

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



PROYECTO DE GRADO

**“SISTEMA DE CONTROL DE INVENTARIOS CON ALERTAS
TEMPRANAS PARA PEDIDOS EN LINEA**

CASO: ASOCIACIÓN MIXTA DE ORFEBRES Y RELOJEROS A.M.O.R.”

**PARA OPTAR AL TÍTULO DE LICENCIATURA EN INFORMÁTICA
MENCIÓN: INGENIERIA DE SISTEMAS INFORMÁTICOS**

POSTULANTE: UNIV. JORGE ARMANDO DURAN MORALES
TUTOR METODOLÓGICO: LIC. JAVIER HUGO REYES PACHECO
ASESOR: LIC. JUAN GONZALO CONTRERAS CANDIA

LA PAZ – BOLIVIA

2016



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



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

LICENCIA DE USO

El usuario está autorizado a:

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

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

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

DEDICATORIA:

Este trabajo está dedicado a mis padres, Don Jorge Durán y Doña Amalia Morales, y a mis hermanos Melina, Rosa, Gastón, por sus consejos, comprensión y apoyo constante.

AGRADECIMIENTOS:

Quiero agradecer a Lic. Reyes y Lic. Contreras por la colaboración muy importante en la corrección de este proyecto, así también a la Asociación de Mixta de Orfebres y Relojeros "A.M.O.R." por facilitarme toda la información requerida y finalmente a la Carrera de Informática por todo el conocimiento impartido a mi persona.

e-mail: duranmoralesjorge@gmail.com

RESÚMEN

La Asociación Mixta de Orfebres y Relojeros, ha venido desempeñando sus labores desde el año 1.955, durante ese periodo de seis décadas ha ido evolucionando las técnicas de la orfebrería, además la globalización gracias al avance tecnológico, han hecho posible la incursión de prendas y productos de gran valor y exquisita calidad de parte de los artesanos. Esta habilidad y destreza ha sido conocida a nivel mundial, incursionando varios de los artesanos a países del extranjero, sin embargo esta actividad económica en la que se involucran muchas familias de la urbe paceña, no ha estado rindiendo éxito en los últimos años, esto debido a que la competencia de otros países como Perú, Chile, Etc. Han incursionado y copiado muchos de los productos según señala el presidente de la institución. El poder llevar a sistema WEB, en donde los productos de muchos artesanos sean apreciados y se pueda alcanzar otros mercados, para ello fue necesario presentar el proyecto como un todo además de ser integral, en donde se controlaran los productos, se llevara un control de todos los productos, previendo la cantidad que se requerirá de parte del comprador, este control al no tener llevaba a demorar los envíos y pedidos de los clientes, quienes optaban por hacerlo a otros artesanos, perdiendo la Asociación contratos por la falta de la implementación de un sistema Web. Gracias al diseño y desarrollo del mismo conjuntamente con la directiva se pudo alcanzar el objetivo planificado para este propósito. El desarrollo del presente proyecto ha sido avalado por el directorio quienes han mostrado su conformidad del mismo. Es necesario mencionar las técnicas que se han empleado para ello y metodologías empleadas para este fin.

ABSTRACT

The Joint Association of Goldsmiths and Watchmakers, has been performing their duties since 1955, during that period of six decades has evolved techniques jewelry, plus globalization thanks to technological advances have made possible the incursion of garments and products valuable and exquisite quality of the artisans. This ability and skill has been known worldwide, dabbling number of artisans to foreign countries, however this economic activity in which many families in the La Paz city are involved, has not been yielding success in recent years, this due to competition from other countries such as Peru, Chile, etc., they have dabbled and copied many of the products as noted by the president of the institution. The power take WEB system, where the products of many craftsmen are appreciated and can reach other markets, it was necessary to present the project as a whole as well as being an integral, where the products were controlled, control is take all products, providing the amount that will be required from the buyer, this control by not having had to delay shipments and customer orders, who chose to do other artisans, losing the Association contracts for the lack of implementation of a web system. Thanks to the design and development thereof together with the directive could achieve the objective planned for this purpose. The development of this project has been endorsed by the board who have shown their compliance thereof. It should be mentioned techniques have been employed for this purpose and methodologies employed for this purpose.

ÍNDICE

1.1	INTRODUCCIÓN.....	1
1.2	ANTECEDENTES.....	1
1.2.1	MISIÓN.....	3
1.2.2	VISIÓN.....	3
1.2.3	ANTECEDENTE DE TRABAJOS AFINES.....	4
1.2	PROBLEMA CENTRAL.....	7
1.2.1	PROBLEMAS ESPECIFICOS.....	7
1.3	OBJETIVO CENTRAL.....	8
1.3.1	OBJETIVO ESPECIFICOS.....	8
1.4	DEFINICIÓN DE VARIABLES.....	9
1.5	JUSTIFICACIÓN.....	11
1.5.1	SOCIAL.....	11
1.5.2	TÉCNICA.....	11
1.5.3	ECONÓMICA.....	11
1.6	APORTES.....	11
1.7	ALCANCES Y LÍMITES.....	12
1.7.1	ALCANCES.....	12
1.7.2	LÍMITES.....	13
2.	MARCO TEÓRICO.....	10
2.1	INGENIERÍA DE SOFTWARE.....	10
2.1.1	DEFINICIÓN DE SOFTWARE.....	10
2.1.2	DEFINICIÓN DE INGENIERÍA DE SOFTWARE.....	10
2.1.3	APLICACIONES DE INGENIERÍA DE SOFTWARE.....	11
2.2	METODOLOGÍAS DE DESARROLLO DE SOFTWARE.....	12
2.2.1	METODOLOGÍA ICONIX.....	12
2.2.1.1	CARACTERÍSTICAS DE LA METODOLOGÍA ICONIX.....	13
2.2.1.2	FASES DE LA METODOLOGÍA.....	13
2.3	LENGUAJE DE MODELADO UNIFICADO (UML).....	16
2.3.1	DIAGRAMAS DE LENGUAJE DE MODELADO UNIFICADO.....	16

2.4 PRUEBA DE SOFTWARE.....	20
2.4.1 PRUEBA DE CAJA BLANCA.....	20
2.4.2 PRUEBA DE CAJA NEGRA	21
2.5 INTELIGENCIA ARTIFICIAL.....	21
2.5.1 ÁREAS DE APLICACIÓN.....	22
2.5 COSTO DE SOFTWARE.....	23
2.5.1 MODELO BÁSICO.....	24
2.5.2 MODELO INTERMEDIO.....	25
2.5.3 ATRIBUTOS	26
2.6 METODOLOGÍA DE ANÁLISIS Y DISEÑO.....	27
2.7 PATRÓN DE ARQUITECTURA.....	27
2.7.1 PATRÓN MODELO VISTA CONTROLADOR (MVC)	27
2.8 ARQUITECTURA CLIENTE-SERVIDOR	29
2.8.1 FUNCIONAMIENTO CLIENTE/SERVIDOR	32
2.8.2 APLICACIÓN DEL LADO DEL SERVIDOR	33
2.9 DISEÑO DE LA PÁGINA Y SISTEMA WEB.....	34
2.9.1 SEGURIDAD DE UN SISTEMA WEB	35
2.9.2 TECNOLOGÍAS	36
2.10 LENGUAJE DE PROGRAMACIÓN PHP.....	38
2.10.1 PHP.....	39
2.10.1.1 CARACTERÍSTICAS DE PHP	39
2.10.1.2 VENTAJAS DE USAR PHP5.....	40
2.10.1.3 Desventajas de usar PHP5.....	40
2.10.2.1 Características de MYSQL	41
2.11 MANEJADOR DE BASE DE DATOS	41
2.11.1 MODELOS DE BASES DE DATOS	42
2.11.2 LA BASE DE DATOS	46
2.11.3 DISEÑO DE UNA BASE DE DATOS.....	47
2.11.3.1 ELEMENTOS DE UNA BASE DE DATOS	49
2.11.3.2 TIPOS DE BASE DE DATOS	49
2.11.4 SISTEMA DE GESTIÓN DE BASE DE DATOS (SGBD).....	50
2.11.4.1 FUNCIONES DE UN SISTEMA DE GESTIÓN DE BASE DE DATOS	50

2.12.1	¿QUÉ ES UN FRAMEWORK WEB?	51
2.12.2	PATRÓN MVC Y MODEL 2.....	52
2.12.3	TIPOS DE FRAMEWORK WEB	53
2.12.4	CARACTERÍSTICAS	54
2.12.5	¿QUÉ ES BOOTSTRAP Y CÓMO FUNCIONA EN EL DISEÑO WEB?.....	55
CAPÍTULO III MARCO PRÁCTICO		57
3.1	ANÁLISIS DE LA SITUACIÓN ACTUAL.....	57
3.1.1	Planteamiento del Problema.....	60
3.1.2	Descripción de la Situación Actual.....	60
3.1.2.1	Proceso General de Exportación de Productos	60
3.1.4	Conceptualización	61
3.2	DESARROLLO DEL SISTEMA.....	61
3.2.1	Análisis del Software	61
3.2.1.1	Definición de Requerimientos	61
3.2.1.2	Modelo de Caso de Uso	62
3.2.1.3	Especificación de Casos de Uso	64
3.2.1.4	Diagrama de Robustez	73
3.2.2	Diseño del Sistema WEB.....	76
3.2.4	Diseño del Software.....	76
3.2.4.1	Diseño de la arquitectura del sistema.....	77
3.2.4.2	Diagramas de Secuencia.....	77
3.2.4.3	Diagramas de Clases	81
3.2.4.4	Diseño Conceptual de la Base de Datos	82
3.2.4.5	Diseño Lógico de la Base de Datos.....	83
3.2.4.5	Diseño Navegacional.....	83
3.3	CALIDAD DEL SOFTWARE	87
3.3.1	Escalonamiento De Likert.....	87
3.3.3	Usabilidad.....	89
3.3.4	Flexibilidad.....	89
3.3.5	Adaptabilidad.....	89
3.3.6	La Seguridad	89
CAPÍTULO IV		92

ANALISIS COSTO- BENEFICIO	92
4.1 DETERMINACIÓN DE COSTOS.....	92
4.1.2 Costos de Información	92
4.1.3 Costos de Hardware.....	93
4.1.4 Costos de Elaboración del Software	94
4.1.5 Costos de Licencias de Software	95
4.1.6 Costo de desarrollo de software (Cocomo)	95
4.2 ANALISIS DE BENEFICIOS	97
4.2.1 Beneficios Intangibles.....	98
4.2.2 Beneficios Tangibles	98
CAPÍTULO V	99
CONCLUSIONES Y RECOMENDACIONES.....	99
5.1 CONCLUSIONES.....	99
5.2 RECOMENDACIONES.....	99
BIBLIOGRAFÍA.....	100
ANEXOS.....	102

CAPÍTULO I

GENERALIDADES

1.1 INTRODUCCIÓN

En la mayoría de las organizaciones existe la necesidad de gestionar la información de aquellos recursos para dar abastecimiento de herramientas, bienes y servicios en el momento en que una operación lo necesite; mucho más aun, en la actualidad, cuando se manejan grandes cantidades de datos, el tratamiento de la información automática se convierte en un aspecto realmente útil y necesario.

Hoy en día, el mundo vive la era del ciberespacio, el manejo de información vía web, en un mundo globalizado es casi imprescindible para una organización de negocios y/o servicios, hacerse conocer no solo en un nivel local, sino atravesar las fronteras y expandir sus prestaciones para el crecimiento de la organización. A medida que las empresas crecen y se desarrollan son más ineludibles las necesidades que se presentan día a día, como alcanzar máximos niveles de calidad, satisfacer las necesidades de sus clientes, promover nuevos productos en el mercado, posicionar y consolidar los ya existentes, buscar mayor participación en el mercado, acceder a la información desde cualquier lugar, momento o circunstancia, entre muchas otras.

1.2 ANTECEDENTES

La Asociación Mixta de Orfebres y Relojeros A.M.O.R, ubicada en la Ciudad de La Paz, es una organización privada que tiene 26 años de antigüedad en el mercado Nacional, brindando el servicio de joyas, venta de anillos en general, aretes, collares, etc. a los departamentos de La Paz, Cochabamba, Santa Cruz, Trinidad, Oruro, Tarija, Sucre y Potosí, también se dedica a la importación de materia prima para la elaboración de las mismas productos de otros Países como Brasil, Argentina, etc.

Además la Asociación Mixta de Orfebres y Relojeros A.M.O.R, posee un grupo de socios, también cuenta con sucursal que está ubicado en la ciudad de Santa Cruz y este depende de otro principal que se encuentra en la ciudad de La Paz, que es la

Oficina Central, estas oficinas no comparten información sobre las operaciones que se realizan actualmente respecto a la emisión de facturas que se da, a los clientes cuando realizan alguna compra, debido a que el sistema actual no está desarrollado para compartir información en red.

Además de contar con una Sucursal y Oficina Central, se cuenta con representantes legales que son los encargados de llevar a cabo las ventas de las joyas, a las empresas o personas, los representantes legales se encuentran ubicados en: Oficinas de la Plaza Alonzo de Mendoza, La Paz, y en los demás departamentos hay un representante des gremio, no se tiene un representante legal, debido a que este no representa un mercado atractivo para este negocio.

La venta de joyas labradas de manera artesanal se realiza a personas naturales y/o jurídicas (empresas), Las empresas que más frecuente realizan una compra son: las unidades educativas Matrimonios, Bautizos, etc. Para realizar las ventas se cuenta con socios, vendedores que son encargados de vender y distribuir insumos a la sucursal de Santa Cruz.

Además. Cuenta con departamentos que son los encargados de llevar una determinada función dentro la organización, como mejorar el desempeño humano, la calidad de productos, y como mejorar las ventas, etc.

Departamento de Ventas este se encarga de supervisar el trabajo de cada artesano, planifica el trabajo de ventas, estudia el mercado de trabajo y lleva el acumulado diario de las ventas. Se encarga también de verificar cuantos productos se han logrado vender durante el día, semana, mes, año. Verifica si hubo ingresos durante esos periodos o en caso que no fuera así se verifica si hubo pérdidas, y ver que ocasiono esas pérdidas durante esos periodos, en este se encuentra falencias en los resultados que muestra sobre los ingresos y egresos durante el periodo, el sistema actual que se tiene dentro la Empresa no logra mostrar con seguridad esos datos, que los administradores necesitan para poder llevar a cabo un funcionamiento favorable para la organización.

Departamento de Contabilidad se encarga del papeleo interno para dar entrada en el inventario a los insumos solicitados a los proveedores, comprueba también los

asientos y las sumas en las facturas y registra los detalles financieros de todas las operaciones. Al momento de llevar a cabo los estados financieros se genera malos resultados que el sistema actual muestra.

El Departamento Administrativo es el encargado de velar por los intereses de la Empresa, ve como se está administrando cada departamento, y los recursos que utiliza para cumplir con los fines y objetivos trazados por la organización, además como planificar, cada una de las actividades que va a realizar la Distribuidora durante un determinado periodo de tiempo.

Departamento de Comercialización es el encargado de controlar la existencia de mercaderías que están en almacén, Revisa las condiciones de la mercadería adquirida, ve como se distribuirá la mercadería a la sucursal de santa cruz, registra también las entradas y salidas de cada producto.

En este departamento se ve que no existe un control adecuado sobre la mercadería que se encuentra almacenada en almacén de la Distribuidora, no se logra saber a tiempo con que insumos cuenta y así para poder después realizar el pedido correspondiente a los proveedores.

Departamento de gerencia es donde se vigila el normal funcionamiento y cumplimiento de las labores de cada una de los departamentos anteriormente mencionados, es decir esta es el caudal, donde desembocan todos y cada uno de los por menores, para ser canalizados y efectuar la toma de decisiones que el caso amerite.

1.2.1 MISIÓN

La misión de la Asociación Mixta de Orfebres y Relojeros A.M.O.R, es de facilitar a los socios las técnicas, herramientas, material y demás asuntos relacionados en la creación de joyas para la comercialización en los mercados disponibles.

1.2.2 VISIÓN

En una organización privada de interés público, que presta servicios artesanales con fines de lucro, queriendo expandir las fronteras en donde se puedan comercializar sus productos a un precio justo y rentable para los mismos.

1. 2.3 ANTECEDENTE DE TRABAJOS AFINES

Realizando una revisión preliminar de trabajos afines, se pudo evidenciar que existen trabajos que se relacionan de alguna manera con el presente proyecto.

Título:	Sistema de información para el control de ventas y cobros y calidad de materia prima para el laboratorio "Blames".
Autor	Costas Delgadillo, Raúl Ricardo
Año:	2008
Objetivo:	Desarrolla sistema de información automatizada para el control de ventas cobros y cantidad de materia prima del laboratorio que facilite el proceso de información de diferentes departamentos de tomas
Carrera:	Ingeniería de Sistemas
Lugar:	Universidad Salesiana

Título:	Sistema de información vía Web para la promoción de servicios para EMISTUR
Autor:	Quisbert Saavedra Ingrid Nathaly
Año:	2010
Objetivo:	Colabora con el desarrollo de una adecuada estrategia de promoción y comunicación, de la empresa mejorando la asistencia que presta la empresa en cuanto a la emisión de información: Turística como de los servicios que ofrece y también mejorar el manejo de los datos que son de total importancia para la institución.
Carrera:	Informática
Lugar:	UMSA

Título:	Sistema de información basado en la web para la carrera de informática de la UMSA.
Autor:	Suseth Vania Romero Bedoya
Año:	2009
Objetivo:	Satisfacer requerimientos específicos de información y comunicación de la comunidad de informática de la universidad, las entidades con quienes mantiene una relación de trabajo, y el hecho de informar a todos los interesados que desean conocer los servicios que presta la Carrera de informática.
Carrera:	Informática.
Lugar:	UMSA

Sistema de inventarios

Con frecuencia los precios sufren variaciones en cada compra de producto o mercancía que se hace en el ciclo contable. Esto dificulta el fácil cálculo del costo de los productos vendidos y el de los productos disponibles. Para evitar estas dificultades existen varios tipos de sistemas, donde los más importantes son:

- ✓ Sistema de inventario Perpetuo.
- ✓ Sistema de inventario Periódico.

Sistema de inventario Perpetuo

Es aquel que lleva en continuo recuento con las existencias de productos en almacén, por medio de un registro detallado, donde se lleva los importes de unidades monetarias y cantidades físicas, es decir, requiere el mantenimiento o registro en los cuales indica el importe y cantidad de todos los productos, en cualquier momento.

Sistema de inventario Periódico

Por medio de este sistema de inventarios solo puede determinarse a través de un conteo o verificación física de los productos existentes. Cuando los inventarios de productos determinan solo el chequeo físico a intervalos específicos, se dice que es un inventario periódico, es decir, requiere contar la cantidad de los productos, en fecha determinada y posteriormente determinar el importe de los mismos.

Método de valuación de inventarios

Existe numerosas bases aceptables para. La valuación de inventarios, algunas de ellas se consideran aceptables solamente en circunstancias especiales, en tanto que otras son de aplicación general.

Entre las cuestiones relativas a la valuación de los inventarios, la principal importancia es la consistencia: La información contable debe ser obtenida mediante la aplicación de los mismos principios durante todo el periodo contable y diferentes periodos contables de manera que sea factible comparar los Estados Financieros de diferentes periodos y conocer la evolución de la entidad económica

Método Costo Promedio

Como su nombre lo indica la forma de determinarse es sobre la base de dividir el importe acumulado de las erogaciones aplicables entre el número de artículos adquiridos o producidos. El costo de los artículos disponibles para la venta se divide entre el total de las unidades. El promedio resultante se emplea entonces para valorizar el inventario final.

Método Primero en Entrar, Primero en Salir o “PEPS”

Este método identificado también como “PEPS”, se basa en el supuesto de que los primeros artículos o materias primas en entrar al almacén o a la producción son los primeros en salir de él se ha considerado conveniente este método porque da lugar a una evaluación del inventario concordante con la tendencia de los precios, puesto que se presume que el inventario está integrado por las compras más recientes y esta valorizado a los también más recientes la valorización sigue entonces la dependencia del mercado

Método Último en Entrar, Primero en Salir "UEPS"

Este método parte de la suposición de que las últimas entradas en el almacén o al proceso de producción, son los primeros artículos o materias primas en salir.

El método "UEPS", asigna los costos a los inventarios bajo el supuesto que las mercancías que se adquieren de ultimo son las primeras en utilizarse o venderse, por lo tanto el costo de las mercadería vendida quedara en los últimos precios de compra con que fueron adquiridos los artículos y de forma contraria, el inventario final es valorado a los precios de compra de cada artículo en el momento que se dio la misma

1.2 PROBLEMA CENTRAL

La unidad de almacén de la Asociación Mixta de Orfebres y Relojeros A.M.O.R, no cuenta con una administración, seguimiento de control de inventarios, debido a que el manejo de la información es realizada manualmente, imposibilitando acceder a la generación de informes, reportes solicitados, factura y entrega de pedidos, generando un retardo en la toma de decisiones, además existe dificultad con los pedidos que realiza cada vendedor cuando se encuentra fuera de la organización o fuera del departamento ya que los pedidos se hacen manualmente y generando demora en hacer los cálculos correspondientes del pedido.

1.2.1 PROBLEMAS ESPECIFICOS

- Los clientes se encuentran ubicados en distintos departamentos, provincias y zonas lo cual imposibilita la interacción entre los mismos y la distribuidora de forma directa.
- Llenado manual de toda la información que llega a almacén tanto de entrada como de salida, lo cual ocasiona demoras en el control del departamento de ventas.

- Pedidos innecesarios de productos, ocasionado por el control no oportuno de stock de almacenes.
- Demora en ubicar los productos existentes en la distribuidora, lo cual ocasiona que un vendedor tarde en ubicar dicho producto
- La solicitud de pedidos que realiza el vendedor es realizada manualmente, haciendo que el proceso de atención sea demoroso y tarde en la entrega del mismo a los clientes.
- No se cuenta con informe sobre los productos más vendidos y menos vendidos durante el día, semana y año esto afecta al departamento de comercialización que no pueda tomar una decisión adecuada.
- No se cuenta con un seguimiento en la entrega de pedidos, solo se mantiene un control de las mismas mediante la verificación de la firma del cliente en la nota de remisión.
- Se desconoce los productos existentes en cada distribuidora y no hay un control de adecuado de la emisión de factura.

1.3 OBJETIVO CENTRAL

Desarrollar un sistema de control de inventarios con alertas tempranas para pedidos en línea de la Asociación Mixta de Orfebres y relojeros A.M.O.R.

1.3.1 OBJETIVO ESPECIFICOS

- Desarrollar un formulario vía web para que las diferentes empresas y representantes legales, puedan realizar una solicitud de las joyas artesanales, productos de forma directa.
- Desarrollar un módulo que registre los productos que ingresan y salen del almacén

- Desarrollar un módulo que permita la generación de reportes de stock y movimiento de productos, para así evitar pedidos innecesarios y haciendo más eficiente los pedidos.
- Realizar opciones de búsqueda de los productos que se encuentran en el almacén, según su clasificación de una manera más eficiente y más rápida
- Realizar el seguimiento de solicitudes de productos que realiza cada vendedor,
- Proporcionar al departamento de comercialización un informe sobre los productos más vendidos y menos vendidos, a través del módulo de ventas y este permita que los encargados de este departamento puedan tomar una decisión acertada al caso.
- Desarrollar formularios de control de pedidos o reserva de los productos tanto de entradas y salidas.
- Desarrollar un módulo de gestión por sucursal, para controlar e integrar los productos existentes en cada una de las sucursales, así también tener control de la emisión factura.

1.4 DEFINICIÓN DE VARIABLES

- **Nivel de ingresos:** La empresa en el ejercicio de su actividad presta servicios y bienes al exterior. A cambio de ellos, percibe dinero o nacen derechos de cobro a su favor, que hará efectivos en las fechas estipuladas (Aparicio, 2001).
- **Nivel de Stock:** Cantidad de mercancías que se tienen en depósito. Cantidad de productos, materias primas, herramientas, etc., que es necesario tener almacenadas para compensar la diferencia entre el flujo del consumo y el de la producción. Constituye una inversión que permite asegurar en condiciones óptimas la continuidad de las ventas, las fabricaciones y la explotación normal de la empresa (Guardia, 2005).

- **Producto:** El producto es un conjunto de atributos que el consumidor considera que tiene un determinado bien para satisfacer sus necesidades o deseos (Bonta, 2001).
- **Venta:** La venta es una de las actividades más pretendidas por empresas, organizaciones o personas que ofrecen algo (productos, servicios u otros) en su mercado meta, debido a que su éxito depende directamente de la cantidad de veces que realicen ésta actividad, de lo bien que lo hagan y de cuán rentable les resulte hacerlo (Aparicio, 2011)

OPERACIONALIZACION DE VARIABLES

Variables	Dimensiones	Indicadores	Instrumento
Nivel de Ingreso	Perdida Ganancia	Decremento de capital Ingreso de Dinero	Observación Observación
Nivel de Stock	Existe en almacén No existe en almacén	Productos disponibles para la venta. Productos no disponibles para la venta	Observación Observación
Producto	Entradas	Aumenta la cantidad de productos	Libro de registro de entrada de productos
Venta	Número de ventas efectuadas en por día	Cantidad de productos vendidos durante el día	Libro de ventas

Tabla 1 OPERACIONALIZACION DE VARIABLES (elaboración propia)

1.5 JUSTIFICACIÓN

1.5.1 SOCIAL

La implementación del sistema propuesto beneficiara a todo el personal encargada de forma de los departamentos de almacenes comercialización y contabilidad, permitirá además que los distribuidores ,clientes y personas pertenecientes a la institución puedan contar con una información confiable oportuna y segura de todo el flujo de información de vital importancia para un amplio campo de actividades referente al crecimiento de mercado consumidor y las correspondientes proyecciones que se harán para la toma de toma futuras decisiones en beneficio de la organización.

1.5.2 TÉCNICA

El Proyecto a desarrollar, se realizara por la necesidad que tiene la Distribuidora, ya que no cuenta con un buen control de los insumos que se tiene en almacén, además de compartir la información que se genera en la sucursal de Santa Cruz, además de las ventas que se efectúan en un lapso de tiempo y control de empleados. Así optimizando el servicio que se presta.

1.5.3 ECONÓMICA

La parte económica es justificada ya que la Distribuidora mejorara sus ingresos económicos al tener información oportuna y precisa para la toma de decisiones, dando a conocer información del manejo de la compra y venta de los productos, mejorando los procesos al conocerse toda la información financiera.

1.6 APORTES

La implementación del presente proyecto será un aporte significativo a la Asociación Mixta de Orfebres y Relojeros A.M.O.R, ya que esta será de gran ayuda para la toma de decisiones del gerente, agilizando el trabajo de la empresa, debido a que el personal cuenta con conocimientos sobre el manejo de Software, sin embargo se

pretende dar una capacitación al personal encargado que trabajara directamente con toda la información. Por otra parte, el desarrollo de un sistema web que ayude a mejorar los procesos de producción, aplicando métodos, técnicas y herramientas adecuadas, minimizando costos y maximizando los beneficios para la institución, además de mejorar la comunicación entre los diferentes puntos de distribución con los que cuenta la empresa.

1.7 ALCANCES Y LÍMITES

1.7.1 ALCANCES

El proyecto a desarrollar permitirá realizar un mejor control automatizando diferentes procesos o tareas con módulos como:

- Módulo de administración: realizará una oportuna y fiable información de los procesos de la empresa. Además de la actualización del portal web, control de usuarios y control general del contenido.
- Módulo de control de almacén: controlará y se realizará un seguimiento de todos los productos que ingresan y salen de la empresa, a través de un sistema de administración de inventarios.
- Módulo de consultas y reportes: permitirá realizar de manera automática la emisión de reportes a través de consultas para una mejor toma de decisiones por parte de la gerencia y la administración.
- Módulo de control de órdenes de servicios, se encargará de hacer seguimiento de los pedidos que se elaboran, los que están en almacenes y de los que se venden.
- Módulo de administración de usuarios y clientes, que se encargara de controlar el acceso, ver el historial de usuarios y formularios de registro de tener una correcta toma de decisiones.
- Módulo de información, se encargará de hacer conocer al público en general, acerca de la actividades que realiza la empresa, los nuevos productos, los precios, los puntos de venta y más datos que la empresa vea por conveniente mostrar en la portal web.

Además contará con niveles de seguridad, restringiendo al personal que no esté autorizado en la manipulación de los diferentes módulos del sistema.

1.7.2 LÍMITES

Para el desarrollo del presente proyecto, no se considera los módulos de: Recursos Humanos, Ventas de productos vía internet (comercio electrónico), módulos de contabilidad, planillas de sueldos, además de no contemplar la facturación automática.



CAPITULO II MARCO TEÓRICO

2. MARCO TEÓRICO

2.1 INGENIERÍA DE SOFTWARE

2.1.1 DEFINICIÓN DE SOFTWARE

Software son todos los documentos asociados y la configuración de datos que se necesitan para que funcionen de manera correcta. Por lo general un sistema de software consiste en diversos programas independientes, archivos de configuración que se utilicen para ejecutarlos, la documentación para el usuario que explica cómo utilizar el sistema [SOMMEERVILLE, 2005].

2.1.2 DEFINICIÓN DE INGENIERÍA DE SOFTWARE

La ingeniería de software es una disciplina que comprende todos los aspectos de la producción del sistema desde las etapas iniciales de la especificación, hasta el mantenimiento de éste después de que se utiliza. En esta definición, existen dos frases clave [SOMMEERVILLE, 2005].

- **Disciplina de la Ingeniería.** Los ingenieros hacen que las cosas funcionen. Aplican teorías, métodos y herramientas donde sean convenientes, pero las utilizan de forma selectiva y siempre tratando de descubrir soluciones a los problemas.
- **Todos los Aspectos de Producción de Software.** La ingeniería del software no sólo comprende los procesos técnicos del desarrollo, sino también con actividades tales como la gestión de proyectos y el desarrollo de herramientas, métodos y teorías para el apoyo a la producción.

El objetivo de la ingeniería de software es lograr productos de calidad (tanto en su forma final como durante su elaboración), mediante un proceso apoyado por métodos y herramientas.

A continuación explicara algunas aplicaciones de la ingeniería de software como va avanzando la tecnología.

2.1.3 APLICACIONES DE INGENIERÍA DE SOFTWARE

Las aplicaciones de las ingenierías de software son: [WWW1, 2014]

- **Software de Sistemas.** Es un conjunto de programas que han sido escritos para servir a otros programas.
- **Software de Tiempo Real.** Coordina, analiza, controla sucesos del mundo real conforme ocurren, se denomina de tiempo real.
- **Software de Gestión.** El proceso de la información comercial constituye la mayor de las áreas de aplicación del software.
- **Software de Ingeniería y Científico.** Esta caracterizado por los algoritmos de manejo de números. Las aplicaciones van desde la astronomía a la vulcanología, del análisis de la presión de los automotores a la dinámica orbital de las lanzaderas espaciales y de la biología molecular a la fabricación automática.
- **Software Empotrado.** Los productos inteligentes se han convertido en algo común en casi todos los mercados de consumo e industriales. Consiste en residir en la memoria sólo lectura y se utiliza para controlar productos y sistemas de los mercados industriales y de consumo.
- **Software de Computadoras Personales.** El procesamiento de textos, las hojas de cálculo, los gráficos por computadora, multimedia, entretenimientos, gestión de bases de datos, aplicaciones financieras, de negocios y personales y redes o acceso de datos externas son algunas de los cientos de aplicaciones.
- **Software Basado en Web.** Las páginas Web buscadas por un explorador son software que incorpora instrucciones ejecutables por ejemplo, CGI, HTML, Perl, o Java, y datos se tiene, hipertexto y una variedad de formatos de audio y visual. En esencia, la red viene a ser una gran computadora que proporciona un recurso software casi ilimitado que puede ser accedido por cualquiera con un modem.

- **Software de Inteligencia Artificial.** Hace uso de algoritmos no numéricos para resolver problemas complejos para los que no son adecuados el cálculo o el análisis directo. Los sistemas expertos, también llamados sistemas basados en el conocimiento, reconocimiento de patrones (imágenes y voz), redes neuronales artificiales, prueba de teoremas, y los juegos son representativos de las aplicaciones de esta categoría.

El presente proyecto se enfocara en el software de sistemas y de inteligencia artificial para el desarrollo del sistema experto basado en el teorema de bayes.

A continuación se describe la metodología de desarrollo de software escogida para el trabajo de grado.

2.2 METODOLOGÍAS DE DESARROLLO DE SOFTWARE

Las metodologías de desarrollo de software en ingeniería, son un marco de trabajo utilizado para estructurar, planificar y controlar el proceso de desarrollo de un sistema. [PRESSMAN, 2005].

2.2.1 METODOLOGÍA ICONIX

ICONIX¹ es una metodología pesada ligera de desarrollo del Software, es simplificada en comparación a otras más tradicionales, la cual unifica un conjunto de métodos de orientación a objetos con el objetivo de tener un control estricto sobre todo el ciclo de vida del producto a realizar, cuenta con una secuencia de pasos que se deben seguir y determina claramente las actividades a desarrollar en cada etapa del proyecto.

Propone un enfoque aerodinámico del desarrollo de software que está basado en el uso de diagramas UML conjuntamente con técnicas que se toman de los casos de uso para codificar rápida y eficazmente. [ROSENBERG & STEPHENS & COLLINS-COPE, 2005].

¹ [RUP](#) (Proceso Unificado de Rational) [XP](#) (Programación Extrema)

A continuación se describirá las características de la metodología escogida para el desarrollo del presente proyecto.

2.2.1.1 CARACTERÍSTICAS DE LA METODOLOGÍA ICONIX

Las tres características fundamentales de ICONIX son: [QUISPE VERA, 2012].

- **Iterativo e Incremental:** Varias iteraciones ocurren entre el desarrollo del modelo del dominio y la identificación de los casos de uso. El modelo estático es incrementalmente refinado por los modelos dinámicos.
- **Trazabilidad:** Cada paso está referenciado por algún requisito. Se define trazabilidad como la capacidad de seguir una relación entre los diferentes artefactos producidos.
- **Dinámica del UML:** La metodología ofrece un uso “dinámico del UML” como los diagramas del caso de uso, diagramas de secuencia y de colaboración.

Es así que Rosenberg y Scott definen que el proceso de ICONIX se define a través de cuatro fases: Revisión de los Requisitos y Análisis de Requisitos, Revisión del Diseño Preliminar y Diseño Preliminar, Revisión crítica del diseño e Implementación. A continuación se describirá cada fase, conjuntamente a sus tareas y productos entregables.

2.2.1.2 FASES DE LA METODOLOGÍA

Las 4 fases de la metodología Iconix se utilizará para el desarrollo del proyecto es la siguiente:

a) **Revisión de los Requisitos y Análisis de Requisitos:**

Identificar en el mundo real, los objetos y todas las relaciones de agregación y generalización entre ellos. Se deben analizar todos los requisitos formarán parte del sistema y con estos construir el diagrama de clases, que representa las agrupaciones funcionales que estructurarán el procedimiento de desarrollo. [KENDALL SCOTT & DOUG ROSENBERG., 2001].

Para esta fase se utilizan 3 herramientas:

- **Modelo de Dominio:** Esto se refiere a identificar objetos y cosas del mundo real que intervienen con nuestro sistema estático.
- **Modelo de Casos de Uso:** Describe las acciones o el comportamiento que un usuario realiza dentro del sistema. Comprende de actores, casos de uso y el sistema.
- **Prototipo de Interfaz de Usuario:** Implica la creación de modelos operativos del trabajo de un sistema, en el que analistas y clientes deben estar de acuerdo. Dinámico que los usuarios se hacen participantes activos en el desarrollo

b) Revisión del Diseño Preliminar y Diseño Preliminar

En esta fase a partir de cada caso de uso se obtendrán una ficha de caso de uso, la cual no pertenece a UML, está formada por un nombre, una descripción, una precondición que debe cumplir antes de iniciarse, una pos condición que debe cumplir al terminar correctamente. Se deben describir los casos de uso, como un flujo principal de acciones, pudiendo contener los alternativos y los de excepción. La principal sugerencia de Iconix, en esta actividad es que no se debe perder mucho tiempo con la especificación textual, debería usarse un estilo consistente que sea adecuado al contexto del proyecto.

A realizar Diagrama de Robustez: Es un híbrido entre un Diagrama de Clases y un Diagrama de Actividades. Es una herramienta que nos permite capturar el que hacer y a partir de eso él como hacerlo. Facilita el reconocimiento de objetos y hace más sencilla la lectura del sistema. Ayuda a identificar los objetos que participan en cada caso de uso. [KENDALL SCOTT & DOUG ROSENBERG., 2001].

El diagrama de Robustez se divide en:

- **Objetos Fronterizos:** Usado por los actores para comunicarse con el sistema.
- **Objetos Entidad:** Son objetos del modelo del dominio.
- **Objetos de Control:** Es la unión entre la interfaz y los objetos de entidad.

- **Diagrama de Clases:** Describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos

c) Revisión Crítica del Diseño

En esta fase se reconocen todos los elementos que forman parte de nuestro sistema, Diagramas de Secuencia: muestra los métodos que llevarán las clases del prototipo. Muestra todos los cursos alternos que pueden tomar todos los casos de uso. Se debe terminar el modelo estático, añadiendo los detalles del diseño en el diagrama de clases y verificar si el diseño satisface todos los requisitos identificados. [KENDALL SCOTT & DOUG ROSENBERG, 2001].

d) Implementación

En esta fase a partir del buen diseño logrado se creará el software; que posteriormente se entregará. Se debe utilizar el diagrama de componentes si fuera necesario para apoyar el desarrollo, es decir mostrar una distribución física de los elementos que componen la estructura interna del sistema. Así como escribir y generar el código.

Pero además se debe tener en cuenta factores como:

- **Reusabilidad:** Es la posibilidad de hacer uso de los componentes en diferentes aplicaciones.
- **Extensibilidad:** Consiste en modificar con facilidad el software.
- **Confiabilidad:** Realización de sistemas descartando las posibilidades de error.
- **Realizar Pruebas:** Test de unidades, de casos, datos y resultados también de integración con los usuarios para verificar la aceptación de los resultados.

2.3 LENGUAJE DE MODELADO UNIFICADO (UML)

El lenguaje UML se ha convertido en ese estándar tan ansiado para representar y modelar la información con la que se trabaja en las fases de análisis y especialmente de diseño, tiene una notación gráfica muy expresiva que permite representar en mayor o menor medida todas las fases de un proyecto informático: desde el análisis con los casos de uso, el diseño de clases, objetos, etc., hasta la implementación y configuración con los diagramas de despliegue

La falta de estandarización en la manera de representar gráficamente un modelo impedía que los diseños gráficos realizados se pudieran compartir fácilmente entre distintos diseñadores, se necesitaba por tanto un lenguaje no sólo para comunicar las ideas a otros desarrolladores sino también para servir de apoyo en los procesos de análisis de un problema.

Se utilizara esta notación para realizar el análisis y diseño del sistema del sistema experto para el diagnóstico de osteoporosis, porque como se mencionara posteriormente UML permite tener una estructura y organización claras para la realización de proyectos [DOCUMENTOS, 2013].

2.3.1 DIAGRAMAS DE LENGUAJE DE MODELADO UNIFICADO

UML² cuenta con varios tipos de diagramas los cuales muestran diferentes aspectos de las actividades representadas. A continuación se muestran una descripción de los diagramas que se utilizaran para este sistema experto.

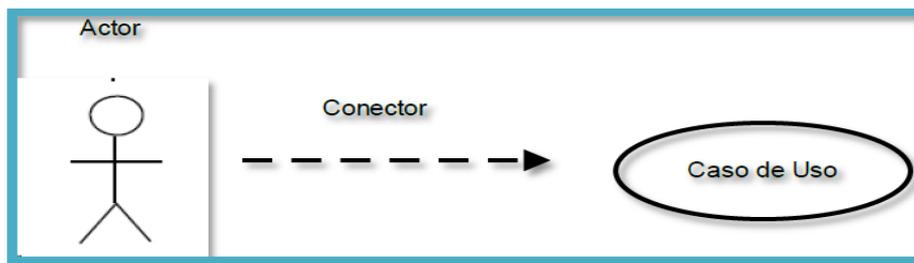
A) Diagrama de Caso de Uso.- Un diagrama de casos de uso, es definido a principalmente través del modelado de casos de uso que básicamente son utilizados para capturar los requisitos funcionales del sistema y lograr documentar como el usuario percibe el funcionamiento del prototipo. Los diagramas de caso de uso, están formados por tres elementos:

- **Actor:** Que representa cualquier elemento que intercambia información con el sistema, por lo que está fuera de él.

² Lenguaje de Modelo Uificado - Language Model Uificado

- **Caso de Uso:** Es una secuencia de intercambios en diálogo con el sistema que se encuentran relacionadas por su comportamiento.
- **Los Arcos:** Entre los actores y los casos de uso se denominan arcos de comunicación.

Figura N° 2.1 Diagrama de Caso de Uso

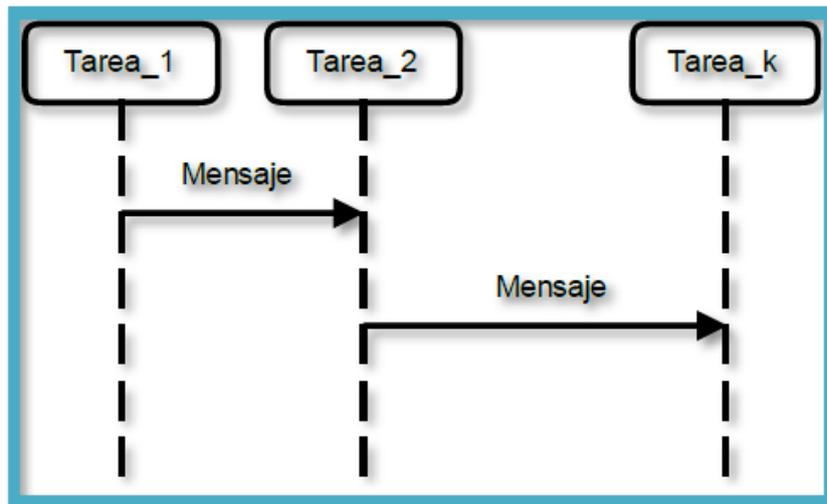


Fuente: Elaboración Propia en Base a Diagrama de Casos de Uso, Fowler & Scott

B) Diagrama de Secuencia. Es uno de los diagramas más efectivos para modelar interacción entre objetos en un sistema se modela para cada caso de uso. Mientras que el caso de uso permite el modelado de una vista “de negocio” del escenario, el diagrama de secuencia contiene detalles de implementación, incluyendo los objetos y clases que se usan para implementar el escenario, y mensajes pasados entre los objetos.

Muestra los objetos que intervienen en el escenario con líneas discontinuas verticales, y los mensajes pasados entre los objetos como vectores horizontales. Los mensajes se dibujan cronológicamente desde la parte superior y parte inferior; la distribución horizontal es arbitraria. [POPKIN SOFTWARE AND SYSTEMS, 2002].

Figura N° 2.2 Diagrama de Secuencia

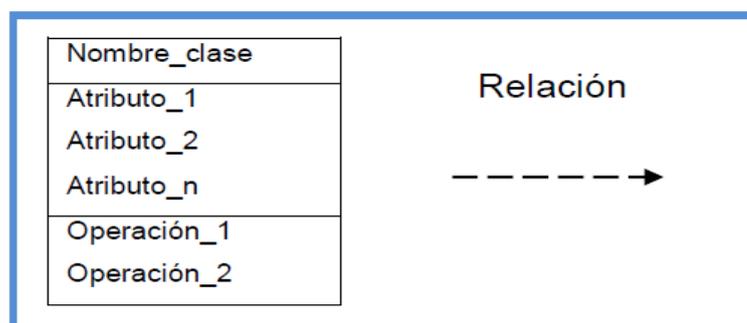


Fuente: Elaboración Propia en Base a Diagrama de Secuencia, Fowler & Scott

C) Diagrama de Clases. Es el diagrama principal en la fase de diseño, debido a que describen la estructura estática del sistema. Las cosas que existen y que nos rodean se agrupan naturalmente en categorías o grupo de cosas que tienen atributos (propiedades) y acciones similares, está formado por varios rectángulos de este tipo conectados por líneas que representan las asociaciones o maneras en que se relacionan entre sí.

Muestra un conjunto de clases, interfaces, y colaboraciones y sus relaciones entre ellos: clase tiene a atributos, métodos y visibilidad; relaciones que puede ser de herencia, composición, agregación, asociación y uso. [POPKIN SOFTWARE AND SYSTEMS, 2002].

Figura N° 2.3: Diagrama de Clases

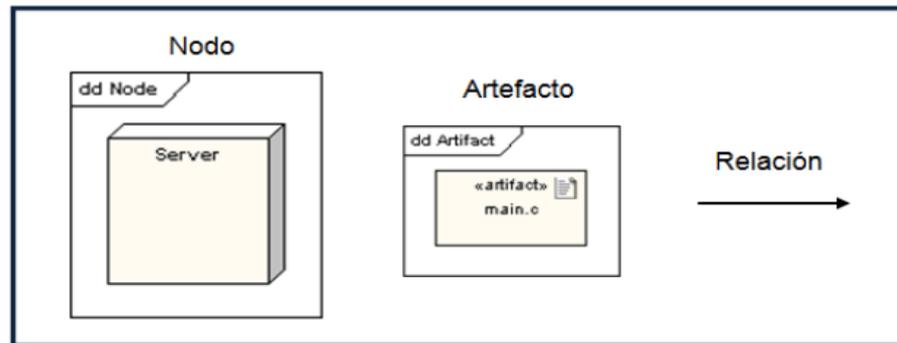


Fuente: Elaboración Propia en Base a Diagrama de Clases, Fowler & Scott, 2000

D) Diagrama de Despliegue. Muestran las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos

nodos, la vista de despliegue representa la disposición de las instancias de ejecución en los nodos conectados por enlaces de comunicación. Un nodo es un recurso de ejecución tal como un computador, un dispositivo o memoria.

Figura N° 2.4 Diagrama de Despliegue

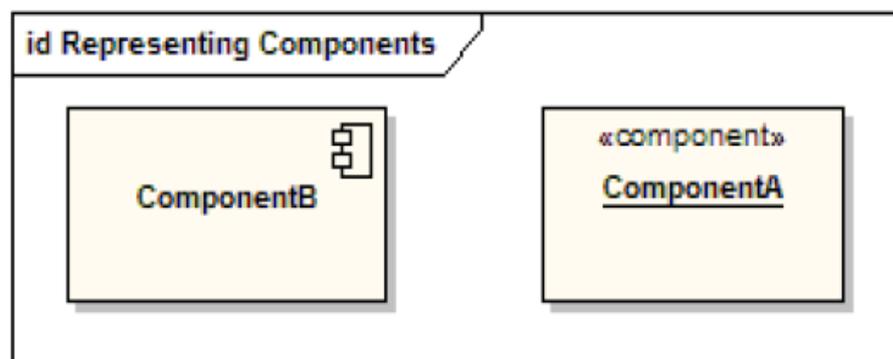


Fuente: Elaboración Propia en Base a Diagrama de Despliegue, Fowler & Scott, 2000

E) Diagrama de Componentes. Son las dependencias lógicas entre componentes software, sean éstos componentes fuentes, binarios o ejecutables, controladores embebidos, etc. Se relacionan con los diagramas de clases, ya que un componente normalmente se corresponde con una o más, interfaces o colaboraciones pero el de componentes tiene un nivel más alto de abstracción que la de clase [POPKIN SOFTWARE AND SYSTEMS, 2002].

Los componentes se representan como un clasificador rectangular con la clave «componente», opcionalmente se puede mostrar como un rectángulo con un icono en la esquina derecha arriba.

Figura N° 2.5 Diagrama de Componentes



Fuente: Elaboración Propia en Base a Diagrama de Componente, Fowler & Scott, 2000

2.4 PRUEBA DE SOFTWARE

La prueba es el proceso de ejecución de un programa con la intención de descubrir un error. Un buen caso de prueba es aquel que tiene una sola probabilidad demostrar un error no descubierto hasta entonces. [PRESSMAN, 2005]

Para el diseño de pruebas de software, se tomarán en cuenta dos para el desarrollo del sistema experto: caja blanca, se centra en la organización interna del programa y de caja negra, se centra en las funciones, entradas y salidas.

2.4.1 PRUEBA DE CAJA BLANCA

La prueba de caja blanca, en ocasiones llamada prueba de caja de cristal, es un método de diseño que usa la estructura de control descrita como parte del diseño al nivel de componentes para derivar los casos de prueba. Al emplear los métodos de prueba de caja blanca, el ingeniero del software podrá derivar casos de prueba que: [PRESSMAN, 2005]

1. Garanticen que todos las rutas independientes dentro del módulo se han ejercitado por lo menos una vez.
2. Ejerciten los lados verdadero y falso de todas las decisiones lógicas.
3. Ejecuten todos los bucles en sus límites y dentro de sus límites operacionales.
4. Ejerciten estructuras de datos internos para asegurar su validez.

Se basa en el estudio minucioso de toda la operatividad de una parte del sistema, considerando los detalles procedurales. Se plantean distintos caminos de ejecución alternativos y se llevan a cabo para observar los resultados y contrastarlos con lo esperado [MOLINA, SÁNCHEZ, LETELIER & SÁNCHEZ, 1997].

2.4.2 PRUEBA DE CAJA NEGRA

Las pruebas de caja negra, también denominadas, pruebas de comportamiento, se concentran en los requisitos funcionales del software. Es decir, permiten al ingeniero de software derivar conjuntos de condiciones de entrada que ejercitarán por completo todos los requisitos funcionales de un programa. No es una opción frente a las técnicas de caja blanca. Es, en cambio, un enfoque complementario que tiene probabilidades de descubrir una clase diferente errores de los que se descubrirían con los métodos de caja blanca. [PRESSMAN, 2005]

Las pruebas de caja negra tratan de encontrar errores en las siguientes categorías:

1. Funciones incorrectas o faltantes.
2. Errores de interfaz.
3. Errores en estructuras de datos o en acceso a bases de datos externas.
4. Errores de comportamiento o desempeño.
5. Errores de inicialización y término.

Después de describir las características importantes de la ingeniería de software que serán fundamento teórico para la construcción del sistema experto y base de datos.

Los métodos de la caja negra se enfocan los requisitos funcionales del software permitiendo al ingeniero el disponer de conjuntos de valores de entrada que ejercitan de forma completa todos los requisitos del programa [MOLINA, SÁNCHEZ, LETELIER & SÁNCHEZ, 1997].

2.5 INTELIGENCIA ARTIFICIAL

Es la rama de la ciencia que se encarga del estudio de la inteligencia en elementos artificiales y, desde el punto de vista de la ingeniería, propone la creación de elementos que posean un comportamiento. Dicho de otra forma, pretende construir sistemas y máquinas que presenten una conducta que si fuera llevado a cabo por una persona, se diría que es inteligente.

El aprendizaje, la capacidad de adaptación a entornos cambiantes, la creatividad, etc., son facetas que usualmente se relacionan con el comportamiento inteligente. Además, la IA es muy interdisciplinar, y en ella intervienen disciplinas tan variadas como la Neurociencia, la Psicología, las Tecnologías de la Información, la Ciencia Cognitiva, la Física, las Matemáticas, etc. [ROMERO, GOMEZ, DAFONCE, & PENOUSAL, 2007].

2.5.1 ÁREAS DE APLICACIÓN

Diversas son las áreas de la Inteligencia artificial se presenta en mayor o menor medida, a continuación brevemente algunas de las áreas: [ALFONSO GALIPIENSO, CARZORIA, COLOMINA PARDO, ESCOLANO RUIZ, & LOZANO ORTEGA, 2003]

- **Tratamiento de Lenguajes Naturales:** En este campo se puede englobar aplicaciones que realizan traducciones entre idiomas, interfaces hombre-máquina que permiten interrogar una base de datos o dar órdenes a un sistema operativo, etc., de manera que la comunicación sea más amigable al usuario.
- **Aprendizaje:** Modelización de conductas para su posterior implantación en computadoras.
- **Problemas de Percepción:** Visión y habla, Reconocimiento de objetos y del habla, detección de defectos en piezas por medio de visión, apoyo en diagnósticos Administradores, etc.
- **Robótica:** Navegación de robots móviles, control de brazos de robots, ensamblaje de piezas, etc.
- **Sistemas Expertos:** En esta área están englobados aquellos sistemas donde la experiencia de personal cualificado se incorpora a dichos sistemas para conseguir deducciones más cercanas a la realidad.

El presente proyecto de grado abarcará el área de los Sistemas Expertos, con el objeto de desarrollar un sistema que logre demostrar la aplicabilidad de este tipo de sistema en el área de la salud, permitiendo el diagnóstico adecuado para

personas entre 45 a 65 años. A continuación se la definición de sistema experto con cual se trabajara para el desarrollo del sistema.

2.5 COSTO DE SOFTWARE

El Modelo Constructivo de Costes (o COCOMO, por su acrónimo del inglés Contractiva Cost Model) es un modelo de estimación de costes de software que incluye tres sub-modelos donde cada uno ofrece un nivel de detalle y aproximación cada vez mayor, a medida que avanza el proceso de desarrollo del software: básico, intermedio y detallado. [UPV, 2006].

Este será el modelo utilizado al finalizar el proyecto para estimar los costos del mismo. Pertenece a la categoría de modelos de subestimaciones basados en estimaciones matemáticas. Está orientada a la magnitud del producto final, midiendo el "tamaño" del proyecto, en líneas de código principalmente.

El modelo COCOMO está compuesto por tres sub-modelos de estimación según el nivel de detalle empleado en su utilización: básico, intermedio y detallado. En los tres sub-modelos se pretende calcular tres importantes factores para determinar el costo total del software:

- **Esfuerzo (E)** necesario para concretar un proyecto de desarrollo de software, que es expresado en persona/mes y representa los meses de trabajo de una persona requeridos para desarrollar el proyecto.
- **Tiempo de Desarrollo (Tdev)** necesario para concertar el desarrollo del software, expresado en meses.
- **Cantidad de Personas (P)** promedio para concretar el desarrollo del software, expresado en personas.

Las ecuaciones que se utilizan en los tres modelos son:

$$E = (aKl)^b * m(X), \text{ en personas meses}$$

$$Tdev = c * (E)^d, \text{ en meses}$$

$$p = \frac{E}{Tdev}, \text{ en personas}$$

Dónde:

E es el esfuerzo requerido por el proyecto, en persona-mes

Tdev es el tiempo requerido por el proyecto, en meses

P es el número de personas requerido por el proyecto

a, b, c y d son constantes con valores definidos en una tabla, según cada submodelo

Kl es la cantidad de líneas de código, en miles.

m(X) Es un multiplicador que depende de 15 atributos

Por otra parte cada sub-modelo contiene una sub-clasificación de tres modos de desarrollo, que pueden ser del tipo: orgánico, semi-libre y empotrado. [RUIZ, 1999, p. 33]

- **Modo Orgánico:** Proyectos donde el tamaño del software no es más de 50 KLCD15 (50.000 líneas de código), desarrollan en base a áreas muy específicas y bien conocidas por el equipo participante (normalmente un pequeño grupo de programadores experimentados)
- **Modo Semi-Libre o Semi-Empotrado:** Proyectos donde el tamaño del software tiene un tamaño menor a 300 KLCD (300.000 líneas de código), el grupo de trabajo de desarrollo puede incluir una mezcla de personas experimentadas y no experimentadas.
- **Modo Empotrado o Rígido:** Proyectos donde el tamaño del software es superior a 3000 KLCD, normalmente compuesto por un grupo de trabajo de desarrollo numeroso que no necesariamente son experimentadas, los proyectos correspondientes a este modo suelen presentar exigencia de altos niveles de fiabilidad y restricciones que pueden estar relacionadas con la funcionalidad y/o características técnicas.

2.5.1 MODELO BÁSICO

Se utiliza para obtener una primera aproximación rápida del esfuerzo, y hace uso de la tabla 2.1 de constantes para calcular distintos aspectos de costes:

Tabla N° 2.1 Valores para Modelo Básico

MODO	A	B	C	D
Orgánico	2,40	1,05	3,20	1,05
Semi libre	3,00	1,12	3,00	1,12
Rígido	3,60	1,20	2,80	1,20

Fuente: (UPV, 2006)

Estos valores son para las fórmulas:

- Personas necesarias por mes para llevar adelante el proyecto (MM) = $a \cdot (Klb)$
- Tiempo de desarrollo del proyecto (TDEV) = $c \cdot (MMd)$
- Personas necesarias para realizar el proyecto (CosteH) = $MM/TDEV$
- Costo total del proyecto (CosteM) = $CosteH \cdot \text{Salario medio entre los programadores y analistas.}$

Se puede observar que a medida que aumenta la complejidad del proyecto (modo), las constantes aumentan de 2.4 a 3.6, que corresponde a un incremento del esfuerzo del personal. Hay que utilizar con mucho cuidado el modelo básico puesto que se obvian muchas características del entorno.

2.5.2 MODELO INTERMEDIO

Este añade al modelo básico quince modificadores opcionales para tener en cuenta en el entorno de trabajo, incrementando así la precisión de la estimación. Para este ajuste, al resultado de la fórmula general se lo multiplica por el coeficiente surgido de aplicar los atributos que se decidan utilizar. Los valores de las constantes a reemplazar en la fórmula se ilustran en la tabla 2.2.

Tabla N° 2.2: Valores para Modelo Intermedio MODO A B Orgánico

MODO	A	B
Orgánico	3.20	1.05
Semilibre	3.00	1.12
Rígido	2.80	1.20

Fuente: (UPV, 2006)

Se puede observar que los exponentes son los mismos que los del modelo básico, confirmando el papel que representa el tamaño; mientras que los coeficientes de los modos orgánico y rígido han cambiado, para mantener el equilibrio alrededor del semi libre con respecto al efecto multiplicador de los atributos de coste.

2.5.3 ATRIBUTOS

Cada atributo se cuantifica para un entorno de proyecto. La escala es muy baja - bajo - nominal - alto - muy alto - extremadamente alto. Dependiendo de la calificación de cada atributo, se asigna un valor para usar de multiplicador en la fórmula (por ejemplo, si para un proyecto el atributo DATA es calificado como muy alto, el resultado de la fórmula debe ser multiplicado por 1000). El significado de los atributos es el siguiente, según su tipo:

- De software

RELY: Garantía de funcionamiento requerida al software. Indica las posibles consecuencias para el usuario en el caso que existan defectos en el producto. Va desde la sola inconveniencia de corregir un fallo (muy bajo) hasta la posible pérdida de vidas humanas (extremadamente alto, software de alta criticidad).

DATA: Tamaño de la base de datos en relación con el tamaño del programa. El valor del modificador se define por la relación: donde D corresponde al tamaño de la base de datos en bytes y K es el tamaño del programa en cantidad de líneas de código.

CPLX: Representa la complejidad del producto. De hardware.

TIME: Limitaciones en el porcentaje del uso de la CPU.

STOR: Limitaciones en el porcentaje del uso de la memoria.

VIRT: Volatilidad de la máquina virtual.

TURN: Tiempo de respuesta requerido. De personal.

ACAP: Calificación de los analistas.

AEXP: Experiencia del personal en aplicaciones similares.

PCAP: Calificación de los programadores. De proyecto.

MODP: Uso de prácticas modernas de programación.

TOOL: Uso de herramientas de desarrollo de software.

SCED: Limitaciones en el cumplimiento de la planificación.

El valor de cada atributo, de acuerdo a su calificación, se muestra en la tabla 2.3.

Tabla N° 2.3: Escala de Valores de Cada Atributo

Atributos	Valor					
	Muy Bajo	bajo	Nominal	Alto	Muy Alto	Extra alto
Atributos de software						
Fiabilidad	0.75	0.88	1.00	1.15	1.40	
Tamaño de Base de datos		0.94	1.00		1.16	
Complejidad	0.70	0.85	1.00		1.30	1.65
Atributos de Hardware						
Restricciones de tiempo de ejecución			1.00	1.11	1.30	1.66
Restricción de memoria virtual			1.00	1.06	1.21	1.56
Vitalidad de la máquina virtual		0.87	1.00	1.15	1.30	
Tiempo de respuesta		0.87	1.00	1.07	1.15	

Fuente: Elaboración Propia en Base a Software de calidad UPV, 2006

2.6 METODOLOGÍA DE ANÁLISIS Y DISEÑO

A principios de la décadas de los 80's los métodos y lenguajes de programación orientados a objetos (OO), obtuvieron una amplia difusión entre la comunidad de la ingeniería de software, y en la década de los 90's Jacobson, Rumbaugh y Booch, desarrollan el proceso unificado (PRESSMAN, 2006), un marco de trabajo para la ingeniería de software orientada a objetos.

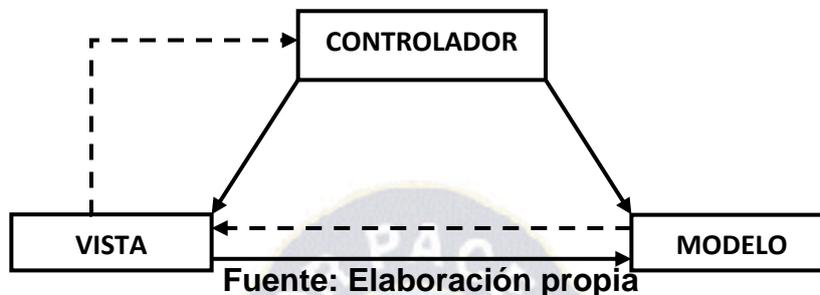
2.7 PATRÓN DE ARQUITECTURA

2.7.1 PATRÓN MODELO VISTA CONTROLADOR (MVC)

El patrón MVC (Modelo-Vista-Controlador) es un patrón arquitectónico de 3 capas conceptuales. Definido para sistemas usuario-máquina, aplicado en la actualidad a los sistemas web. No define exactamente las 3 capas clásicas de las

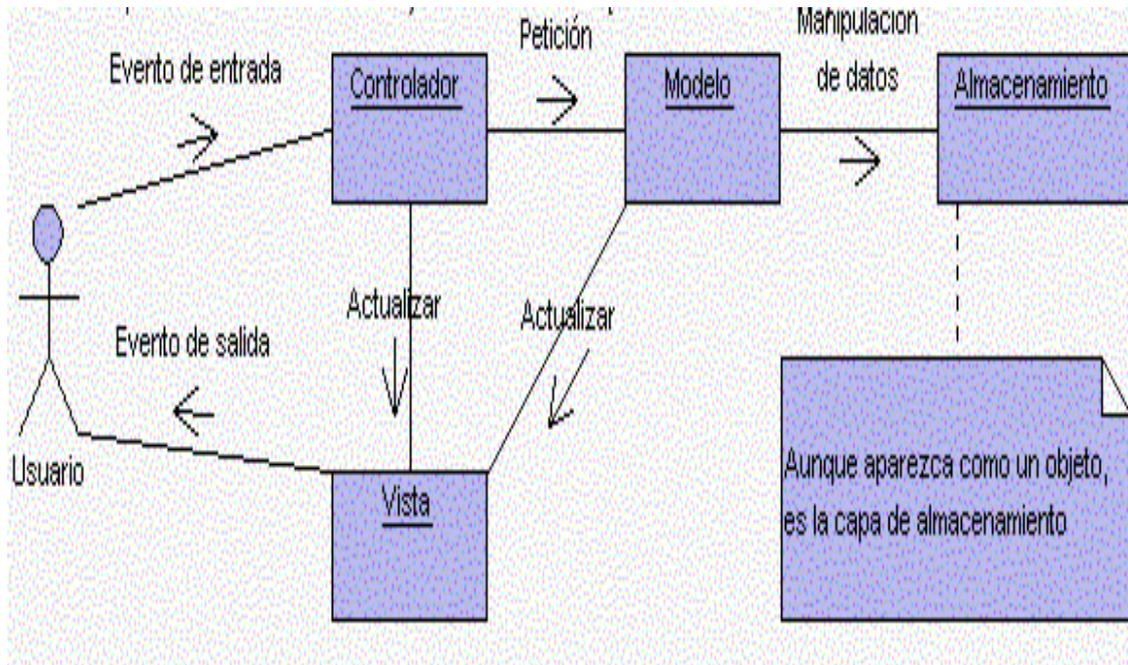
arquitecturas, Presentación, lógica de negocios y datos. En su lugar, define las responsabilidades y las dependencias dependiendo de los objetivos que representa en tres paradigmas: Modelo, Vista y Controlador.

Gráfico 2.9: Diagrama del Modelo Vista Controlador



- **Modelo:**
 - Representa a toda la información con la que opera la aplicación.
 - Gestiona el comportamiento y los datos del dominio.
 - Responde a las peticiones de información sobre el estado, que vienen de la Vista.
 - Responde a instrucciones de cambio de estado, provenientes del Controlador.
- **Vista:**
 - La Vista gestiona la presentación de la información de la aplicación. Todo lo relativo a la interfaz de usuario, los datos de que dispone para seguir interactuando con la aplicación.
 - Desde la interfaz gráfica a los estímulos que recibe del usuario, visual, auditiva o sensitivamente.
- **Controlador:**
 - Respuesta a eventos invocados desde la Vista.
 - Llama a la lógica de negocio para procesar y producir una respuesta.
 - Interpreta las entradas del usuario, informando al modelo y/o a la vista de los cambios que supongan esas entradas.

Gráfico 2.10: Ejemplo de Modelo Vista Controlador



[Fuente: INTERNET. WEB. Prestashop (29 marzo, 2010).]

2.8 ARQUITECTURA CLIENTE-SERVIDOR

Independientemente de la arquitectura física toda aplicación informática consta de al menos dos niveles funcionales

- 1- Presentación:** el nivel de presentación del refiere a la interfaz entre el usuario y el sistema. Esta interfaz es la responsable de aceptar los datos de entrada.
- 2- Lógica de negocio:** en este nivel se validan los datos se realizan los cálculos, etc.
- 3- Nivel de acceso:** en este nivel se realiza el almacenamiento físico y la extracción de los datos.

Informática de min friendo o host: toda la potencia de almacenamiento se encontraba en una maquina el usuario interactivo con los datos a través de una terminal no inteligente
Informática de redes área local: en esta arquitectura el nivel de presentación de negocio residen lógicamente en una pc local. El nivel de acceso de datos reside en otra máquina situada dentro de la red no están adaptadas a los entornos multiusuarios.

Cliente servidor tradicional: en esta arquitectura tradicional la arquitectura de negocio reside en la aplicación cliente. Cuando la lógica cambiadores modificarse la aplicación y redistribuirla a todas las estaciones de trabajo.

Arquitectura de n niveles: implica separar los tres niveles funcionales en tres niveles lógicos el objetivo principal consiste en separar la lógica de negocios del nivel de presentación como el acceso a datos empleando un conjunto de datos reutilizables denominados objetos de negocio. Pueden hacerse cambios en la base de datos y la lógica de negocio sin la necesidad de redistribuir las aplicaciones clientes.

Administración de BD: consiste en la valides consistencia y exactitud de los datos además de la protección contra el acceso y la modificación

a) CLIENTE

Un cliente es todo proceso que reclama servicios de otro. Una definición un poco más elaborada podría ser la siguiente: cliente es el proceso que permite al usuario formular los requerimientos y pasarlos al servidor. Se lo conoce con el término front-end.

Éste normalmente maneja todas las funciones relacionadas con la manipulación y despliegue de datos, por lo que están desarrollados sobre plataformas que permiten construir interfaces gráficas de usuario (GUI), además de acceder a los servicios distribuidos en cualquier parte de la red. Las funciones que lleva a cabo el proceso cliente se resumen en los siguientes puntos:

- Administrar la interfaz de usuario.
- Interactuar con el usuario.
- Procesar la lógica de la aplicación y hacer validaciones locales.
- Generar requerimientos de bases de datos.
- Recibir resultados del servidor.
- Formatear resultados.

La funcionalidad del proceso cliente marca la operativa de la aplicación (flujo de información o lógica de negocio). De este modo el cliente se puede clasificar en:

- Cliente basado en aplicación de usuario. Si los datos son de baja interacción y están fuertemente relacionados con la actividad de los usuarios de esos clientes.
- Cliente basado en lógica de negocio. Toma datos suministrados por el usuario y/o la base de datos y efectúa los cálculos necesarios según los requerimientos del usuario.

b) SERVIDOR

Un servidor es todo proceso que proporciona un servicio a otros. Es el proceso encargado de atender a múltiples clientes que hacen peticiones de algún recurso administrado por él. Al proceso servidor se lo conoce con el término back-end. El servidor normalmente maneja todas las funciones relacionadas con la mayoría de las reglas del negocio y los recursos de datos. Las principales funciones que lleva a cabo el proceso servidor se enumeran a continuación:

- Aceptar los requerimientos de bases de datos que hacen los clientes.
- Procesar requerimientos de bases de datos.
- Formatear datos para transmitirlos a los clientes.
- Procesar la lógica de la aplicación y realizar validaciones a nivel de bases de datos.

Puede darse el caso que un servidor actúe a su vez como cliente de otro servidor. Existen numerosos tipos de servidores, cada uno de los cuales da lugar a un tipo de arquitectura Cliente/Servidor diferente.

El término "servidor" se suele utilizar también para designar el hardware, de gran potencia, capacidad y prestaciones, utilizado para albergar servicios que atienden a un gran número de usuarios concurrentes. Desde el punto de vista de la arquitectura cliente/servidor y del procesamiento cooperativo un servidor es un servicio software que atiende las peticiones de procesos software clientes.

2.8.1 FUNCIONAMIENTO CLIENTE/SERVIDOR

Los sistemas Cliente/Servidor se pueden ver de la siguiente manera, los clientes piden que una tarea sea realizada; el servidor realiza dicha tarea y regresa la información al cliente a través de la red.

a) Ventajas

- Costos. El enfoque cliente/servidor es económico, sobre todo cuando está unido al concepto de racionalización.
- Acceso a la información. Si bien el acceso a los datos es posible por otros medios, la arquitectura Cliente/Servidor constituye el ambiente ideal para facilitar el acceso a la información.
- Ergonomía. Un buen sistema Cliente/Servidor no se concibe sin una interfaz gráfica de usuario y sin una transparencia total. Se concentra en el trabajo que debe realizar más que en la tecnología.
- Buena tecnología en el lugar adecuado. En teoría, un ambiente Cliente/Servidor puede conformarse de varias plataformas, sistemas operativos, Bases de Datos, etc.
- Modularidad. En un ambiente Cliente/Servidor, es factible agregar o eliminar estaciones de trabajo y servidores, puesto que el sistema puede ser más o menos fácil de volver a configurar

b) Desventajas

- Capacitación. En casi todos los casos de implantación del modelo Cliente/Servidor, la principal dificultad es la capacitación de los usuarios.
- Costos. Si bien el costo es uno de los principales factores que inclinan la balanza en favor de la arquitectura Cliente/Servidor. La implantación del modelo Cliente/Servidor comprende varios elementos. En primer lugar, se debe contar con una arquitectura completa de telecomunicación.
- En efecto, es necesario disponer de funciones como administración de archivos en red, subordinación de trabajos, mensajería, comunicación entre aplicaciones, etc. Además, sería útil contar con

una base de datos distribuida

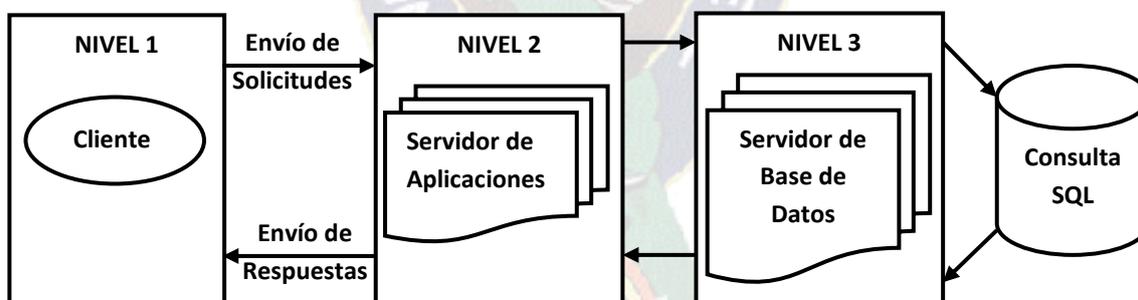
- Es casi imposible implementar con éxito un proyecto Cliente/Servidor sin contar con el apoyo de los sectores superiores de la administración para dar respaldo al proyecto y controlar la observancia de los planes del proyecto por parte de la organización.

En la **arquitectura en 3 niveles**, existe un nivel intermediario. Esto significa que la arquitectura generalmente está compartida por:

Un cliente, es decir, el equipo que solicita los recursos, equipado con una interfaz de usuario (generalmente un navegador Web) para la presentación. El servidor de aplicaciones (también denominado software intermedio), cuya tarea es proporcionar los recursos solicitados, pero que requiere de otro servidor para hacerlo.

El servidor de datos, que proporciona al servidor de aplicaciones los datos que requiere:

Gráfico 2.11: Arquitectura Cliente/Servidor de 3 Niveles



[Fuente: Elaboración propia]

2.8.2 APLICACIÓN DEL LADO DEL SERVIDOR

Una aplicación del lado del servidor es cualquier programa o conjunto de instrucciones diseñadas con la finalidad de que un Servidor Web las procese para realizar alguna acción. Las aplicaciones del lado del servidor están escritas mediante algún lenguaje de programación, entre los que destacan:

Lenguaje	Fecha de primera versión estable	Sistema operativo
PHP	1995	Multiplataforma
ASP.Net	1998	Windows (Algunas versiones)
Perl	1987	Multiplataforma
Python	1991	Multiplataforma
Ruby	1995	Multiplataforma

Existen numerosas aplicaciones que facilitan la instalación automática de servidores web Apache y aplicaciones adicionales como Mysql y PHP (entre otros), de forma conjunta, como XAMPP, JAMP o EasyPHP. Estas aplicaciones reciben el nombre de LAMP cuando se instalan en plataformas Linux, WAMP en sistemas Windows y MAMP en sistemas Apple Macintosh.

2.9 DISEÑO DE LA PÁGINA Y SISTEMA WEB

En los primeros días de la Web, los sitios Web consistían de páginas estáticas, permitiendo una interacción limitada con el usuario. Al comienzo de los años 90, estas limitaciones fueron superadas cuando los servidores Web fueron reemplazados para permitir comunicaciones a través del desarrollo de fragmentos de código que eran ejecutados del lado del servidor. A partir de entonces las aplicaciones dejaron de ser estáticas y solamente editadas por aquellos “gurúes” del HTML y se permitieron a usuarios normales interactuar con las aplicaciones por primera vez.

Este fue un paso fundamental para llegar a la Web que hoy en día conocemos. Sin la interacción no existiría el comercio electrónico (Ej.: paypal.com), el Web-mail (Ej.: Gmail), Internet-banking, blogs, forums o comunidades online.

La tendencia evolutiva de la Web continúa con el advenimiento de la “Web 2.0”, un término que indica una segunda versión de la Web, presentando un incremento notable en la interacción con el usuario a través de las llamadas interfaces ricas de usuario.

2.9.1 SEGURIDAD DE UN SISTEMA WEB

La seguridad es un elemento de primer nivel que entra en juego desde la concepción inicial de un sistema y participa desde un principio en las decisiones de diseño. Los requisitos de seguridad deben considerarse explícitamente durante todo el proceso de desarrollo, lo que da lugar a la inclusión de fases o actividad dedicada a la seguridad.

El objetivo principal de la seguridad es mantener las tres características primordiales de la información:

- Confidencialidad
- Integridad
- Disponibilidad

Las medidas de seguridad suelen centrarse principalmente en la eliminación o reducción de las vulnerabilidades del sistema

Requisitos mínimos de seguridad (Campos, 2006, p. 19)

- La seguridad de una contraseña se degrada con el uso. Todas las contraseñas, especialmente si se utilizan mucho, deben cambiarse periódicamente ya que los descuidos cotidianos pueden dar lugar a que diferentes personas las conozcan.
- Asesorarse. Establece un plan de asesoría o consultoría permanente aunque pienses que el sistema está bien y no lo necesita.
- Aprovechar la experiencia de otras personas. Prestar atención cuando se escuche de fallos de sistemas similares.
- Actualizar el sistema operativo. A veces es una lata descargar e instalar los "Service Packs" del sistema operativo, pero ahí vienen siempre correcciones y protecciones necesarias que pueden evitar dolores de cabeza en el futuro.

- Actualiza el antivirus. Los creadores de virus nunca descansan y cada vez salen virus más ingeniosos que solamente un antivirus actualizado puede detectar y detener.
- Modernizar el hardware. Muchos equipos o dispositivos tienen un periodo de vida, y no nos enteramos de ellos hasta que llegan a su fin. Anticiparse a esto y procura planear con tiempo la actualización del mismo.
- Llevar un registro de actividades lo más detallado posible. El registro debe de incluir al menos el registro con fecha y hora de todos los movimientos así como las entradas y salidas de los usuarios.
- Establecer políticas de seguridad para las personas responsables. No confiar en que todos saben que deben y no deben hacer. Establecer las políticas por escrito y asegurarse de que todos las conocen y las aplican.
- No instalar ni permitir que se instale software de dudosa procedencia. Esto incluye software pirata, gratis descargado de sitios no oficiales, plugins para el mensajero o navegador, juegos, ni nada que no se necesites ya que puede, desde poner en riesgo el sistema, hasta simplemente bajarle el rendimiento.

2.9.2 TECNOLOGÍAS

Inicialmente, era difícil la construcción de aplicaciones sofisticadas. La primera generación de aplicaciones Web era primitiva, en general basada en formularios con información y aplicaciones de búsqueda. Incluso estas aplicaciones básicas requerían de un alto conocimiento para su construcción.

A través del tiempo, el conocimiento necesario para construir aplicaciones ha sido reducido. Hoy en día, es relativamente sencillo construir aplicaciones sofisticadas utilizando las modernas plataformas y lenguajes, como ser PHP, .NET o Java.

- **Primera generación – CGI**

Common Gateway Interface (CGI) fue la tecnología reinante desde aproximadamente 1993 hasta fines de los '90 cuando los lenguajes de scripting comenzaron a ganar importancia.

CGI trabaja encapsulando la información provista por el usuario en variables de ambiente. Estas luego son accedidas por scripts o programas desarrollados

comúnmente en Perl o C. Estos programas procesan la información provista por los usuarios, y luego envían código HTML con la información procesada a la salida estándar, que a su vez es capturada por el servidor Web y pasada al usuario.

- **Scripting**

La falta de manejos de sesiones y control de autorización por parte de CGI impidió el desarrollo de aplicaciones Web comerciales con esa tecnología.

Los desarrolladores Web comenzaron entonces a utilizar lenguajes de script, como ser JavaScript o PHP para resolver esos problemas. Básicamente los lenguajes de script son ejecutados en el servidor Web y como son no compilados son desarrollados e implementados más fácilmente.

Los lenguajes de script incluyen, ASP, Perl, ColdFusion y PHP. De cualquier manera, muchos de esos podrían ser considerados como lenguajes interpretados híbridos, en particular las últimas versiones de PHP y ColdFusion.

Para seleccionar un lenguaje de Script las cuatro grandes características que debe cumplir son: Velocidad, estabilidad, seguridad y simplicidad.

- **Velocidad:** No sólo la velocidad de ejecución, la cual es importante, sino además no crear demoras en la máquina. Por esta razón no debe requerir demasiados recursos de sistema. PHP se integra muy bien junto a otro software, especialmente bajo ambientes Unix, cuando se configura como módulo de Apache, está listo para ser utilizado.
- **Estabilidad:** La velocidad no sirve de mucho si el sistema se cae cada cierta cantidad de ejecuciones. Ninguna aplicación es 100% libre de bugs, pero teniendo de respaldo una increíble comunidad de programadores y usuarios es mucho más difícil para lo bugs sobrevivir. PHP utiliza su propio sistema de administración de recursos y dispone de un sofisticado método de manejo de variables, conformando un sistema robusto y estable.
- **Seguridad:** El sistema debe poseer protecciones contra ataques. PHP provee diferentes niveles de seguridad, estos pueden ser configurados desde el archivo .ini

- **Simplicidad:** Se les debe permitir a los programadores generar código productivamente en el menor tiempo posible. Usuarios con experiencia en C y C++ podrán utilizar PHP rápidamente.

2.10 LENGUAJE DE PROGRAMACIÓN PHP

Para la realización de este proyecto se usara:

El PHP: Hypertext Pre-processor es un lenguaje de programación utilizado para crear páginas web dinámicas, necesita que un servidor web con capacidad PHP sea instalado y funcionando para poder ejecutar las páginas PHP. El servidor va a compilar el código PHP y tratarlo en tiempo real, con la información viniendo del cliente web, para regresarle una página web adaptada, en tiempo real. El servidor tiene también capacidad de seguir el camino de un usuario, así como de identificarlo.

Servidor WEB Apache.

El servidor Apache HTTP, también llamado Apache, es un servidor web HTTP de código abierto para la creación de páginas y servicios web. Es un servidor multiplataforma, gratuito, muy robusto y que destaca por su seguridad y rendimiento.

Para entender mejor lo que es Apache, primeramente definiremos lo que es un servidor web. La definición más sencilla de servidor web, que es un programa especialmente diseñado para transferir datos de hipertexto, es decir, páginas web con todos sus elementos (textos, widgets, banners, etc). Estos servidores web utilizan el protocolo http.

Son estas características que nos permiten elegir esta tecnología para montar el servicio de la Plataforma Web para la Administración de Recursos Humanos para la Cliente Boliviana

El servidor Web será instalado en la infraestructura computacional que cuenta actualmente cuenta la Cliente Boliviana.

2.10.1 PHP.

La Plataforma Web de Administración de Recursos Humanos para la Cliente Boliviana será desarrollada en el lenguaje de programación PHP (Hypertext Pre-processor), mismo que no requiere la adquisición de licencias para su empleo, además esta es una de las herramientas más popular y adecuada para el desarrollo web y que además puede ser incrustado en HTML.

Php es un programación interpretado (Lenguaje de alto rendimiento), diseñado originalmente para la creación de web dinámicas pero actual mente puede ser utilizado en la creación de otros tipos de programas, las siglas PHP significan (Hypertext Pre-processor). Fue creado por original mente por RasmusLerdor en 1994 avenida evolucionando en el trascurso de los años ahora con su nueva versión PHP 5 la cual está enfocada en la orientación a objetos o programación poo.

2.10.1.1 CARACTERÍSTICAS DE PHP

PHP posee varias características que lo hacen indicado para el software que se desarrollara de las cuales se pueden mencionar las siguientes:

- ✓ Orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos.
- ✓ El código fuente escrito en PHP es invisible al navegador web y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable.
- ✓ Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.
- ✓ Es libre, por lo que se presenta como una alternativa de fácil acceso para todos y reduce los costos para la empresa que desee una herramienta informática para sus procesos.
- ✓ Permite aplicar técnicas de programación orientada a objetos ideales para el software a desarrollar.
- ✓ Tiene manejo de excepción

2.10.1.2 VENTAJAS DE USAR PHP5

Algunas de las ventajas de usar PHP como lenguaje de programación son las siguientes:

- ✓ PHP corre en (casi) cualquier plataforma utilizando el mismo código fuente, pudiendo ser compilado y ejecutado en algo así como 25 plataformas, incluyendo diferentes versiones de Unix, Windows 95,98,NT,ME,2000,XP, 7 y Macs. Como en todos los sistemas se utiliza el mismo código base, los scripts pueden ser ejecutados de manera independiente al OS.
- ✓ PHP es completamente expandible. Está compuesto de un sistema principal (escrito por Zend), un conjunto de módulos y una variedad de extensiones de código.
- ✓ Puede interactuar con muchos motores de bases de datos tales como MySQL, MS SQL, Oracle, Informix, PostgreSQL, y otros muchos. Siempre podrás disponer de ODBC para situaciones que lo requieran.
- ✓ Rapidez. PHP generalmente es utilizado como módulo de Apache, lo que lo hace extremadamente veloz. Está completamente escrito en C, así que se ejecuta rápidamente utilizando poca memoria.

2.10.1.3 Desventajas de usar PHP5

- ✓ El manejo de errores no es tan sofisticado como ColdFusion o ASP.
- ✓ Como es un lenguaje que se interpreta en ejecución, para ciertos usos puede resultar un inconveniente que el código fuente no pueda ser ocultado. La ofuscación es una técnica que puede dificultar la lectura del código pero no necesariamente impide que el código sea examinado.
- ✓ Debido a que es un lenguaje interpretado, un script en PHP suele funcionar considerablemente más lento que su equivalente en un lenguaje de bajo nivel, sin embargo este inconveniente se puede minimizar con técnicas de cache tanto de en archivos y memoria.
- ✓ Las variables al no ser tipeadas dificulta a los diferentes IDEs para ofrecer asistencias para el tipeado del código, aunque esto no es realmente un

inconveniente del lenguaje en sí. Esto es solventado por Zend Studio añadiendo un comentario con el tipo a la declaración de la variable.

2.10.2 MYSQL

MYSQL es un servidor de base de datos relacionales con el cual se administra la misma, desarrollado y proporcionado por MYSQL AB. MYSQL Ab es una empresa cuyo negocio consiste en proporcionar servicios en torno al servidor de base de datos MYSQL.

2.10.2.1 Características de MYSQL

Algunas de las características más importantes que posee MYSQL son las siguientes:

- ✓ Los clientes se conectan al servidor MySQL usando sockets TCP/IP en cualquier plataforma. En sistemas Windows se pueden conectar usando ñame pipes y en sistemas Unix usando ficheros socket Unix.
- ✓ Seguridad: ofrece un sistema de contraseñas y privilegios seguro mediante verificación basada en el host y el tráfico de contraseñas está cifrado al conectarse a un servidor.
- ✓ Soporta gran cantidad de datos. MySQL Server tiene bases de datos de hasta 50 millones de registros.
- ✓ Se permiten hasta 64 índices por tabla (32 antes de MySQL 4.1.2). Cada índice puede consistir desde 1 hasta 16 columnas o partes de columnas. El máximo ancho de límite son 1000 bytes (500 antes de MySQL 4.1.2).

2.11 MANEJADOR DE BASE DE DATOS

Una base de datos o banco de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. (Rodríguez, 2009, p. 14)

Según la variabilidad de los datos almacenados se clasifican en:

- **Bases de datos estáticas:** Éstas son bases de datos de sólo lectura,

utilizadas primordialmente para almacenar datos históricos que posteriormente se pueden utilizar para estudiar el comportamiento de un conjunto de datos a través del tiempo, realizar proyecciones y tomar decisiones.

- **Bases de datos dinámicas:** Éstas son bases de datos donde la información almacenada se modifica con el tiempo, permitiendo operaciones como actualización y adición de datos, además de las operaciones fundamentales de consulta.

Según el contenido:

- **Bases de datos bibliográficas:** Solo contienen un subrogante (representante) de la fuente primaria, que permite localizarla. Un registro típico de una base de datos bibliográfica contiene información sobre el autor, fecha de publicación, editorial, título, edición, de una determinada publicación, etc.
- **Bases de datos de texto completo:** Almacenan las fuentes primarias, como por ejemplo, todo el contenido de todas las ediciones de una colección de revistas científicas.

2.11.1 MODELOS DE BASES DE DATOS

Además de la clasificación por la función de las bases de datos, éstas también se pueden clasificar de acuerdo a su modelo de administración de datos.

Un modelo de datos es básicamente una "descripción" de algo conocido como contenedor de datos (algo en donde se guarda la información), así como de los métodos para almacenar y recuperar información de esos contenedores. Los modelos de datos no son cosas físicas: son abstracciones que permiten la implementación de un sistema eficiente de base de datos; por lo general se refieren a algoritmos, y conceptos matemáticos.

- **Bases de datos jerárquicas:** Éstas son bases de datos que, como su nombre indica, almacenan su información en una estructura jerárquica. En este modelo los datos se organizan en una forma similar a un árbol (visto al revés), en donde un nodo padre de información puede tener varios hijos. El nodo que no tiene padres es llamado raíz, y a los nodos que no tienen

hijos se los conoce como hojas.

- **Base de datos de red:** Éste es un modelo ligeramente distinto del jerárquico; su diferencia fundamental es la modificación del concepto de nodo: se permite que un mismo nodo tenga varios padres (posibilidad no permitida en el modelo jerárquico).
- **Base de datos relacional:** En este modelo, el lugar y la forma en que se almacenen los datos no tienen relevancia (a diferencia de otros modelos como el jerárquico y el de red). Esto tiene la considerable ventaja de que es más fácil de entender y de utilizar para un usuario esporádico de la base de datos. La información puede ser recuperada o almacenada mediante "consultas" que ofrecen una amplia flexibilidad y poder para administrar la información.
- **Bases de datos multidimensionales:** Son bases de datos ideadas para desarrollar aplicaciones muy concretas, como creación de Cubos OLAP. Básicamente no se diferencian demasiado de las bases de datos relacionales (una tabla en una base de datos multidimensional podría serlo también en una base de datos relacional), la diferencia está más bien a nivel conceptual; en las bases de datos multidimensionales los campos o atributos de una tabla pueden ser de dos tipos, o bien representan dimensiones de la tabla, o bien representan métricas que se desean estudiar.
- **Bases de datos orientadas a objetos:** Este modelo, bastante reciente, y propio de los modelos informáticos orientados a objetos, trata de almacenar en la base de datos los objetos completos (estado y comportamiento). Una base de datos orientada a objetos es una base de datos que incorpora todos los conceptos importantes del paradigma de objetos:

Encapsulación - Propiedad que permite ocultar la información al resto de los objetos, impidiendo así accesos incorrectos o conflictos.

Herencia - Propiedad a través de la cual los objetos heredan comportamiento dentro de una jerarquía de clases.

Polimorfismo - Propiedad de una operación mediante la cual puede ser aplicada a distintos tipos de objetos.

- **Bases de datos documentales:** Permiten la indexación a texto completo, y en líneas generales realizar búsquedas más potentes. Taurus es un sistema de índices optimizado para este tipo de bases de datos.
- **Base de datos deductivas:** Un sistema de base de datos deductivas, es un sistema de base de datos pero con la diferencia de que permite hacer deducciones a través de inferencias. Se basa principalmente en reglas y hechos que son almacenados en la base de datos. También las bases de datos deductivas son llamadas base de datos lógica, a raíz de que se basan en lógica matemática.

Obviamente la creación de aplicaciones Cliente/Servidor está asociada a la utilización de servidores de bases de datos relacionales, con sistema de gestión de base de datos (SGBD).

Los sistemas de gestión de base de datos (por ejemplo, phpMyAdmin) permiten almacenar y posteriormente acceder a los datos de forma rápida y estructurada, este servicio permite crearse sus propias bases de datos y acceder a las mismas utilizando cualquier lenguaje de programación.

Para la gestión de base de datos usaremos el phpMyAdmin el cuales una herramienta de administración de bases de datos MySQL a través Internet (mediante una página web). Se encuentra escrita en PHP, bajo licencia GPL y está disponible en más de 50 idiomas. Permite crear y eliminar bases de datos; crear, eliminar y alterar tablas; borrar, editar y añadir campos; ejecutar cualquier sentencia SQL; administrar claves en campos; administrar privilegios; exportar datos en varios formatos; etc.

Funcionalidades del phpMyAdmin:

- visualizar y borrar bases de datos, tablas, vistas, campos e índices
- mostrar múltiples resultados a través de procedimientos almacenados o consultas
- crear, copiar, borrar, renombrar y alterar bases de datos, tablas, campos e índices
- realizar labores de mantenimiento de servidor, bases de datos y tablas, dando consejos acerca de la configuración del servidor
- ejecutar, editar y marcar cualquier expresión SQL, incluyendo consultas en lote
- carga tablas con el contenido de ficheros de texto
- crea y lee volcados de tablas
- exporta datos a varios formatos: CSV, XML, PDF, ISO/IEC 26300 - «OpenDocument Text and Spreadsheet», Microsoft Word 2000 y LATEX
- importar datos y estructuras MySQL de planillas Open Document así como también archivos XML, CSV y SQL
- administrar múltiples servidores
- gestionar privilegios y usuarios de MySQL
- comprobar la integridad referencial en las tablas MyISAM
- mediante Query-by-example (QBE), crear consultas complejas conectando automáticamente las tablas necesarias
- crear gráficos PDF del diseño de su base de datos
- buscar globalmente o solamente en una parte de una base de datos
- transformar los datos almacenados a cualquier formato usando un conjunto de funciones predefinidas, como mostrar objetos binarios (BLOBs) como imágenes o enlaces de descarga
- visualizar cambios en bases de datos, tablas y vistas
- capacidad de trabajar con tablas InnoDB y claves foráneas
- capacidad de utilizar mysqli, la extensión MySQL mejorada
- crear, editar, ejecutar y eliminar funciones y procedimientos almacenados («storedprocedures»)
- crear, editar, exportar y eliminar eventos y disparadores

- comunicarse en 62 idiomas distintos

MySQL: MySQL es un sistema de gestión de bases de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones como software libre en un esquema de licenciamiento dual.

SQL: El **lenguaje de consulta estructurado** o **SQL** (por sus siglas en inglés structured query language) es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en ellas. Una de sus características es el manejo del álgebra y el cálculo relacional que permiten efectuar consultas con el fin de recuperar de forma sencilla información de interés de bases de datos, así como hacer cambios en ella

2.11.2 LA BASE DE DATOS

Especifica del problema que se está resolviendo, que incluye toda la información proporcionada por el usuario a los sistemas, la información obtenida de los sensores y todas las conclusiones que el programa inteligente ha sido capaz de obtener [GAMEZ MARTIN, JOSE ANTONIO, PUERTA, JOSE MIGUEL, 1998].

Son muchas las consideraciones a tomar en cuenta al momento de hacer el diseño de la base de datos, quizá las más fuertes sean:

- La velocidad de acceso.
- El tamaño de la información.
- El tipo de la información.
- Facilidad de acceso a la información.
- Facilidad para extraer la información requerida.

El comportamiento del manejador de bases de datos con cada tipo de información. Pueden desarrollarse sistemas de procesamiento de archivo e

incluso manejadores de bases de datos basándose en la experiencia del equipo de desarrollo de software logrando resultados altamente aceptables, siempre es recomendable la utilización de determinados estándares de diseño que garantizan el nivel de eficiencia más alto en lo que se refiere a almacenamiento y recuperación de la información.

De igual manera se obtiene modelos que optimizan el aprovechamiento secundario y la sencillez y flexibilidad en las consultas que pueden proporcionarse al usuario.

2.11.3 DISEÑO DE UNA BASE DE DATOS

El diseño de una Base de Datos (**BD**) no es un proceso sencillo. Habitualmente, la complejidad de la información y la cantidad de requisitos de los sistemas de información hacen que sea complicado. Por este motivo, cuando se diseñan bases de datos es interesante aplicar la estrategia de dividir para vencer.

Por lo tanto, conviene descomponer el proceso del diseño en varias etapas; en cada una se obtiene un resultado intermedio que sirve de punto de partida de la etapa siguiente, y en la última etapa se obtiene el efecto deseado. De este modo no hace falta resolver de golpe toda la problemática que plantea el diseño, sino que en cada etapa se afronta un solo tipo de sub problema. Así se divide el problema y, al mismo tiempo, se simplifica el proceso.

Descompondremos el diseño de bases de datos en tres etapas:

1) **Diseño Conceptual**: Se obtiene una estructura de la información de la futura BD independiente de la tecnología que hay que emplear. No se tiene en cuenta todavía qué tipo de base se utilizará relacional, orientada a objetos, jerárquica, etc. en consecuencia, tampoco se tiene en cuenta con qué Sistema de Gestión de una Base de Datos (**SGBD**) ni con qué lenguaje concreto se implementará la base de datos.

Así pues, la etapa del diseño conceptual nos permite concentrarnos únicamente en la problemática de la estructuración de la información, sin tener que preocuparnos al mismo tiempo de resolver cuestiones tecnológicas.

El resultado de la etapa del diseño conceptual se expresa mediante algún modelo de datos de alto nivel. Uno de los más empleados es el modelo entidad interrelación (entity- relationship), que abreviaremos con la sigla ER³.

2) **Diseño Lógico:** Es parte del resultado del conceptual, que se transforma de forma que se adapte a la tecnología que se debe emplear. Más concretamente, es preciso que se ajuste al modelo del SGBD con el que se desea implementar la base de datos.

Por ejemplo, si se trata de un SGBD relacional, esta etapa obtendrá un conjunto de relaciones con sus atributos, claves primarias y claves foráneas. Esta etapa parte del hecho de que ya se ha resuelto la problemática de la estructuración de la información en un ámbito conceptual, y permite concentrarnos en las cuestiones tecnológicas relacionadas con el modelo de base de datos.

3) **Diseño Físico:** Se transforma la estructura obtenida en la etapa del diseño lógico, con el objetivo de conseguir una mayor eficiencia; además, se completa con aspectos de implementación física que dependerán del SGBD.

Por ejemplo, si se trata de una base de datos relacional, la transformación de la estructura puede consistir en lo siguiente: tener almacenada alguna relación que sea la combinación de varias relaciones que se han obtenido en la etapa del diseño lógico, partir una relación en varias, añadir algún atributo calculable a una relación, etc. Los aspectos de implementación física que hay que completar consisten normalmente en la elección de estructuras física, la selección del tamaño de las memorias intermedias (buffers) o de las páginas, etc.

En la etapa del diseño físico con el objetivo de conseguir un buen rendimiento de la BD se deben tener en cuenta las características de los procesos que consultan y actualizan la base de datos, como por ejemplo los caminos de acceso que utilizan y las frecuencias de ejecución. También es necesario considerar los volúmenes que se espera tener de los diferentes datos que se quieren almacenar.

En resumen, el diseño de una BD consiste en definir la estructura de los datos que debe tener la base de datos de un sistema de información determinado. En el

³ Entidad Relación

caso relacional, esta estructura será un conjunto de esquemas de relación con sus atributos, dominios de atributos, claves primarias, claves foráneas, etc.

Base de datos

Una base de datos es un conjunto de información estructurada en registros y almacenada en un soporte electrónico legible desde un ordenador. Cada registro constituye una unidad autónoma de información que puede estar a su vez estructurada en diferentes campos o tipos de datos que se recogen en dicha base de datos.

Las bases son cualquier conjunto de datos organizados para su almacenamiento en la memoria de un ordenador o computadora, diseñado para facilitar su mantenimiento y acceso de una forma estándar. Los datos suelen aparecer en forma de texto, números o gráficos. Desde su aparición en la década de 1950, se han hecho imprescindibles para las sociedades industriales.

2.11.3.1 ELEMENTOS DE UNA BASE DE DATOS

- Datos: Es la parte esencial de la información, es decir, la información que llega a la base de datos.
- Atributos: Son los diferentes campos que conforman la estructura de una base de datos.
- Campos: Es la unidad más pequeña de datos.
- Registro: Es un conjunto de campos o atributos relacionados entre sí.
- Archivo: Es un conjunto de registros relacionados.

2.11.3.2 TIPOS DE BASE DE DATOS

- Base de Datos Jerárquica: Es aquella donde los datos se presentan en nivel múltiples que represente con raíz y sus ramificaciones.
- Bases de Datos Red: Es aquella que permite la conexión de los nodos en forma multidireccional, por lo que cada nodo puede tener varios dueños a la vez.
- Base de Datos Relacional: En informática, tipo de base de datos o sistema de administración de bases de datos, que almacena información en tablas

(filas y columnas de datos) y realiza búsquedas utilizando los datos de columnas especificadas de una tabla para encontrar datos adicionales en otra tabla. En una base de datos relacional, las filas representan registros (conjuntos de datos acerca de elementos separados) y las columnas representan campos (atributos particulares de un registro). Al realizar las búsquedas, una base de datos relacional hace coincidir la información de un campo de una tabla con información en el campo correspondiente de otra tabla y con ello produce una tercera tabla que combina los datos solicitados de ambas tablas.

2.11.4 SISTEMA DE GESTIÓN DE BASE DE DATOS (SGBD)

Los Sistemas de Gestión de Base de Datos (DataBase Management System) son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta.

2.11.4.1 FUNCIONES DE UN SISTEMA DE GESTIÓN DE BASE DE DATOS

Ullman & Widom (1997) establecen que las funciones básicas de un sistema de gestión de base de datos son:

1. Permitir a los usuarios crear nuevas bases de datos y especificar su estructura, utilizando un lenguaje o interfaz especializado, llamado lenguaje o interfaz de definición de datos.
2. Dar a los usuarios la posibilidad de consultar los datos y modificarlos, utilizando un lenguaje o interfaz apropiado, generalmente llamado lenguaje de consulta o lenguaje de manipulación de datos.
3. Permitir el almacenamiento de grandes cantidades de datos durante un largo periodo de tiempo, manteniéndolos seguros de accidentes o uso no autorizado y permitiendo un acceso eficiente a los datos para consultas y modificaciones.

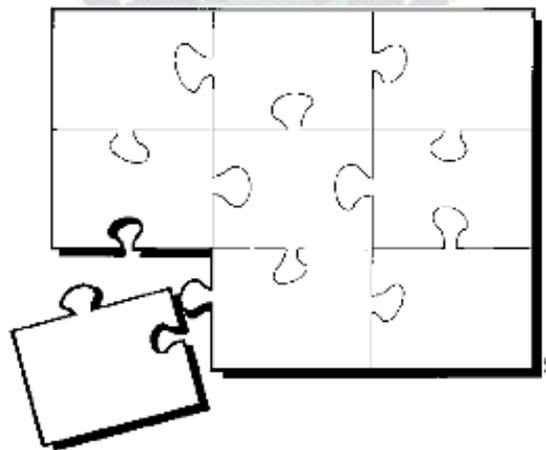
4. Controlar el acceso a los datos de muchos usuarios a la vez, impidiendo que las acciones de un usuario puedan afectar a las acciones de otro sobre datos diferentes y que el acceso simultáneo no corrompa los datos.

2.12 FRAMEWORK WEB

2.12.1 ¿QUÉ ES UN FRAMEWORK WEB?

El concepto framework se emplea en muchos ámbitos del desarrollo de sistemas software, no solo en el ámbito de aplicaciones Web. Podemos encontrar frameworks para el desarrollo de aplicaciones médicas, de visión por computador, para el desarrollo de juegos, y para cualquier ámbito que pueda ocurrírse nos.

En general, con el término framework, nos estamos refiriendo a una estructura software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, un framework se puede considerar como una aplicación genérica incompleta y configurable a la que podemos añadirle las últimas piezas para construir una aplicación concreta.



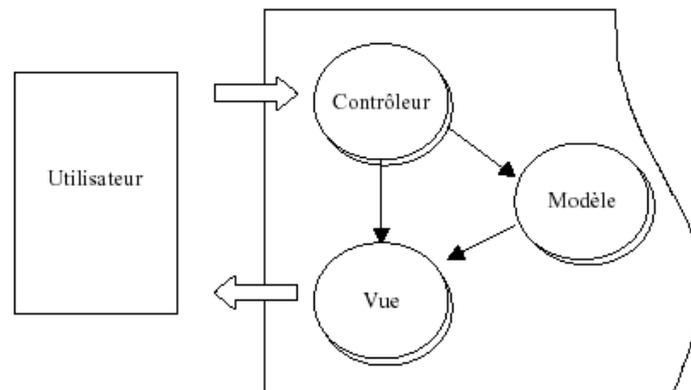
Los objetivos principales que persigue un framework son: acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones.

Un framework Web, por tanto, podemos definirlo como un conjunto de componentes (por ejemplo clases en java y descriptores y archivos de

configuración en XML) que componen un diseño reutilizable que facilita y agiliza el desarrollo de sistemas Web.

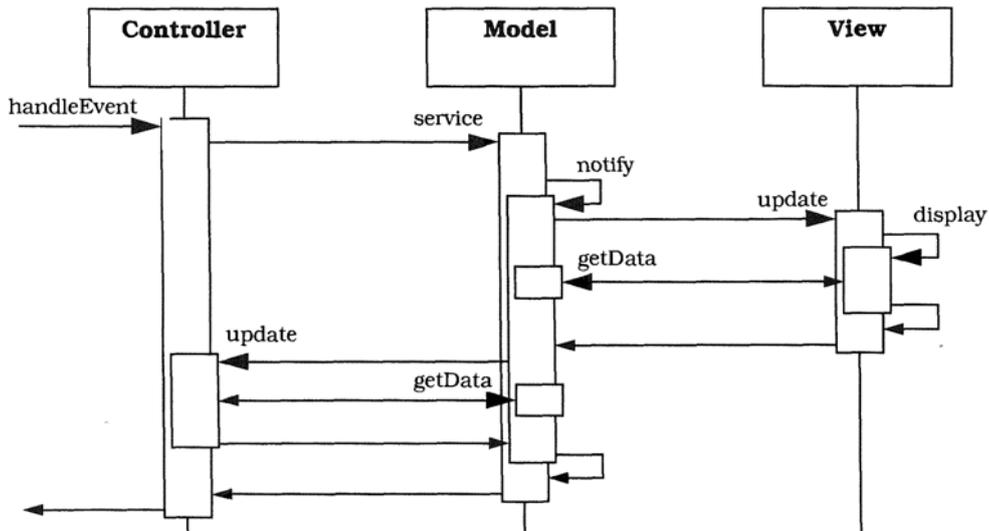
2.12.2 PATRÓN MVC Y MODEL 2

Para comprender como trabajan los frameworks Web existentes es imprescindible conocer el patrón MVC.

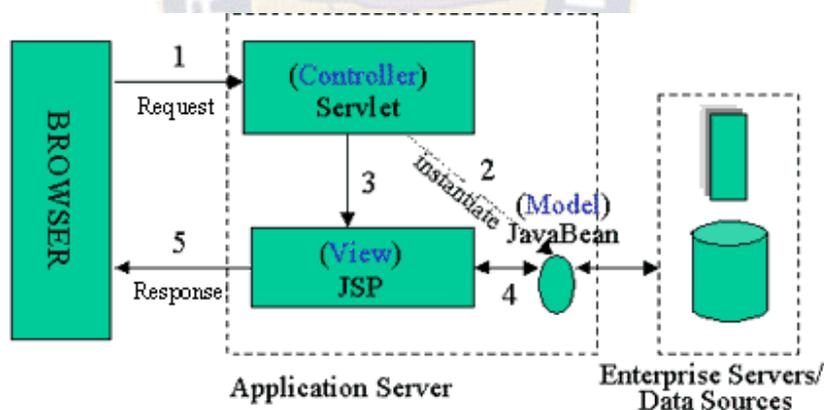


Modelo MVC

El patrón Modelo-Vista-Controlador es una guía para el diseño de arquitecturas de aplicaciones que ofrezcan una fuerte interactividad con usuarios. Este patrón organiza la aplicación en tres modelos separados, el primero es un modelo que representa los datos de la aplicación y sus reglas de negocio, el segundo es un conjunto de vistas que representa los formularios de entrada y salida de información, el tercero es un conjunto de controladores que procesa las peticiones de los usuarios y controla el flujo de ejecución del sistema.



La mayoría, por no decir todos, de los frameworks para Web implementan este patrón. Una aplicación de este patrón en entornos Java para programación Web es lo que se conoce con el nombre de arquitectura model2.



Esta arquitectura consiste, a grandes rasgos, en la utilización de servlets para procesar las peticiones (controladores) y páginas JSP para mostrar la interfaz de usuario (vistas), implementando la parte del modelo mediante JavaBeans o POJOs.

2.12.3 TIPOS DE FRAMEWORK WEB

Existen varios tipos de frameworks Web: orientados a la interfaz de usuario,

como Java Server Faces, orientados a aplicaciones de publicación de documentos, como Cocoon, orientados a la parte de control de eventos, como Struts y algunos que incluyen varios elementos como Tapestry.

La mayoría de frameworks Web se encargan de ofrecer una capa de controladores de acuerdo con el patrón MVC o con el modelo 2 de Servlets y JSP, ofreciendo mecanismos para facilitar la integración con otras herramientas para la implementación de las capas de negocio y presentación.

2.12.4 CARACTERÍSTICAS

A continuación enunciamos una serie de características que podemos encontrar en prácticamente todos los frameworks existentes.

<i>Abstracción de URLs y sesiones.</i>	No es necesario manipular directamente las URLs ni las sesiones, el framework ya se encarga de hacerlo.
<i>Acceso a datos.</i>	Incluyen las herramientas e interfaces necesarias para integrarse con herramientas de acceso a datos, en BBDD, XML, etc.
<i>Controladores.</i>	La mayoría de frameworks implementa una serie de controladores para gestionar eventos, como una introducción de datos mediante un formulario o el acceso a una página. Estos controladores suelen ser fácilmente adaptables a las necesidades de un proyecto concreto.
<i>Autenticación y control de acceso</i>	Incluyen mecanismos para la identificación de usuarios mediante login y password y permiten restringir el acceso a determinadas páginas a determinados usuarios.
<i>Internacionalización.</i>	
<i>Separación entre</i>	

2.12.5 ¿QUÉ ES BOOTSTRAP Y CÓMO FUNCIONA EN EL DISEÑO WEB?

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

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. Lo que se usa con más frecuencia, y que a mi opinión personal me gusta más, es el uso de media queries, que es un módulo de CSS3 que permite la representación de contenido para adaptarse a condiciones como la resolución de la pantalla y si trabajas las dimensiones de tu contenido en porcentajes, puedes tener una web muy fluida capaz de adaptarse a casi cualquier tamaño de forma automática.

Pero si no quieres nada que ver con los media queries, otra muy buena opción es el uso del framework de Bootstrap, que como te dije te ayudará a desarrollar tus sitios adaptativos.

Aun ofreciendo todas las posibilidades que ofrece Bootstrap a la hora de crear interfaces web, los diseños creados con Bootstrap son simples, limpios e intuitivos, esto les da agilidad a la hora de cargar y al adaptarse a otros dispositivos. El Framework trae varios elementos con estilos predefinidos fáciles de configurar: Botones, Menús desplegables, Formularios incluyendo todos sus elementos e integración jQuery para ofrecer ventanas y tooltips dinámicos.

Bootstrap tiene un soporte relativamente incompleto para HTML5 y CSS 3, pero es compatible con la mayoría de los navegadores web. La información básica de compatibilidad de sitios web o aplicaciones está disponible para todos los dispositivos y navegadores. Existe un concepto de compatibilidad parcial que hace disponible la información básica de un sitio web para todos los dispositivos y navegadores. Por ejemplo, las propiedades introducidas en CSS3 para las esquinas redondeadas, gradientes y sombras son usadas por Bootstrap a pesar de la falta de soporte de navegadores antiguos. Esto extiende la funcionalidad de la herramienta, pero no es requerida para su uso.

Desde la versión 2.0 también soporta diseños sensibles. Esto significa que el diseño gráfico de la página se ajusta dinámicamente, tomando en cuenta las características del dispositivo usado (Computadoras, tabletas, teléfonos móviles).



CAPÍTULO III MARCO PRÁCTICO

3.1 ANÁLISIS DE LA SITUACIÓN ACTUAL

El desarrollo de un sistema WEB para la venta de Joyas, precisa una metodología de desarrollo, que establezca los pasos a seguir para cumplir con la finalidad del presente proyecto en base a la Metodología ICONIX para el desarrollo del Sistema de gestión de la WEB. A continuación la Tabla 3.1 se define los objetivos, tareas y productos, entregables de la metodología.

Tabla Nº 3.1 Especificación de la Metodología de Desarrollo

Etapa	Fases	Objetivo	Tarea o Actividad	Productos Entregables
ANÁLISIS	Análisis de la Situación Actual	-Reconocer los procesos actuales que la Asociación de Orfebres y Relojeros A.M.O.R. -Obtener datos de los artesanos, de los productos que ellos elaboran.	- Entrevistar a los Administradores para identificar el problema actual del proceso.	-Descripción del problema actual
			-Identificar el proceso actual para la exportación y venta de La Paz al mundo.	-Diagrama de flujo de proceso de venta. -Diagrama de flujo de proceso de exportación
			Identificar los procesos, agentes y técnicas actuales para diagnosticar las necesidades.	-Proceso de adquisición del conocimiento -Tabla de factores de riesgos -Tabla Antecedentes y técnicas de adquisición de insumos

	Análisis del software	Análisis de requerimientos.	Definir los requerimientos del sistema	- Tabla de requerimientos funcionales del sistema
			Definir y especificar los actores y tareas del sistema.	- Tabla de actores del sistema
			Definir y especificar el modelamiento de casos de uso del sistema.	-Diagrama de caso de uso -Diagrama de robustez
DISEÑO DEL SISTEMA	Diseño del Sistema WEB	-Definir los campos de la base de datos	-Diseñar formalmente el conocimiento de los artesanos para hacer la representación en una base de datos.	Tabla de definición de variables -Tipo de campos - Formas Normales
		-Definir el funcionamiento del FrameWorks	- Calcular los Niveles de Seguridad	-Definir Logotipos -Definir Páginas a utilizar -Definir las interfaces
	Diseño del software	Definir la estructura dinámica y estática del sistema	Modelar la interacción de los componentes del sistema	- Diagramas de secuencia
			-Diseño de la base de datos	- Diagrama de clases - Diagrama Entidad Relación - Diagrama relacional de la Base de Datos

			Diseñar la integración de los componentes del sistema	- Diagramas de componentes
			-Diseñar las interfaces graficas del prototipo	-Interfaces graficas de usuario (GUI)
			-Construir pantallas muertas del sistema.	- Primer prototipo no funcional del sistema.
			-Codificar los componentes del sistema	- Prototipo final del sistema
VALIDACION	Validación	- Verificar las funciones del producto de software según lo diseñado	Se realizan pruebas no dirigidas del sistema con los usuarios	- Pruebas de Unidad - Pruebas de Integración - Pruebas de Validación

Fuente: Elaboración Propia

Realizar el análisis del sistema WEB propuesto del presente proyecto, requiere que en forma previa se defina la problemática que el sistema en el internet pretende resolver y el respectivo análisis de la situación actual en la cual se encuentra sumergida. Para la realización del análisis del Sistema, se seguirá las siguientes fases:

- Planteamiento del problema.
- Descripción de la situación actual.

- Obtención del conocimiento
- Conceptualización

A continuación se describen detalladamente cada una de estas fases.

3.1.1 Planteamiento del Problema

La descripción del problema del objeto de estudio y la determinación del mismo se concretó en el Capítulo 1, Sección 1.3, a través del análisis general de la situación actual del proceso del diagnóstico que realizaron en la Asociación de Orfebres y Relojeros, a continuación se detalla la descripción de los procesos que se realiza dentro de la Institución.

3.1.2 Descripción de la Situación Actual

Para realizar las etapas de análisis, diseño y construcción del sistema WEB propuesto es necesario tener un conocimiento exacto de los procesos de venta, y de exportación de las joyas y relojes que se elaboran en la asociación, dejando a un lado los aspectos que se deseen implementar o mejorar.

3.1.2.1 Proceso General de Exportación de Productos

El proceso general de Exportación de los productos que se elaboran, se ha ido contactando a través de las diferentes asociaciones que existen en los departamento del Estado plurinacional de Bolivia. Consiste en que cada artesano, lleva sus productos a las ferias, muchas veces es necesario realizar viajes largos para que se pueda exponer los productos que producen cientos de orfebres y relojeros.

A continuación se identifica el personal involucrado:

- **Artesano (Orfebre o relojero):** Encargado en realizar sus productos para exponerlos a la venta.

- **Personal de Administrativo:** Encargado en realizar los contactos con ferias y demás eventos en donde pueden ofertar los productos.
- **Escuela:** Encargado de impartir clases de diferentes técnicas sobre la actividades de orfebrería y relojería.
- **Cajero (tesorero):** Encargado de llevar las cuentas de los ingresos, cobrar y ofrecer una contabilidad transparente.

3.1.4 Conceptualización

Los conceptos claves y relaciones, obtenidos durante el apartado anterior son hechos explícitos en la fase de conceptualización; se diagrama estos conceptos y relaciones a manera de hacer permanente la base de datos relacional para el sistema prototipo, por ello, la etapa se lleva a cabo empleando tablas y grafos.

3.2 DESARROLLO DEL SISTEMA

En la siguiente fase se va describir el análisis del sistema y los diagramas utilizados para el desarrollo del software.

3.2.1 Análisis del Software

En la fase de análisis del software se define las necesidades de los usuarios finales, acompañado con la definición de los mismos a través del modelo de casos de uso, prototipo de las interfaces gráficas de usuario y el modelado preliminar del comportamiento del sistema.

3.2.1.1 Definición de Requerimientos

Los requerimientos del sistema fueron determinados de acuerdo a las necesidades de los dirigentes de la Asociación Mixta de Orfebres y Relojeros y la problemática del objetivo de estudio. A continuación se determina los requerimientos para realizar el Sistema de base de Datos por la WEB, se exponen en la Tabla 3.2:

Tabla N° 3.2 Requerimientos del Sistema

No.	Código	Requerimientos
1	R1	Proceso de autenticar al usuario para acceder al sistema
2	R2	Proceso de registro de usuario (Administrador y Artesano)
3	R3	Actualizar, modificar o eliminar datos del usuario
4	R4	Registro datos de los productos (Datos generales, peso en gramos, oro, plata, etc.)
5	R5	Guardado de datos del Artesano (Datos generales, antecedentes, especialidades)
6	R6	Brindar información necesaria al Artesano para la exportación de sus productos.
7	R7	Brindar información necesaria a través de reportes.

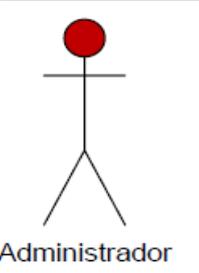
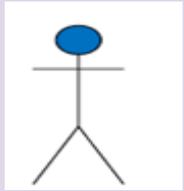
Fuente: Elaboración Propia

Para representar los requerimientos identificados y solicitados por la institución, se procedió a modelar al sistema para esto se utilizan diagramas de Casos de Uso que comprenden los Actores Relacionados.

3.2.1.2 Modelo de Caso de Uso

En base a los requerimientos obtenidos y especificados en la sección anterior se realiza el diseño de los diagramas de Casos de Uso descritos por UML (Lenguaje de Modelado Unificado); se identifica los tipos de usuarios o actores que interactúan con el sistema, que son el artesano y el administrador. A continuación, en la tabla 3.3, se describe las características de cada uno de ellos.

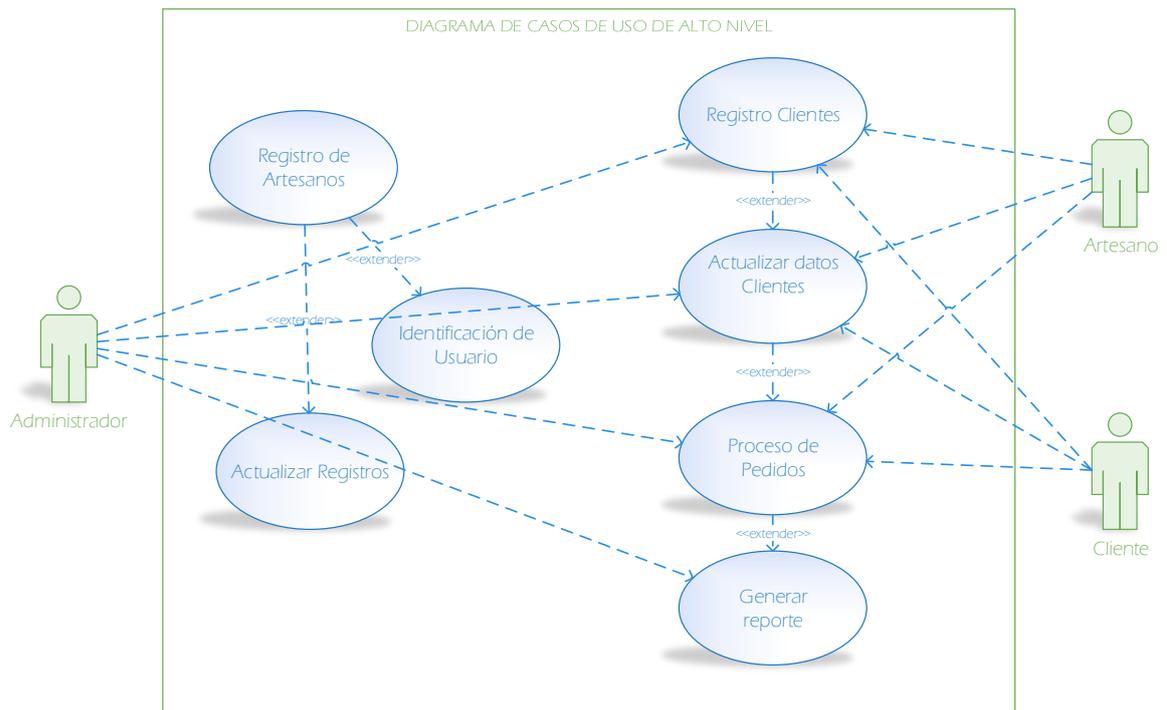
Tabla N° 3.3 Descripción de Actores del Sistema

IDENTIFICADOR	ACTOR	DESCRIPCION
 <p>Administrador</p>	<p>Administrador</p>	<p>El administrador es representado por el usuario que gestiona el sistema. Este actor posee los privilegios suficientes para realizar modificaciones en el sistema, mediante la creación y eliminación de usuario.</p>
 <p>Artesano</p>	<p>Artesano</p>	<p>El artesano es aquel actor experto en la elaboración de joyas y relojes. Los mismos que se desean sean vendidos en otros países.</p>
 <p>Cliente</p>	<p>Clientes</p>	<p>Son aquellas personas que no tiene ningún privilegio para gestionar el sistema pero indirectamente se sirve u obtiene un catálogo de los productos que la asociación AMOR oferta. Con la ayuda del sistema WEB.</p>

Fuente: Elaboración Propia

En la siguiente Figura 3.4, se describirá los diagramas de caso de uso de Alto Nivel y los diagramas de casos de uso expandidos.

Figura N° 3.4 Diagrama de Caso de Uso de Alto Nivel del Sistema



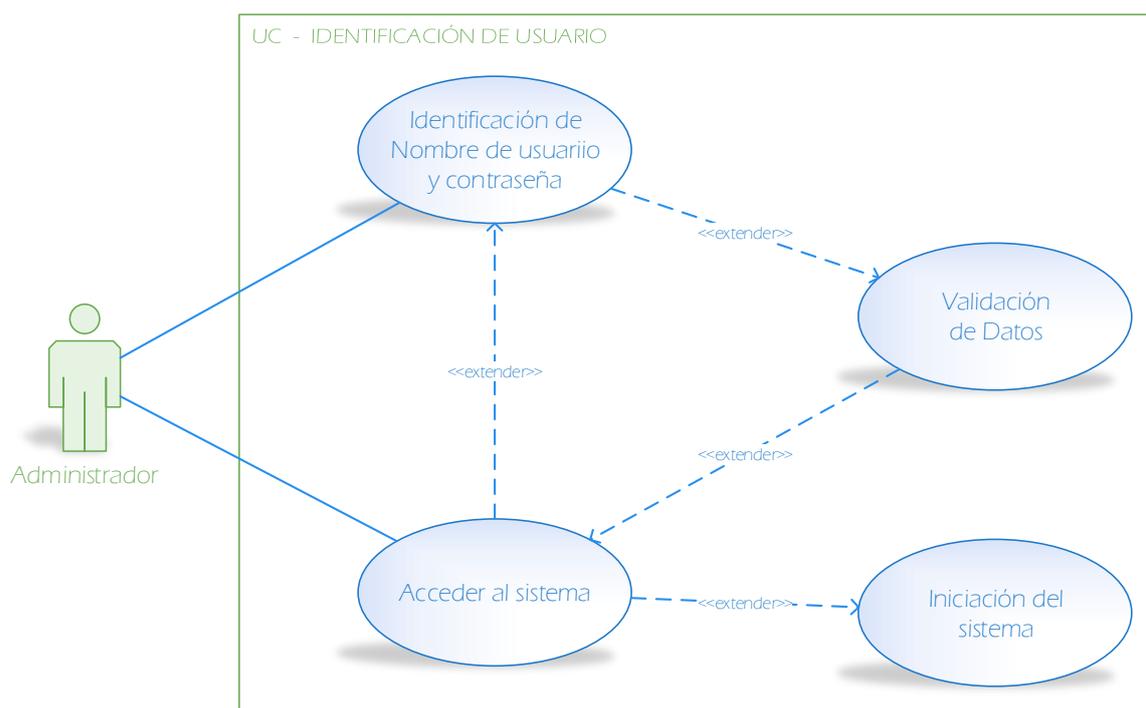
Fuente: Elaboración Propia

El diagrama de caso de uso de alto nivel del sistema ilustra los tipos de actores que interactuarán con el mismo, que son el Administrador y el artesano, ambos limitados a funcionalidades específicas dentro del sistema propuesto por otro lado el cliente quien hará los pedidos.

3.2.1.3 Especificación de Casos de Uso

Después de presentar el Caso de Uso de Alto Nivel, se procede a la elaboración de los Casos de Uso Expandidos, los cuales representan de manera detallada los procesos que se llevarán a cabo en el sistema, para su mejor comprensión. Los casos de Uso Expandidos se representan de dos formas, la primera gráficamente y a continuación en forma de tablas, para tener una idea más clara de cada una de ellas.

Figura N° 3.5 Diagrama de Caso de Uso de Identificación de Usuario



Fuente: Elaboración Propia

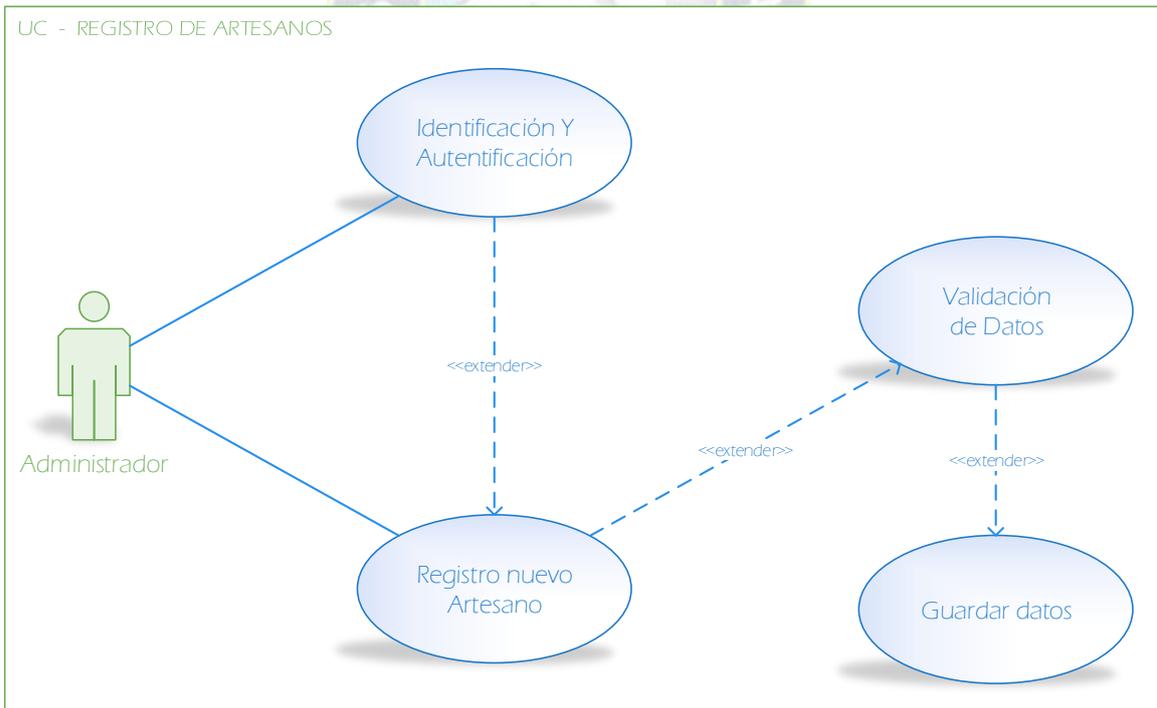
Tabla N° 3.6 Caso de Uso Identificación de Usuario

CASO DE USO:	Identificación de Usuario	
DESCRIPCIÓN:	El usuario que se encuentre utilizando el software, ya sea artesano o administrador, debe ingresar su nombre de usuario y password que se le asignó para tener acceso al sistema.	
ACTOR:	Administrador o Artesano	
CURSO NORMAL DE ACCIONES	Actor	Sistema
	1. El actor ingresa al icono que le da acceso a inicializar la	2. Muestra la Ventana de Inicio que permite autenticarse al sistema.

	aplicación	
	3. El actor ingresa los datos de autenticación: usuario y contraseña.	4. El sistema válido si los datos de identificación son correctos.
		5. Visualización de la ventana principal
	6. El usuario cierra sesión	7. El Sistema cierra sesión

Fuente: Elaboración Propia

Figura Nº 3.7 Diagrama de Caso de Uso Registro de Artesanos



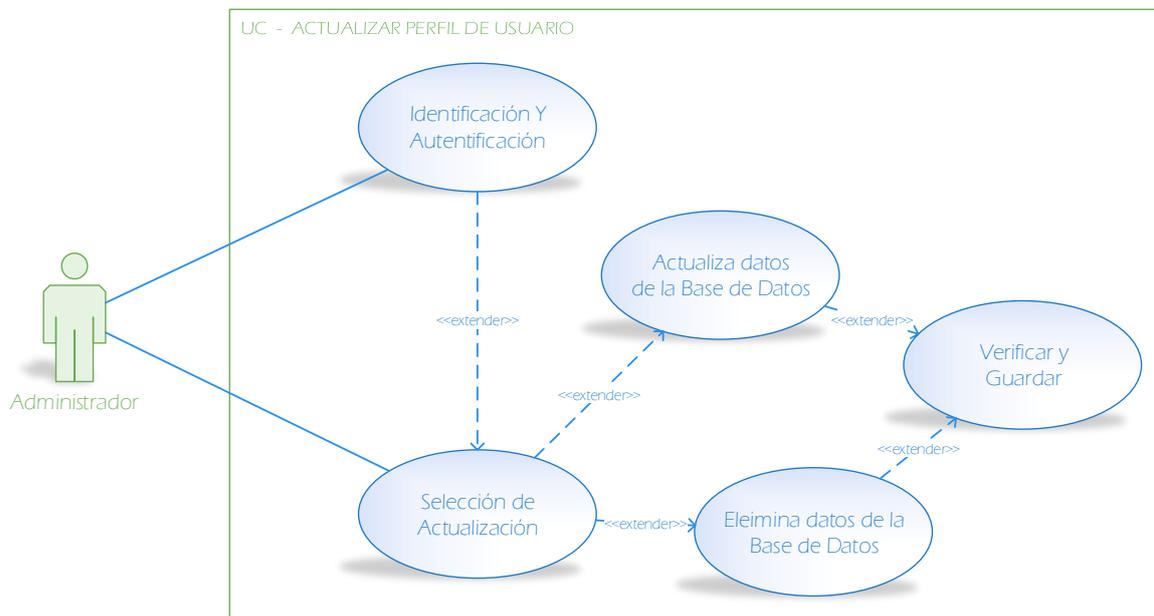
Fuente: Elaboración Propia

Tabla N° 3.8 Caso de Uso Registro de Artesanos

CASO DE USO:	Registro de Artesanos	
DESCRIPCIÓN:	Caso de uso que permite al administrador crear nuevo perfiles de usuario para el artesano asignándole privilegios.	
ACTOR:	Administrador	
CURSO NORMAL DE ACCIONES	Actor	Sistema
	1. El Administrador hace clic al botón de “Registrar Nuevo” dentro la Ventana Principal de Administrador	2. Presenta la ventana para registrar el perfil de usuario de un nuevo artesano.
	3. El Administrador ingresa los datos del nuevo artesano:	
	5. El actor acepta los datos introducidos en el sistema, con un clic en el botón “Crear”.	4 El sistema valida si los datos ingresados son correctos
		5 Guarda los datos introducidos

Fuente: Elaboración Propia

Figura N° 3.9 Diagrama de Caso de Uso Actualizar Perfil de Usuario Artesano.



Fuente: Elaboración Propia

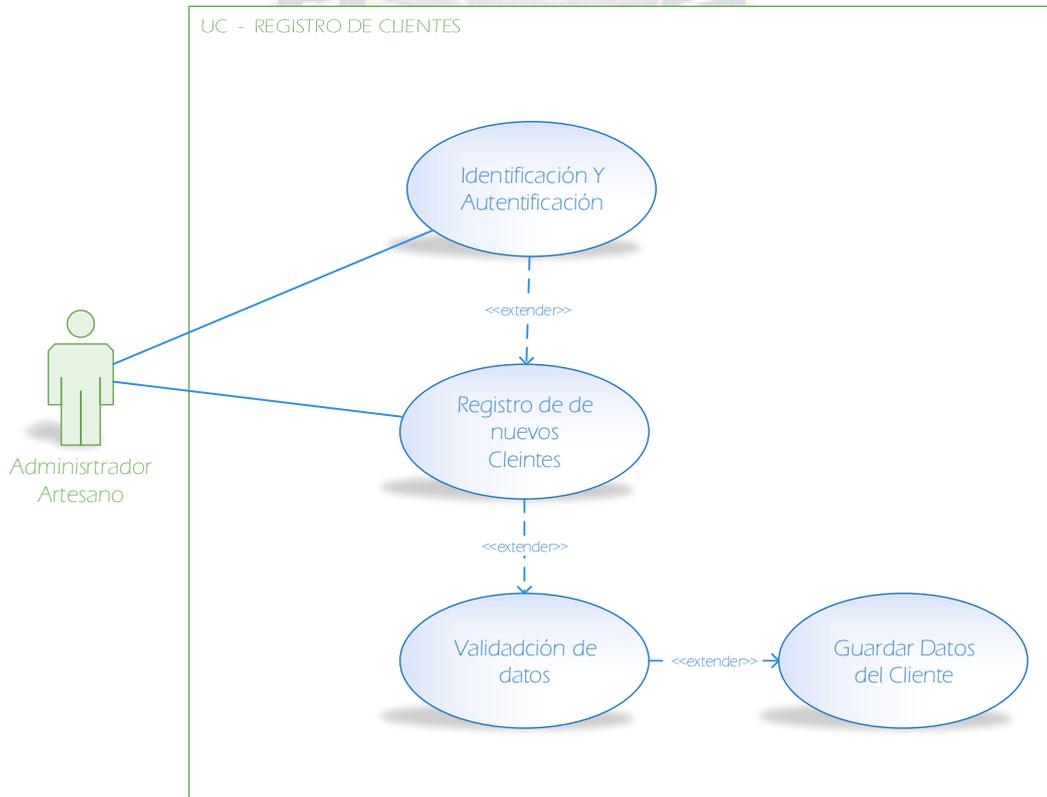
Tabla N° 3.10 Caso de Uso Actualizar Perfil de Usuarios

CASO DE USO:	Actualizar Perfil de Usuarios	
DESCRIPCIÓN:	Caso de uso que permite al Administrador modificar datos o eliminar perfiles del artesano.	
ACTOR:	Administrador	
CURSO NORMAL DE ACCIONES	Actor	Sistema
	1. El Administrador hace clic al botón de “Buscar artesano”.	2. Presenta la pantalla de búsqueda de Artesanos
	3. El Administrador ingresa los datos de búsqueda, de	4. Realiza la búsqueda y valida los datos de entrada, y visualiza los

	acuerdo al criterio requerido.	datos en la pantalla
	5. El Administrador hace clic en la opción “Eliminar”.	6 Elimina el perfil del Administrador y visualiza la pantalla principal
	7. El administrador puede modificar y actualizar	8 Visualiza la ventana de Ficha del Perfil de artesano seleccionado con los datos ya registrados

Fuente: Elaboración Propia

Figura N° 3.11 Diagrama de Caso de Uso Registro de Clientes

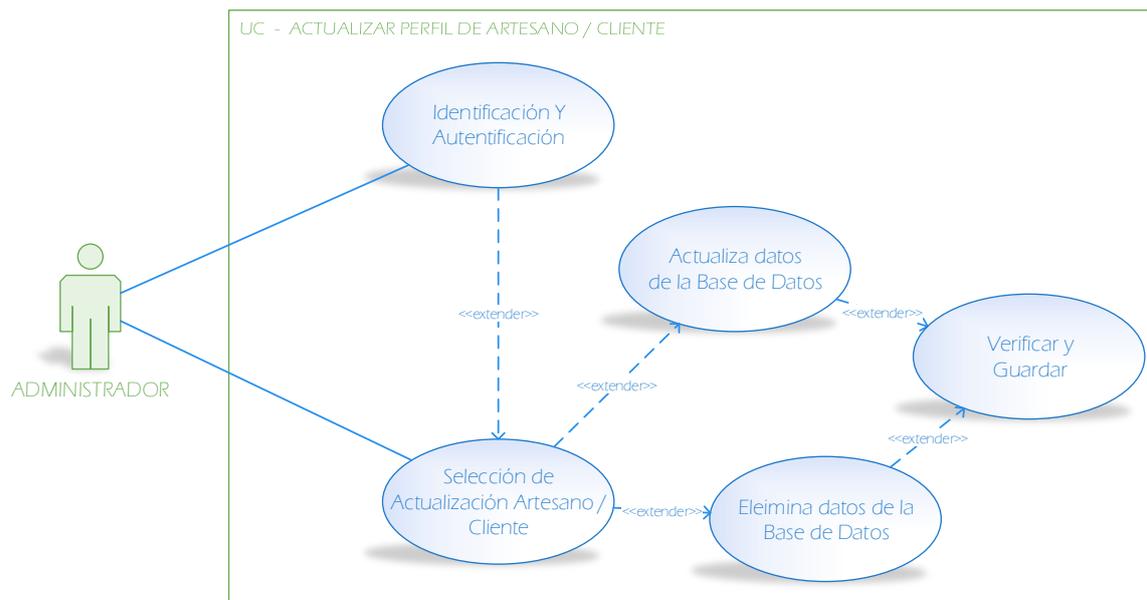


Fuente: Elaboracion Propia

Tabla N° 3.12 Caso de Uso Registro de Clientes

CASO DE USO:	Registro de Clientes	
DESCRIPCIÓN:	Caso de uso que permite al Administrador registrar nuevos clientes que ingresan para la consulta.	
ACTOR:	Cliente	
CURSO NORMAL DE ACCIONES	Actor	Sistema
	1. El Administrador hace clic al botón de “Registrar Nuevo”	2. El sistema presenta la pantalla para registrar al nuevo cliente.
	3. El Administrador ingresa los datos personales del cliente. “Continuar Registro”	4. El sistema valida los datos ingresados y muestra la siguiente pantalla.
	5 El Artesano ingresa los datos del cliente, su rubro y datos importantes y guarda los datos.	6 El sistema valida si los datos ingresados son correctos, y procede a guardar los datos ingresados.

Fuente: Elaboración Propia



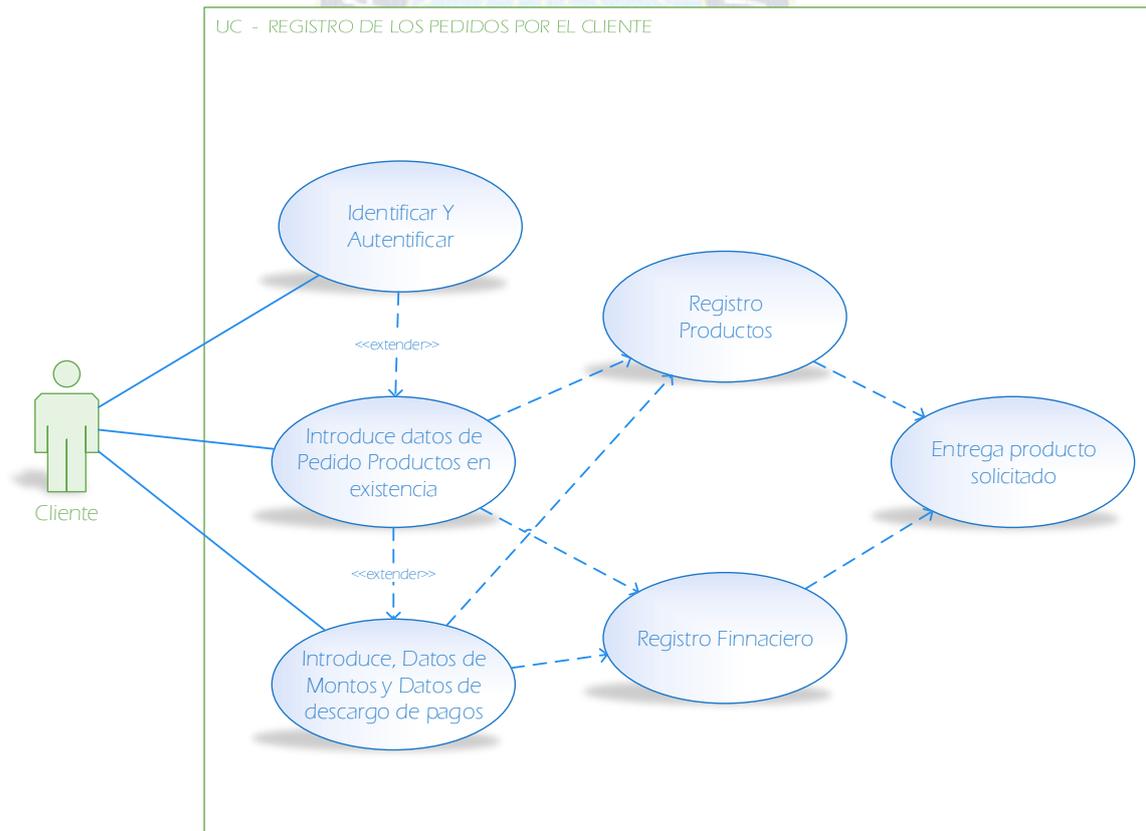
Fuente: Elaboración Propia

Tabla N° 3.14 Caso de Uso Actualizar Datos de Artesano / Cliente

CASO DE USO:	Actualizar Datos del Postulante	
DESCRIPCIÓN:	Caso de uso que permite al Administrador actualizar la información de los postulantes, es decir, modificar el registro e estudios y demás méritos del funcionario.	
ACTOR:	Administrador	
CURSO NORMAL DE ACCIONES	Actor	Sistema
	1. El Administrador ingresa "Buscar Artesano/Cliente"	2. El sistema muestra la pantalla de búsqueda del actor.
	3. El Administrador ingresa los datos del Artesano/Cliente de	4. Realiza la búsqueda y valida los datos de entrada

	búsqueda.	
	5. El Administrador hace clic en la opción “Editar”.	6. Edita el registro del Artesano/Cliente seleccionado y visualiza la pantalla principal del postulante.
	7. El Administrador actualiza datos del postulante, y hace clic en “Guardar”	8. Valida los datos de entrada para actualizar.

Fuente: Elaboración Propia



Fuente: Elaboración Propia

Tabla Nº 3.16 Caso de Uso Registro de Pedidos

CASO DE USO:	Registro de los Análisis del Pedido	
DESCRIPCIÓN:	Proceso de Pedidos de los Clientes de joyas	
ACTOR:	Administrador- Artesano	
CURSO NORMAL DE ACCIONES	Actor	Sistema
	1. El Administrador RRHH hace clic al botón de “WEB ⁴ ” de la pantalla de Muestra inicio de realizar análisis	2. El sistema presenta la pantalla inicial del proceso
	3. El Administrador introduce datos de cursos, méritos, deméritos y antecedentes del Cliente.	4. Presenta reporte Final

Fuente: Elaboración Propia

Concluida la especificación de los casos de uso, a continuidad se definen los diagramas de robustez, para realizar un diseño más formal del sistema.

3.2.1.4 Diagrama de Robustez

Los diagramas de robustez definen el diseño preliminar del sistema.

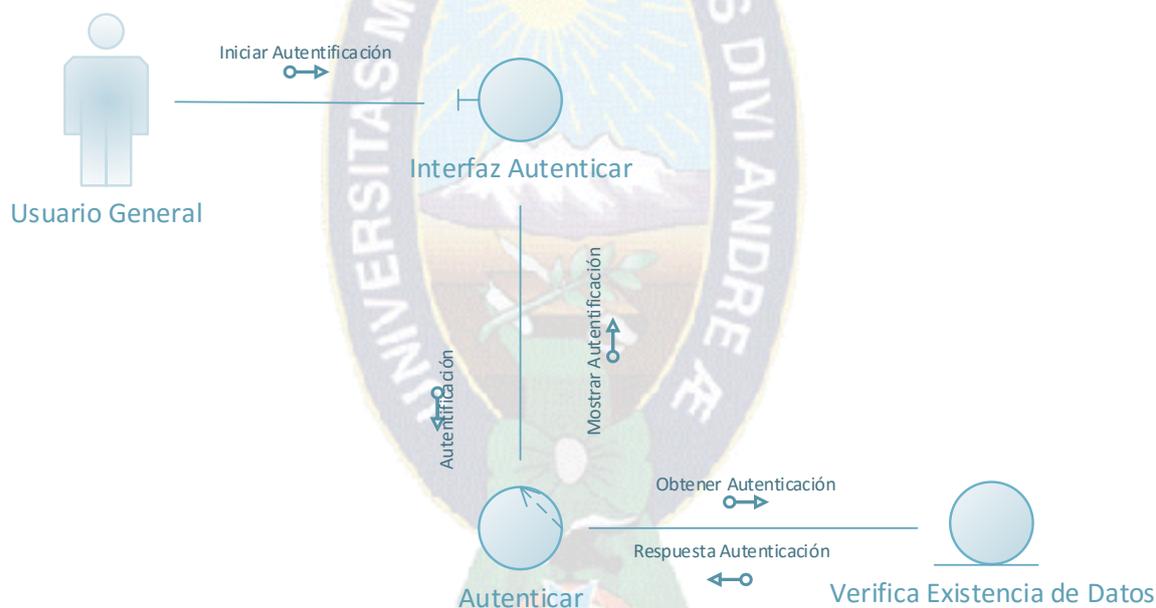
⁴ Sistema Experto de destinos policiales

Representación de los diagramas de robustez guarda relación con la especificación de los casos de uso, debido a que el diseño de ambos debe ser retroalimentado y actualizado según se realice el modelado. A continuación se muestra el modelado para cada caso de uso del sistema.

Diagrama de robustez: Autenticar al usuario para acceder al sistema.

La Figura 3.16 corresponde al diagrama de robustez de la ficha de especificación del Caso de Uso Identificación de Usuario (ver Tabla 3.7), el diagrama especifica las interfaces que serán desplegadas en el proceso de autenticación, los procesos internos del sistema y las salidas esperados.

Figura 3.17 Diagrama de Robustez de Identificación de Usuario

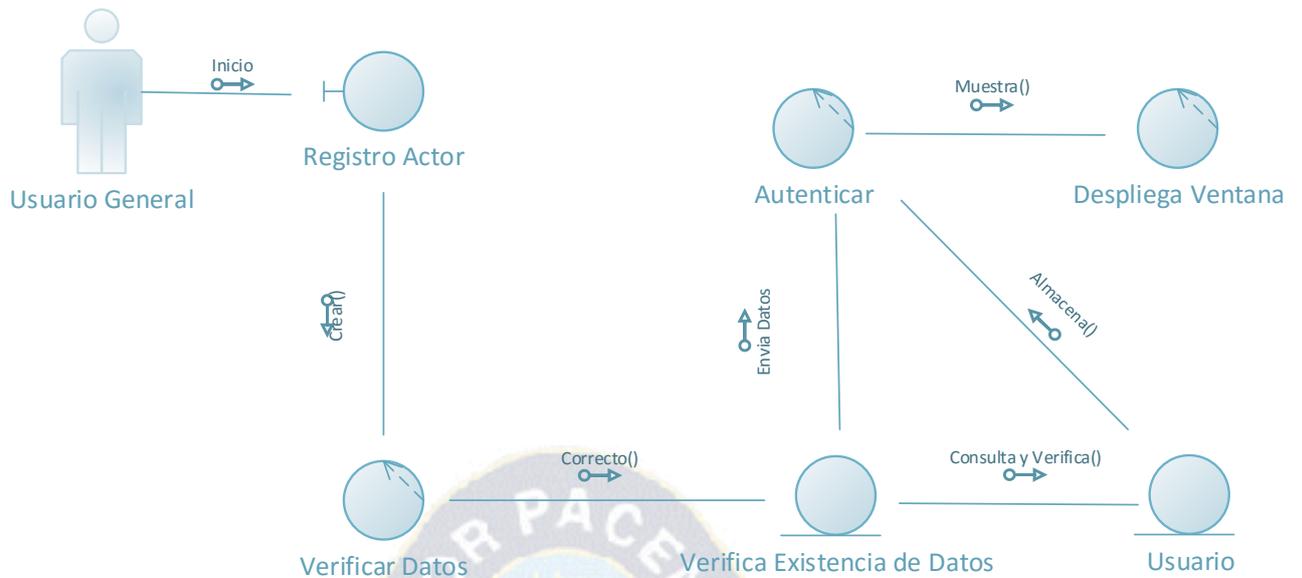


Fuente: Elaboración Propia

Diagrama de Robustez: Realizar el registro de Actores.

La Figura 3.18 corresponde al diagrama de robustez de la ficha de especificación del registro de Administradores (ver Tabla 3.8), el diagrama especifica las interfaces que serán desplegadas en el proceso de registro de un nuevo actor y las modificaciones que se realizarán.

Figura N° 3.18 Diagrama de Robustez Registro de Actores



Fuente: Elaboración Propia

Diagrama de robustez: Actualizar el perfil de Actores.

La Figura 3.19 corresponde al diagrama de robustez de la ficha de especificación de Actualización de Perfil del Usuario, el diagrama especifica las interfaces que serán desplegadas en el proceso de eliminación y modificación de datos de los actores.

Figura N° 3.19 Diagrama de Robustez Actualizar Perfil de Usuario X

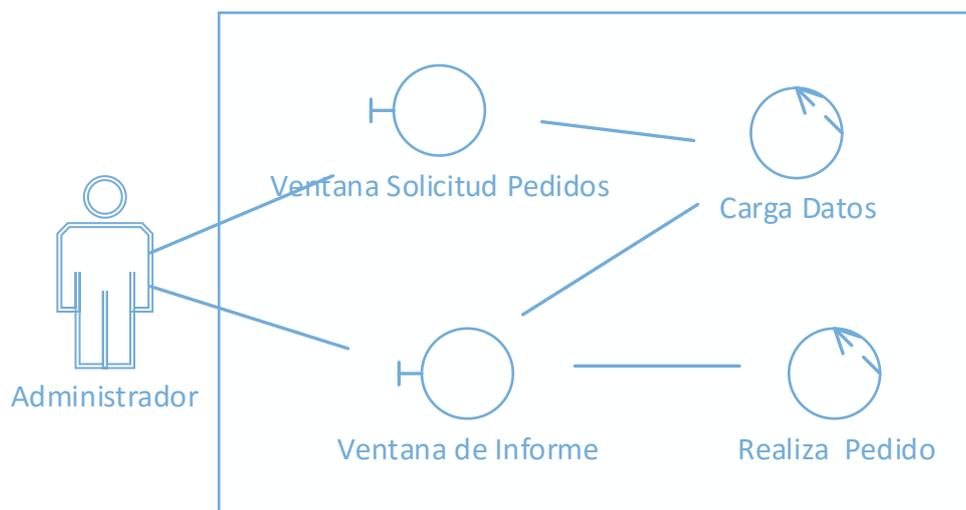


Fuente Elaboración Propia

Diagrama de robustez: Registro de los análisis para la designación.

La Figura 3.20 corresponde al diagrama de robustez de la ficha de especificación (ver Tabla 3.12), el diagrama especifica las interfaces que serán desplegadas en el proceso de consulta del diagnóstico y tratamiento, los procesos internos del sistema y los mensajes o parámetros de salida esperados.

Figura N° 3.20 Diagrama de Robustez Registro de los Pedidos de clientes



Fuente Elaboración Propia

3.2.2 Diseño del Sistema WEB

Se define los elementos del sistema WEB propuesto de acuerdo a sus características y funcionalidades definidas en el Capítulo Dos. A continuación se desarrolla la estructura de cada uno de los componentes.

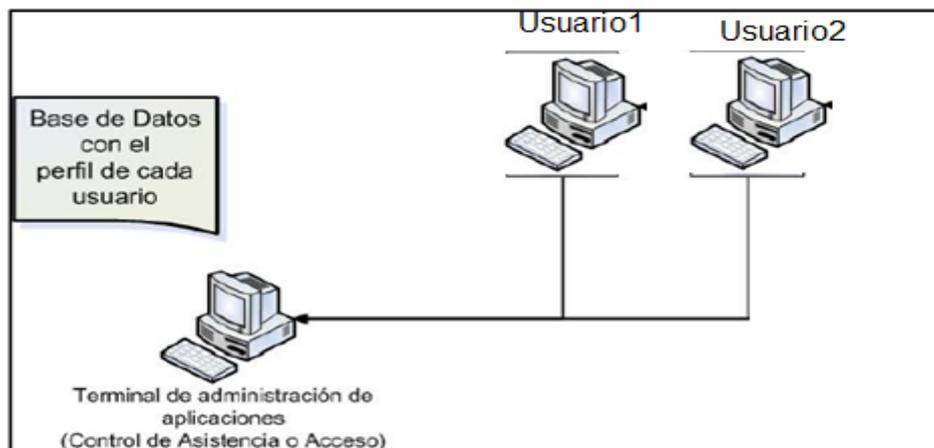
3.2.4 Diseño del Software

El diseño del software comprende la elaboración de diagramas UML estáticos, dinámicos, estructurales del sistema, que a continuación se presentan.

3.2.4.1 Diseño de la arquitectura del sistema

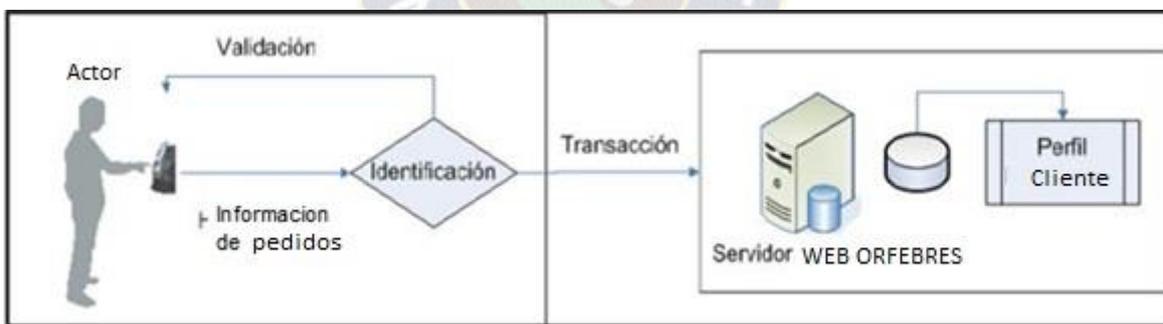
La etapa del diseño de la arquitectura se refiere a la planificación del hardware, software, y a la infraestructura de comunicaciones para el nuevo sistema como se puede observar en las figuras siguientes.

Figura 3.22 Diseño de la arquitectura del sistema



Fuente: Elaboración Propia

Figura 4.27 Proceso de Diagnostico del Usuario

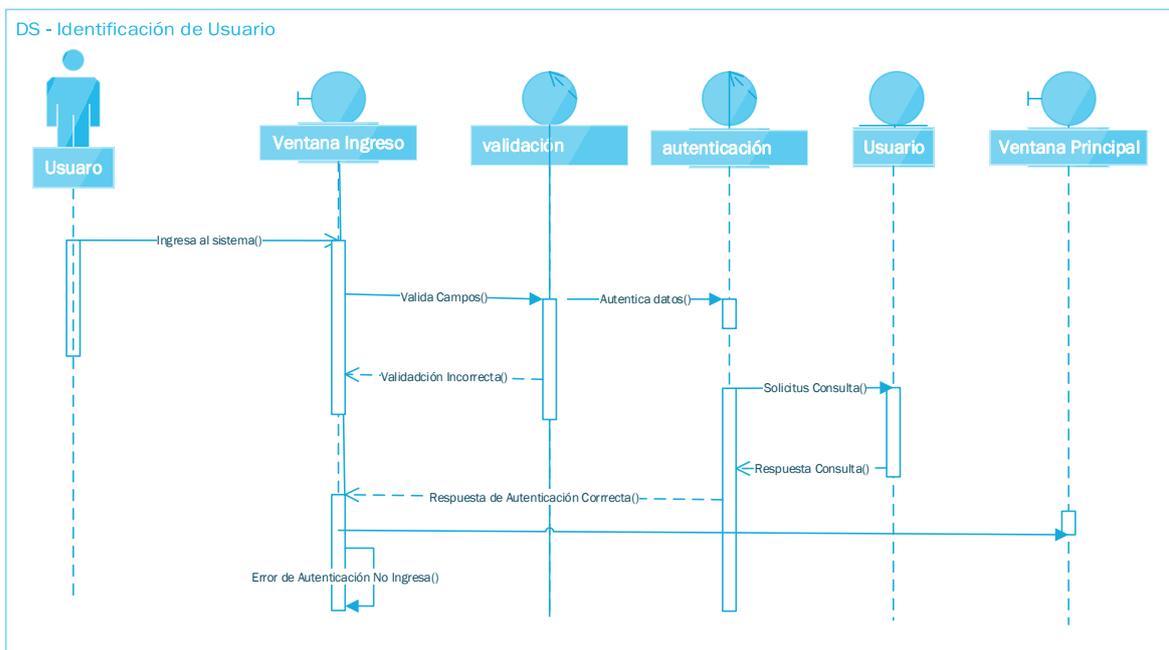


Fuente: Elaboración Propia

3.2.4.2 Diagramas de Secuencia

Los diagramas de secuencia reflejan el comportamiento del sistema y la relación con objetos internos del sistema, es decir, interfaces gráficas de usuario, procesos de control y entidades. Los diagramas de secuencia fueron definidos en base a los diagramas de robustez (ver Sección 3.2.1.4), como se describe a continuación.

Figura N° 3.23 Diagrama de Secuencia de Identificación de Usuarios

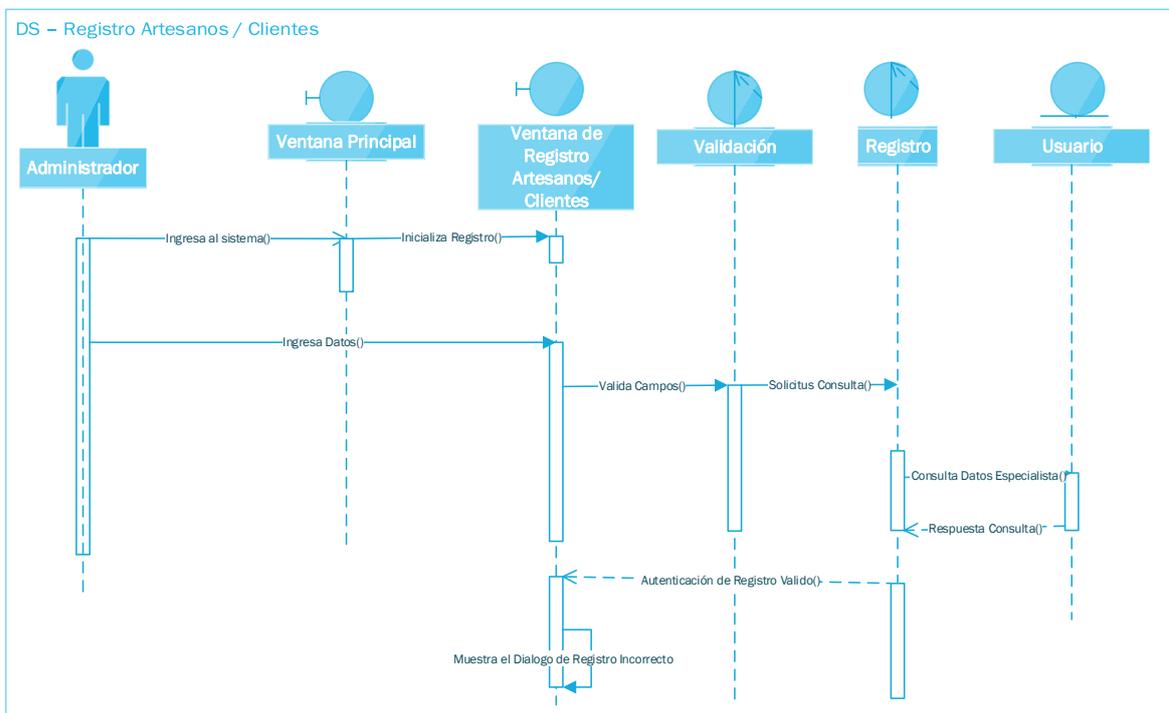


Fuente: Elaboración Propia

El Diagrama de Secuencia de la Figura 3.37 describe la Identificación de usuario para el ingreso del sistema.

1. El usuario ingresa al sistema cual se despliega la ventana de ingreso.
2. El usuario ingresa al sistema proporcionando datos de usuario y contraseña, hace clic en ingresar pero caso contrario no podrá acceder.
3. El sistema valida los campos, envía datos para la validación de autenticación, caso contrario no ingresa y envía un mensaje (login incorrecto).
4. El método de autenticación verifica los parámetros de Usuario y Contraseña si existe en la base de datos ingresa a la ventana principal, caso contrario no ingresará al sistema.

Figura N° 3.24 Diagrama de Secuencia de Registro de Artesanos

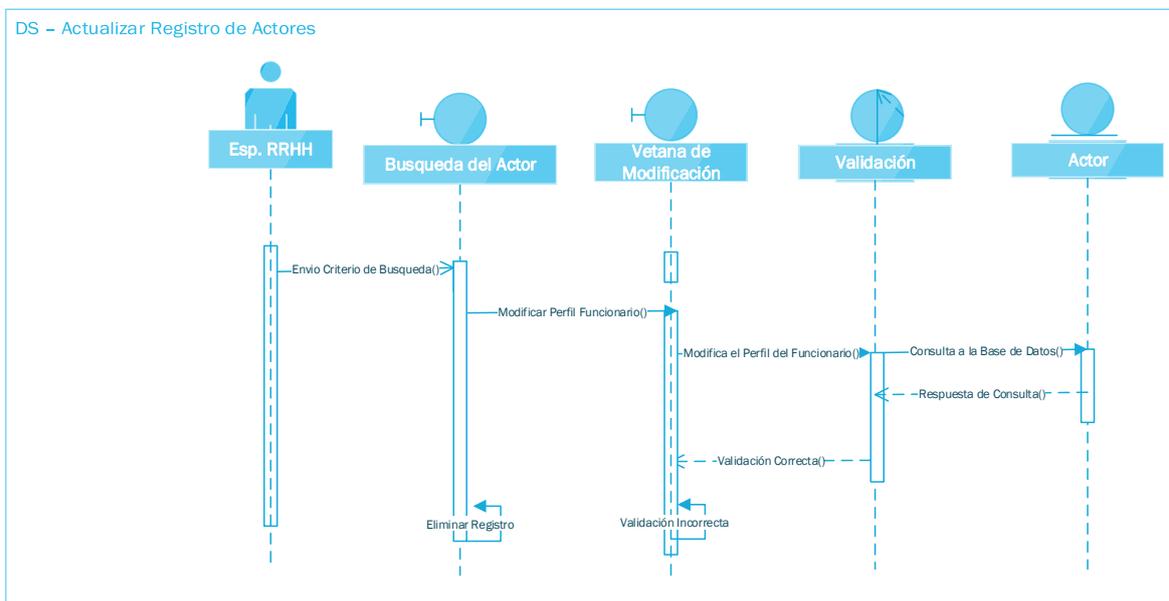


Fuente: Elaboración Propia

El Diagrama de Secuencia de la Figura 3.24 muestra el registro de Artesanos.

1. El administrador ingresa en el perfil de ventana principal, hace clic en registro del Artesano o Cliente, cual despliega la ventana de registro.
2. El administrador ingresa los datos de requeridos, realiza clic en guardar caso contrario cancelar para salir del registro.
3. El sistema realiza la validación de los campos si es correcta, el sistema enviará los parámetros necesarios al método registrar; caso contrario, se mostrará un cuadro de diálogo de “validación incorrecta”.
4. El registro consultará a la base de datos de la entidad Artesano / Cliente, enviando los datos ingresados, en caso de que el registro sea correcta mostrará cuadro de diálogo de “Registro Correcto”, caso contrario “Registro Incorrecto”.

Figura N° 3.25 Diagrama de Secuencia de Actualizar Perfil de Usuarios



Fuente: Elaboración Propia

En el Diagrama de Secuencia de la Figura 3.25 se muestra el proceso de actualización de todos los actores.

1. El administrador hace clic en la ventana de búsqueda del actor e ingresa los criterios necesarios de búsqueda.
2. El administrador hace clic en modificar y se despliega la ventana de modificación.
3. El administrador ingresa los datos requeridos del Administrador que desea modificar, y hace clic en guardar, y se inicializa la validación de datos del actor.
4. Si es correcta la validación se mostrará un cuadro de diálogo de “Modificación Correcta”, caso contrario se mostrará “Modificación Incorrecta” o “Validación Incorrecta”.
5. El Administrador hace clic en eliminar y se despliega la ventana de cuadro de diálogo de información.

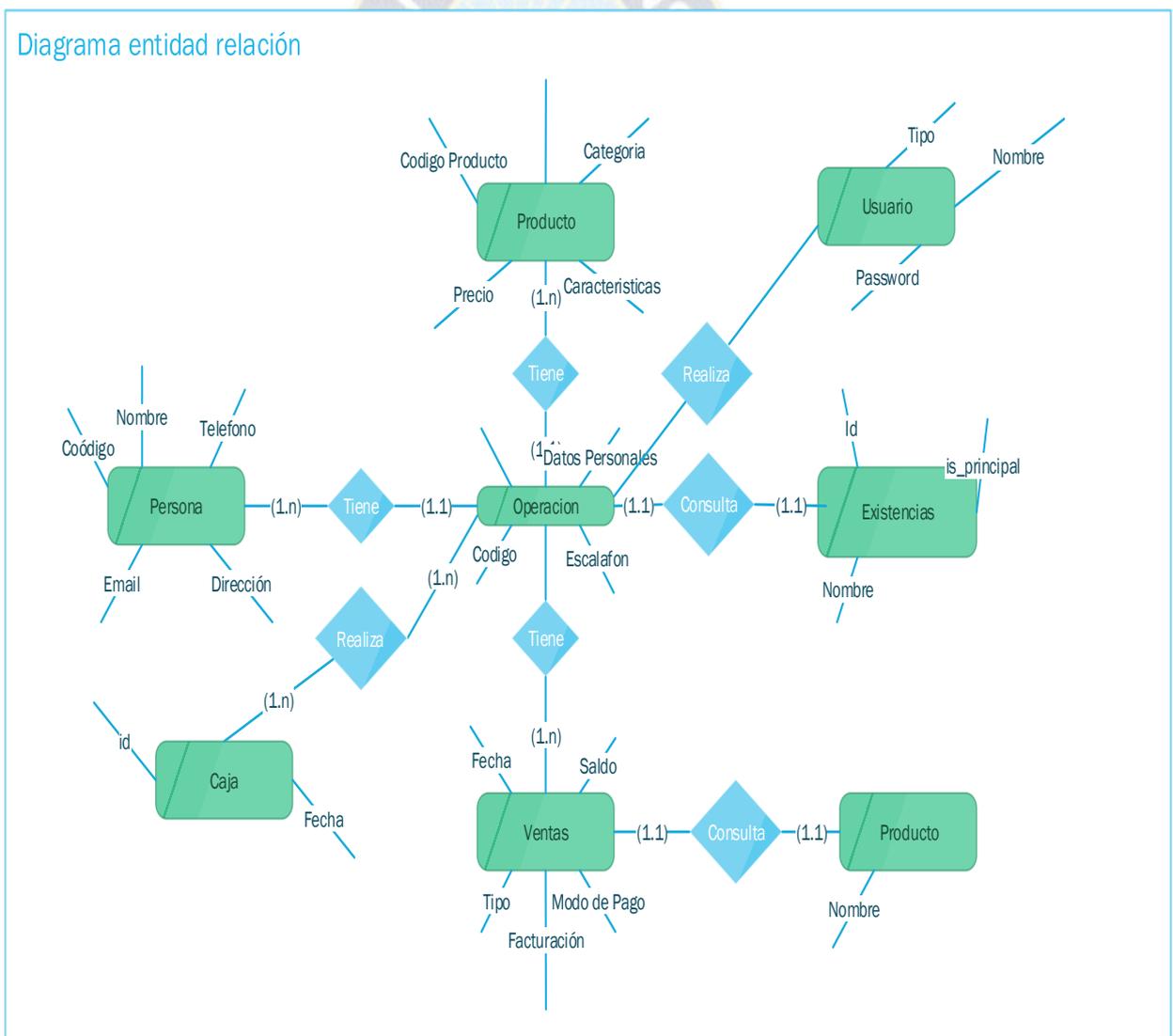
Con la determinación de los diagramas de secuencia, a continuación se determinan los diagramas de componentes del sistema.

3.2.4.4 Diseño Conceptual de la Base de Datos

Es el proceso que determina la organización de la base de datos, incluidos su estructura, contenido y las aplicaciones que se han de desarrollar. El diseño de la base de datos está compuesto por tres etapas: diseño conceptual, diseño lógico y diseño físico.

El modelo conceptual del sistema es representado a través del modelo entidad-relación, que posee conceptos como ser: entidades, relaciones, atributos, dominios de atributos, identificadores y jerarquías de generalización. En la Figura 3.23 se muestra el diagrama entidad – relación.

Figura N° 3.26 Diagrama Entidad-Relación de la Base de Datos



Fuente Elaboración Propia

3.2.4.5 Diseño Lógico de la Base de Datos

El diagrama conceptual de base de datos, en la presente sección se describirá el modelo lógico, que tiene como objetivo convertir los esquemas conceptuales locales en un esquema lógico global que se ajuste al modelo de sistema de gestión de base de datos.

Los modelos de bases de datos más extendidos son el modelo relacional, el modelo de red y el modelo jerárquico. Para el diseño de la base de datos se utilizó el modelo relacional, el cual considera la base de datos como una colección de relaciones, donde la relación es considerada una tabla compuesta por filas, y cada fila con un conjunto de campos.

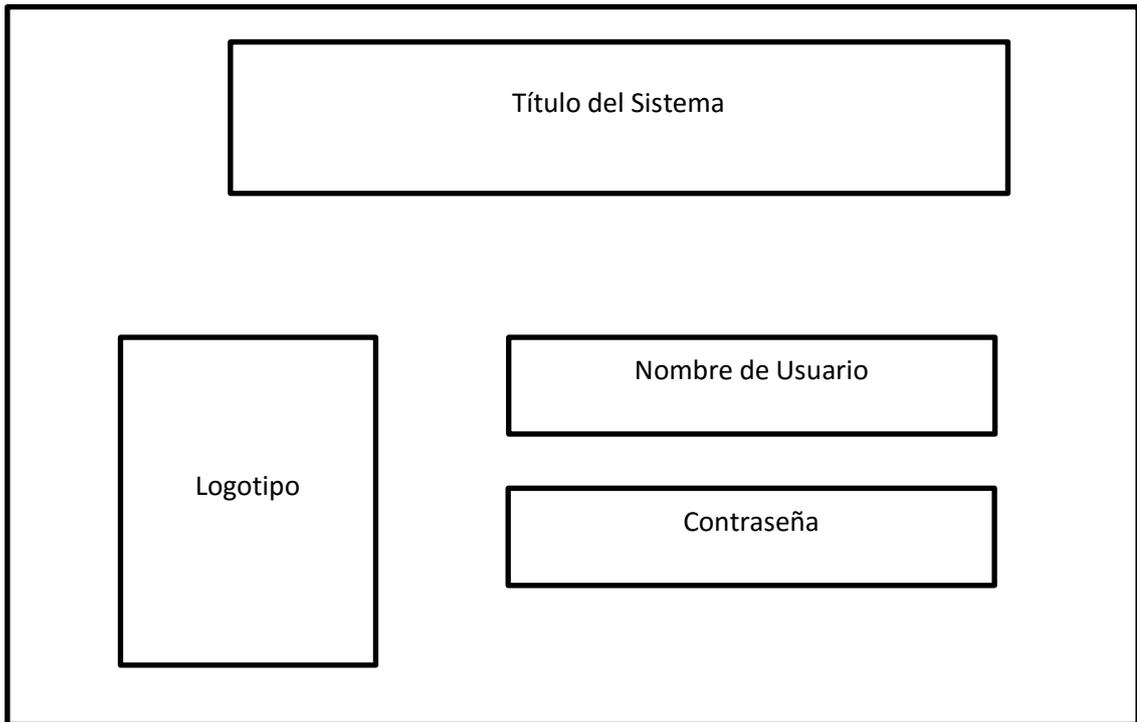
3.2.4.5 Diseño Navegacional

Para el sistema definen la relación interna entre los objetos físicos, es decir, archivos, cabeceras, bibliotecas compartidas, módulos, ejecutables, paquetes, librerías, etc. que hacen posible el funcionamiento modular o conjunto del software. A continuación se determinan los diagramas de componentes.

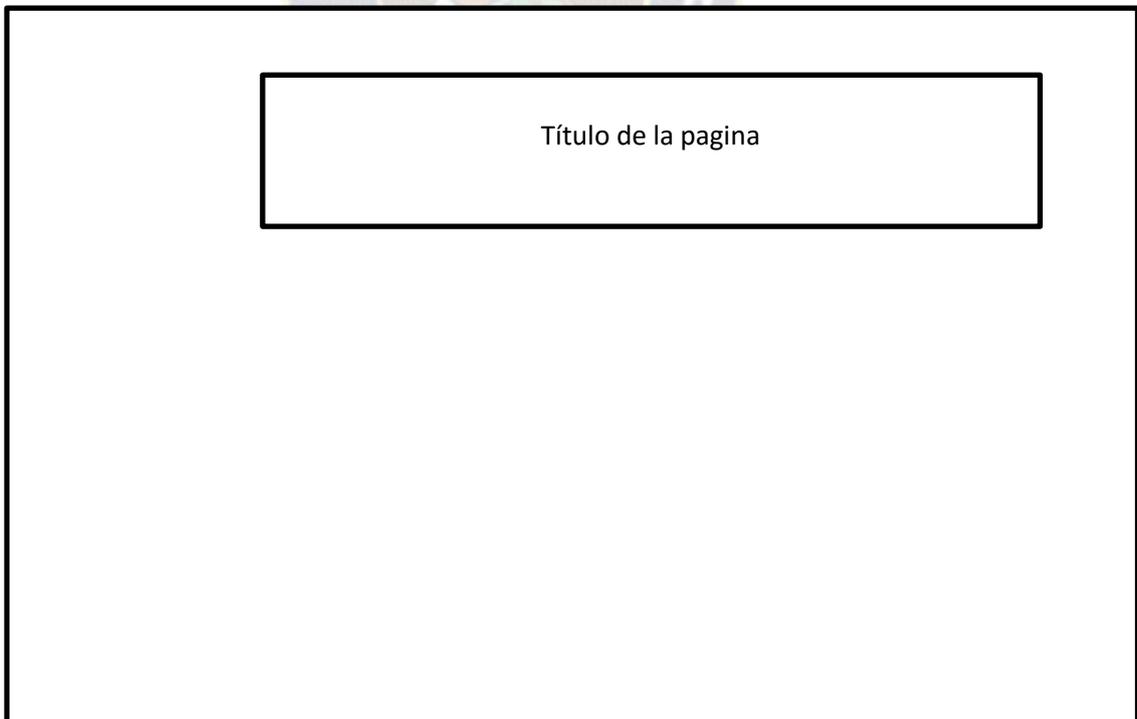
Fuente Elaboración Propia

Interfaces Gráficas de Usuario

La interfaz de usuario es el enlace entre el Sistema WEB y el usuario que es el



artesano y también cliente. Para el desarrollo del sistema que sea una



herramienta efectiva, debe incorporar mecanismos eficientes para mostrar y obtener de forma fácil ya agradable.

El presente sistema tiene opciones para el usuario pueda realizar las debidas consultas, modificación y borrado de datos mediante menús adecuados para tener una mejor comunicación del usuario y el sistema.

Conforme a la metodología de ICONIX a continuación se presentan las Interfaces Gráficas de Usuario (GUI) diseñadas para perfil de usuario que considera el sistema web, es decir: administrador, artesano y cliente.

Interfaz de Inicio

En la primera ventana se visualiza al iniciar la aplicación, es la ventana que da acceso a través de datos de usuario y contraseña.

Fuente Elaboración Propia

Fuente Elaboración Propia



Fuente Elaboración Propia



Fuente Elaboración Propia

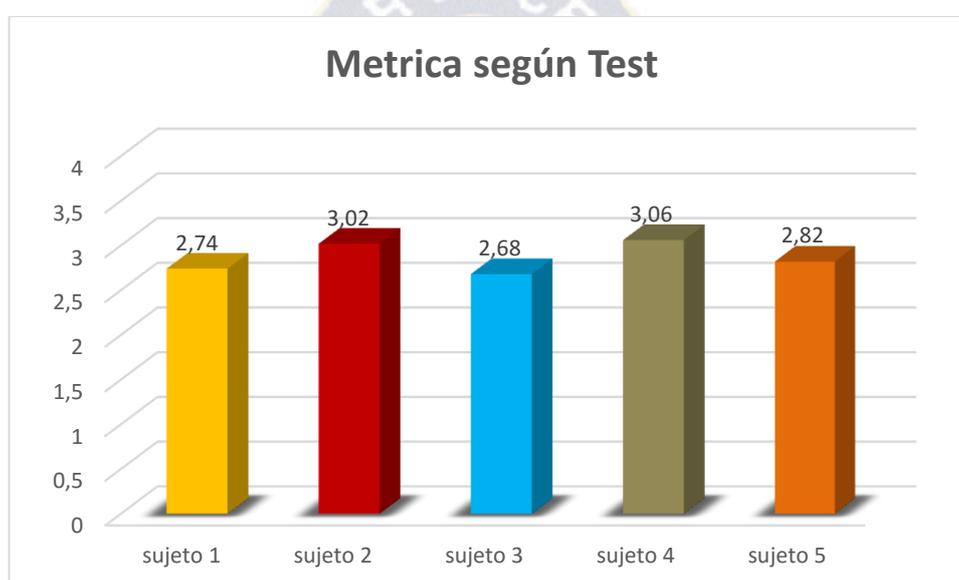
Para más detalle vea el anexo A

3.3 CALIDAD DEL SOFTWARE

3.3.1 Escalonamiento De Likert

Se ha visto por conveniente utilizar el método escalonamiento de Likert, el cual realiza, la evaluación de la calidad del programa desarrollado. En base a encuestas realizadas a personas entendidas para ello, es decir que se tomaron encuestas a informáticos y estudiantes de la carrera de informática, los que evaluaron. Este test de 34 preguntas las que contenían especificaciones del sistema WEB. Para ver encuesta refiérase al apéndice B.

Las conclusiones a las que se llegaron según el análisis de datos son:



Por ejemplo el sujeto 1, califico, las siguientes puntuaciones, respecto al sistema WEB:

11 preguntas con respuesta muy adecuado	→ 4 x 11	= 44
13 preguntas con respuestas adecuado	→ 3 x 13	= 39
3 preguntas con respuesta poco adecuado	→ 2 x 3	= 6
4 preguntas con respuesta Nada adecuado	→ 1 x 4	= 4
	TOTAL	= 93 PTS.

$$Sujeto 1 = \frac{\sum puntos}{\text{Número de preguntas}} = \frac{93}{34} = 2,74$$

Si porcentualizamos esto:

$$Sujeto 1 = \frac{2,74}{4} * 100 = 68,25 \%$$

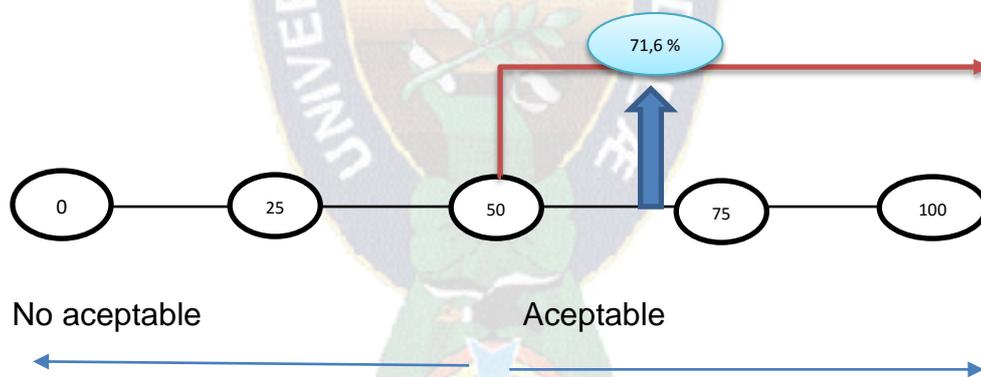
Que se interpreta que el software tiene un 68,25% de grado de aceptabilidad. Por lo tanto obteniendo un promedio general de las personas que evaluaron el sistema WEB, obtenemos:

$$Promedio Total = \frac{\sum sujetos i}{\text{Número de evaluadores}} = \frac{914,32}{5} = 2,86$$

Porcentualizando:

$$Pt = \frac{2,86}{4} * 100 = 71,6 \%$$

La escala que se utiliza para determinar si es aceptable el sistema WEB es el siguiente:



3.3.2 Equipo Requerido

Antes de instalar el programa, el equipo de cómputo deberá tener las siguientes características:

- Un ordenador personal con, al menos el procesador Pentium 4^o superior.
- Un disco Duro con al menos 500 MB.
- Una memoria RAM disponible de 512 MB.
- Plataforma cualquiera.
- Teclado, Mouse.

3.3.3 Usabilidad

Medida en que el sistema es fácil de aprender y fácil de utilizar. Se examinan los siguientes aspectos de usabilidad.

Facilidad de aprendizaje, el usuario experimento, comprendió como utilizar inicialmente el sistema y como a partir de ello a poder aprovechar las bondades y facilidades del mismo.

Se establece el sistema como predictivo, es decir que los conocimientos adquiridos por el usuario son suficientes para determinar los resultados de sus futuras interacciones, además de familiar, la correlación entre el conocimiento que tiene el usuario y el conocimiento que necesita para una interacción efectiva.

3.3.4 Flexibilidad

Las formas en las que el usuario y el sistema WEB, intercambia información. Se vio que el sistema tiene una iniciativa en la conducción del dialogo, existiendo la libertad para iniciar una acción al sistema, siendo su adaptabilidad congruente en todo aspecto.

3.3.5 Adaptabilidad

Se refiere, en qué medida el software imponen obligaciones para su uso, la misma que se evaluó basándose en los artículos que maneja la asociación.

Los Materiales, el sistema impone el uso de colocar características de las joyas y relojes para su exposición y venta.

Limitaciones para el usuario. El programa ofrece una navegación modular con el fin de efectuar una acción determinada dentro del sistema.

3.3.6 La Seguridad

En el sistema WEB, es de mucha importancia contar con una seguridad de primer nivel. Los requisitos de seguridad se consideraron explícitamente durante todo el proceso de desarrollo, lo que da lugar a la inclusión de fases o actividad dedicada a la seguridad.

El objetivo principal de la seguridad fue el de mantener las tres características primordiales de la información:

- Confidencialidad
- Integridad
- Disponibilidad

Las medidas de seguridad suelen centrarse principalmente en la eliminación o reducción de las vulnerabilidades del sistema

Se cuenta en tres niveles a saber:

1. Del equipo
2. De la Plataforma
3. Del Sistema WEB



CAPÍTULO IV

ANALISIS COSTO- BENEFICIO

4.1 DETERMINACIÓN DE COSTOS

En el desarrollo de cualquier investigación científica se toma en cuenta factores de costo en términos de tiempo, dinero y esfuerzos.

A continuación se detallan cada uno de los costos privados.

4.1.1 Costos de Material de Escritorio.

Los costos de los materiales de escritorio son requeridos para la elaboración del presente proyecto se describen en la tabla 4.1.

Tabla N° 4.1 Costos de Material de Escritorio
(Expresado en Bolivianos)

DESCRIPCION	Unidad	Cantidad	Costo Unitario (Bs.)	Costo Total (Bs.)
Papel Bond	Paquete	9	35	315
Cartuchos de Tinta Negra para impresora	Pieza	5	120	600
Otros	--	--	200	200
			Total Bs	1115

Fuente: Elaboración Propia

4.1.2 Costos de Información

Los costos de la información requerida para la elaboración del presente proyecto de grado que se describen en la siguiente tabla 4.2.

Tabla N° 4.2 Costos de Información

(Expresado en Bolivianos)

Detalle	Unidad	Cantidad	Costo Unitario (Bs.)	Costo Total (Bs.)
Internet	Hora	100	2	200
Libros	Libro	5	100	500
Fotocopias	Hoja	1000	0.1	100
Entrevistas	Entrevista	10	3	30
Total				830

Fuente: Elaboración Propia

4.1.3 Costos de Hardware

Previo a la determinación del equipo computacional a ser utilizado para el desarrollo del prototipo de software.

Por lo tanto el tamaño de memoria (RAM) y la velocidad del procesador son fundamentales en las corridas del sistema. A continuación se presenta los costos de hardware que con el que se desarrolla y lleva a cabo las corridas del software.

Una computadora personal con las siguientes características principales:

Tabla N° 4.3 Características de Hardware

(Expresado en Bolivianos)

Especificaciones	Equipo requerido
-------------------------	-------------------------

Microprocesador	Intel Core i5-4460S
Memoria RAM	4 GB
Disco Duro	500 GB
Monitor	LCD de 16 pulgadas
Dispositivos de Entrada/Salida	Mouse/Teclado/Lector de CD-ROM

Fuente: Elaboración Propia

El costo promedio de una computadora personal con estos mínimos requerimientos según precios del mercado es de **Bs 3.758**.

Sin embargo, por otra parte entre los requerimientos mínimos de hardware también se considera una impresora, la cual no posee la institución, es así que a continuación se describe las especificaciones mínimas del mismo adjunto al costo promedio de adquisición.

Tabla N° 4.4 Requerimientos y Costo de la impresora
(Expresado en Bolivianos)

Especificaciones	Descripción	Costo (Bs)
Modelo Sugerente	Epson L210	1.200
Tinta	1 Cartucho de color negro, PG-40	300
Costo total	1500	

Fuente: Elaboración Propia

4.1.4 Costos de Elaboración del Software

El modelo de proceso de software que se utilizó para desarrollar el software fue el **modelo de construcción de prototipos**; el costo total viene dado por la cantidad de tiempo y esfuerzo empleada en cada iteración del mismo.

4.1.5 Costos de Licencias de Software

Se tomara en cuenta a los recursos de software necesarios para el proyecto, el sistema operativo, el lenguaje de programación y el gestor de base de datos, para determinar el costo total de licencias de software. Que se detallan en la siguiente tabla 4.5:

Tabla 4.5 Costos de Licencias de Software

(Expresado en Bolivianos)

Software	Características	Costos (BS.)
Sistema Operativo	Windows	1.983,6
Lenguaje de Programación	PHP 5.0	0
Gestor de Base de Datos	MySql	0
Total		1.983,6

Fuente: [Microsoft, 2007]

4.1.6 Costo de desarrollo de software (Cocoma)

Para el proyecto en cuestión se utilizara para la estimación del costo de desarrollo de software el modelo:

COMPOSION DE APLICACIONES	DISEÑO TEMPRANO	POST ARQUITECTURA
Tomando las primeras etapas del prototipo	La estimación de los anteriores resultados para utilizando de un modelo post arquitectura	El desarrollo y el mantenimiento

El modelo de post-arquitectura predice el esfuerzo de desarrollo del software, personas-mes (PM),

Orgánico: proyectos relativamente sencillos, menores de 50 KDLC líneas de código, en los cuales se tiene experiencia de proyectos similares y se encuentran en entornos estables aproximadamente.

PROYECTO DE SOFTWARE	a	b	c	d
Orgánico	2,4	1,05	2,5	0,38
Semi-acoplado	3,0	1,12	2,5	0,35
Empotrado	3,6	1,20	2,5	0,32

La ecuación del esfuerzo de **COCOMO** es:

$E = \text{Esfuerzo} = a \text{ KLDC}^b \text{ (PERSONA X MES)}$

KLCD Es El Número De Líneas De Código En Miles

La **ecuación del tiempo de desarrollo** es:

$T = \text{Tiempo de duración del desarrollo} = c \text{ Esfuerzo}^d \text{ (mes)}$

Permite estimar el esfuerzo costo y duración de cualquier proyecto de software

Proyecto tiene = 2.800 líneas de código

$$E = 2.4 * KLOC^{1.05}$$

$$E = 2.4 * 2.8^{1.05}$$

$$E = 7 \text{ hombre-mes}$$

Para calcular la duración del proyecto usamos la estimación de esfuerzo aproximadamente:

$$D = c * E^{0.38}$$

$$D = 2.5 * 7^{0.38}$$

$$D = 5 \text{ meses}$$

El valor de la duración del proyecto permite al planificador recomendar un número de personas N para el proyecto.

$$N = E / D$$

$$N = 7 / 5$$

$$N = 1,4 \text{ personas} \gg 1 \text{ persona}$$

Por lo tanto con los resultados del presente proyecto debería terminarse en aproximadamente 5 meses por una personas.

El costo promedio de un programador es de 1.050 bs/mes

Así tomando en cuenta obtenemos un total de:

$$\text{Costo} = D * \text{PRO}$$

Dónde:

D= Duración del proyecto

PRO=Costo promedio de un programador

$$\text{Costo} = (5) * (1050) = 5250 \text{bs/mes}$$

El factor multiplicador es 0.51 que debe ser tomado en cuenta para obtener el producto final

Tabla 4.6 Costo Total
(Expresado en Bolivianos)

DESCRIPCIÓN	COSTO Bs
Costo hardware	3.358
Costo impresora	1.500
Costo de software	1.983,6
Costo del material de escritorio	343
Costo de Información	830
Desarrollo de producto	5.250
Total	13.664,6

Fuente: Elaboración Propia

4.2 ANALISIS DE BENEFICIOS

El análisis de beneficios consiste en determinar los beneficios tangibles e intangibles que brinda la implementación del sistema WEB a la Asociación Mixta de Orfebres y Relojeros A.M.O.R.

4.2.1 Beneficios Intangibles

Los beneficios intangibles se entienden como aquellos beneficios de carácter cualitativo y no cuantificable, debido a que describen ventajas que brinda el sistema WEB en tiempo de ejecución.

Entre los beneficios intangibles principales se encuentran los siguientes:

- Permite determinar la probabilidad de crear nuevos mercados para los artesanos, con resultados económicos favorables.
- La toma de decisiones que debe realizar el Artesano respecto a la determinación de cual producto es el más requerido por los clientes.
- El sistema permite sistematizar el proceso de registro de pedidos, logrando un ahorro en recursos de almacenamiento (papel) y esfuerzo de recursos humanos (personal involucrado).
- Facilitará el almacenamiento de la información para determinar reportes inmediatos, en línea.
- Agilizara que el servicio de atención al cliente que ofrece la Asociación de orfebres, para que sea más competitivo, así manteniendo una buena imagen del profesional de la asociación.
- Accederá el manejo del historial e inventarios en soporte digital lo que facilite su acceso por parte de los administrativos.

4.2.2 Beneficios Tangibles

Los beneficios tangibles se entienden como aquellos beneficios que pueden ser medidos en términos monetarios y que representan una ventaja en cuanto a costos que se pueden ahorrar con la utilización del sistema WEB.

De acuerdo a la finalidad del sistema WEB, los beneficios tangibles que brinda son características asociadas en mayor magnitud a la sistematización del proceso de recolección y búsqueda de información, reducción de tiempo, pedidos inmediatos y respuestas inmediatas.

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

- Se desarrolló un formulario vía web para que los clientes realcen sus pedidos, solicitudes que son rápidamente atendidas.
- Se cuenta con un módulo que registra los productos que ingresan y salen del almacén con la finalidad de hacer un control al inventario.
- Se tiene un módulo que permite la generación de reportes de stock y movimiento de productos, para de esta manera, evitar pedidos mal efectuados, haciendo más eficiente los pedidos.
- La búsqueda de los productos que se encuentran en el almacén, según su clasificación de una manera más eficiente y más rápida
- Se proporcionó al departamento de comercialización la disposición de diferentes informes sobre los productos e inventario de la asociación de orfebres.

5.2 RECOMENDACIONES

- Que se puedan implementar el pago a través de la WEB.
- La Asociación Mixta de Orfebres y Relojeros, debe de invertir en la compra de un dominio a la brevedad posible, alojamiento y SEO, para el buen funcionamiento del sistema.
- Todos los socios deberán tener un acceso constante para modificar sus productos y de eta manera interactuar con el sistema para mejorar SEO.

BIBLIOGRAFÍA

- [KENDALL SCOTT & DOUG ROSENBERG., 2001]** Kendall Scott & Doug Rosenberg. (2001). Applying Use Case Driven Object Modeling with UML: An Annotated e-Commerce Example. Addison Wesley.
- [MOLINA, SÁNCHEZ, LETELIER & SÁNCHEZ, 1997].** Antonio Molina Marco, Pedro Sánchez Palma, Patricio Letelier Torres, Juan Sánchez Díaz, Universidad Politécnica de Valencia. Departamento de Sistemas Informáticos y Computación, Metodología y tecnología de la programación, Ed. Univ. Politéc. Valencia, 1997
- [LAHOZ-BELTRA, 2004]** LAHOZ-BELTRA, Rafael. Bioinformática: Simulación, vida artificial e inteligencia artificial. Ediciones Díaz de Santos, S.A. Madrid, España, 2004.
- [POPKIN SOFTWARE AND SYSTEMS, 2002]** Popkin Software and Systems Modelado de Sistemas con UML. Popkin Software and Systems, 2002. <http://es.tldp.org/Tutoriales/doc-modelado-sistemas-UML/doc-modelado-sistemas-uml.pdf>
- [PRESSMAN, 2010].** PRESSMAN Roger S. (2005), Ingeniería de Software. Un enfoque Práctico. Séptima Edición, Mexico D.F.
- [QUISPE VERA, 2012].** Quispe Vera, I. (s.f.). Metodología de Desarrollo de Software. Obtenido de Metodología de Desarrollo de Software: <http://jqv-sistemas.xtrweb.com/docs/Metodologia-ICONIX->

2012.pdf

[ROMERO, GOMEZ, DAFONCE, & PENOUSAL, 2007]. Romero, J. J., Gomez, A., Dafonte, C., & Penousal, F. J. (2007). Inteligencia Artificial y Computacion Avanzada. Fundacion Alfredo Brañas.

[ROSENBERG & STEPHENS & COLLINS-COPE, 2005] ROSENBERG Doug & STEPHENS Matt & COLLINS-COPE Mark (2005). Agile Development with ICONIX Process. People, Process, and Pragmatism. Estados Unidos.

[RUIZ, 1999] Ruiz González, F. (1999). COCOMO v2: Modelo de estimación de costes para proyectos software.

[SOMMEERVILLE, 2005] Sommeerville, I. (2005). Ingenieria de Software (Septima ed.). Madrid: Pearson Educacion S.A.

[UPV, 2006] UPV. (2006). Modelo Cocomo. <http://www.sc.ehu.es/jiwdocoj/mmis/cocomo.htm> .

Referencia de la Web

[WWW1, 2014] Ingeniería de Software, 2001, Recuperado el 03 de 10 de 2016, <http://ingenieraupoliana.blogspot.com/2010/09/aplicaciones-del-software.html>

[WWW2, 2011] Ingeniería de conocimiento, Recuperado el 01 de 10 de 2016, <http://laboratorios.fi.uba.ar/lsi/rgm/CD-IC/Ingenieria-del-Conocimiento.pdf>

ANEXOS



The logo of Universidad Mayor Pacensis Divi Anselmi is a circular emblem. The top half features a sun with rays rising over a mountain range. The bottom half shows a green landscape with a blue cross in the center. The text 'UNIVERSITAS MAJOR PACENSIS DIVI ANSELMI' is written around the perimeter of the circle.

ANEXOS

ANEXO A. MATRIZ DE MARCO LÓGICO

Variables	Dimensiones	Indicadores	Instrumento
Nivel de Ingreso	Perdida Ganancia	Decremento de capital Ingreso de Dinero	Observación Observación
Nivel de Stock	Existe en almacén No existe en almacén	Productos disponibles para la venta. Productos no disponibles para la venta	Observación Observación
Producto	Entradas	Aumenta la cantidad de productos	Libro de registro de entrada de productos
Venta	Número de ventas efectuadas en por día	Cantidad de productos vendidos durante el día	Libro de ventas

ANEXO B: ÁRBOL DE PROBLEMAS

Pérdida de tiempo y mala información en los reportes, datos que coadyuvan en la toma de decisiones, para la adquisición de materia

La solicitud de pedidos que realiza el vendedor es realizada .manualmente,

Las ventas no son eficaces, provocan demoras en las entregas y pedidos que se realizan.

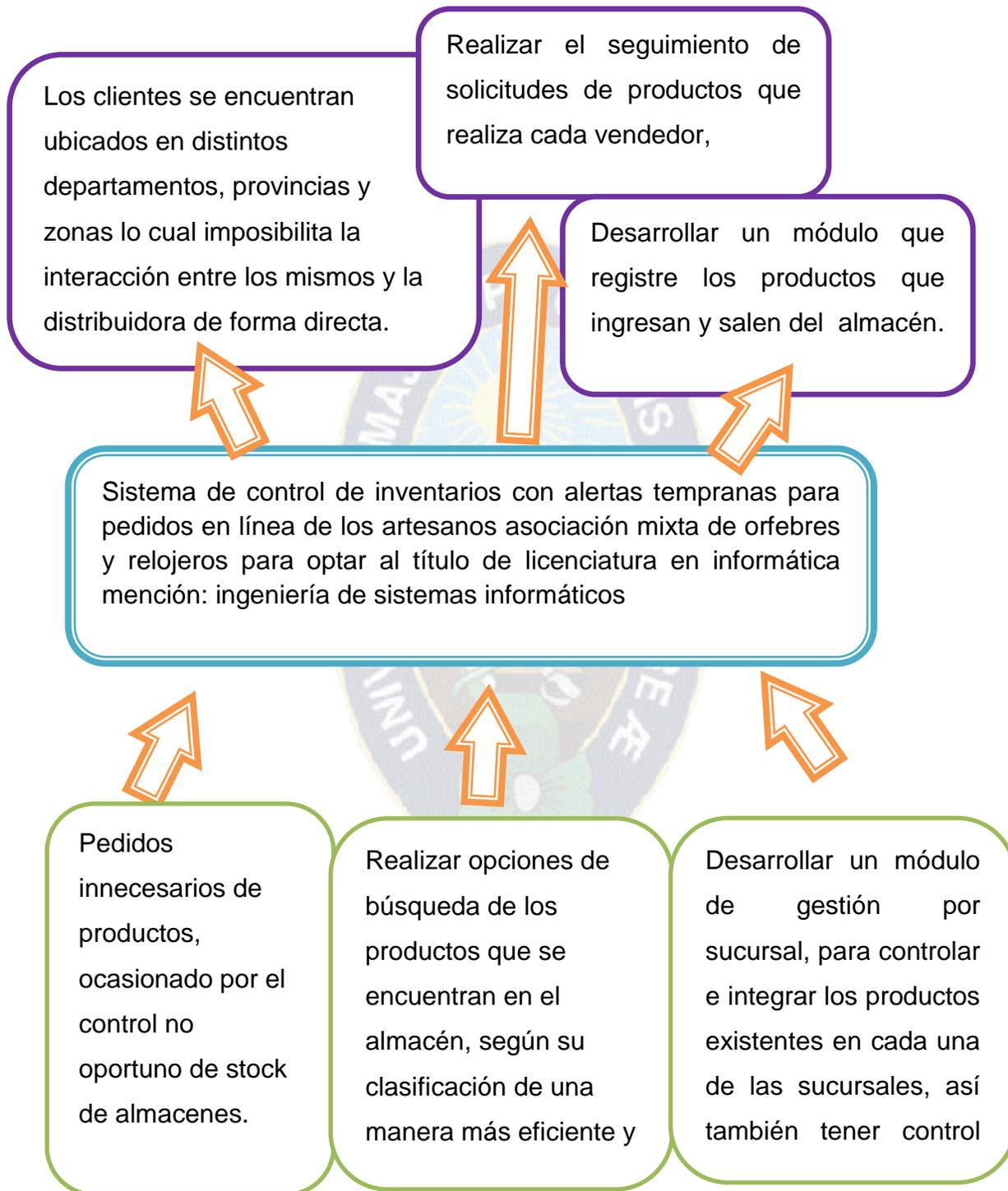
La unidad del almacén de La Asociación Mixta de Orfebres y Relojeros A.M.O.R., no cuenta con una administración, seguimiento de control de inventarios, debido a que el manejo de la información es realizada manualmente, imposibilitando acceder a la generación de informes, reportes solicitados, factura y entrega de pedidos, generando un retardo en la toma de decisiones, además existe dificultad con los pedidos que realiza cada vendedor cuando se encuentra fuera de la organización o fuera del departamento ya que los pedidos se hacen manualmente y generando demora en hacer los cálculos correspondientes del pedido.

La dificultad de tiempo de consulta del estudio detallado y el análisis de los resultados de ventas y pedidos con lleva a la demora.

Los clientes se encuentran ubicados en distintos departamentos, provincias y zonas lo cual imposibilita la interacción entre los mismos y la distribuidora de forma directa.

Demora en ubicar los productos existentes en la distribuidora, lo cual ocasiona que un vendedor tarde en ubicar dicho producto.

ANEXO C: ÁRBOL DE OBJETIVOS



ANEXO "D"

The screenshot shows the main interface of the A.M.O.R. S.R.L. system. The header includes the company name and logo, the text 'Fundada el 5 de marzo 1955', and the user role 'Administrador (Administrador)'. A sidebar on the left lists navigation options: Inicio, Pedidos, Vender, Inventario, Ventas, Compras, Datos, Finanzas, Reportes, and Administracion. The main content area displays 'Alerta de Inventario - Principal' with a table of inventory alerts. The table has columns for Observaciones, Nombre del producto, En Stock, and Estado. One row shows 'Reloj Bañado oro' with an 'En Stock' value of 10 and a state of 'Relojes de Plata'. A 'Limite' field is set to 10. A 'Nuevo Pedido' button is visible in the top right corner.

Observaciones	Nombre del producto	En Stock	Estado
Fecha terminado.	Reloj Bañado oro	10	Relojes de Plata

Gráfico N° 1 Captura pantalla principal al Ingreso

The screenshot shows the 'Pedidos' (Orders) section of the A.M.O.R. S.R.L. system. The header and sidebar are identical to the previous screenshot. The main content area displays 'Cotizaciones' (Quotes) with a table of order entries. The table has columns for Ver, Folio, Producto, Pago, Entrega, Total, Fecha, and Acción. One entry is shown with Folio #1, Producto 1, Pago Pendiente, Entrega Pendiente, Total Bs. 4,500, and Fecha 2016-11-16 11:28:39. Action buttons for 'Procesar' and 'Cancelar' are visible. A 'Nuevo Pedido' button is in the top right corner.

Ver	Folio	Producto	Pago	Entrega	Total	Fecha	Acción
<input type="checkbox"/>	#1	1	Pendiente	Pendiente	Bs. 4,500	2016-11-16 11:28:39	<input type="checkbox"/> Procesar <input type="checkbox"/> Cancelar

Gráfico N° 2 Captura pantalla Pedidos

ASOCIACIÓN MIXTA DE ORFEFRES Y RELOJEROS
A.M.O.R.
Fundada el 5 de marzo 1955

A.M.O.R. S.R.L. Administrador (Administrador)

ADMINISTRACION

- Inicio
- Inicio
- Pedidos
- Vender
- Inventario
- Ventas
- Compras
- Datos
- Finanzas
- Inventario
- Reportes
- Administracion

Pagina 1 de 1 Imprimir

Alerta de Inventario - Principal

Observaciones	Nombre del producto	En Stock	Estado
Fecha terminada.	Reloj Bañado oro	10	Relojes de Plata

Limite 1

Activar Windows
Ir a Configuración de PC para activar Windows.

Gráfico N° 3 Captura de alerta al inventario

ASOCIACIÓN MIXTA DE ORFEFRES Y RELOJEROS
A.M.O.R.
Fundada el 5 de marzo 1955

A.M.O.R. S.R.L. Administrador (Administrador)

ADMINISTRACION

- Inicio
- Inicio
- Pedidos
- Vender
- Inventario
- Ventas
- Compras
- Datos
- Finanzas
- Inventario
- Reportes
- Administracion

Reporte de Inventario

Principal -- PRODUCTOS -- 01/11/2016 24/11/2016 Imprimir

Imprimir en Excel (.xlsx)

Id	Producto	Cantidad	Operacion	Fecha
4	Reloj Bañado oro	8	salida	2016-11-24 14:46:12
3	Reloj Bañado oro	10	entrada	2016-11-24 14:38:03
2	Aretes de Oro 18 kl	10	salida	2016-11-16 11:28:40
1	Aretes de Oro 18 kl	100	entrada	2016-11-16 11:28:09

Activar Windows
Ir a Configuración de PC para activar Windows.

Gráfico N° 4 Captura reporte de inventario

ANEXO “E” EVALUACIÓN DEL SOFTWARE

NOMBRE: Universidad:

IDENTIFICACIÓN DEL PROGRAMA: SISTEMA DE CONTROL DE INVENTARIOS CON ALERTAS TEMPRANAS PARA PEDIDOS EN LÍNEA. NOMBRE COMERCIAL: SCIAT. FABRICANTE: Jorge Durán VESIÓN AÑO: 1 – 2016. EQUIPO MÍNIMO REQUERIDO: PENTIUM IV. PLATAFORMA: CUALQUIERA. USUARIOS POTENCIALES: PERSONAL DE A.M.O.R.

VALORACIÓN DE ELEMENTOS

ELEMENTO	MUY ADECUADO	ADECUADO	POCO ADECUADO	NADA ADECUADO
Guía rápida				
Manual				
Instalación y aprendizaje de uso				
Secuenciación de contenidos				
Accesibilidad de los datos				
Apoyo gráfico				
Apoyo de video				
Apoyo Oral				
Autosuficiencia de Software (No es necesario el uso del manual)				
La interactividad				
Iconos intangibles (analogía entre la imagen y la función que representa)				
Posibilidad de personalización				
Intuitivo				
Atractivo				
Sencillo				
Facilidad de salida y reinicio				
Permite la experimentación (Su propio uso genera aprendizaje)				
Da orientación ante errores				
Guía de aprendizaje				
Manejo fácil				
Motiva / estimula				
Concentra la atención sobre la tarea				
Facilita el aprendizaje				
Individualiza el aprendizaje				
Posibilita la conexión a redes				
Posibilita la conexión a base de datos				
Adecuación a la actividad				
Secuencia				
Ejemplificaciones				
Aprendizaje de uso				
Resistencia a errores				
Favorece al trabajo en equipo				
Desarrolla creatividad				
Favorece el esfuerzo personal				

