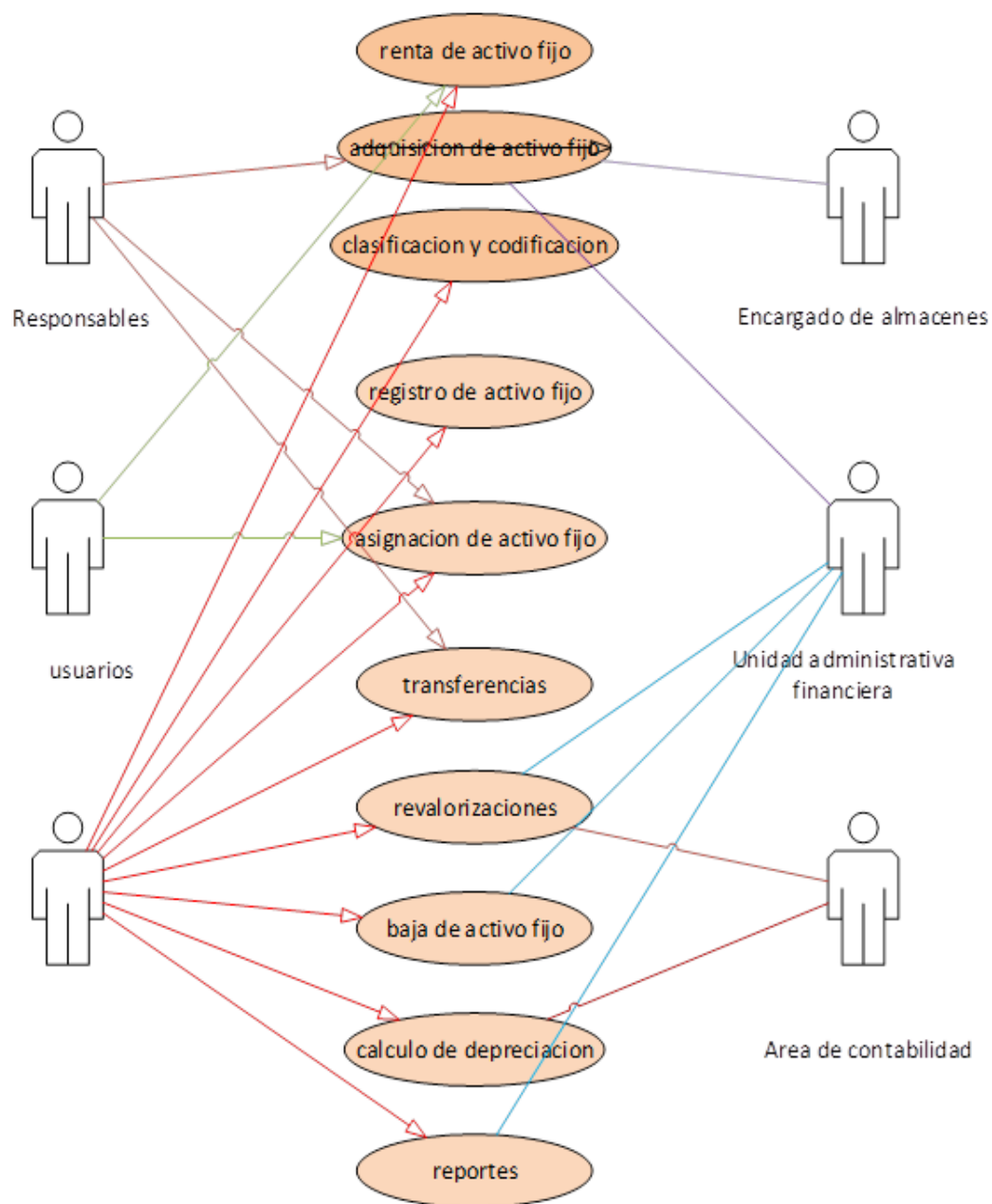
3.2 análisis situación actual.

fig. 3 1 Modelo del negocio del sistema de control de almacenes e inventarios



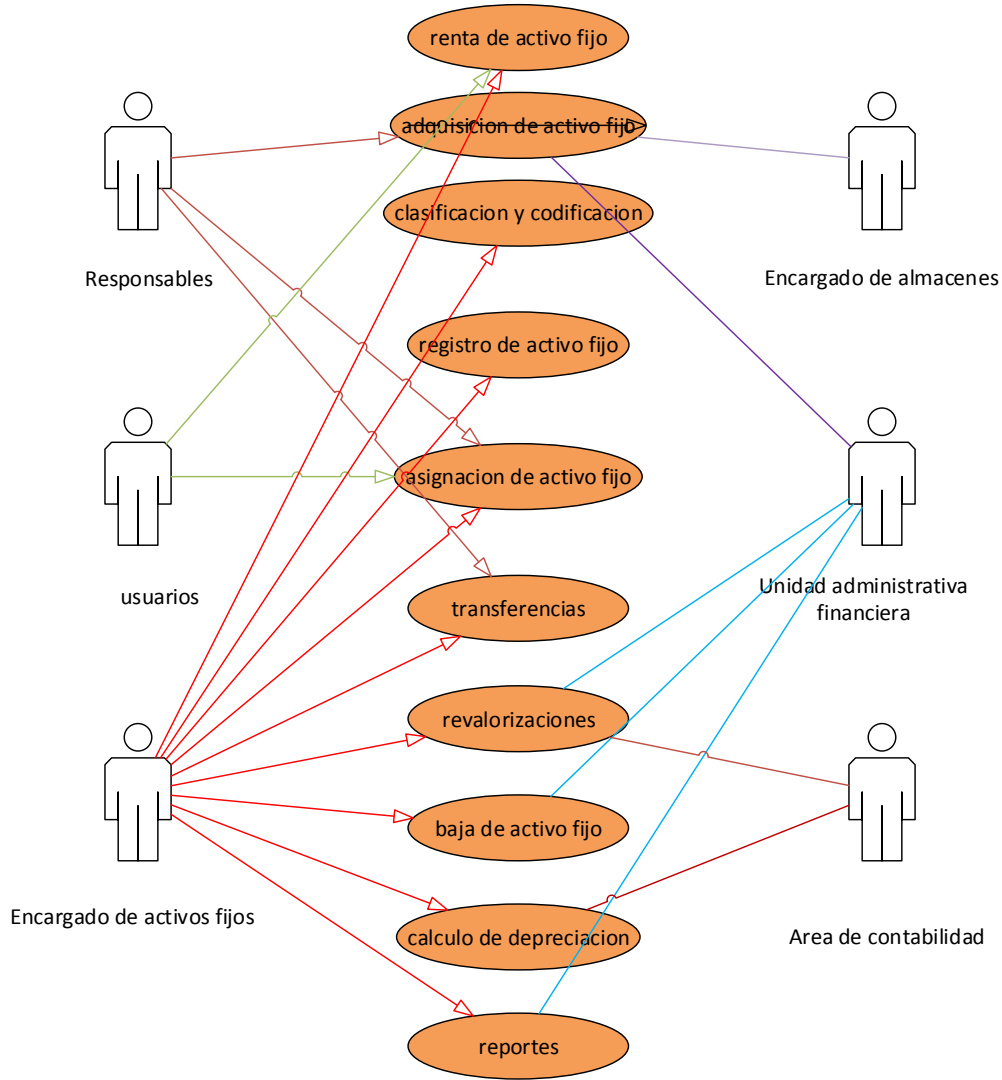
Fuente: [Elaboración Propia]

fig. 3.2 Casos de uso del Modelo del Negocio del Sistema Actual: Gestión de Control de Almacenes y activos fijos



Fuente: [Elaboración Propia]

Fig. 3 3 Casos de uso del Modelo del Negocio del Sistema Actual: Gestión de Activos Fijos



Fuente: [Elaboración Propia]

3.3 Análisis de requerimientos. -

Para la captura de requerimientos de aquellas funciones que deberá realizar el producto software, las mismas que deben estar diferenciadas en categorías entre

aquellas que son funciones evidentes para el usuario, de aquellas que son ocultas. Hay que identificar todos los requerimientos que el sistema debe cumplir con el objetivo de satisfacer las necesidades de los usuarios finales.

Tabla3 1 requerimientos de Control de Almacenes e Inventarios

Ref.#	Función	Categoría
R1	Registrar login, fecha y hora para cada proceso	oculto
R2	Registrar correctamente el Ingreso de materiales	Evidente
R3	Registrar correctamente la salida de materiales	
R4	Control eficiente de todos los almacenes (insumos, materias primas, etc.)	Evidente
R5	Determinar Inventario por fecha de vencimiento y punto de orden	Evidente
R6	Determinar con exactitud las Adquisiciones realizadas	Evidente
R7	Determinar con exactitud las ventas realizadas	Evidente
R8	Registrar los datos de los proveedores	Evidente
R9	Registrar los datos de los compradores	Evidente
R10	Reportar si materia prima esta vencida	Evidente
R11	Reportar si materia prima no hay en existencia	Evidente
R12	Reportar la cantidad de producto terminado ORI	Evidente

Fuente: [Elaboración Propia]

Tabla3 2 requerimientos de Control de Activos Fijos

Ref. #	Función	Categoría
R 1	Registrar la asignación de un activo fijo a un responsable de una determinada unidad.	Evidente
R 2	Registrar la liberación de un activo de un responsable	Evidente
R 3	Generar Acta de entrega y Recepción de activos fijos para un determinado responsable.	Evidente
R 4	Registrar la transferencia de responsabilidad de un activo de un usuario a otro.	Evidente
R 5	Generar acta de entrega y recepción de activos fijos para un determinado cliente.	Evidente
R 6	Registrar la asignación de un activo fijo a un cliente por motivo de la renta.	Evidente
R 7	Registrar la aceptación de transferencia de activos fijos.	Evidente
R 8	Generar acta de transferencia con Vo. Bo. Encargado de activos fijos.	Evidente
R 9	Registrar el informe técnico de mantenimiento.	Evidente
R 10	Registrar login, fecha y hora para cada proceso de gestión y transacciones que se registre para cada activo.	Oculto
R 11	Generar reporte de historial de transacciones.	Evidente

R 12	Registrar el inventariado de activos fijos que se realiza a un responsable determinado.	Evidente
R 13	Generar reporte de inventario de activos fijos de un determinado responsable.	Evidente
R 14	Generar reporte de usuarios.	Evidente
R 15	Generar reporte de Unidades y/o Oficinas.	Evidente
R 16	Generar reporte de responsables.	Evidente
R 17	Generar reporte de activos fijos.	Evidente
R 18	Generar reporte de incorporación por compra de activos fijos.	Evidente
R 19	Generar reporte de Revalúo de activos fijos.	Evidente
R 20	Generar reporte de activos fijos, por grupo y/o auxiliar contable	Evidente
R 21	Generar reporte de activos fijos, por unidad y/o responsable.	Evidente
R 22	Generar reporte de transferencias.	Evidente
R 23	Generar reporte de activos dados de baja.	Evidente
R 24	Generar reporte de historial de mantenimiento de un activo.	Evidente
R 25	Generar reporte de activos no asignados.	Evidente
R 26	Registrar y codificar el nuevo activo fijo. Que es incorporado a la institución, se incluye el grupo y auxiliar contable, fuente y organismo financiador.	Evidente
R 27	Registro de baja de activo fijo.	Evidente

R 28	Registro de la revalorización del activo fijo.	Evidente
R 29	Reduce cantidades del inventario cuando se da de baja un activo.	Evidente
R 30	Mostrar los días consumidos, factor actual, valor actual, depreciación de la gestión, depreciación acumulada, valor neto del activo registrado.	Evidente

Fuente: [Elaboración Propia]

3.3.1 Funciones del sistema

Descripción de los casos de uso del modelado del sistema actual

Control de almacenes e inventarios

Recepción de materiales o productos - Consiste en planear, dirigir, y controlar la entrada física de insumos y materias primas y otros bienes adquiridos por la empresa. Buscar la mayor eficacia de los métodos de descarga, inspección y verificación a través de notas de entrada

Despacho y envió. - Planear, dirigir y controlar operaciones de despacho de bienes y supervisar que se lleven a cabo con mayor eficacia. El despacho engloba una serie de actividades como el surtido, verificación de empaque, destino, etiquetado. Respecto al despacho, debe preverse un área para estibar (cargar) los artículos ya empacados próximos a ser embarcados para su transporte y el despacho adecuado para el estacionamiento de los vehículos de transporte.

Registro de salidas y entradas –Este proceso consiste en planear, organizar, dirigir y los procedimientos de registro, con el fin de mantener al día la información acerca de la recepción, existencia, despacho, costo y localización de los bienes.

Almacenamiento de materiales. - Es la tarea física de almacenamiento de materiales como reactivos, insumos, materias primas, material de envase y productos terminados en sus respectivos almacenes y su respectivo registro en kardex.

Mantenimiento y control -. Consiste en controlar, organizar y dirigir operaciones para tener el mejor manejo y control de la mercancía dentro del almacén, para lograr la mayor protección y conservación de los bienes bajo su cuidado según características de los bienes y asegurar su fácil rápida identificación, así como el optimizar el espacio.

Procesos y Muestras. - Es una actividad de verificación y clasificación de procesos y muestras para la producción.

Control de Calidad. - Etapa final para certificar la calidad de los productos terminados.

Coordinación del almacén con otros departamentos. - Almacenes e Inventarios coordina funciones con Gerencia y las unidades de Compras y Ventas y la unidad Administrativa.

Activos fijos

Renta de Activos Fijos. - Este proceso consiste en la renta de un activo fijo a un cliente.

Adquisición de Activos Fijos. - Este proceso consiste en la adquisición de un activo fijo, tras la solicitud de un usuario a la Unidad Administrativa Financiera que da curso a la adquisición del bien, para luego depositarlo en Almacenes hasta que el encargado de Activos fijos registre, clasifique y asigne al usuario solicitante.

Clasificación y Codificación de Activos Fijos. - El encargado de Activos Fijos es el responsable de clasificar el activo según normas y reglamentos de las NB-SABS y codificarlo según la clasificación realizada.

Registro de Activos Fijos. - El encargado de Activos fijos registra datos del Activo Fijo manualmente.

Asignación de Activos Fijos. Se registra los datos de asignación manualmente, se transcribe un acta de entrega y recepción de activos fijos, se realiza la entrega física de los activos asignados ya sea al responsable o al cliente y se archiva la copia del acta.

Transferencias. -

Este proceso consiste en la transferencia de un activo de un usuario a otro, por parte del encargado de activos fijos, se verifica el registro del activo en el acta de entrega y recepción, para luego transcurrir un acta de transferencia al nuevo responsable del activo, se le entrega el activo físicamente.

Revalorizaciones. -

Amparado bajo una resolución este proceso consiste en dar un valor agregado a un activo fijo después de una valuación al bien. El valor nuevo es calculado por el área de contabilidad, para que el encargado de activos fijos actualice su respectivo registro.

Baja de Activos Fijos. -Tras la depreciación total del activo fijo o por algún motivo, amparado bajo resolución, generalmente se realiza la baja gestionada por la Unidad Administrativa y Financiera, para luego pasar a su destrucción física, si amerita el caso y posterior registro.

Calculo de Depreciaciones. - El área de contabilidad realiza el cálculo de las depreciaciones para luego transcribir un reporte de depreciaciones.

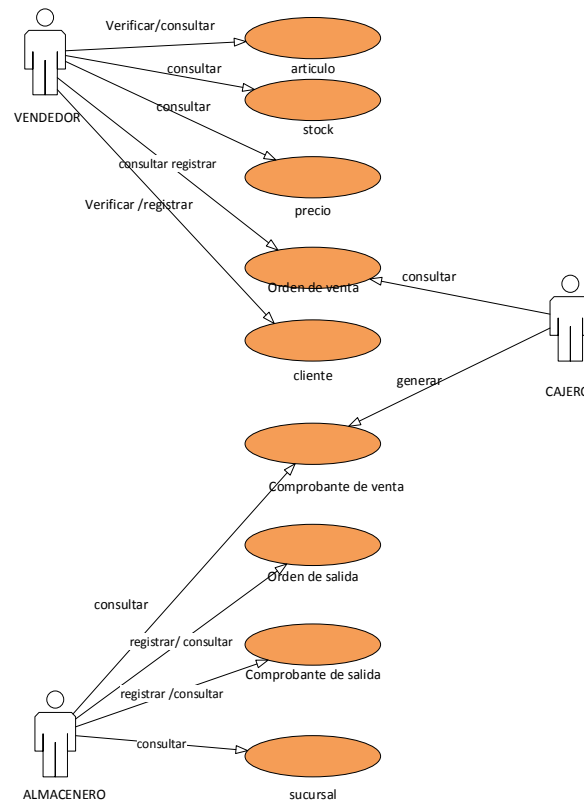
Reportes. - Este proceso consiste en transcribir y dar un informe según se requiere de los activos con los que cuenta la empresa.

3.3.2 Requerimientos funcionales

Casos de uso expandido del sistema de control de almacenes

Se identifican los casos de uso luego de haber identificado el rol de los actores y las funciones en las cuales participa, sintetizando en un caso de uso las funciones principales que debe contar el Sistema de Control de Almacenes.

fig. 3 4 Casos de uso del Modelo del Negocio del Sistema Actual: Gestión de Almacenes



Ingreso a Almacenes.

Caso de uso extendido: **Ingreso de materiales al Almacén**

Descripción del caso de uso extendido: Ingreso de materiales al Almacén

Tabla3 3 caso de uso almacenero

Caso de Uso:	Ingreso de materiales al Almacén
Actores :	Almacenero
Propósito :	Realiza el ingreso físico de materiales a los diferentes almacenes
Encargado de Almacén:	Registra el volumen de materiales ingresados
Sistema:	Solicita Almacén, Fecha de Ingreso, Departamento, Tipo Cpbte, Número y serie del Documento de Ingreso físico.
Encargado de Almacenes:	Guarda Comprobante sistema y comprobante de ingreso físico
Tipo:	Primario
Referencias Cruzadas	R1, R2, R5

Fuente: [Elaboración Propia]

Tabla 3 4 Caso de Uso encargado almacén

Caso de Uso:	Ingreso de materiales al Almacén
Actores :	Encargado de Almacenes
Propósito :	Realiza Control y Adm. De todos los almacenes, Coordinando con la Unidad Adm Financiera, Contabilidad. y Producción.
Adm. Almacenes:	Registra el volumen de materiales ingresados y enviados
Sistema:	Reporta Listados de Ingreso y Egresos y kardex actualizado
Adm. Almacenes:	Guarda Comprobante de sistema, de ingreso físico y Reportes
Tipo:	Primario
Referencias Cruzadas	R1, R2, R5,R6,R7

Tabla3 5 caso de uso laboratorista

Caso de Uso:	Ingreso de materiales al Almacén
Actores :	Laboratorista
Propósito :	Realiza Pedido de insumos y materiales para la producción, devolviendo el material sobrante.
Jefe Almacenes:	Registra el volumen de materiales pedido

Sistema:	Solicita Almacén para registrar de entrada y salida de insumos y materia prima.
Jefe Almacenes:	Guarda Solicitud de pedido
Tipo:	Primario
Referencias Cruzadas	R1, R2, R5,R6,R7

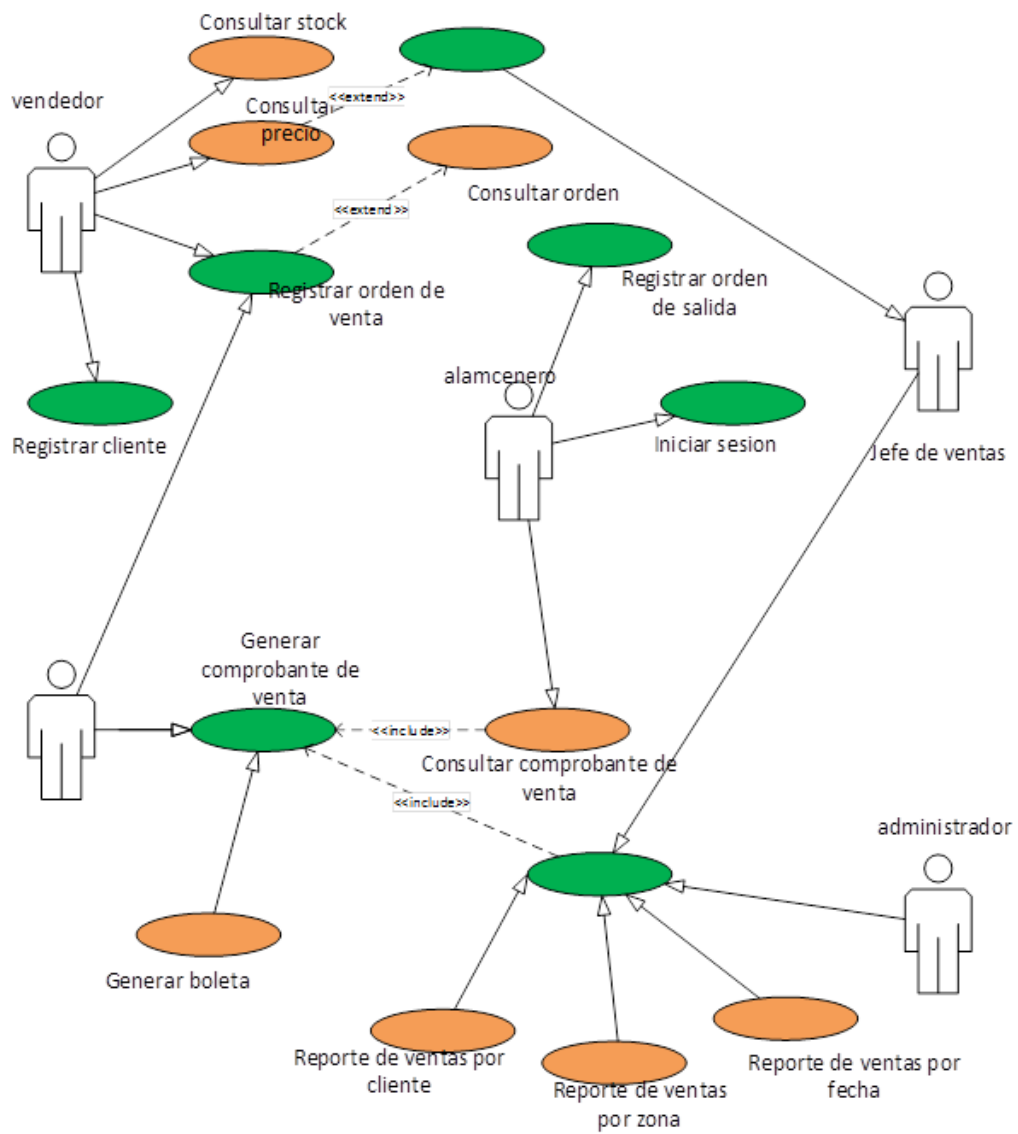
Fuente: [Elaboración Propia]

Tabla3 6 caso de uso de uso Gerencia

Caso de Uso:	Ingreso de materiales al Almacén
Actores :	Gerencia
Propósito :	Tener información pertinente, oportuna y de calidad.
Adm Almacenes:	Reporta productos en cuarentena, reactivos, materias primas y producción
Sistema:	Reporta Reportes
Adm. Almacenes:	Entrega reporte a Gerencia
Tipo:	Primario
Referencias Cruzadas	R1, R2, R5,R6,R7

Fuente: [Elaboración Propia]

fig. 3 5 Casos de uso del Modelo del Negocio del Sistema Actual: Gestión de Salidas y Ventas de Almacenes.



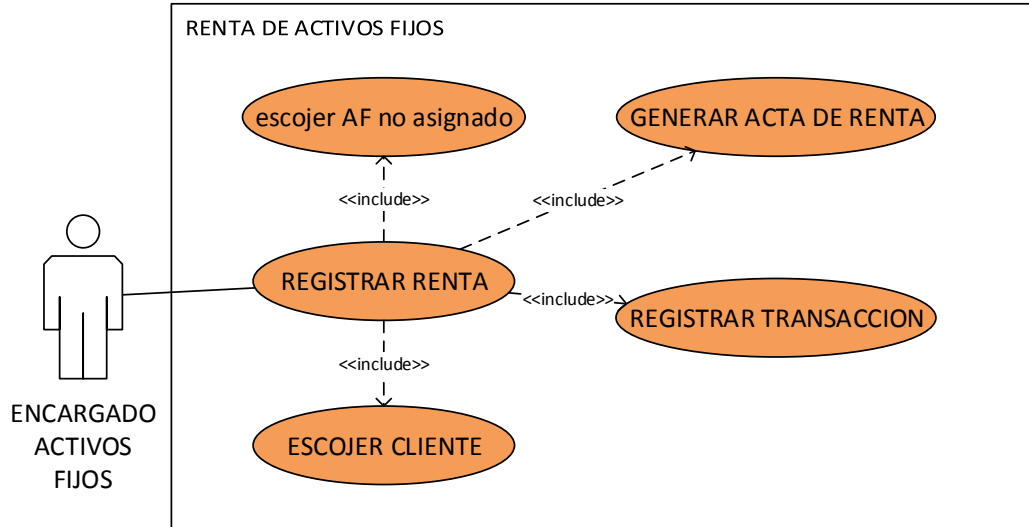
Fuente: [Elaboración Propia]

Tabla3 7 Salida de Materiales de Almacén

Caso de Uso:	Salida de materiales de Almacén
Actores :	Almacenero
Propósito :	Realiza el ingreso físico de materiales a los diferentes almacenes
Encargado de Almacén:	Registra el volumen de materiales ingresados
Sistema:	Solicita Almacén, Fecha de Ingreso, Departamento, Tipo Cpbte, Número y serie del Documento de Ingreso físico.
Encargado de Almacenes:	Guarda Comprobante sistema y comprobante de ingreso fisico
Tipo:	Primario
Referencias Cruzadas	R1, R2, R5
Caso de Uso:	Salida de materiales de Almacén
Actores :	Encargado de Almacenes
Propósito :	Realiza Control y Adm. De todos los almacenes, Coordinando con la Unidad Adm Financiera, Contabilidad. y Producción.
Adm Almacenes:	Registra el volumen de materiales ingresados y enviados
Sistema:	Reporta Listados de Ingreso y Egresos y kardex actualizado
Adm.Almacenes:	Guarda Comprobante de sistema, de ingreso físico y Reportes
Tipo:	Primario
Referencias Cruzadas	R1, R2, R5,R6,R7

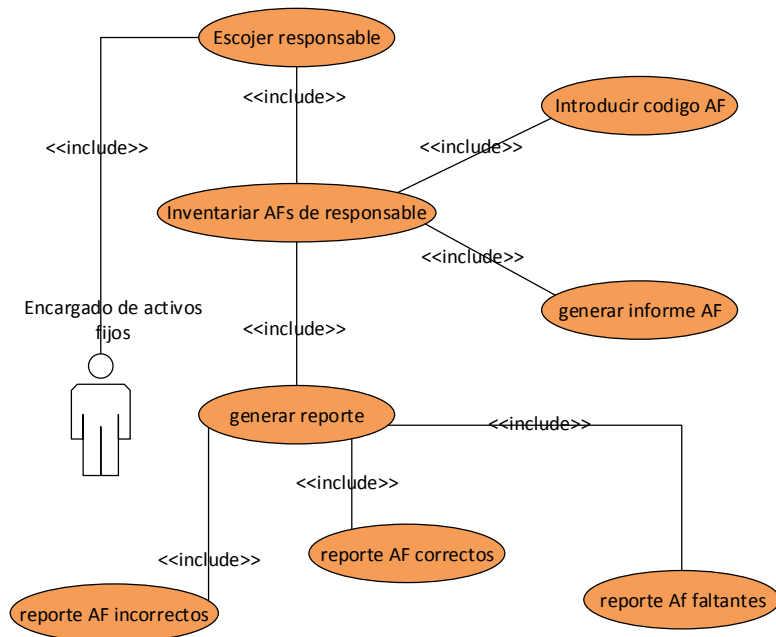
3.2.4 Casos de uso expandido del sistema de control activos fijos

fig. 3 6 Diagrama de Uso: Inventario de Activos Fijos



Caso de Uso expandido: Inventario de activos fijos

fig. 3 7 Diagrama de caso de Uso: Renta de activos Fijos.



Descripción del caso de Uso Expandido: Renta de Activos fijos

Tabla3 8 caso de uso renta de activos fijos

Caso de Uso:	Renta de Activos Fijos
Actores:	Encargado de Activos Fijos
Propósito:	Asignar Activo Fijo a Un determinado Usuario cliente
Adm. AF	Registra la asignación de un activo a un cliente en calidad de renta, y a un empleado en calidad de responsable
Sistema	Solicita Nro. De documento de respaldo de la asignación del activo del cliente o del empleado.
Admin. AF	Ingresa el Nro. De documento de respaldo y acepta la asignación del activo.
Sistema	Guarda el registro de la asignación del cliente o responsable, esta operación se registra en el historial de transacciones.
Tipo:	Primario
Referencias cruzadas	R1, R2, R,6, R26, R27, R32

Fuente: [Elaboración Propia]

fig. 3 8 Caso de Uso Expandido Inventario de Activos Fijos

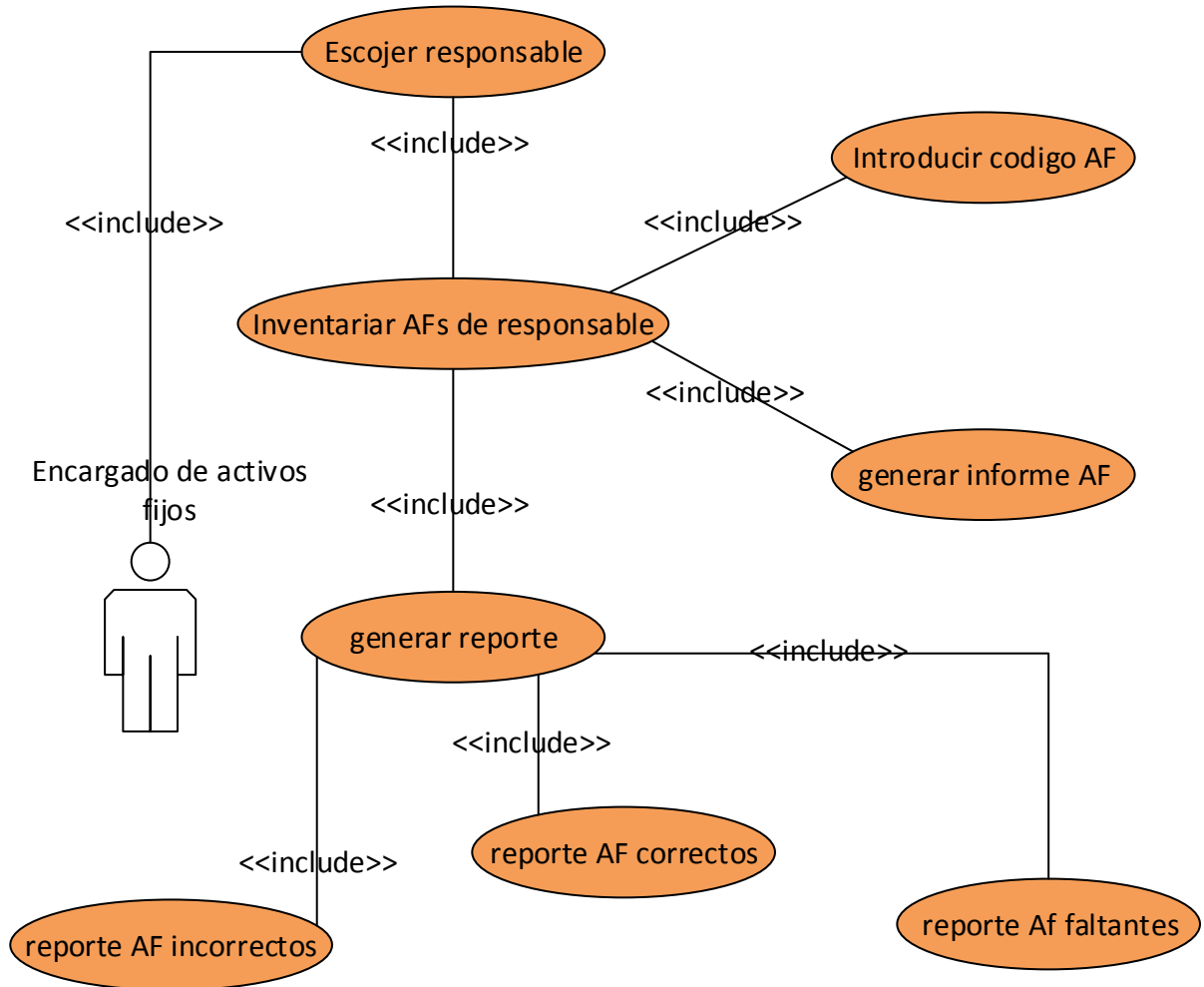


Diagrama de Caso de Uso: Inventario de activos fijos
Fuente: [elaboración propia]

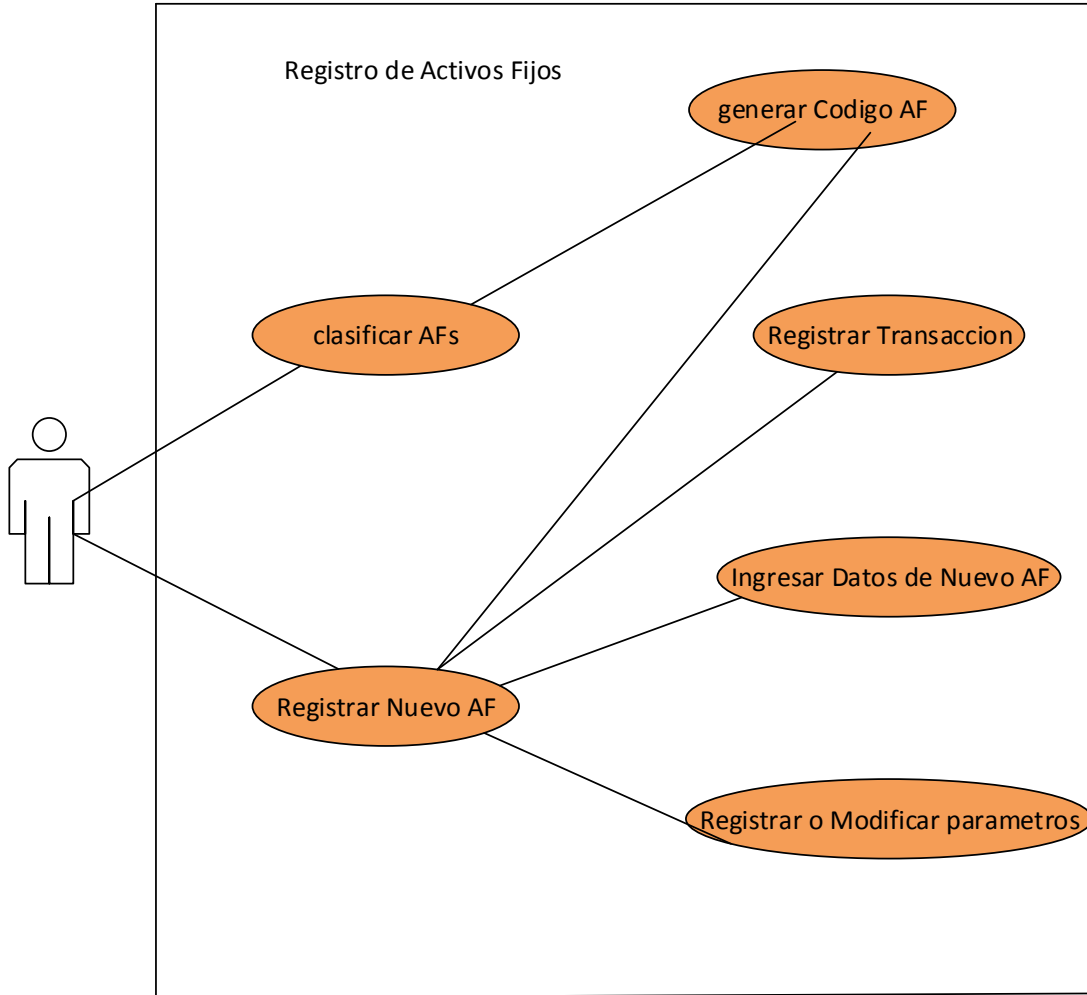
Descripción del caso de uso Expandido: Inventario Activos Fijos

Tabla3 9 Caso de Uso Inventario de Activos Fijos

Caso DE	Inventario de Activos Fijos
Uso	
Actores:	Encargado de Activos Fijos
Propósito:	Inventariar los activos fijos asignados a un determinado responsable o cliente
Admin. AF	Registra el inventario del activo fijo. Escoge el cliente y/o responsable del activo fijo
Sistema	Genera informe parcial para cada activo fijo
Admin. AF	Introduce los códigos de los activos fijos a inventariar que se encuentra en custodia del cliente y/o responsable.
Sistema	Al terminar el inventario se genera el reporte de los activos fijos Incorrectos, activos fijos correctos y activos fijos Faltantes.
Tipo:	Primario
Cruzadas:	R1, R3, R5, R12, R13, R32.

Fuente: [Elaboración Propia]

fig. 3 9 Caso Expandido: Registro de Activos Fijos



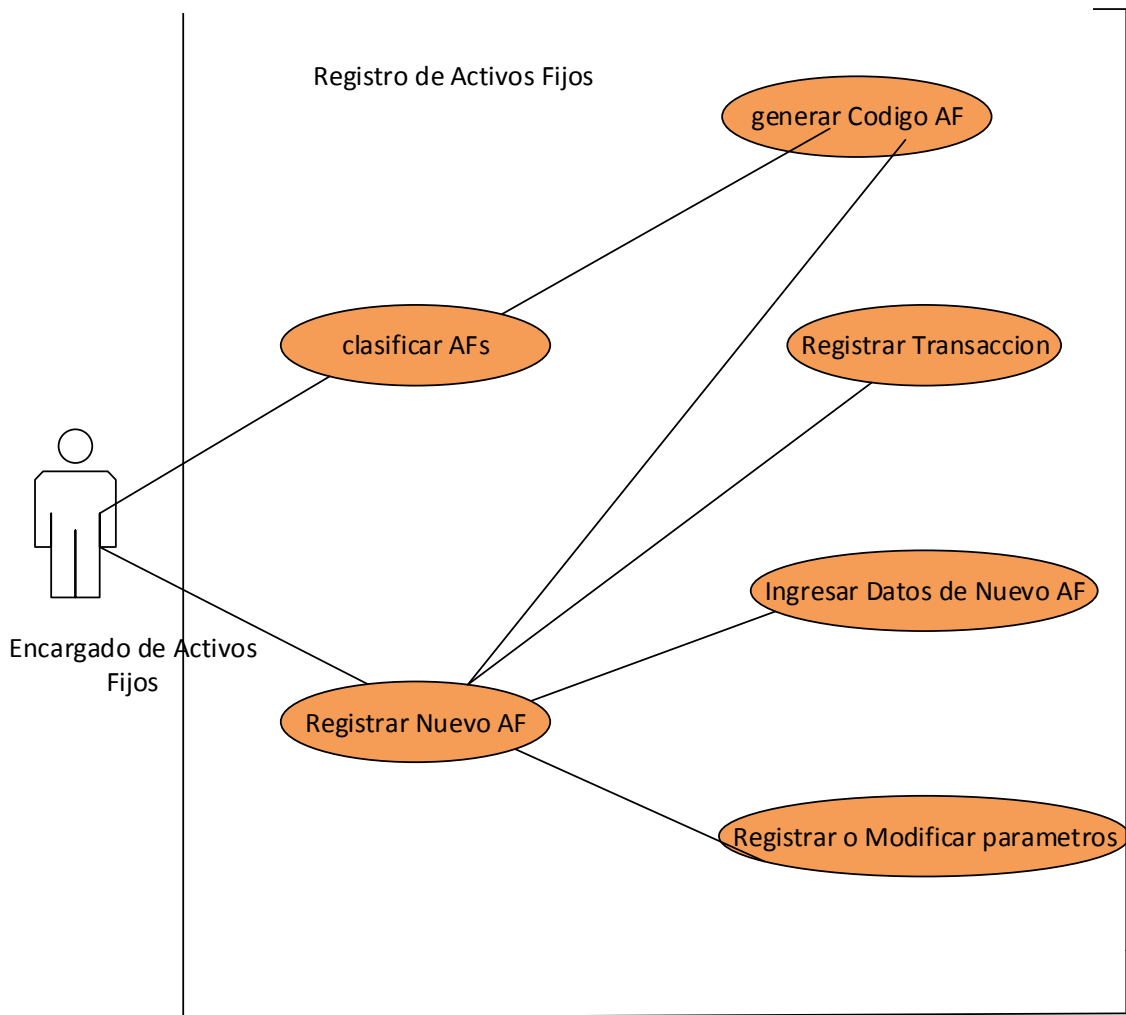
Fuente: [Elaboración Propia]

Tabla3 10 caso de uso registro de activos fijos

Caso de Uso	Registro de Activos Fijos
Actores:	Encargado de activos fijos
Propósito:	Registrar un Activo Fijo
Admin. AF	Registra el ingreso de un activo fijo, clasificándolo por su grupo y auxiliar contable y otros datos del activo fijo
Sistema	Genera el código tentativo del Activo Fijo
Admin. AF	Acepta el registro y el código tentativo generado o lo modifica
Sistema	Guarda el registro del activo fijo. Esta operación se registra también en el historial de transacciones.
Tipo:	Primario
Referencias	
Cruzadas	R 10, R 26, R 32.

Fuente: [Elaboración Propia]

fig. 3 10 diagrama de caso de uso registro activos fijos



Fuente: [Elaboración Propia]

3.3 Diseño

3.3.1 Evaluación de la fase de inicio

Se avanzó satisfactoriamente el Capítulo anterior de acuerdo a lo planificado, se estableció el modelado del negocio y desarrollo sustancial de la captura de los requerimientos del sistema. Como conclusión el estudio de factibilidad realizado, se puede establecer que el sistema propuesto es viable.

3.3.2 Planificación fase de elaboración.

En esta fase transcurrirá a través de los flujos de trabajo: El análisis y diseño del sistema, se desarrollará las interfaces de usuario.

En La Tabla 3.11 se presenta la planificación para la fase de elaboración del proyecto, detallando la fecha inicio, fin y fecha de aprobación.

Tabla3 11 Tiempo de desarrollo de Software

Actividad Generada Durante la Fase de Elaboración	Semana	Inicio	Fin	Aprobación
Análisis / Diseño				
Revisar el modelo de Análisis y Diseño	2 - 5	10-08-20	10-09-20	10-10-20-20
Implementación	1-2	15-08-20	25-08-20	27-10-20-
Prototipos de Interfaces de usuario	2-3	19-09-20	10-10-20	11-10-20
Modelo de Implementación	1	22-10-20		23-10-20

Plan de Actividades de la Fase de Elaboración

Fuente: [Elaboración Propia]

3.4 Desarrollo del modelo

El modelado del sistema consiste en obtener una visión del sistema, modelando los requisitos funcionales planteados en la captura de requerimientos.

En esta fase, se usó el diagrama de secuencias para conseguir una comprensión más precisa de los requisitos, refinándolos y estructurándolos, logrando una descripción más detallada del funcionamiento de cada módulo del sistema.

3.4.1 Modelo Conceptual

3.4.1.1 Diagrama de Secuencias.

Desarrollado los casos de uso expandidos, elaboramos los diagramas de secuencias relacionados a dichos casos de uso, en los que se realizan las operaciones que requieren los actores del sistema.

fig. 3 11 Diagrama de Secuencia: Gestionar Ingresos a Almacén

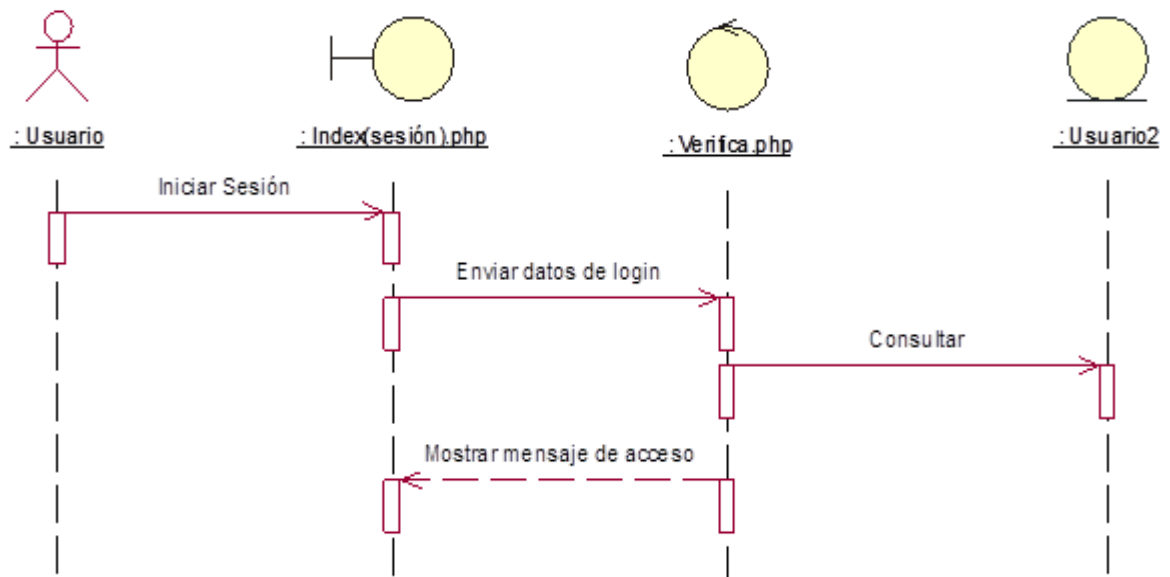


fig. 3 12 Diagrama de Colaboración: Gestionar Ingresos a Almacén

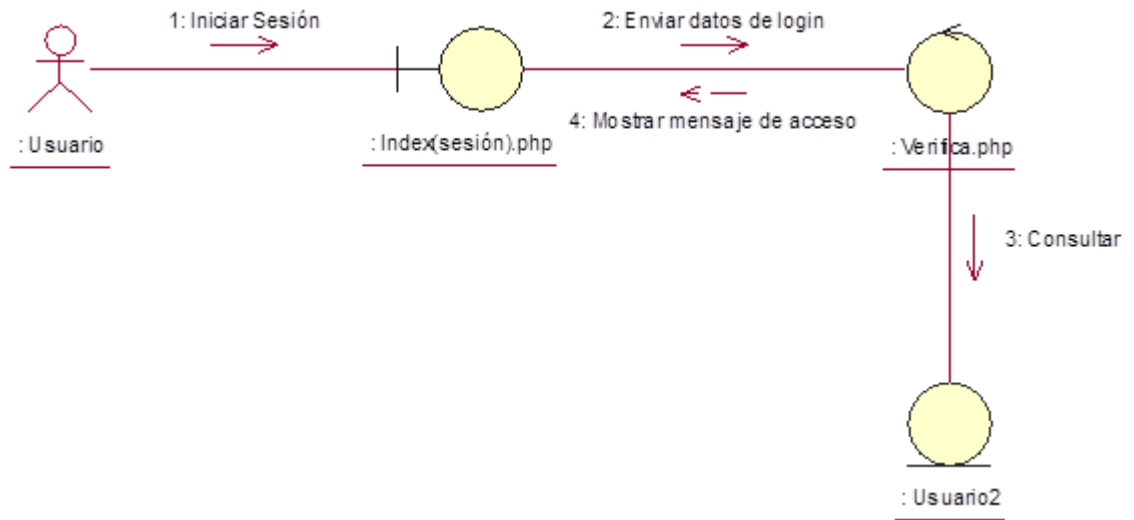


fig. 3 13 Diagrama de Secuencia: Registro de Materiales

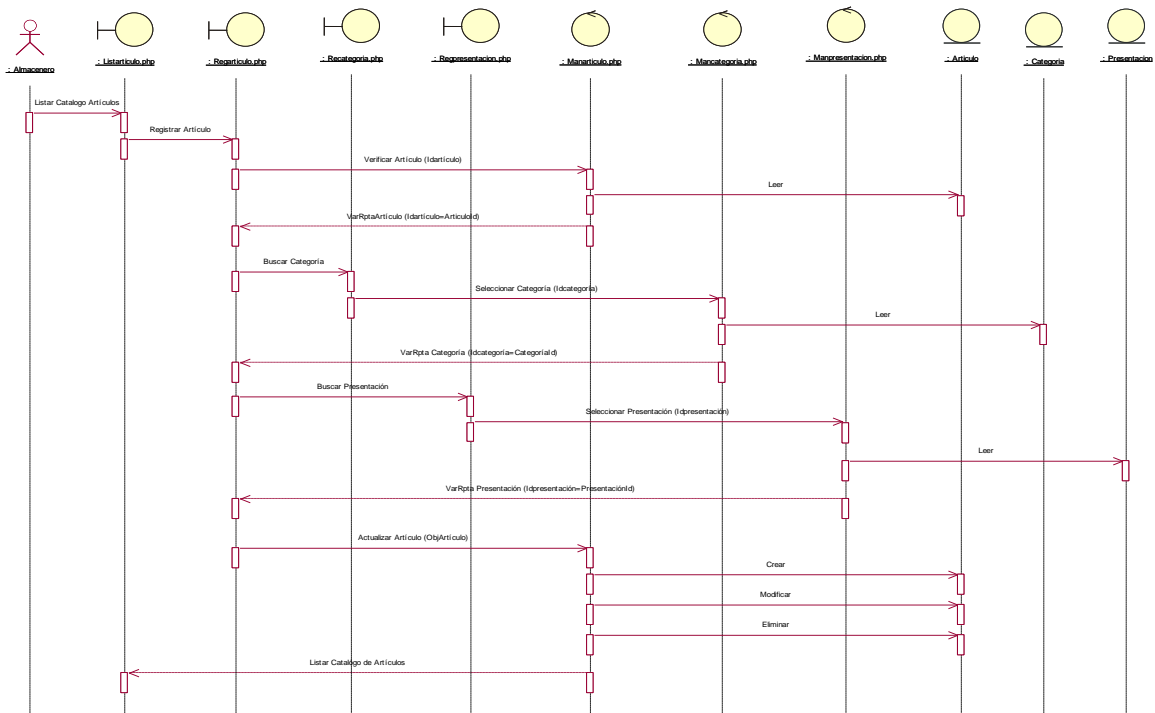


Diagrama de Secuencias: Registro Materiales
Fuente: [Elaboración Propia]

fig. 3 14 Diagrama de Secuencia: Actualizar kardex

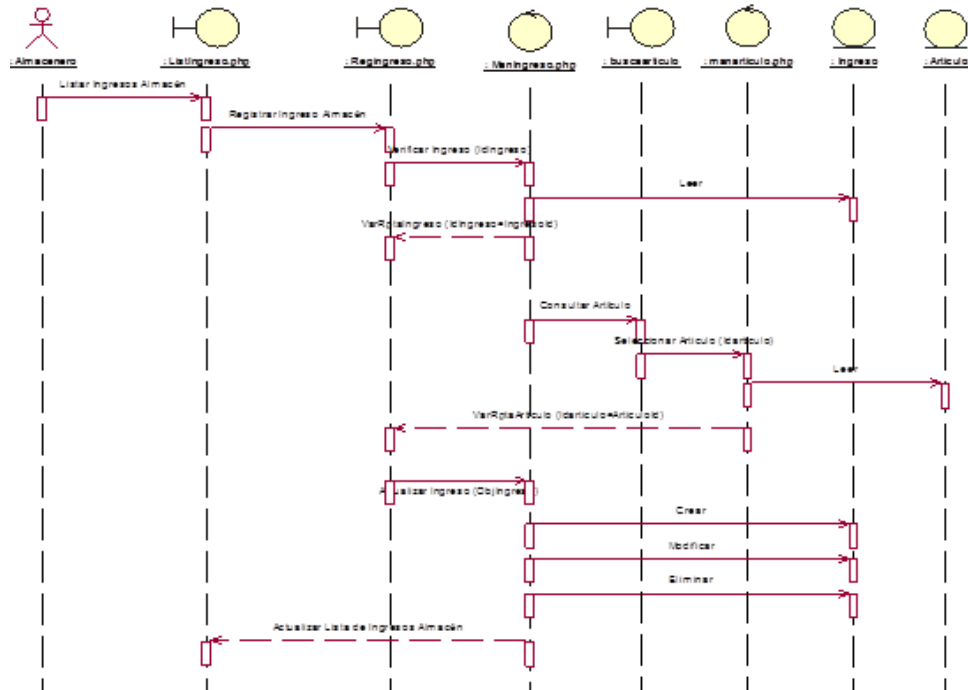


fig. 3 15 Diagrama de Colaboración: Actualizar Kardex

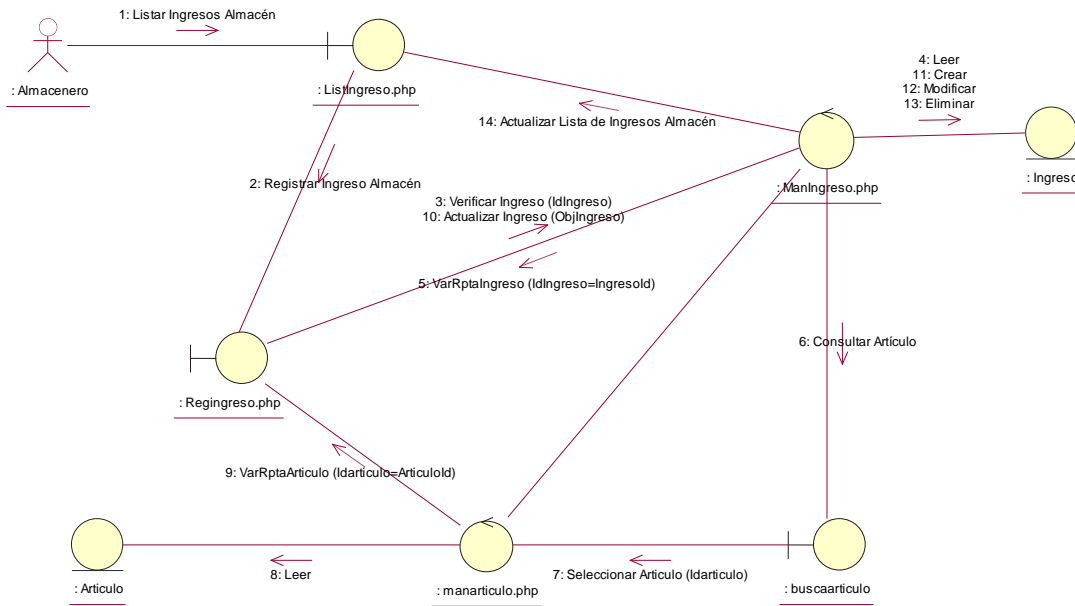


fig. 3 16 Diagrama de Secuencias: Órdenes de Compra

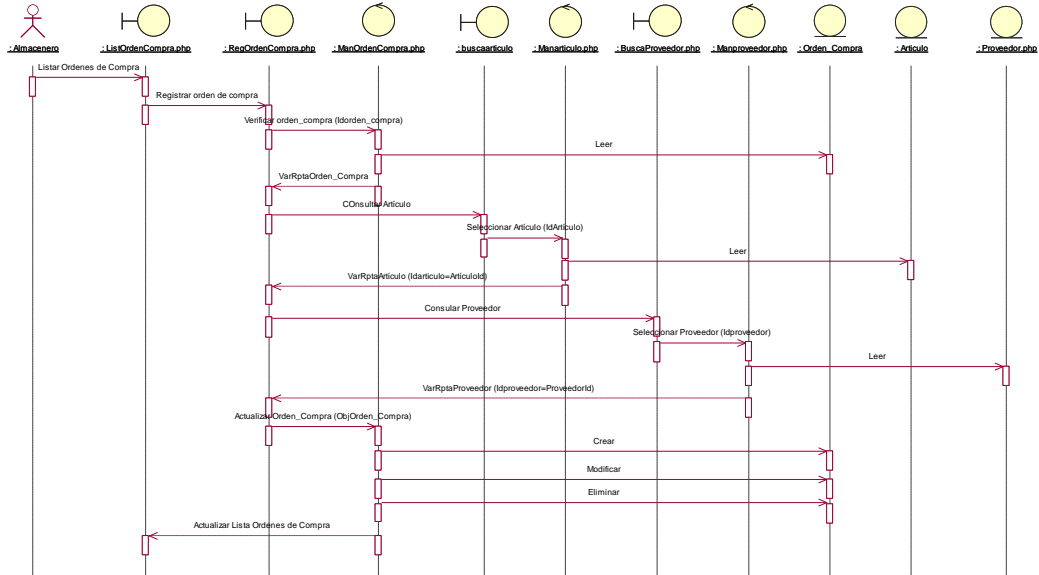


fig. 3 17 Diagrama de Colaboración: Órdenes de Compra

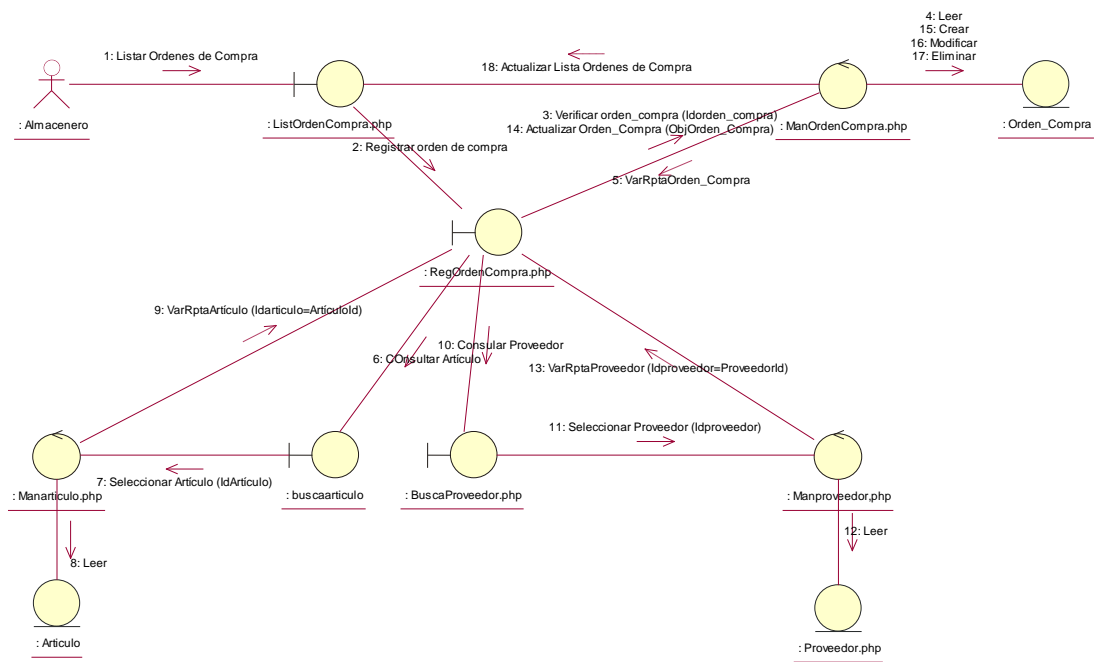


fig. 3 18 Diagrama de Secuencia: Órdenes de Venta

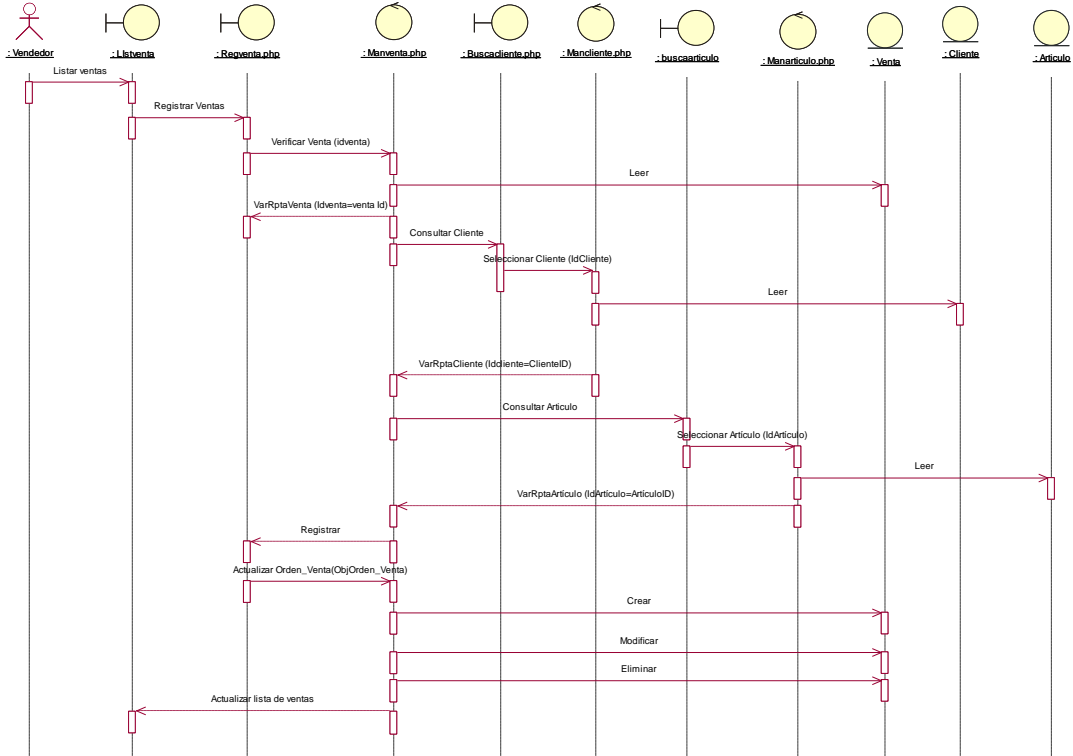


fig. 3 19 Diagrama de Colaboración: Órdenes de Compra

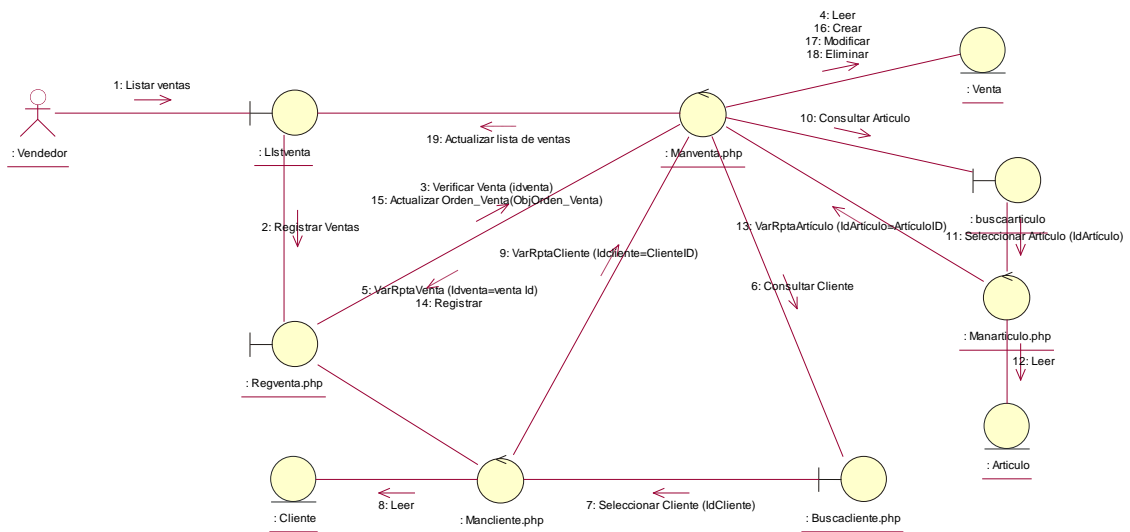
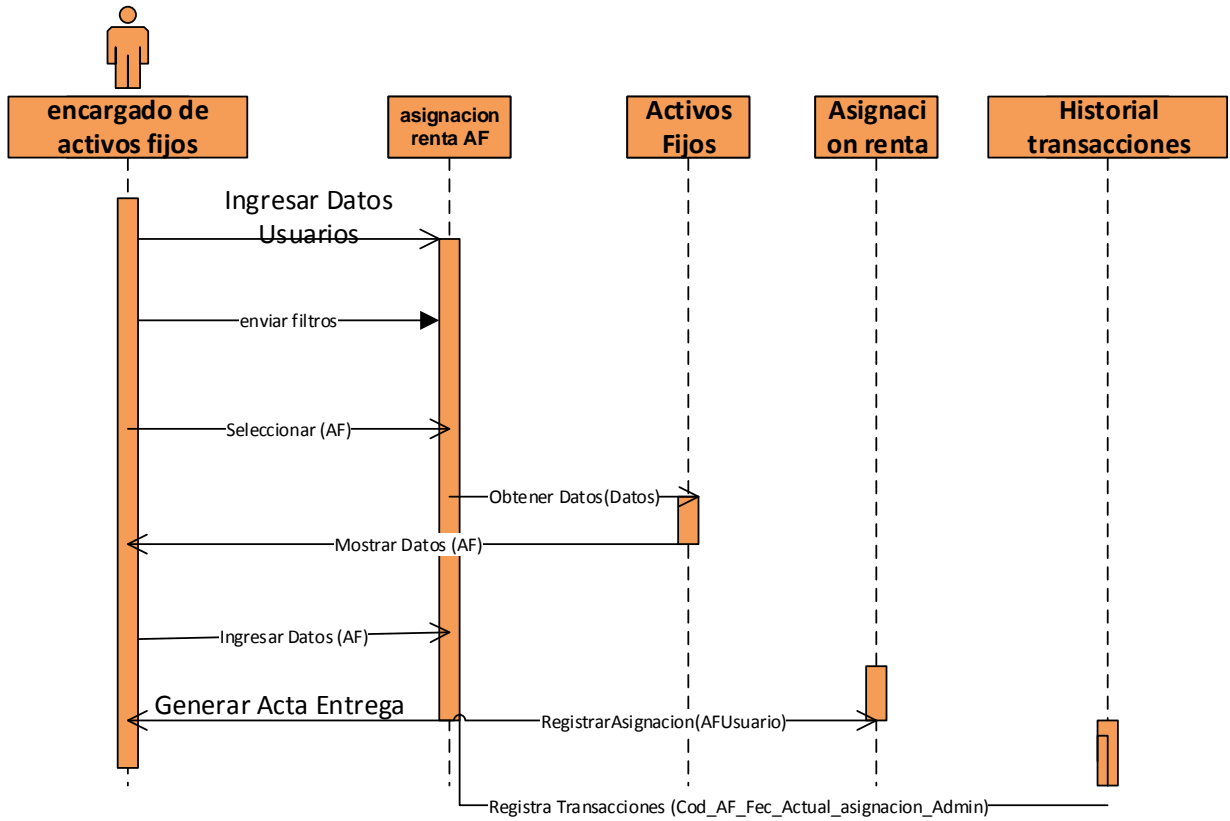


fig. 3 20 Diagrama de secuencia: Renta de Activos fijos



3.5 Fase de Construcción

fig. 3 21 Diagrama de Clases: Control de Almacenes e Inventarios

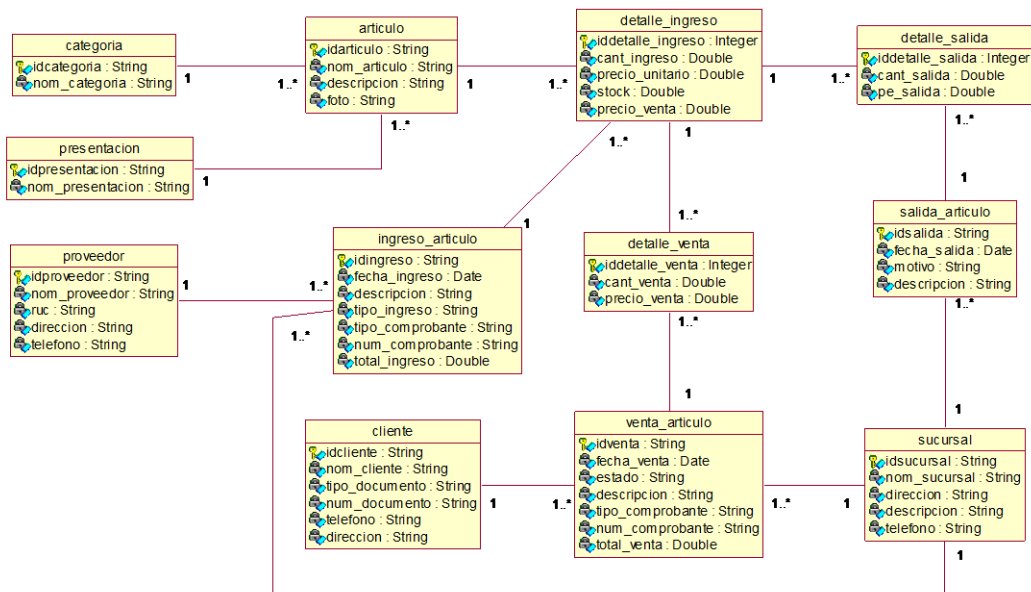
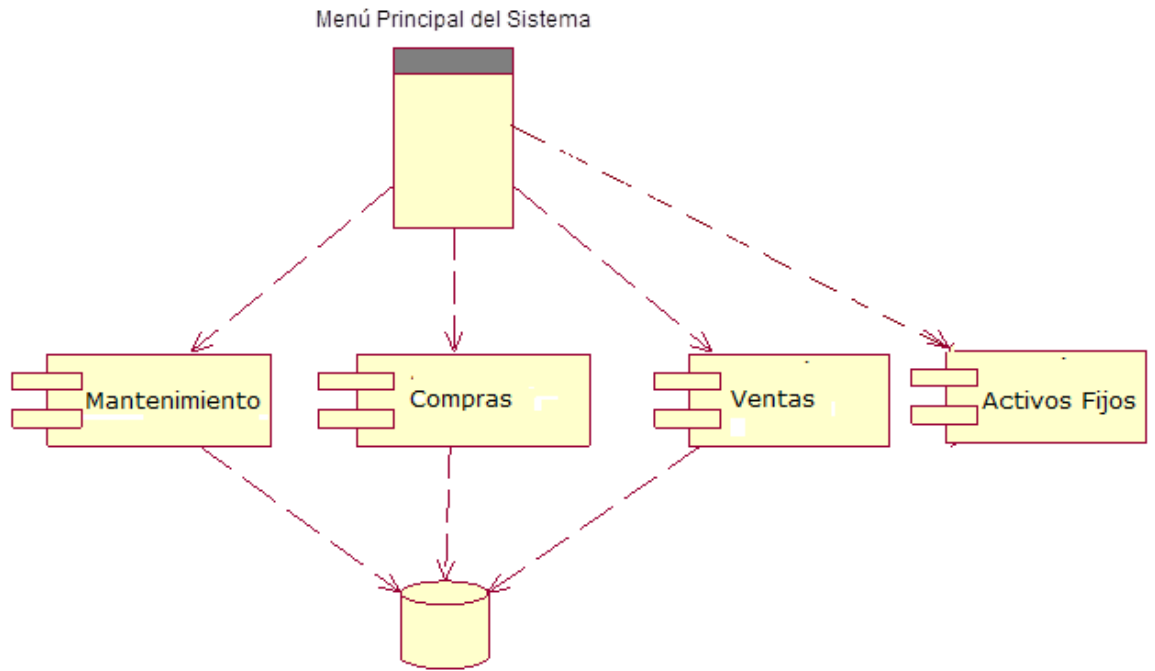


fig. 3.22 Diagrama de Componentes: Control de Almacenes e Inventarios

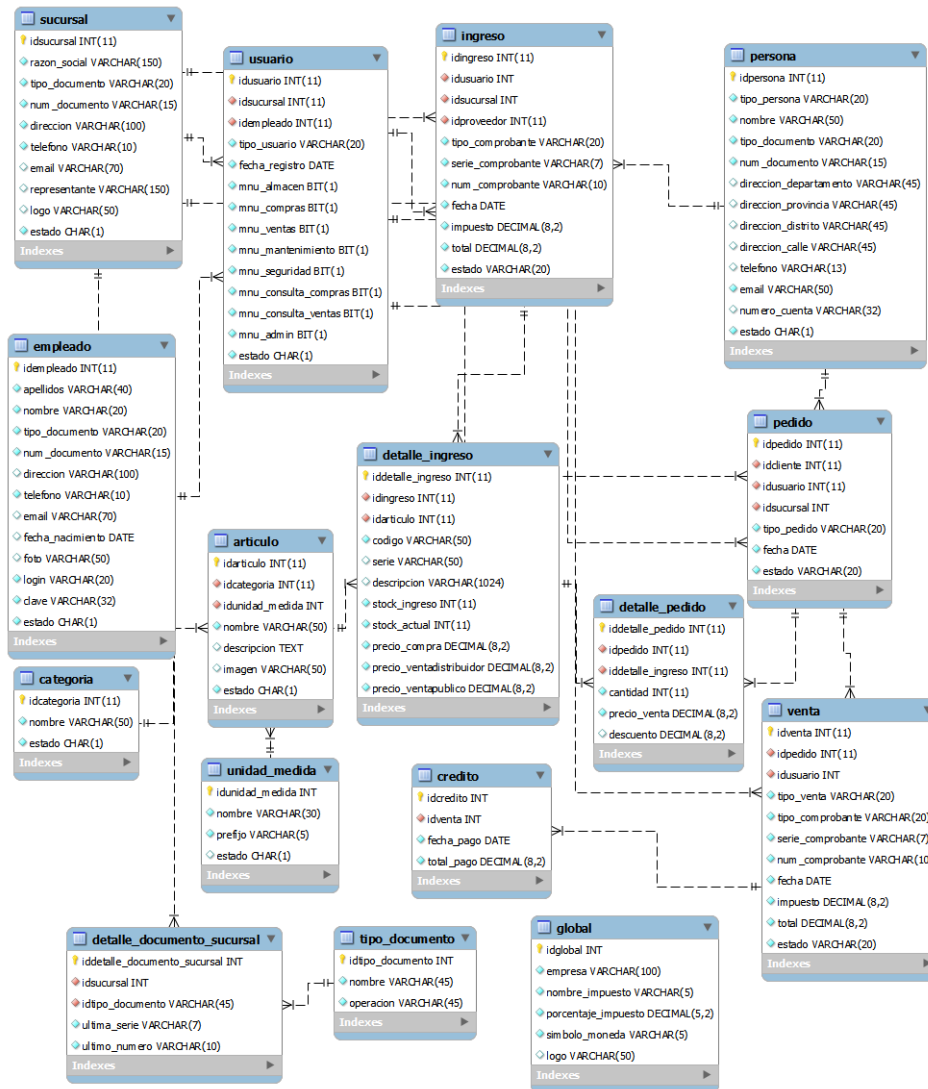


3.5.1 Base de datos

3.5.2.1 Modelo Entidad Relación: Control de Almacenes

En el Modelo Entidad Relación se plasma una abstracción del diagrama de clases fig.3.23. En la que se diseñó las asociaciones, Clases asociadas, atributos que se requiere para construir la base de datos, también se incluye todos los métodos analizadas en el diagrama de secuencias.

fig. 3 23 diseño de base de datos del sistema



3.5.2 Patrón Modelo Vista Controlador

Uno de los patrones que ha demostrado ser fundamental a la hora de diseñar aplicaciones Web es el Modelo-Vista-Controlador [MVC]. Las capas de este modelo son:

Modelo: Son las estructuras de datos, y la lógica de la aplicación, es decir se trata de cómo funciona la aplicación.

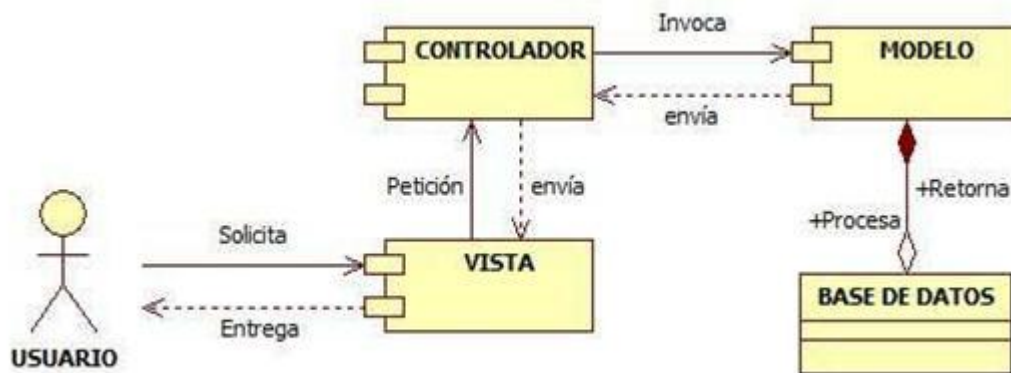
Vista: Es la interfaz que ve el usuario. Dependiendo de la implementación, esta

interfaz se genera en base a los datos que se obtienen del modelo o con la información que proporciona el controlador.

Controlador: Es la capa que se encarga de procesar los eventos. Esto se realiza en base a lo que indica el usuario y se produce modificación en el modelo. Es decir es la traducción de las acciones del usuario en cambios de estado del modelo.

En la figura 3.24 se observa la relación entre una aplicación Web y algunas de las capas del patrón: la vista es la Web que ve el usuario en su navegador, el controlador es el código que interpreta el formulario o la petición del navegador y lo traduce a una llamada a una función modelo, para finalmente devolver una vista a lo que es una página Web.

fig. 3 24 Movimiento de (MVC)



3.5.3 Diseño de Interfaces

En esta fase, se implementa el diseño con todos los modelos que fueron construidos en el anterior capítulo de forma independiente de la plataforma de implementación; en esta fase se ha tenido en cuenta el entorno particular en el cual se va a correr la aplicación.

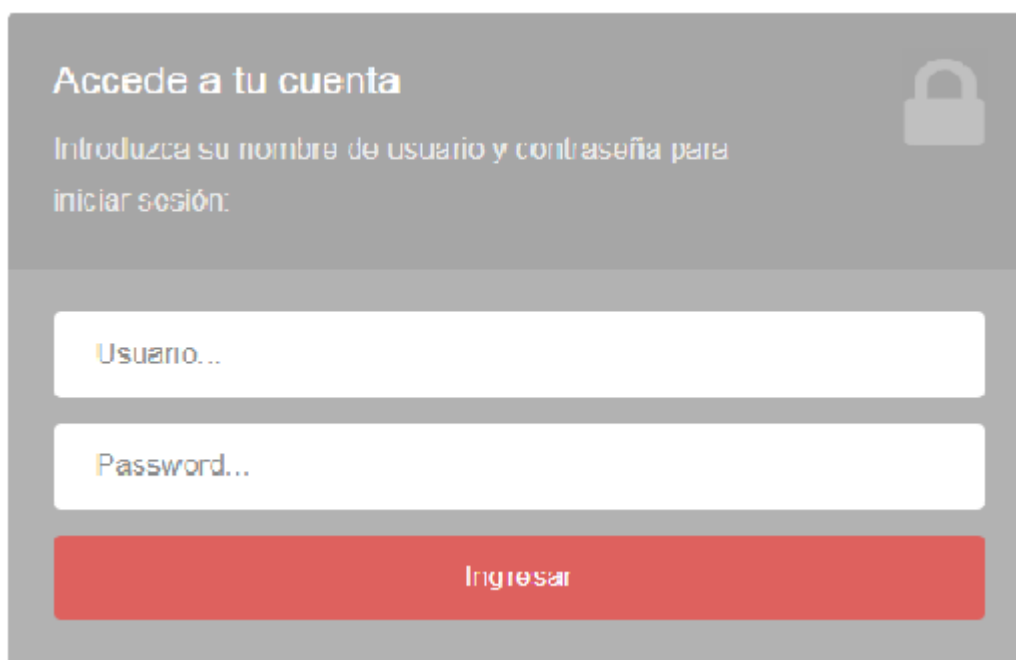
Interfaz de usuario

La interfaz de usuario es la forma en que los usuarios pueden comunicarse con

una computadora, y comprende todos los puntos de contacto entre el usuario y el equipo.

Luego de haber culminado las etapas de Análisis, Diseño y la construcción del “Sistema de Gestión Administrativo Portuario”, se mostrará a continuación las interfaces con que trabajará el usuario.

3.5.3.1 Pantalla Inicial de Presentación



The image shows a login interface with a dark gray background. At the top left, the text 'Accede a tu cuenta' is displayed in white. To the right of this text is a white padlock icon. Below the title, the instruction 'Introduzca su nombre de usuario y contraseña para iniciar sesión:' is written in white. There are two white input fields: the first is labeled 'Usuario...' and the second is labeled 'Password...'. At the bottom of the form is a prominent red button with the white text 'Ingresar'.


Esta es la pantalla de ingreso al Sistema de Control de Almacenes, el cual restringe

El acceso a los usuarios no autorizados.


El usuario accede al sistema ingresando el nombre del usuario y su respectiva contraseña de acceso.


3.5.3.2 Interfaz para Acceder a Sucursales


Datos del Empleado





Javier Calle Gutierrez
Administrador

Documento: CI 123456789 

Telefono: 72565061 


Direccion: Av. Jaimes Freyre 

Email: jcalle@gmail.com 




Usuario: admin 


[Cerrar Sesión](#)

Acceso a las Sucursales

Opción	Sucursal	Logo
Acceder	Laboratorios Estasa	

3.5.3.3 Interfaz Menu Principal Para Administrar Control de Almacenes

CONTROL ALMACENES  Hoy  Sucursal - Laboratorios Estasa 

 **Javier Calle Gutierrez**
Online

- Mantenimiento <
- Almacén <
- Compras **C**
- Ventas **V**
- Activos Fijos **A**
- Consultas Compras <
- Consultas Ventas <
- Consultas de Activos Fijos **A**
- Seguridad <
- Ayuda **PDF**
- Acerca De...
- Cerrar Sesión **SOL**

3.6 FASE DE TRANSICION

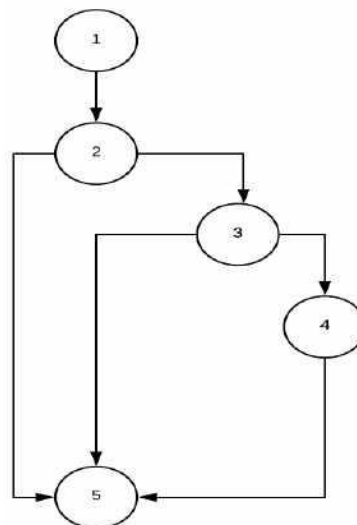
3.6.1 Pruebas de la caja blanca

Estas pruebas se centran en los detalles procedimentales del software. Se escogen una serie de conjuntos de entradas para cubrir los distintos flujos de ejecución del código.

Una técnica de este tipo de pruebas, es el **camino de prueba básico**, en el que se crea un flujo que representa el código fuente y se visualizan cuantos casos de prueba se deben definir para cubrir la funcionalidad en su totalidad, según su complejidad ciclomática.

Los flujos y los caminos resultantes de los EndPoints definidos para la aplicación, no contemplarán la respuesta de error de servidor (HTTP Error 500).

El flujo para el EndPoint `/api/register` es:



Las condiciones que hay en el flujo son:

- El nodo 2, cuándo el *nombre de usuario* o *correo electrónico* ya están dados de alta en el sistema.
- En el nodo 3, cuándo los datos introducidos no son los correctos.

La complejidad ciclotómica de este método es:

$$V(G) = e - n + p = 6 - 5 + 2 = 3$$
$$V(G) = \text{número de condiciones} + 1 = 3$$

Procedemos a encontrar los caminos independientes:

$$\begin{aligned} \underline{\text{Camino 1}} &= 1, 2, 5 \\ \underline{\text{Camino 2}} &= 1, 2, 3, 5 \\ \underline{\text{Camino 3}} &= 1, 2, 3, 4, 5 \end{aligned}$$

Los casos de prueba para cada uno de los caminos serán:

- *Camino 1*: Datos de usuario con un nombre de usuario ya registrado, no se da de alta.
- *Camino 2*: Datos de usuario con nombre vacío, no se da de alta.
- *Camino 3*: Datos de usuario correctos y se da de alta.

3.6.1 Pruebas de Caja Blanca Login Al sistema

A continuación, se realizarán pruebas de caja Blanca al módulo de Logueo del Sistema de Control de Almacenes e Inventarios y Activos Fijos.

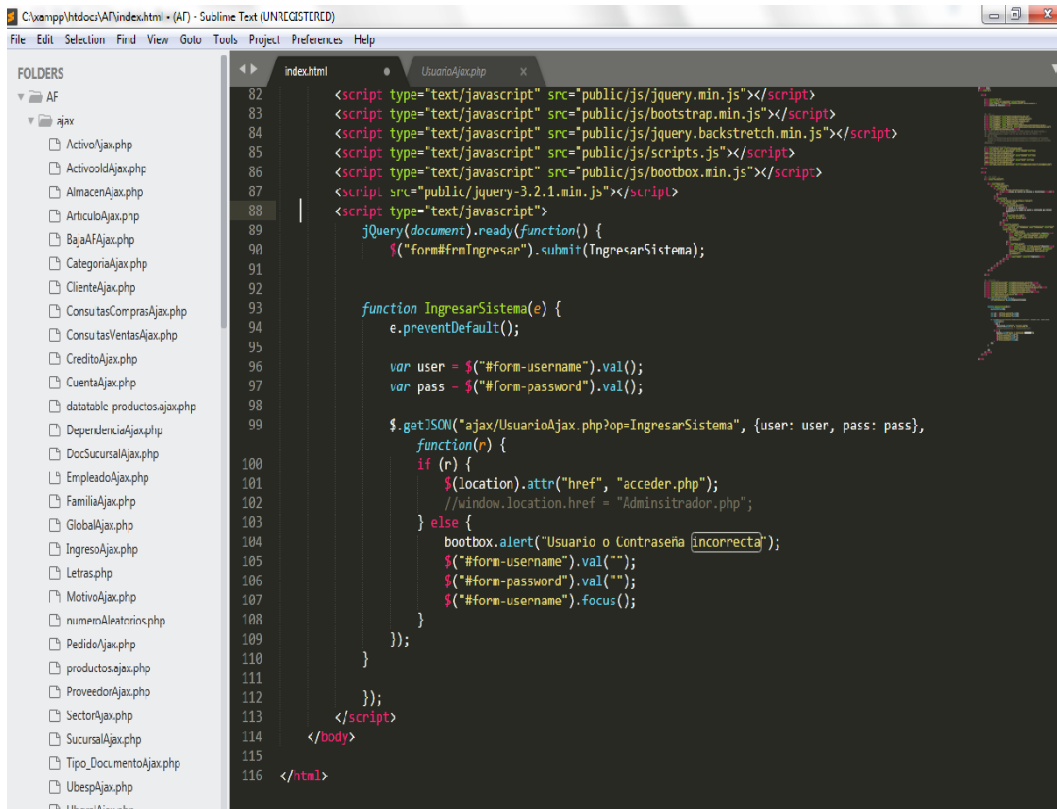
El módulo de Logueo del usuario, consiste en 2 eventos, el primero verifica que el usuario ya se encuentra registrado en la base de datos, surgiendo con lo anterior la primera estructura de decisión del módulo ya que si el usuario no se encuentra registrado procederá a un formulario para tomar los datos principales del usuario como lo son:

- Nombre del Usuario
- Apellido del Usuario
- Clave identificación del Usuario
- Telefónico

Si por el contrario el usuario ya se encuentra registrado procederá al Ingreso al sistema.

A continuación, se verificará que lo anteriormente descrito se esté cumpliendo en cada una de las líneas de código.

fig. 3 25 Función ingresar código fuente

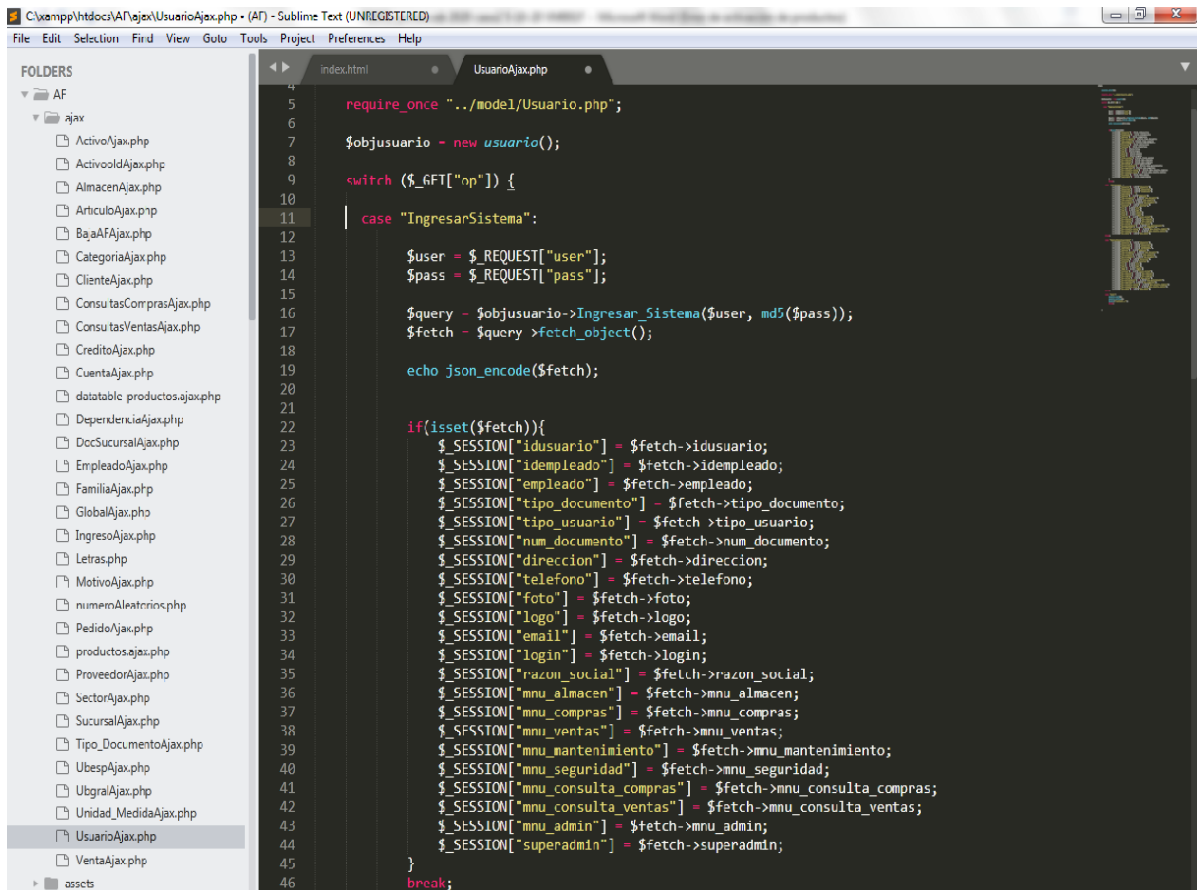


```
82 <script type="text/javascript" src="public/js/jquery.min.js"></script>
83 <script type="text/javascript" src="public/js/bootstrap.min.js"></script>
84 <script type="text/javascript" src="public/js/jquery.backstretch.min.js"></script>
85 <script type="text/javascript" src="public/js/scripts.js"></script>
86 <script type="text/javascript" src="public/js/bootbox.min.js"></script>
87 <script src="public/jquery-3.2.1.min.js"></script>
88 <script type="text/javascript">
89     jQuery(document).ready(function() {
90         $("#frmIngresar").submit(IngresarSistema);
91     });
92
93     function IngresarSistema(e) {
94         e.preventDefault();
95
96         var user = $("#form-username").val();
97         var pass = $("#form-password").val();
98
99         $.getJSON("ajax/UsuarioAjax.php?op=IngresarSistema", {user: user, pass: pass},
100             function(r) {
101                 if (r) {
102                     $(location).attr("href", "acceder.php");
103                     //window.location.href = "Administrador.php";
104                 } else {
105                     bootbox.alert("Usuario o Contraseña incorrecta");
106                     $("#form-username").val("");
107                     $("#form-password").val("");
108                     $("#form-username").focus();
109                 }
110             });
111     }
112 </script>
113 </body>
114 </html>
115
116 </html>
```

Al darle clic en el botón Ingresar el deberá validar si la identificación del Usuario está en la base de datos para eso deberá remitir a la capa de “ajax/UsuariosAjax.php?op=IngresoSistema” para realizar esa información.

Debido a lo anterior y para un mejor performance de la página la función **Ingreso Sistema** que está realizada en una función Ajax le enviara la tarea de verificar Usuario para que sea validada por medio de la función Switch.

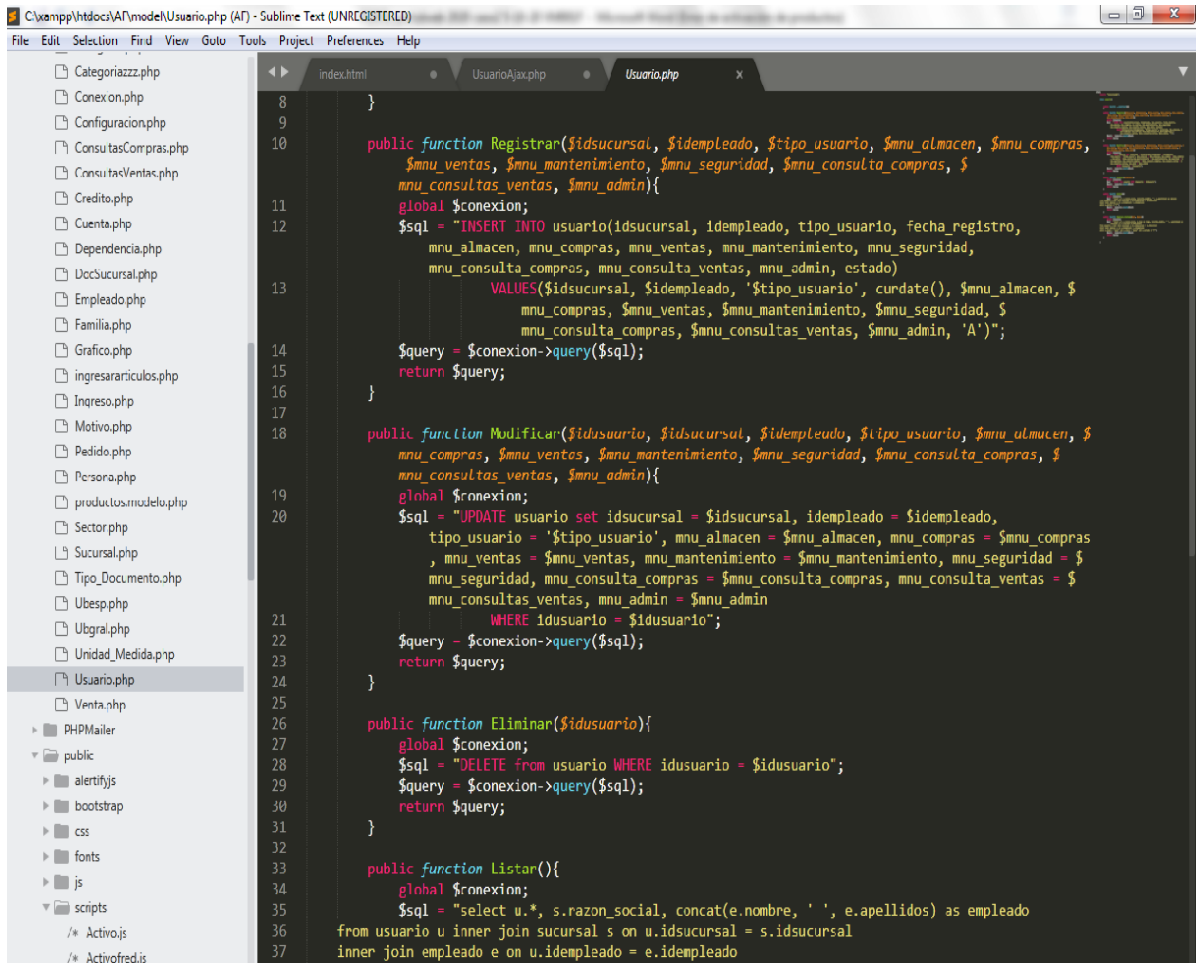
fig. 3 26 condigo fuente función ingreso con Ajax



```
4
5 require_once "../model/Usuario.php";
6
7 $objusuario = new usuario();
8
9 switch ($_GET["op"]) {
10
11     case "IngresarSistema":
12
13         $user = $_REQUEST["user"];
14         $pass = $_REQUEST["pass"];
15
16         $query = $objusuario->Ingresar_Sistema($user, md5($pass));
17         $fetch = $query->fetch_object();
18
19         echo json_encode($fetch);
20
21
22         if(isset($fetch)){
23             $_SESSION["idusuario"] = $fetch->idusuario;
24             $_SESSION["idempleado"] = $fetch->idempleado;
25             $_SESSION["empleado"] = $fetch->empleado;
26             $_SESSION["tipo_documento"] = $fetch->tipo_documento;
27             $_SESSION["tipo_usuario"] = $fetch->tipo_usuario;
28             $_SESSION["num_documento"] = $fetch->num_documento;
29             $_SESSION["direccion"] = $fetch->direccion;
30             $_SESSION["telefono"] = $fetch->telefono;
31             $_SESSION["foto"] = $fetch->foto;
32             $_SESSION["logo"] = $fetch->logo;
33             $_SESSION["email"] = $fetch->email;
34             $_SESSION["login"] = $fetch->login;
35             $_SESSION["razon_social"] = $fetch->razon_social;
36             $_SESSION["mnu_almacen"] = $fetch->mnu_almacen;
37             $_SESSION["mnu_compras"] = $fetch->mnu_compras;
38             $_SESSION["mnu_ventas"] = $fetch->mnu_ventas;
39             $_SESSION["mnu_mantenimiento"] = $fetch->mnu_mantenimiento;
40             $_SESSION["mnu_seguridad"] = $fetch->mnu_seguridad;
41             $_SESSION["mnu_consulta_compras"] = $fetch->mnu_consulta_compras;
42             $_SESSION["mnu_consulta_ventas"] = $fetch->mnu_consulta_ventas;
43             $_SESSION["mnu_admin"] = $fetch->mnu_admin;
44             $_SESSION["superadmin"] = $fetch->superadmin;
45         }
46         break;
```

La función Ingresar Sistema verifica el usuario para luego llamar el método de la clase **usuario** que realiza el Query a la base datos tomando como valor principal el id del usuario.

fig. 3 27 código fuente función registrar



```
8 }
9
10 public function Registrar($idsucursal, $idempleado, $tipo_usuario, $mnu_almacen, $mnu_compras,
    $mnu_ventas, $mnu_mantenimiento, $mnu_seguridad, $mnu_consulta_compras, $
    mnu_consultas_ventas, $mnu_admin){
11     global $conexion;
12     $sql = "INSERT INTO usuario(idsucursal, idempleado, tipo_usuario, fecha_registro,
        mnu_almacen, mnu_compras, mnu_ventas, mnu_mantenimiento, mnu_seguridad,
        mnu_consulta_compras, mnu_consulta_ventas, mnu_admin, estado)
13     VALUES($idsucursal, $idempleado, '$tipo_usuario', cdate(), $mnu_almacen, $
        mnu_compras, $mnu_ventas, $mnu_mantenimiento, $mnu_seguridad, $
        mnu_consulta_compras, $mnu_consultas_ventas, $mnu_admin, 'A')";
14     $query = $conexion->query($sql);
15     return $query;
16 }
17
18 public function Modificar($idusuario, $idsucursal, $idempleado, $tipo_usuario, $mnu_almacen, $
    mnu_compras, $mnu_ventas, $mnu_mantenimiento, $mnu_seguridad, $mnu_consulta_compras, $
    mnu_consultas_ventas, $mnu_admin){
19     global $conexion;
20     $sql = "UPDATE usuario set idsucursal = $idsucursal, idempleado = $idempleado,
        tipo_usuario = '$tipo_usuario', mnu_almacen = $mnu_almacen, mnu_compras = $mnu_compras
        , mnu_ventas = $mnu_ventas, mnu_mantenimiento = $mnu_mantenimiento, mnu_seguridad = $
        mnu_seguridad, mnu_consulta_compras = $mnu_consulta_compras, mnu_consulta_ventas = $
        mnu_consultas_ventas, mnu_admin = $mnu_admin
21     WHERE idusuario = $idusuario";
22     $query = $conexion->query($sql);
23     return $query;
24 }
25
26 public function Eliminar($idusuario){
27     global $conexion;
28     $sql = "DELETE from usuario WHERE idusuario = $idusuario";
29     $query = $conexion->query($sql);
30     return $query;
31 }
32
33 public function Listar(){
34     global $conexion;
35     $sql = "select u.*, s.razon_social, concat(e.nombre, ' ', e.apellidos) as empleado
36     from usuario u inner join sucursal s on u.idsucursal = s.idsucursal
37     inner join empleado e on u.idempleado = e.idempleado
```

El método realizar **Logueo** recibe la identificación del usuario, realiza la consulta en la Base de Datos y dependiendo de la Consulta envía mensaje de “**Usuario o Contraseña Incorrecta**”, Caso contrario envía lo datos relevantes del usuario en variables a través de Ajax para su posterior registro.

CAPITULO IV

4 Calidad y seguridad

4.1 Evaluación de calidad

Necesidad de evaluar para comprender y mejorar

Muchas características y atributos de calidad de artefactos Web tales como usabilidad, navegabilidad, seguridad, características de búsquedas y recuperación de información, mantenibilidad, performance, accesibilidad, no-deficiencia, y aspectos de funcionalidades específicas del dominio, por citar algunas, deben ser sistemáticamente tenidas en cuenta durante el proceso de desarrollo de modo de garantizar la calidad en consideración de la satisfacción de los requerimientos de los diferentes perfiles de usuario.

4.1.2 Definiendo metas de evaluación

La meta principal es “verificar que el presente proyecto tiene un nivel de cumplimiento en cuanto a los atributos y, características de calidad de la aplicación” PORTAL WEB PARA CONTROL DE ALMACENES Y ACTIVOS FIJOS.”

4.2 Metodología de evaluación de calidad para aplicaciones web

Para el proceso de evaluación se ha tomado en cuenta el perfil de usuario anteriormente mencionado, para ello se tomara en cuenta las cuatro características que plantea Olsina los cuales son:

- Usabilidad
- Funcionalidad
- Confiabilidad.

- Eficiencia.

- **Usabilidad.** Grado en que el software es fácil de usar. Viene reflejado por los siguientes subatributos: facilidad de comprensión, facilidad de aprendizaje y operatividad.
- **Funcionalidad.** El grado en que el software satisface las necesidades indicadas por los siguientes subatributos: idoneidad, corrección, interoperabilidad, conformidad y seguridad.
- **Confiabilidad.** Cantidad de tiempo que el software está disponible para su uso. Está referido por los siguientes subatributos: madurez, tolerancia a fallos y facilidad de recuperación.
- **Eficiencia.** Grado en que el software hace optimo el uso de los recursos del sistema. Está indicado por los siguientes subatributos: tiempo de uso y Recursos utilizados.

A continuación se detalla el árbol de requerimientos de calidad correspondiente al dominio Portal Web para el control de almacenes y activos fijos.

Árbol de requerimientos de calidad para el sistema de control de almacenes

Tabla 4 1 Usabilidad

CÓDIGO	DESCRIPCIÓN
1.1	Comprensibilidad Global del Sitio
1.1.1	Esquema de Organización Global
1.1.1.1	<i>Mapa del Sitio</i>
1.1.1.2	<i>Tabla de Contenidos</i>
1.1.1.3	<i>Índice Alfabético</i>
1.1.2	<i>Calidad en el Sistema de Etiquetado Visita</i>
1.1.3	<i>Guiada Orientada al cliente Mapa de</i>
1.1.4	<i>Imagen (Campus/Edificio)</i>
1.2	Mecanismos de Ayuda y Retroalimentación en línea
1.2.1	Calidad de la Ayuda
1.2.1.1	<i>Ayuda Explicatoria Orientada al Usuario</i>
1.2.1.2	<i>Ayuda de la Búsqueda</i>
1.2.2	Indicador de Última Actualización
1.2.2.1	<i>Global (de todo el sitio Web)</i>
1.2.3	<i>Restringido (por subsitio o página)</i>
1.2.3.1	Directorio de Direcciones <i>Directorio E-</i>
1.2.3.2	<i>mail</i>
1.2.3.3	<i>Directorio TE-Fax</i>
1.2.4	<i>Directorio Correo Postal</i>
1.2.5	<i>Facilidad FAQ</i>
1.2.5.1	Retroalimentación
1.2.5.2	<i>Cuestionario</i>
1.2.5.3	<i>Libro de Invitados</i>
1.3	Aspectos
1.3.1	<i>Comentarios/Sugerencias</i>
1.3.2	de Interfaces y Estéticos
1.3.2.1	<i>Cohesividad al Agrupar los Objetos de Control Principales</i>
1.3.2.2	<i>Permanencia y Estabilidad en la Presentación de los Controles Principales</i>
1.3.3	<i>Permanencia de Controles Directos</i>
1.3.3.1	<i>Permanencia de Controles Indirectos</i>
1.3.3.2	<i>Aspectos de Estilo</i>
1.3.3.3	
1.3.4	
1.4	
1.4.1	
1.4.2	

Tabla 4 2 Funcionalidad

CODIGO	DESCRIPCION
2.1	Aspectos de Búsqueda y Recuperación
2.1.1	Mecanismo de Búsqueda en el Sitio Web
2.1.1.1	Búsqueda Restringida
2.1.1.1.1	<i>de Perforaciones de</i>
2.1.1.1.2	
2.1.1.1.3	<i>Comunidades de</i>
2.1.1.2	<i>Instituciones</i>
2.1.2	
2.1.2.1	<i>Búsqueda Global</i>
2.1.2.2	
2.2	Mecanismos de Recuperación
2.2.1	<i>Nivel de Personalización</i>
2.2.1.1	<i>Nivel de Retroalimentación en la Recuperación</i> Aspectos
2.2.1.1.1	de Navegación y Exploración Navegabilidad
2.2.1.1.2	Orientación
2.2.1.2	<i>Indicador del Camino</i>
2.2.2	<i>Etiqueta de la Posición Actual Promedio</i>
2.2.2.1	<i>de Enlaces por Página</i> Objetos de
2.2.2.1.1	Control Navegacional
2.2.2.1.2	Permanencia y Estabilidad en la Presentación de los Controles Contextuales
2.2.2.2	(Subsitio)
2.2.2.2.1	<i>Permanencia de los Controles Contextuales</i>
2.2.2.2.2	<i>Estabilidad</i>
2.2.3	Nivel de Desplazamiento
2.2.3.1	
2.2.3.2	<i>Desplazamiento Vertical</i>
2.3	<i>Desplazamiento Horizontal</i>
2.3.1	
2.3.1.1	Predicción Navegacional
2.3.1.1.1	
2.3.1.1.2	<i>Enlace con Título (enlace con texto explicatorio)</i>
2.3.1.2	<i>Calidad de la Frase del Enlace</i>
2.3.1.2.1	Aspectos del Dominio orientados al Estudiante
2.3.1.2.2	Relevancia de Contenido
2.3.1.3	Información de la Dirección
2.3.1.3.1	<i>Índice de las Áreas que comprende la DSBVI Sub-</i>
	<i>sitios de las Áreas.</i>
	Información de Registro de datos de cada una de las Áreas.
	<i>Información de los Requerimientos de Ingreso</i>
	<i>Formulario para Rellenar/Bajar</i>
	Información de las Perforaciones realizadas en las Comunidades.

Tabla 4 3 Confidencialidad

Codigo	Descripción
2.3.1.3.2	<i>Índice de Áreas</i>
2.3.1.3.3	<i>Descripción de Comunidades</i>
2.3.1.3.4	<i>Plan de Perforaciones en las Comunidades</i>
2.3.1.3.4.1	Descripción de Requisitos.
2.3.1.3.4.3	<i>Comentarios</i>
2.3.1.4	<i>Programación Seminarios</i>
2.3.1.4.1	Información de Servicios que se le da a la Comunidad.
2.3.1.4.2	<i>Información de</i>
2.3.1.4.3	<i>Convenios Información</i>
2.3.1.4.4	<i>de Salud Información de</i>
2.3.1.4.5	<i>Tarifas</i>
2.3.1.5	<i>Información de Seminarios y Conferencias</i>
2.3.1.5.1	<i>Información de publicaciones.</i>
2.3.1.5.2	Información de Infraestructura
2.3.2	<i>Información de Bibliotecas</i>
2.3.2.1.	<i>Información de Laboratorios</i>
2.3.2.2	Servicios On-line
2.3.2.3	<i>Que serán aprobadas.</i>
2.3.2.4	<i>Servicio de Páginas Web</i>
	<i>Servicio FTP</i>
	<i>Servicio de Grupo de Noticias</i>

Tabla 4 4 Eficiencia

CODIGO	DESCRIPCION
4.1	Performancia
4.1.1	<i>Páginas de Acceso Rápido</i>
4.2	Accesibilidad
4.2.1	Accesibilidad de Información
4.2.1.1	<i>Soporte a Versión sólo Texto</i>
4.2.1.2	Legibilidad al desactivar la Propiedad Imagen del Browser
4.2.1.2.1	<i>Imagen con Título Legibilidad</i>
4.2.1.2.2	
4.2.2	<i>Global Accesibilidad de Ventanas</i>
4.2.2.1	
4.2.2.2	<i>Número de Vistas considerando</i>
	<i>Marcos (frames) Versión sin</i>
	<i>Marcos</i>

4.3 Análisis de costo

En la actualidad tenemos diferentes herramientas para hacer el análisis de los costos

En la que se hace una relación entre el esfuerzo del trabajo y el tiempo de desarrollo.

4.3.1 Método de estimación de costos COCOMO II

En el ámbito de la ingeniería de software, la estimación de costos radica básicamente en estimar la cantidad necesaria de personas para hacer el desarrollo de software. A diferencia de otras disciplinas de la ingeniería, en las cuales, el costo de los materiales es el primer componente a ser estimado, así para nuestro proyecto utilizaremos cocomoII.

Donde el método más utilizado, y donde trabaja en función del tamaño del software tomando como parámetro las líneas de código, interpretadas en KLDC es decir kilo líneas de código. Al analizar las líneas de código de mi proyecto de grado con nombre Portal Web para el Control de Almacenes y Activos Fijos caso Lab. ESFASA y realizar la sumatoria de todos los lenguajes, framework, base de datos.

Líneas de código del proyecto: 23893

$$KLDC = \frac{LDC}{1000}$$

$$KLDC = \frac{23893}{1000} = 23,893$$

$$KLDC = (23,893)KLDC$$

Con el primer dato obtenido KLDC voy a empezar a estimar el costo y los beneficios vamos a utilizar las siguientes variables, ecuaciones, en unidades y tiempo.

Tabla 4 5 Ecuaciones de COCOMO

VARIABLE	ECUACION	TIPO/UNIDAD
Esfuerzo requerido	$E = \alpha(KLDC)^b * FAE =$	Personas /mes
Tiempo Requerido	$T = C * (E)^d =$	Meses
Número de personas	$NP = E/Tdev$	Personas
Costo Total	$CT = \text{Sueldo Mes} \times NP \times T$	\$us

Donde:

E = es el esfuerzo requerido por el proyecto, en personas por mes.

T = es el tiempo requerido por el proyecto, en mes.

NP = es el número de personas que requiere el proyecto.

a, b, c, y d son las constantes con valores definidos, según cada sub modelo.

KLCD = es la cantidad de líneas de código, expresados en miles.

FAE = es el multiplicador que depende de la tabla de los 15 atributos

Tabla 4 6 Constantes de Complejidad

MODOS	A	B	C	D
Orgánico	2.40	1.05	2.50	0.38
Semi-libre	3.00	1.12	2.50	0.35
Rígido	3.60	1.20	2.50	0.32

Para el caculo de variable FAE utilizaremos la tabla de 15 atributos, que según el factor de las especificaciones de experiencia del personal inmiscuido, el hardware y los aspectos sobre la logística del trabajo es la siguiente.

Tabla 4 7 Factores de Trabajo

FACTOR		MUY BAJO	BAJO	NORMAL	ALTO	MUY ALTO	EXTRA ALTO
RELY	Fiabilidad	0.75	0.88	1.00	1.15	1.40	X
DATA	Tamaño de BD	X	0.94	1.00	1.08	1.16	X
CPLX	Complejidad	0.70	0.85	1.00	1.15	1.30	1.65
TIME	Limitaciones en el porcentaje de uso	X	X	1.00	1.11	1.30	1.66
STOR	Restricción del almacenamiento principal	X	X	1.00	1.06	1.21	1.56
VIRT	Volatilidad de la plataforma	X	0.87	1.00	1.15	1.30	X
TURN	Tiempo de respuesta requerido	X	0.87	1.00	1.07	1.15	X
ACAP	Capacidad del analista	1.46	1.19	1.00	0.86	0.71	X
AEXP	Experiencia en la aplicación	1.29	1.13	1.00	0.91	0.82	X
PCAP	Capacidad del programador	1.42	1.17	1.00	0.86	0.70	X

VEXP	Experiencia en plataforma	1.21	1.10	1.00	0.90	X	X
LEXP	Experiencia en el lenguaje herramientas	1.24	1.10	1.00	0.95	0.82	X
MODP	Uso de prácticas modernas en programación	1.24	1.10	1.00	0.91	0.83	X
TOOL	Uso de herramientas de software	1.24	1.10	1.00	0.91	0.83	X
SCED	Cronograma requerido para el desarrollo	1.22	1.08	1.00	1.04	1.10	X

TOTAL, FAE = 0.371

Calculando el esfuerzo:

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

$$E = 2.40 * (23.893)^{1.05} * 0.371 = 24.932 \text{ personas por mes}$$

$$E = 24.94 \text{ (personas/Mes)}$$

Calculando el tiempo de Desarrollo

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

$$T = 2.50 * (24,932)^{0.38} =$$

$$T = 8,49 \text{ meses}$$

Estimando el tiempo aproximado de desarrollo en meses a 8 meses.

Calculando promedio

$$NP = \frac{E}{T} =$$

$$NP = \frac{24,932}{8,49} = 2,938$$

NP = 2,938 Equivale / Personas

Estimando el salario promedio mínimo actual de un profesional programador Bs. 4000, número que será tomado en cuenta para la estimación:

CT= Sueldo Mes *NP * T

CT = 4000 * 3 * 8

CT = 96000 Bs.

Por lo tanto el proyecto estima aproximadamente, 3 personas, con 8 meses de trabajo por un costo total de Desarrollo de Sistema de Bs. 96000.

4.3 Pruebas y resultados

Para manifestar los resultados de los procesos que tenía antes de la implementación, del proyecto Portal Web para el Control de Almacenes y Activos fijos caso: LAB. ESFASA S.R.L. Haremos una tabla de comparación:

Tabla 4 8 Resultados del Sistema

<p>Tiempo de respuesta antes</p>	<p>Tiempo de respuesta Después</p>	<p>Descripción</p>
<p>El Gerente solicita a almacén Producto terminado informe de existencia la, demora es larga (20 min) ya que almacenero llena informe manual.</p>	<p>La respuesta es rápida y solo tarda 2 min. El sistema ya devuelve la solicitud vía interna o vía web.</p>	<p>Adecuadamente El sistema no presenta problemas de redundancia y a corto plazo el informe es recibido</p>
<p>El almacenero ingresa compras al almacén y debe, llenar boletas de ingreso la demora de esa situación es de (30 min.)</p>	<p>Para esta situación el sistema ya está listo solo debe pulsar el botón ingreso y anotar el código, tiempo de respuesta 5 min.</p>	<p>con la sistematización los ingresos a almacén solo se anotan la fecha y el código</p>
<p>El laboratorista hace pedido al almacén, insumos para hacer un producto la demora es de 30 min. Ya que antes el almacenero debe llenar una boleta de salida. Y hacer la respectiva autorización al Gerente.</p>	<p>La respuesta es rápida porque el sistema ya tiene los datos de insumos y el tiempo para este proceso es de 2 min.</p>	<p>El sistema automáticamente verifica la cantidad solicitada y brinda un reporte con las cantidades solicitadas</p>
<p>El cliente hace un pedido de un fármaco x por teléfono la demora y el costo la llamada es grande y dicha situación es 1 hora</p>	<p>Con el sistema implementado la respuesta de, 10 min. El cliente solo debe Loguearse y verificar la mercadería solicitada puede hacer un pedido vía web y generar una boleta de pedido</p>	<p>Todos los actores inmiscuidos en el proceso del Portal web tendrán una respuesta ágil y rápida para ventas compras y reportes de precios vía web</p>

CAP V

5. Conclusiones y Recomendaciones

5.1 Conclusiones.

Después de haber finalizado con el desarrollo del proyecto tomando en cuenta la problemática inicial y los objetivos planteados se puede afirmar que se ha alcanzado la meta trazada, las pruebas necesarias del sistema Portal Web para el control de almacenes y activos fijos caso: Lab. ESFASA logrando los requerimientos de la institución.

El entorno tecnológico para desarrollar el sistema, resulto muy apropiado y no presentaron inconvenientes importantes durante el desarrollo del mismo, que podrían contradecir la elección realizada. Si no más bien hicieron de sí mismo como un solo en la inclusión de los principios de Usabilidad.

Así mismo el lenguaje de programación PHP permitió el desarrollo de un software con una interface amigable con el encargado de almacenes, para lo cual quedo satisfecho por la meta alcanzada para beneficio de los usuarios.

Tabla 5 1 Objetivos y Conclusiones

OBJETIVOS	CONCLUSIONES
- Diseñar una base de datos que contenga la información completa sobre las compras, ventas y activos fijos seguro y confiable.	Mediante el sistema agilizaremos los Movimientos de los almacenes en sus Diferentes reparticiones.
diseñar un sistema de reportes sobre compras y ventas, que estará al alcance del usuario con soporte web en cualquier momento y lugar.	el sistema entregara rápidamente los reportes diarios de las compras y ventas a todos los usuarios que así lo requieran

- Diseñar un sistema de fácil manejo con una interfaz amigable.	el sistema devuelve informes detallados de los requerimientos de los usuarios de la plataforma los 365 días del año
- Diseñar un sistema de datos de activos fijos para mejorar prestaciones de los muebles e inmuebles.	La interfaz de administración y usuarios Esta lista para servir a todos
Diseñar la red y configurar el servidor web de LAB. ESFASA S.R.L.	La base de datos activos fijos esta lista para Realizar cualquier consulta sobre cualquier Mueble, maquinaria, objeto. Etc.

5.2 Recomendaciones

El trabajo desarrollado denominado “PORTAL WEB PARA EL CONTROL DE ALMACENES Y ACTIVOS FIJOS CASO LAB. ESFASA S.R.L. fue implementado en las instalaciones de la institución misma

Los requisitos de la entidad fueron satisfechos con el presente proyecto, pero todavía quedan otros requisitos no contemplados en el presente trabajo.

- Implementar un sistema estadístico de todo el sistema, para tener la información más relevante, como por ejemplo el producto más vendido en el mes.
- Como el sistema fue desarrollado con una metodología ágil es adaptable y se puede agregar otros elementos al sistema web.

Bibliografía

- Desarrollo de aplicaciones web Ing. Carles Matéu por la UOC 2004
Ingeniería del software: metodologías y ciclos de vida
Ingeniería de software séptima edición (Ian Sommerville) Pearson educación.
S.A. Madrid 2005.
- Sergio Blanco Cuaresma, ingeniero informático, analista programador,
- profesor en cursos sobre GNU/Linux (iniciación, avanzado y programación)
para universitarios / profesionales y apasionado por el Software Libre/Open
Source en general.
Instituto nacional de tecnologías de la comunicación: marzo 2009
- Ingeniería del software y especificaciones
Dolors costal- M Ribera sancho Ernst Teniente
- Ingeniería de Software: una guía para crear sistemas de información
Alejandro Peña Ayala
(Pressman, 2002) Pressman, R (2002): ingeniería de sistemas. MEXICO.
D.F.:
EDITORIAL MCgrall- Hill, 5ta edición.
- [CITON, 2006]CITON María Laura, Metodología Ágil Scrum aplicado al de
un software de trazabilidad 2006.
- [LADINO, 2010] LADINO Aricapa Martha Isabel, Mecanismo De Consulta En
Línea Sobre Programación Extrema (Xp), Scrum Y Cristal, Metodologías
Agiles Para El Desarrollo De Software, Dirigido A Estudiantes Y
Profesionales. Octubre – 2007.

[LONE, 2011] LONE Sáenz Carlos Augusto, Aplicación de los principios de administración de proyectos del PMI a metodologías Agiles de desarrollo de software. San José, Costa Rica- 2011.

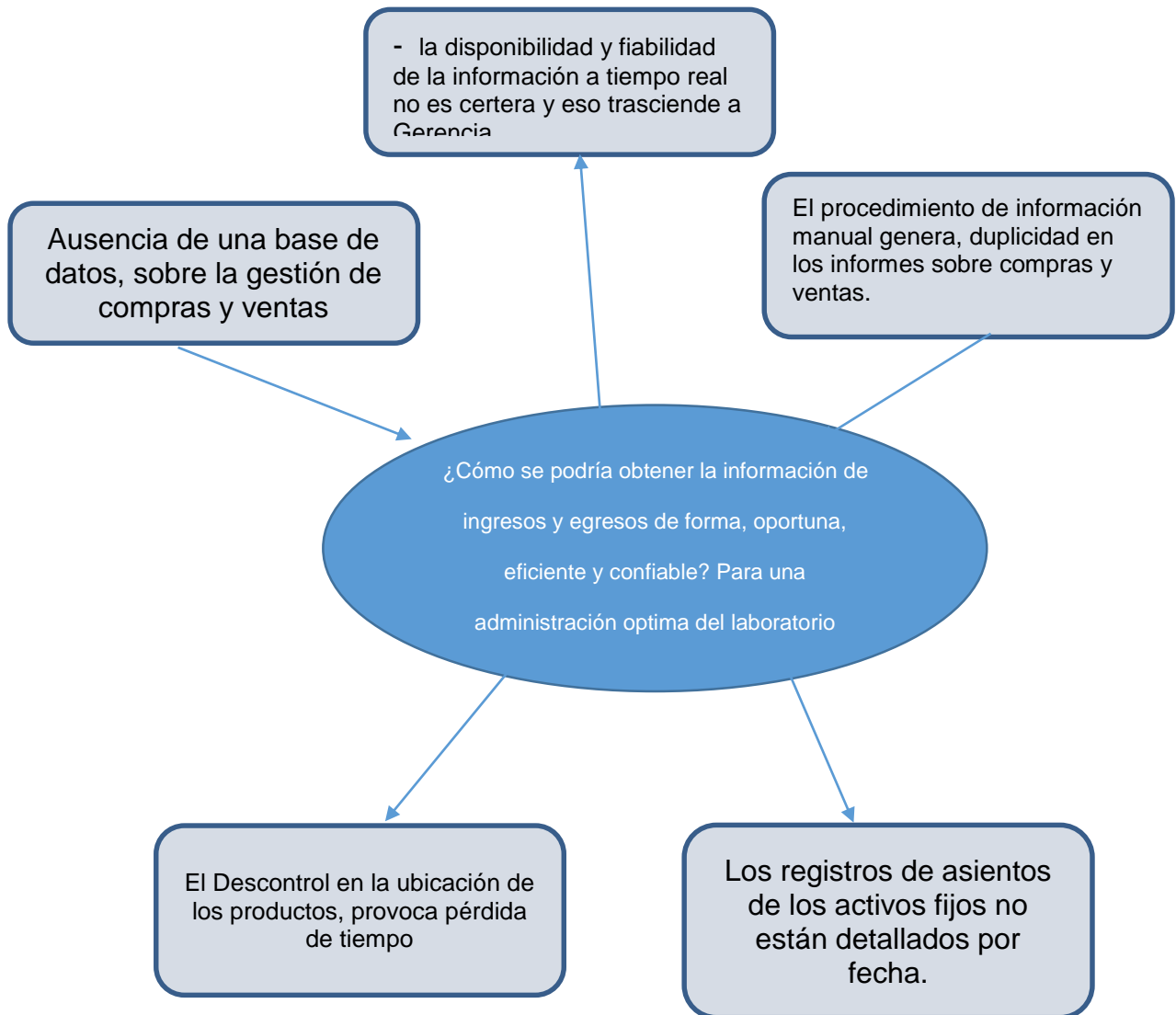
[MITARITONNA, 2010] MITARITONNA, Alejandro Daniel, Una innovadora metodología para el desarrollo de software en ambientes de trabajo virtuales. Argentina, Buenos Aires 2010.

[NÚÑEZ, 2010] NÚÑEZ Morí José German, Usabilidad en metodologías Agiles España, Madrid – 2010

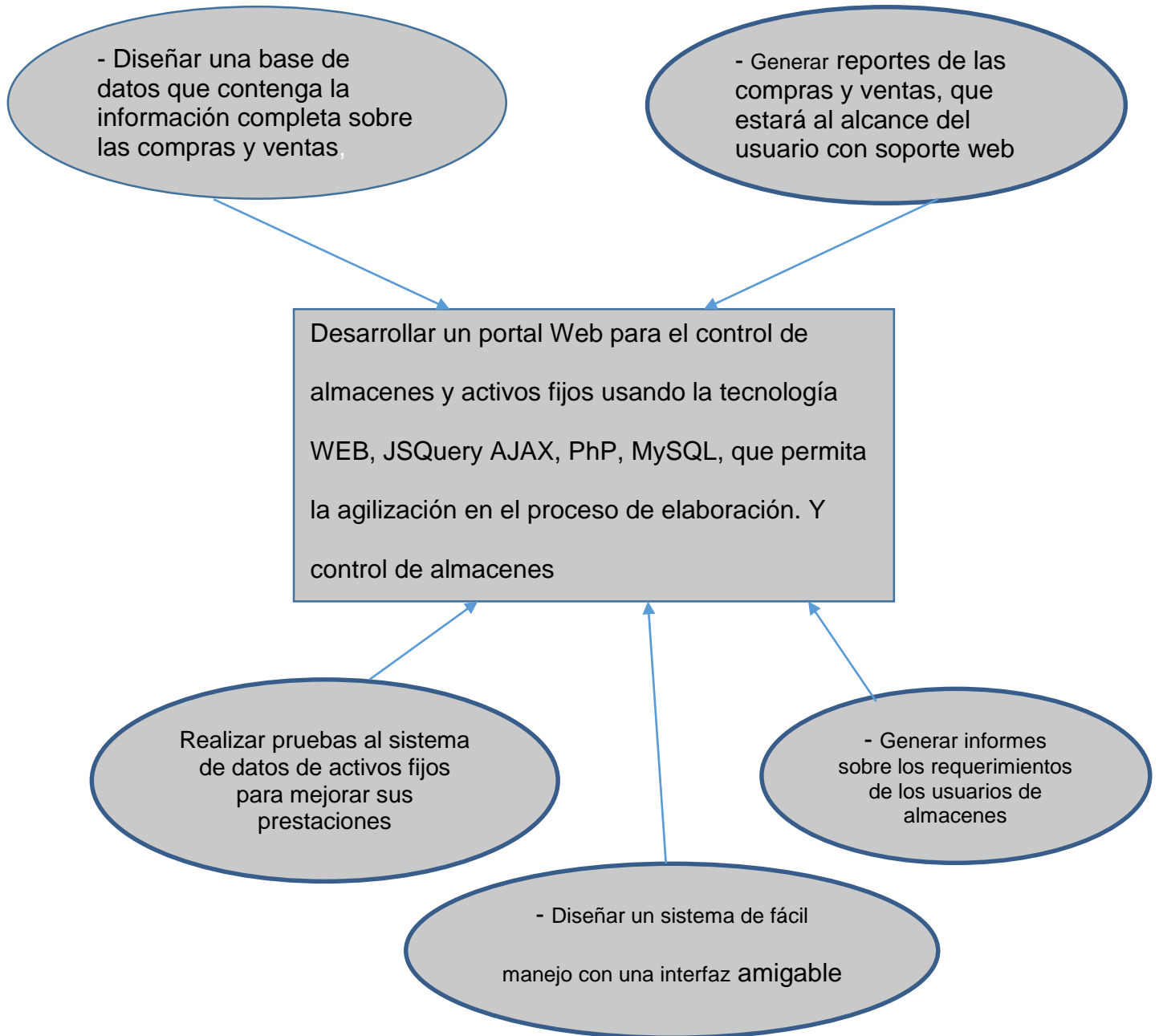
ANEXO A

Anexo A

A.1. árbol de problemas



A2 Árbol de Objetivos



ANEXO B

Anexo B1

CUESTIONARIO DE ENTREVISTA

NOMBRE.....

OCUPACION.....

FECHA.../.../...

Analizando perfil de usuario del sistema

1. ¿cuál es su principal función?
2. ¿en la cadena de producción que actividad realiza?
3. ¿Para quién?
4. ¿Cómo determina la superación de lo que hace?
5. ¿Qué problemas obstaculizan en su tarea?

Analizando el problema

6. ¿qué problema tiene en su tarea para resolver?
7. ¿Por qué cree que existe el problema?
8. ¿Cuáles son las aptitudes del sistema para resolver el problema actualmente?
9. ¿tiene alguna idea de cómo resolver el problema?

10. ¿cree que ayudaría cambiar de sistema?

Analizando y deduciendo la situación del usuario

11. ¿Qué tipo de entidades participan en la elaboración de fármacos?

12. ¿los usuarios tienen experiencias en el manejo de aplicaciones web?

13. ¿tienen alguna aplicación que ayude a la administración del Laboratorio?

14. ¿Qué sistema operativo utilizan en su institución?

15. ¿Cuáles son las expectativas con respecto a tener una aplicación al servicio de su empresa?

16. ¿Qué tipos de información impresa utiliza su empresa?

Síntesis de relevamiento del problema

17. Con sus propias palabras describa el sistema que actualmente utilizan para resolver el problema.

Estimando las soluciones del analista

18. ¿Qué pensaría si lográramos resolver el problema, desarrollando e implementando un portal web para controlar y gestionar los ingresos y egresos de su almacén en un tiempo prudente y eficaz para mejorar las tareas de su empresa?

ANEXO C

ANEXO C1

MANUAL DEL USUARIO Y DEL ADMINISTRADOR

El presente manual tiene la finalidad de dar a conocer sobre la manera de utilizar

El portal web para el control de almacenes y activos fijos caso: Lab. ESFASA

S.R.L.

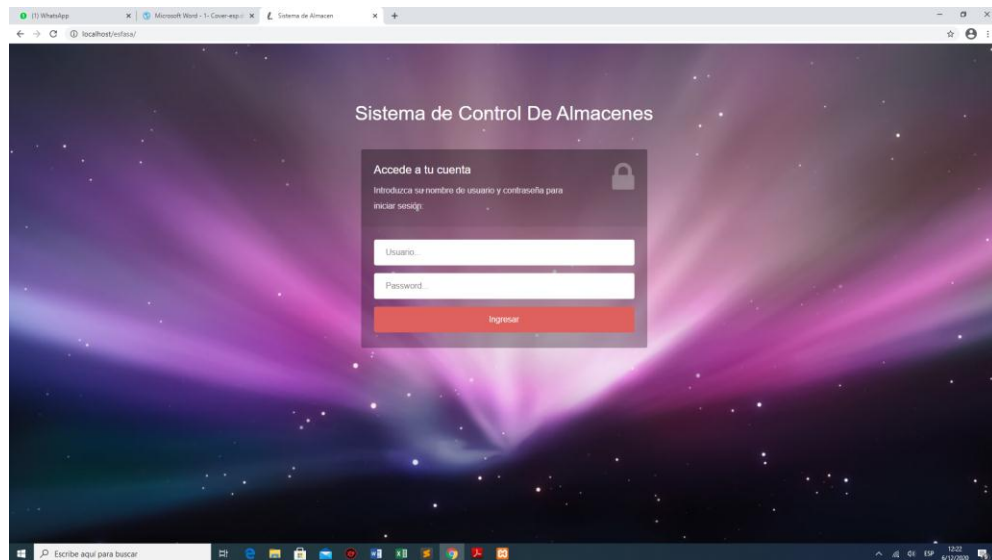
Requisitos técnicos:

- Utilizar o tener un navegador web con particularidad (google Chrome, Mozilla u opera.
- Importante tener conexión a internet.
- Estar registrado en el portal.

ACCESO AL SISTEMA:

Para ingresar al portal web debe dirigirse a la siguiente URL

<http://localhost/esfasa/>



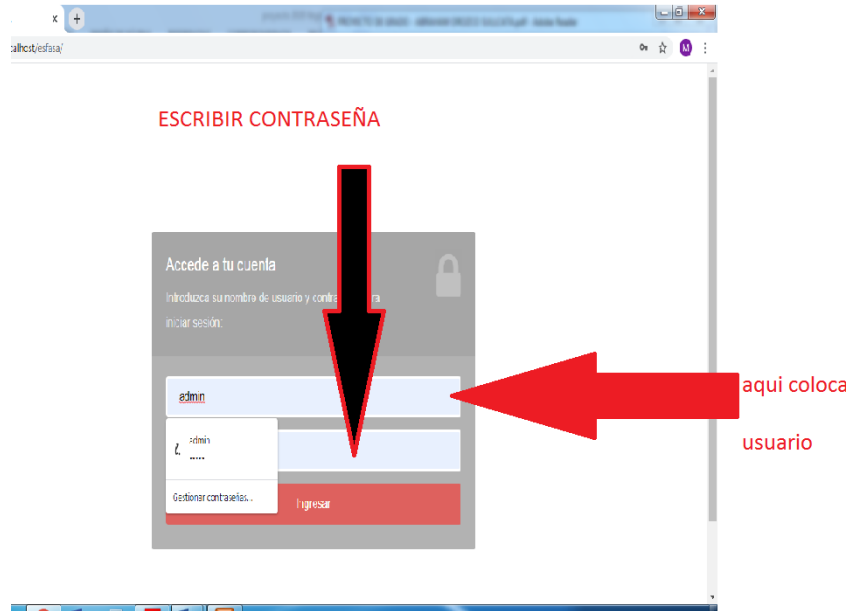
MANUAL DEL USUARIO Y ADMINISTRADOR



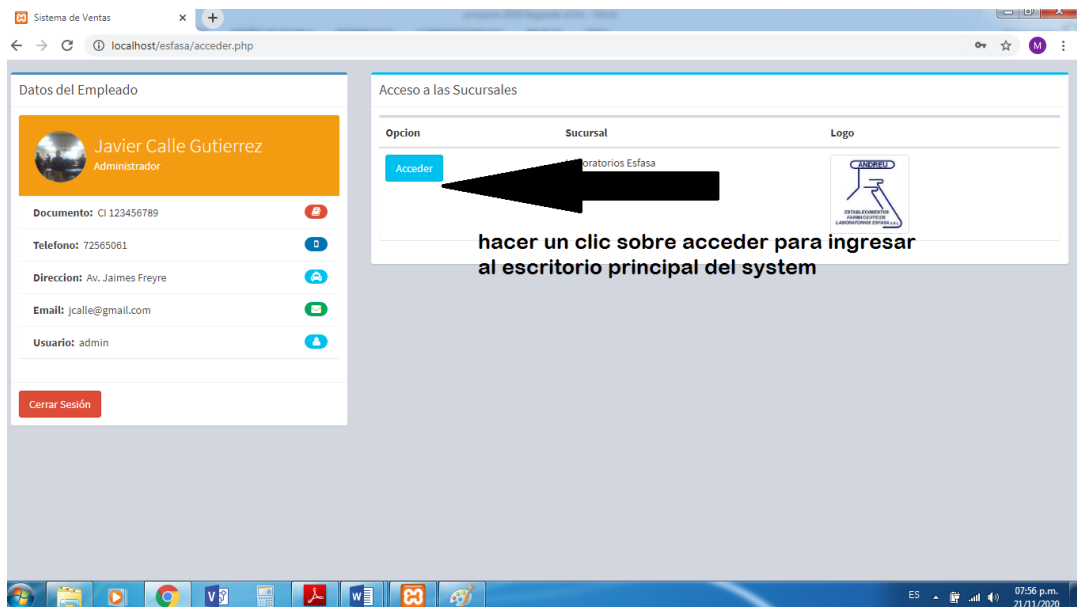
**Portal web para el control de almacenes
y activos fijos caso: LAB. ESFASA**

**UNIV. JAVIER CALLE
GUTIERREZ**

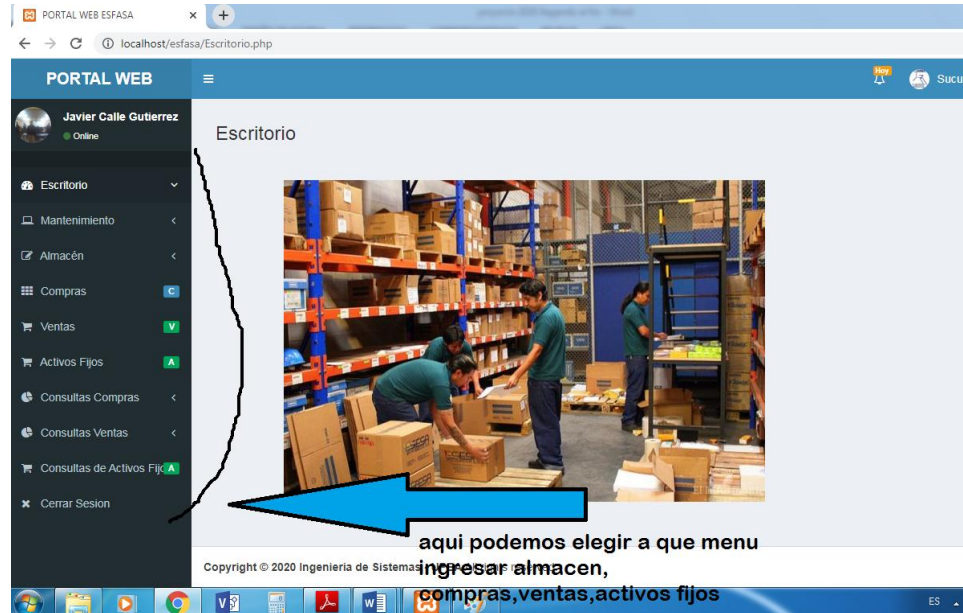
1.- Para empezar el manejo del sistema debe loguearse en la siguiente pantalla:



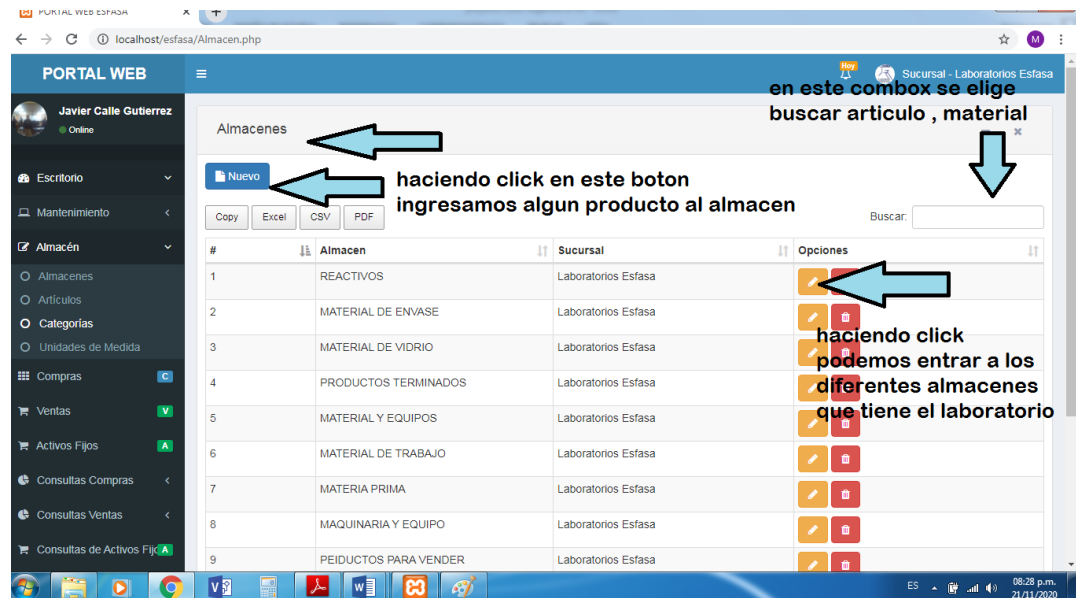
2.- el siguiente paso es ingresar al sistema



3.- El escritorio tiene en lado izquierdo los menús



4.- para ingresar por ejemplo almacén solo tenemos que posicionar el mouse sobre la opción y la manito señalara y también el menú se iluminara.



5.- ingreso al menú artículos donde podemos consultar los artículos que tienen en existencia.

este boton es para ingresar un nuevo articulo al almacen

este boton es de consulta para ver un articulo

el menú articulos se pinta de blanco indicando la visita

#	Categoría	Cod_Ubi	U. Medida	Artículo	Descripción	Imagen	Opciones
1	PRODUCTOS TERMINADOS	0005	Kilo Gramos	hepadif	medicamento		
2	MATERIA PRIMA	0004	Kilo Gramos	sosa caustica	solido		
3	MATERIA PRIMA	0003	Kilo Gramos	acido clorhidrico	reactivo		
4	MATERIAL DE ENVASE	pro08emcgi	Kilo Gramos	Cajas ginecodif	Caja de carton ginecodif		

6.- En esta pantalla mostraremos el ingreso a almacén para buscar proveedores como artículos en sus diferentes almacenes además esta con el impuesto de iva, la fecha actual de ingreso.

con este boton buscamos tambien articulos en almacen

aqui hacemos la busqueda de proveedores

Almacén: REACTIVOS Fecha Ingreso: 22/11/2020 Impuesto IVA: 13.00 % Departamento:

Tipo Comprobante: TICKET Número: Serie/Folio:

Proveedor: Seleccione un Proveedor Q Buscar Q Buscar Artículos

Artículo	Codigo	Medida	Descripción	Stock Ingreso	P. Compra	P. Venta Distribuidor	P. Venta Publico	Opción
----------	--------	--------	-------------	---------------	-----------	-----------------------	------------------	--------

Copyright © 2020 Ingeniería de Sistemas - UPEA All rights reserved. Version 1.0.0

7.- en este menú se registra a los proveedores de las diferentes industrias que abastecen de insumos al LABORATORIO ESFASA

para ingresar al formulario de registro de proveedores

este boton digital para imprimir reportes

aquí para ver datos del proveedor

#	Proveedor	Documento	Email	Teléfono	Dirección	Opciones
1	ABAQUIM	CEDULA 2466751	abaquim@gmail.com	2416623	LA PAZ	[Opciones]
2	Quimica Industrial newman	CEDULA 452256	quimica@mail.com	2455656	santa cruz	[Opciones]
3	Inversiones Nuevo León S.A.C.	RUC 2	nuevomexico@gmail.com	9674565889	Mexico	[Opciones]
4	Importaciones Santa Ana S.A.C.	NIC 12581369852	carlos@gmail.com	257896	Lambayeque	[Opciones]

Mostrando 1 a 4 de 4 entradas

Copyright © 2020 Ingeniería de Sistemas - UPEA All rights reserved. Version 1.0.0

8.- menú ventas ahí registramos nuevas ventas, detalle de ventas y reportes sobre las ventas.

busqueda de ventas

haciendo click en este boton se realiza la venta

ingreso a ventas detalladas

impresion de reporte haciendo click en boton digital

#	Cliente	T. Pedido	Fecha	Total	Estado	Opciones
1	FREDDY DOMINGO BLANCO	Venta	2020-11-07	1.00	CANCELADO	[Opciones]
2	FREDDY DOMINGO BLANCO	Venta	2019-05-22	25.00	CANCELADO	[Opciones]
3	FREDDY DOMINGO BLANCO	Venta	2019-05-20	50.00	CANCELADO	[Opciones]
4	FREDDY DOMINGO BLANCO	Venta	2019-05-12	5.00	CANCELADO	[Opciones]
5	FREDDY DOMINGO BLANCO	Venta	2019-05-11	200.00	ACEPTADO	[Opciones]
6	Javier Calle Gutierrez	Venta	2019-05-10	2.00	ACEPTADO	[Opciones]
7	proyecto 2020 llegando al fin - Word	Venta	2019-05-07	205.00	ACEPTADO	[Opciones]
8		Venta	2019-05-07	615.00	ACEPTADO	[Opciones]
9		Venta	2019-05-07	1000.00	ACEPTADO	[Opciones]

9.- ingreso al menú pedidos, aquí se registra los pedidos nuevos, también imprime reportes hace vista al detalle, y hace búsqueda de pedidos.

ingreso de nuevo pedido

busqueda de un pedido

lista el detalle del pedido

imprime reporte

#	Cliente	T. Pedido	Fecha	Generar Venta
1	FREDDY DOMINGO BLANCO	Pedido	2020-11-08	[+][x][i][X]
2	FREDDY DOMINGO BLANCO	Pedido	2020-11-07	[+][x][i][X]
#	Cliente	T. Pedido	Fecha	

10.- ingreso al menú clientes donde se hace su registro en el sistema

ingreso de nuevos clientes

busqueda por pagina

detalle del cliente registrado

#	Cliente	Documento	Email	Teléfono	Dirección	Opciones
1	Javier Calle Gutierrez	CI 4218722	jcalle@gmail.com	71915302	La Paz	[+][x]
2	FREDDY DOMINGO BLANCO	DNI 75489623	jcarlos.ad7@gmail.com	245862	Chiclayo	[+][x]
#	Cliente	Documento	Email	Teléfono		

11.- ingreso al menú créditos para verificar si algún cliente tiene crédito del Laboratorio.

PORTAL WEB

Javier Calle Gutierrez Online

Escritorio

Mantenimiento

Almacén

Compras

Ventas

Créditos

Deudas Pendientes

Conf. Comprobantes

Activos Fijos

Consultas Compras

Consultas Ventas

Créditos

Copy Excel CSV PDF

Buscar:

#	Tipo de Venta	T. Comprobante	Serie	Numero	Fecha	Impuesto	Total	Total Pagado	Total Deuda	Opcion
1	Credito	FACTURA	28283	2002	2019-05-12	0.00	0.00	0.00	0	

Mostrando 1 a 1 de 1 entradas

Anterior 1 Siguiente

12.- menú deuda pendiente donde se verifica si un cliente tiene alguna deuda.

PORTAL WEB

Javier Calle Gutierrez Online

Escritorio

Mantenimiento

Almacén

Compras

Ventas

Créditos

Deudas Pendientes

Conf. Comprobantes

Activos Fijos

Consultas Compras

Consultas Ventas

Deudas Pendientes

Copy Excel CSV PDF

aqui se consulta si algun cliente tiene deuda

car:

#	Tipo de Venta	T. Comprobante	Serie	Numero	Fecha	Impuesto	Total	Total Pagado	Total Deuda	Opcion
No hay datos disponibles en la tabla										

Mostrando 0 a 0 de 0 entradas

Anterior Siguiente

13.- ingreso a menú comprobantes donde se genera esta clase de documentos.

Comprobantes

Nuevo genera nuevos comprobantes

Copy Excel CSV PDF

Buscar:

#	T. Documento	Ultima Serie	Ultimo Numero	Opciones
1	FACTURA	123	2343436	[Opciones]
#	T. Documento	Ultima Serie	Ultimo Numero	Opciones

Mostrando 1 a 1 de 1 entradas

Anterior 1 Siguiente

14.- ingreso al menú activos fijos donde se registra la compra, y de ahí el sistema calcula su vida útil en el tiempo.

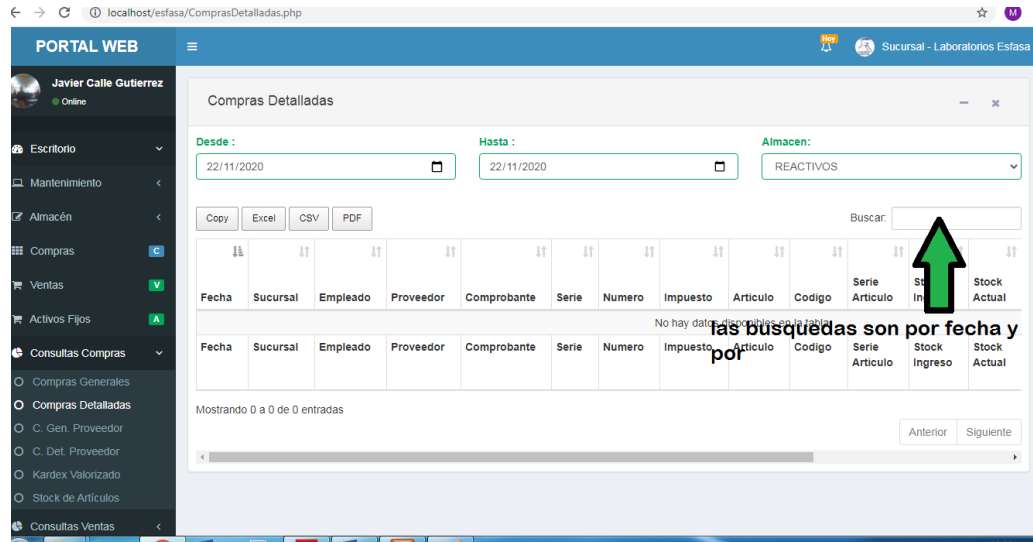
Administrar Activos

Inicio / Administrar Activos

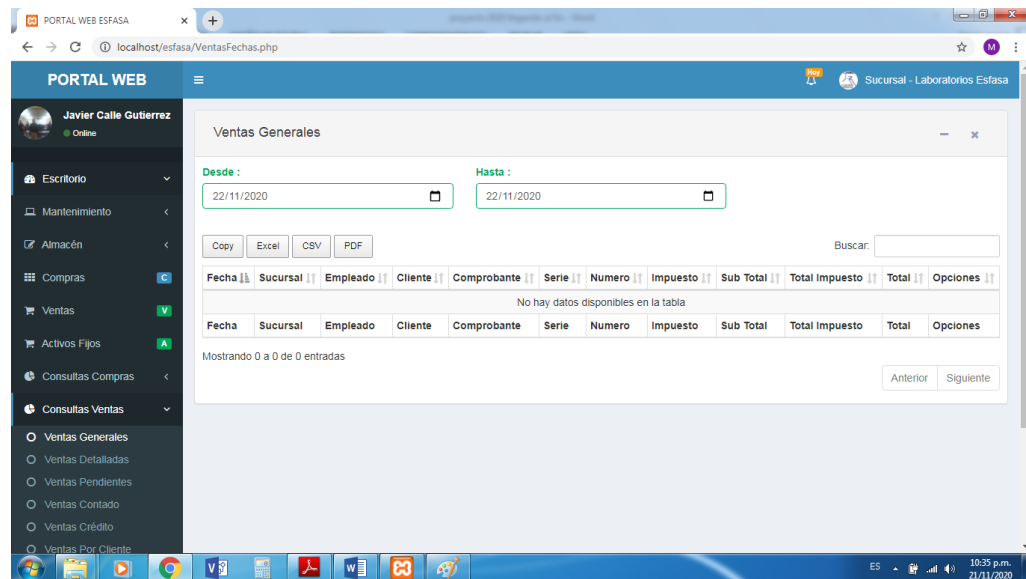
Codigo	Unidad	Descripción	Fecha Compra	Valor	Vida Util	Acciones Editar
SCCHSU.C01.001	Pza	FAX Panasonic negro mod. KXFT21LS Ser:9LARA004098	2004-12-31	4818.58	4.17	[Opciones]
SCCHSU.C06.001	Pza	Cámara fotográfica digital SONY Mod.OSCP72 Ser:904970	2004-06-24	3484	6.00	[Opciones]
SCCHSU.C09.001	Pza	Teléfono interno y externo digital PANASONIC mod.VB5411D	2004-12-31	775.6	3.80	[Opciones]
SCCHSU.C11.001	Pza	Equipo de sonido con dos parlantes SAMSUNG mod. VCDMP3 Ser: ITTWB00745P	2004-12-31	1507.7	6.00	[Opciones]
SCCHSU.C05.001	Pza	Televisor a color 27 SONY Mod.KV27HFR Ser:7323560 con su control remoto	2004-12-31	2921.86	3.30	[Opciones]
SCCHSU.C11.002	Pza	Radio casetera de 5 bandas SILVANO Mod. KC-5110	2005-01-31	1060.19	5.00	[Opciones]
SCCHSU.C09.002	Pza	Teléfono digital PANASONIC (línea externa e interna)	2004-12-31	775.6	4.60	[Opciones]

consulta de activo fijo sobre la fecha de adquisicion

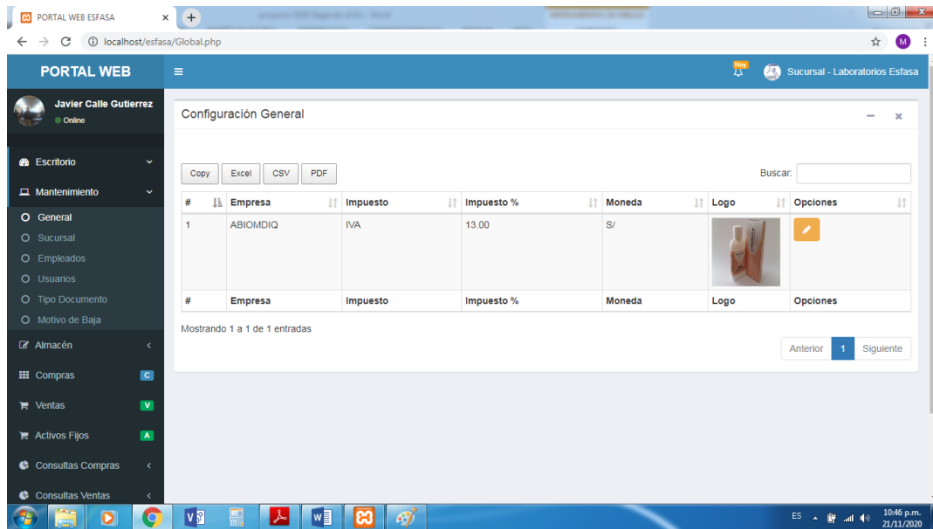
15.-menu consulta de compras donde se registra todo el manejo de compras es decir compras detalladas, compras generales, el stock, y el kardex.



16.- ingreso al menú ventas generales donde se puede consultar todo el movimiento de las ventas ejemplo: ventas al detalle, pendientes, al contado, al crédito, por cliente, por empleado, y por empresas



17.- gracias por su atención, están son las aptitudes del sistema portal web, además su interfaz amigable hace de que el administrador pueda configurarlo a su modo ya que en el menú mantenimiento, podemos de adicionar otras particularidades para acomodarse a cualquier requerimiento por ser un software muy amigable, portable



GRACIAS.

DICCIONARIO DE DATOS

Tablas relacionales Del portal web Lab. ESFASA S.R.L. su

composición de numero de tablas.

TABLA 1 SUCURSAL

CAMPO	TIPO	DESCRIPCION
Idsucursal	int	Clave unica
Razón_social	Varchar	Estado del comercio
Tipo_documento	varchar	Certificación de venta
Num_documento	varchar	Cedula identidad
direccion	varchar	Dirección sucursal
telefono	varchar	Teléfono de susucursal
email	varchar	Correo de susucursal
representante	varchar	Representante sucursal
estado	char	ciudad

TABLA 1 2 Usuario

CAMPO	TIPO	DESCRIPCION
idusuario	int	Clave única
idsucursal	int	Clave sucursal
idempleado	int	Clave única
Tipo_usuario	varchar	Usuario
Fecha_registro	date	Fecha

Mnu_almacen	bit	Código almacen
Mnu_compras	bit	Código compras
Mnu_ventas	bit	Código ventas
Mnu_mantenimiento	bit	Código manten.
Mnu_seguridad	bit	Código seguridad
Mnu_consulta_compras	bit	Código consulta y compras
Mnu_consulta_ventas	bit	Código consulta y ventas
Mnu_admin	bit	Código de admin.
Estado	char	Estado civil usuario

TABLA 1 3 Ingreso

CAMPO	TIPO	DESCRIPCION
idingreso	int	Claveunica
idusuario	int	Claveunica
idsucursal	int	Claveunica
idproveedor	int	Claveunica
Tipo_comprobante	varchar	Documento ingreso
Serie_comprobante	varchar	Numero identificador
Num_comprobante	varchar	Identificador de ingreso
fecha	date	Año
Impuesto	decimal	Interés
total	decimal	Ingreso totales
estado	varchar	Ciudad

TABLA 1 4 Persona

CAMPO	TIPO	DESCRIPCION
idpersona	int	Clave única
Tipo_persona	varchar	Empleado
nombre	varchar	Nombre de persona
Tipo_documento	varchar	Cedula persona
Num_documento	varchar	Numero carnet
Dirección_departamento	varchar	Domicilio persona
Dirección_provincia	varchar	Provincia
Dirección_distrito	varchar	Barrio persona
Dirección_calle	varchar	Domicilio persona
telefono	varchar	Numero
email	varchar	Correo persona
Numero_cuenta	varchar	Cuenta banco
estado	varchar	Estado civil

TABLA 1 5 Empleado

CAMPO	TIPO	DESCRIPCION
idempleado	int	Clave unica
apellidos	varchar	Dato personal
nombre	varchar	Nombre empleado
Tipo_documento	varchar	Cedula identidad
Num_documento	varchar	Código cedula

direccion	varchar	Domicilio empleado
telefono	varchar	Digito comunicación
email	varchar	Correo empleado
Fecha_nacimiento	date	Cumpleaños empleado
foto	varchar	Imagen
login	varchar	Contraseña
estado	varchar	Estado civil empleado
Clave	varchar	Contraseña

TABLA 1 6 Categoría

CAMPO	TIPO	DESCRIPCION
idcategoria	int	Clave única
nombre	varchar	Dato
estado	char	Activo

TABLA 1 7 Artículo

CAMPO	TIPO	DESCRIPCION
idarticulo	int	Clave única
idcategoria	int	Clave única
Idunidad_medida	int	Parámetro
nombre	varchar	Nombre objeto
descripcion	text	Detalle de articulo
imagen	varchar	Foto de articulo
estado	char	Tiempo de uso

TABLA 1 8 Detalle Ingreso

CAMPO	TIPO	DESCRIPCION
Iddetalle_ingreso	int	Clave única
idingreso	int	Clave única
idarticulo	int	Clave única
codigo	varchar	Valor
serie	varchar	Correlativo
descripcion	varchar	Detalle importante
Stock_ingreso	int	Numero
Stock_actual	int	numero
Precio_compra	decimal	Impuesto
Precio_ventadistribuidor	decimal	Porcentaje a la venta
Precio_ventapublico	decimal	Porcentaje

TABLA 1 9 Detalle Pedido

CAMPO	TIPO	DESCRIPCION
Iddetalle_pedido	int	Clave única
idpedido	int	Clave única
Iddetalle_ingreso	int	Clave única
cantidad	int	Proporción de pedido
Precio_venta	decimal	Impuesto
descuento	decimal	Porcentaje de rebaja

TABLA 1 10 Pedido

CAMPO	TIPO	DESCRIPCION
idpedido	int	Clave única
idcliente	int	Clave única
idusuario	int	Clave única
idsucursal	int	Clave única
Tipo_pedido	varchar	Objeto pedido
fecha	date	Dia especial

TABLA 1. 11 Venta

CAMPO	TIPO	DESCRIPCION
idventa	int	Clave única
idpedido	int	Clave única
idusuario	int	Clave única
Tipo_venta	varchar	Contado o crédito
Tipo_comprobante	varchar	Boleta reporte
Serie_comprobante	varchar	Numero definido
Num_comprobante	varchar	Numero unico
fecha	Date	Dia especia
impuesto	decimal	Interés
total	Decimal	Impuesto a la venta
estado	varchar	civil

TABLA 1.12 Unidad de Medida

CAMPO	TIPO	DESCRIPCION
Idunidad_medida	int	Clave única
Nombre	varchar	Denominación de una persona
prefijo	varchar	Nombre exacto
estado	char	Obsoleto

TABLA 1 13 Crédito

CAMPO	TIPO	DESCRIPCION
idcredito	int	Clave única
idventa	int	Clave única
Fecha_pago	date	Dia de pago
Total_pago	decimal	Impuesto

TABLA 1 14 Detalle Documento Sucursal

CAMPO	TIPO	DESCRIPCION
Iddetalle_documento_sucursal	int	Clave única
idsucursal	int	Claveunica
Idtipo_documento	varchar	Clave de tabla
Ultima_serie	varchar	Numero aleatorio
Ultimo_numero	varchar	Numero elegido

TABLA 1.15 Tipo de Documento

CAMPO	TIPO	DESCRIPCION
ldtipo_documento	int	Clave única
nombre	varchar	Primer nombre
operacion	varchar	Almacen

TABLA 1.16 Global

CAMPO	TIPO	DESCRIPCION
idglobal	int	Clave única
empresa	varchar	Razón social
Nombre_impuesto	varchar	Iva
Porcentaje_impuesto	decimal	Tasa
Símbolo_moneda	varchar	Valor
logo	varchar	etiqueta

Tabla 1 17 activos

CAMPO	TIPO	DESCRIPCION
actid	nt(11)	Clave primaria
idsucursal	int(11)	clave primaria
actctaid	int(11)	clave primaria
actcorre	int(11)	Clave primaria

actcodigo	varchar(35)	Clave primaria
Actunidad	Varchar(35)	Lugar especifico
actdescrip	Varchar (35)	Descripción de lugar
actfechac	date	Periodo
actvalorh	float	valor de:
acttc	Decimal(11.2)	
estado	Varchar (10)	Bueno o malo
procedencia	Varchar(35)	De donde:
fuelle	Varchar (35)	
Valhist1	int	Valor de af
deprec	float	Precio de af
valres	float	Valor residual
sino	int	Solo uno

Tabla1 18 asignación

CAMPO	TIPO	DESCRIPCION
asigid	Int(11)	asignacion
actid	Int(11)	Identificaion del activo
admid	Int(11)	Identificion del adminis
asigfecha	date	Fecha de asignacion
Estado	Char(1)	Malo o bueno

Tabla 1 19 bajas

CAMPO	TIPO	DESCRIPCION
actid	int(11)	identificacion
numero	int(11)	valor
actcodigo	text	codigo
actunidad	text	cuantos
actdescripcion	varchar(35)	descripcion de:
actfechac	date	Periodo
actvalorh	float	Valor
actvidautil	float	Vida útil
actvalorr	float	Activo su valor
acttc	float	Tipo de cambio
estado	text	Bueno o malo
procedencia	Varchar(35)	De donde:
fuelle	Varchar(35)	
actfechab	date	Periodo
bajald	smallint	
observ	Varchar(150)	Detalles
acttcb	float	

Tabla 1 20 Estado

campo	tipo	descripcion
idestado	int(11)	estado
estado		

ANEXO D