



Pontificia Universidad
Católica del Ecuador

**FACULTAD DE INGENIERÍA
ESCUELA DE SISTEMAS**

**DISERTACIÓN PREVIA A LA OBTENCIÓN DEL TÍTULO
DE INGENIERO DE SISTEMAS Y COMPUTACIÓN**

**ANÁLISIS, DISEÑO, DESARROLLO E IMPLANTACIÓN DE UN
SISTEMA WEB DE FACTURACIÓN Y CONTROL DE
INVENTARIO APLICADO AL TALLER MECÁNICO
“FRENICENTRO”.**

AUTOR: JOSUÉ CARRANZA ROMÁN

DIRECTOR: ING. JORGE ALARCÓN

Abril, 2017.

DEDICATORIA

El presente trabajo de disertación está dedicado a mis queridos padres y familiares, quienes han sido el pilar fundamental durante este largo camino universitario; en especial a mi mamacita Magdalena, pues con su ejemplo de perseverancia y lucha, me enseñó que, para alcanzar las metas, se debe mantener: valores, principios y jamás perder la esperanza.

No puedo dejar de lado, la guía de la Madre Dolorosa quien ha sido el apoyo más importante en mi vida, ya que siempre me ha brindado sabiduría, salud y ha sido mi fortaleza; sobre todo en esta etapa universitaria.

También se la dedico a mi abuelita, Esther Soria, quien fue un ejemplo de integridad, amor y respeto; sus consejos y cariño siempre me motivaron a perseguir mis sueños y no rendirme.

AGRADECIMIENTO

En el presente trabajo de Disertación quisiera expresar mi agradecimiento a Dios Padre, por bendecirme y por permitirme realizar este sueño tan anhelado.

A la Pontificia Universidad Católica del Ecuador por brindarme la oportunidad de estudiar y ser un digno profesional.

Al Director de Tesis, Ing. Jorge Alarcón, por su dedicación, paciencia y constante apoyo, quien con sus consejos, fue mi guía y motivación para culminar mis estudios y desarrollar este trabajo investigativo.

Un agradecimiento especial a las Autoridades y Personal Docente de la Facultad de Ingeniería, Escuela de Sistemas, por impartirme sus valiosos conocimientos y formación integral en valores, que son y serán mi fortaleza en mi vida personal y profesional.

Exteriorizo mi sincero agradecimiento al Ing. Roberto di Doménico, Gerente Propietario de la Mecánica Automotriz FRENICENTRO, por su valiosa apertura durante todo el desarrollo de la presente tesis, al facilitarme la información requerida para este efecto.

Finalmente, agradezco, muy sinceramente, a mis queridos padres por su incondicional apoyo económico, solidaridad y sabios consejos, que sabré llevarlos hoy y siempre.

TABLA DE CONTENIDOS

RESUMEN	10
---------------	----

CAPÍTULO 1: ANÁLISIS DE PROCESOS ACTUALES

1.1. SITUACIÓN ACTUAL DE LA EMPRESA.	11
1.1.1. Descripción de la Empresa.	11
1.1.2. Misión Empresarial:	11
1.1.3. Visión.....	11
1.1.4. Objetivos Empresariales.....	12
1.1.5. Estrategia Comercial.....	12
1.1.6. Proceso de Venta.	12
1.1.7. Problemas comunes en el Control de Inventarios.	13
1.1.8. Políticas De Control	14
1.1.9. Problemas encontrados Dentro Del Taller Mecánico Frenicentro.	14
1.1.10. Problemas con el Proceso de Facturación.	15
1.1.11. Estructura Organizativa	16
1.2. PROCESOS ACTUALES	17
1.2.1. Análisis de Gestión de Facturas.....	17
1.2.2. Análisis de Control de Inventarios.	18
1.2.3. Uso de sistema para emisión de facturas a través de páginas web.....	19

CAPÍTULO 2: METODOLOGÍAS APLICADAS PARA EL DESARROLLO DEL SISTEMA

2.1. METODOLOGÍA PARA EL ANÁLISIS DE REQUERIMIENTOS ÁNCORA.....	20
2.1.1. Fundamentos de Áncora	20
2.1.2. Alcances de Áncora.....	21
2.1.3. Características Principales.....	22
2.1.3.1. Guiones y Diálogos.	23
2.1.3.1.1. Guiones.....	23
2.1.3.1.2. Diálogos.....	24
2.1.3.2. Tablas y Glosarios	25
2.1.3.2.1. Tablas....	25
2.1.3.2.2. Glosarios.....	25
2.1.4. Modelo Entidad – Relación	25
2.1.5. Modelo de Objetos Semánticos (MOS).....	26
2.1.6. Definición y Características de los Objetos Semánticos.	26
2.1.6.1. Tipos de Objetos Semánticos.	27
2.1.6.2. Modelando con Objetos Semánticos.	27
2.2. Cálculo de Puntos de Función.	28
2.2.1. Asignación de pesos para ALI y AIE.	29
2.2.2. Ajuste de los Puntos de Función.	30

2.2.3. Técnica de Grupo Nominal.....	42
2.2.4. La Reunión de Reflexión y Diseño.....	43
2.2.5. El lugar para la RRD.	44
2.2.6. Los asistentes a RRD.....	44
2.2.7. La duración de la RRD.	44
2.2.8. La conducción de la RRD.....	44
2.2.9. Apoyos físicos para la RRD.	45
2.2.10. Los resultados de la RRD.	45
2.2.11. El Calendario para la RRD.	45
2.2.12. Ciclo de Vida.	45
2.3. Comparativa con otra Metodología.	49
2.4. Metodología de Programación Ágil XP Extreme Programming.....	50
2.4.1. Historia.....	51
2.4.2. Objetivos.....	51
2.4.3. Características.....	52
2.4.4. Ventajas Y Desventajas De XP.	54
2.4.5. Comparativa Con Otras Metodologías.	55

CAPÍTULO 3: HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO DEL SISTEMA.

3.1. LENGUAJE DE PROGRAMACIÓN DEL LADO DEL SERVIDOR.	56
3.1.1. Características Principales.....	56
3.1.2. Ventajas y Desventajas de la Programación del Lado del Servidor.....	58
3.1.3. Curva de Aprendizaje de Lenguaje del Lado del Servidor.....	59
3.2. FRAMEWORK USADO PARA EL DESARROLLO.....	60
3.2.1. Características.....	60
3.2.2. Curva de Aprendizaje de Yii Framework.....	63
3.2.3. Ventajas del Framework.....	64
3.2.4. Comparativa con otros Framework.	65
3.3. LENGUAJE DE PROGRAMACIÓN DE LADO DEL CLIENTE.....	66
3.3.1. Características de programación de Lado del Cliente.	66
3.3.2. Ventajas y Desventajas del Lenguaje del Lado del Servidor.	68
3.3.3. Curva de Aprendizaje del Lenguaje de Lado del Cliente.....	69
3.4. SERVIDOR DE PLATAFORMA XAMPP.....	70
3.4.1. Características.....	70
3.4.2. Ventajas y Desventajas.....	70
3.4.3. Plataforma de gestión de bases de datos MySQL.....	71
3.4.3.1. Introducción.....	71
3.4.3.2. Características de MySQL.....	71

CAPÍTULO 4. CASO DE ESTUDIO

4.1. FASE DE PLANIFICACIÓN DEL PROYECTO.	72
4.1.1. Toma Y Análisis de Requerimientos Utilizando la Metodología Áncora.	72
4.1.2. Elaboración de Reporte de Requerimientos.	73
4.1.3. Documento de Factibilidad.	76
4.2. FASE DE DISEÑO.	77
4.2.1. Diagramas UML.	77
4.2.2. Diagramas de Caso de Uso.	78
4.2.2.1. Nivel de Abstracción Gestión de Clientes.	79
4.2.2.2. Nivel de Abstracción Gestión de Productos.	81
4.2.2.3. Nivel de Abstracción Gestión de Facturas.	84
4.2.2.4. Puntos a tomar en cuenta.	88
4.2.3. Diagrama de Secuencia.	89
4.2.3.1. Gestión de Clientes.	89
4.2.3.2. Gestión de Productos.	92
4.2.3.3. Gestión de Facturas.	95
4.2.4. Diagrama de Clases.	98
4.2.5. Diagrama de Base de Datos.	99
4.2.6. Diccionario de Datos.	100
4.2.6.1. Lista de Tablas.	100
4.2.6.2. Lista de Columnas.	101
4.2.6.3. Lista de Referencias.	103
4.3. FASE DE CODIFICACIÓN.	104
4.3.1. Integración de Sistema.	105
4.3.2. Desarrollo de Funcionalidades del Sistema.	105
4.3.2.1. Integración de la Funcionalidad Gestión de Productos.	105
4.3.2.2. Integración de la Funcionalidad Gestión de Servicios.	107
4.3.2.3. Integración de la Funcionalidad Gestión de Facturas.	108
4.4. Validación de la Versión del Sistema.	109
4.5. Desarrollo de la Interfaz de Usuario.	110
Herramientas Utilizadas.	110
Pasos a seguir para el diseño de una interfaz de usuario.	110
Diseño de la Interacción.	110
Personajes y Reparto.	111
Objetivos.	111
Escenarios.	111
4.6. Especificar el diseño de un sitio web.	112
Usabilidad.	112
Prototipos.	112
4.7. FASE DE PRUEBAS E IMPLANTACIÓN.	116
4.7.1. Pruebas Unitarias.	116
4.7.2. Pruebas del Sistema.	125
4.7.3. Pruebas de Aceptación.	127

4.7.4. Implantación del Sistema.....	127
4.8. DOCUMENTACIÓN.....	128
4.8.1. Manual Técnico.....	128
4.8.2. Manual de Usuario.....	128

CAPÍTULO 5. CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES:	129
5.2. RECOMENDACIONES:	130
5.3. BIBLIOGRAFÍA	131
5.4. ANEXOS.....	; ERROR! MARCADOR NO DEFINIDO.
5.5. GLOSARIO DE TÉRMINOS.....	132

ÍNDICE DE CUADROS.

Cuadro 1. Estructura Organizativa del Taller Mecánico FRENICENTRO.....	16
Cuadro 2: Cuadro Comparativo entre Áncora y DoRCU.....	49
Cuadro 3: Valores del Manifiesto Ágil.....	52
Cuadro 4: Ventajas y Desventajas de XP Extreme Programming.	54
Cuadro 5: Diferencias entre XP Extreme Programming y SCRUM.	55
Cuadro 6: Ventajas y Desventajas de la Programación del Lado del Servidor.	58
Cuadro 7: Comparación entre Yii Framework y Zend.....	65
Cuadro 8: Ventajas y Desventajas de Lenguajes de lado del Cliente.....	68
Cuadro 9: Ventajas y Desventajas de XAMP.....	70

ÍNDICE DE TABLAS.

Tabla 1: Contabilidad de Puntos de la Función.....	29
Tabla 2: Asignación de nivel de dificultad para los Archivos Lógicos Internos o Archivos de Interfaz Externos.	29
Tabla 3: Asignación de nivel de dificultad para las Salidas y Consultas Externas.	30
Tabla 4: Asignación de nivel de dificultad para las Salidas y Consultas Externas.	30
Tabla 5: Asignación de Grados en la Comunicación de Datos.	30
Tabla 6: Asignación de Grados de Procesamiento de Datos.	31
Tabla 7: Asignación de Grados para el Rendimiento.....	32
Tabla 8: Asignación de Grados para la Configuración Altamente Usada.....	33
Tabla 9: Asignación de Grados Promedio de Transacciones.....	34
Tabla 10: Asignación de Grados de Entrada de Datos en Línea.....	35
Tabla 11: Asignación de Grados de Actualización en Línea.....	37
Tabla 12: Asignación de Grados de Procesamiento Complejo.....	37
Tabla 13: Asignación de Grados para la Reusabilidad.....	38
Tabla 14: Asignación de Grados de Facilidad de Instalación.....	39
Tabla 15: Asignación de Grados de Facilidad de Operación.....	40

Tabla 16: Asignación de Grados para Varios Sitios.....	41
Tabla 17: Asignación de Grados de Facilidad de Cambios.....	42
Tabla 18: Etapas y puntos a tratar en la Reunión.	43
Tabla 19: Roles de XP Extreme Programming.	53
Tabla 20: Roles y Actividades de Áncora.	72
Tabla 21: Fase y Artefactos de Áncora.	73
Tabla 22: Guión de Gestión de Venta Actual.....	73
Tabla 23: Guion de Gestión Actual de Control de Inventarios.	74
Tabla 24: Funcionalidades y Actividades a Desarrollar.....	76
Tabla 25: Funcionalidades y Prioridades.....	76
Tabla 26: Tabla de Estados del Sistema.....	88
Tabla 27: Lista de tablas de Base de Datos.	100
Tabla 28: Lista de Datos de la Base de Datos.	102
Tabla 29: Lista de relaciones de Base de Datos.	103
Tabla 30: Descripción de Requerimientos realizados.	125
Tabla 31: Lista de Verificación de requerimientos desarrollados.	127

ÍNDICE DE GRÁFICOS.

GRÁFICO 1: Elementos de la Metodología Áncora.....	¡Error! Marcador no definido.
GRÁFICO 2: Estructura y Elementos de un Guión.	24
GRÁFICO 3: Ciclo de Vida Áncora.....	46
GRÁFICO 4: Arquitectura Lenguaje de Lado del Servidor.....	57
GRÁFICO 5: Ejecución de Peticiones al Servidor.....	58
GRÁFICO 6 : Estadísticas sobre los Lenguajes de lado del Servidor más usados hasta el 2015.	59
GRÁFICO 7: Estadísticas sobre Frameworks PHP más utilizados en la última década.	62
GRÁFICO 8: Estadísticas sobre Frameworks PHP más utilizados.....	64
GRÁFICO 9: Estructura de Página Web en HTML.....	67
GRÁFICO 10: Funcionamiento de un Browser.	67
GRÁFICO 11: Pantalla de Ingreso del Prototipo.	113
GRÁFICO 12: Menú desplegable del Prototipo 1.....	113
GRÁFICO 13: Pantalla de Ingreso del Prototipo 2.	114
GRÁFICO 14: Barra de menú superior del Prototipo 2.	114
GRÁFICO 15: Resultado de Bugs y Vulnerabilidades en el Modelo.	116
GRÁFICO 16: Resultados a detalle de la capa de Modelo.	117
GRÁFICO 17: Modelos probados por Sonar Qube.....	118
GRÁFICO 18: Modelos probados por Sonar Qube.....	119
GRÁFICO 19: Archivos contenidos dentro de la Capa Vista.	120
GRÁFICO 20: Resultados a Detalle de la Capa Vista.	121
GRÁFICO 21: Resultados de la prueba en la Capa Controlador.	122
GRÁFICO 22: Archivos contenidos dentro de la Capa Controlador.	123
GRÁFICO 23: Resultados a Detalle de la Capa Controlador.	124

ÍNDICE DE DIAGRAMAS.

Diagrama 1: Diagrama UML de Requerimientos.	77
Diagrama 2: Caso de Uso.	78
Diagrama 3: Diagrama de Secuencia para Editar un Cliente.	90
Diagrama 4: Diagrama de Secuencia para Eliminar un Cliente.	91
Diagrama 5: Diagrama de Secuencia para el Ingreso de Productos.	92
Diagrama 6: Diagrama de Secuencia para Modificar datos del Producto.	93
Diagrama 7: Diagrama de Secuencia para Eliminar Productos.	94
Diagrama 8: Diagrama de Secuencia para el Ingreso de Factura.	95
Diagrama 9: Diagrama de Secuencia para Modificar datos de la Factura.	96
Diagrama 10: Diagrama de Secuencia para Eliminar Facturas.	97
Diagrama 11: Diagrama de Clases.	98
Diagrama 12: Diagrama de Base de Datos.	99

RESUMEN

La presente Tesis consiste en analizar el proceso comercial de la Mecánica Automotriz Frenicentro, centrándose en el proceso de Facturación y el Control de Inventarios. El objetivo principal es realizar un control de los productos en bodega, aumentando la eficiencia al prestar servicios a los clientes mejorando la atención y entrega de los vehículos. El sistema que se va a desarrollar tiene como objetivo: obtener la información de los inventarios en tiempo real, lo que permitirá mejorar la satisfacción de los clientes y aumentara la competitividad en el mercado. Para conseguir el objetivo final se realizará un proceso de ingeniería siguiendo metodologías de desarrollo y análisis de requerimientos.

Es importante señalar que ambas metodologías tienen características similares y las mismas tuvieron que ser adaptadas, de acuerdo a las limitaciones que se tuvieron durante el desarrollo del sistema. La presente tesis consta de 5 capítulos, en los cuales se fueron analizando cada uno de los componentes del sistema; a continuación se da una breve explicación de cada uno.

En el capítulo uno: se presenta una reseña de la información de la empresa como: misión, visión, objetivos empresariales, estrategia comercial y dentro de los procesos actuales se encuentran: Análisis de la gestión de facturas y control de los inventarios. El capítulo dos: es el marco teórico de la tesis y consta de una explicación de la metodología Ancora utilizada para el análisis de Requerimientos, un análisis y explicación sobre la Metodología de Desarrollo Extreme Programing. El capítulo tres: muestra una reseña de la programación del lado del Cliente y la programación del lado del Servidor. También consta una explicación de las herramientas que se van a utilizar entre las cuales se encuentra el servidor de Aplicaciones, lenguaje de programación con su respectivo análisis, framework utilizado y gestor de base de datos. El capítulo cuatro: es la parte de la aplicación de metodologías y herramientas, este capítulo se centra en las fases de: análisis con la aplicación de la metodología ÁNCORA¹, diseño realizando los modelos UML², secuencia, clases, base de datos y diccionario de Datos, codificación se explica cómo se realizó la integración del sistema y el desarrollo de cada una de las funcionalidades, pruebas se explica y se observa los inconvenientes que tuvo el sistema durante su desarrollo y finalmente la documentación que son los manuales de usuario y técnico. En el capítulo cinco: se detallan las recomendaciones y conclusiones del trabajo de disertación.

¹ Áncora: Metodología para el Análisis de Requerimientos

² UML: Lenguaje unificado de Modelado.

CAPÍTULO 1: ANÁLISIS DE PROCESOS ACTUALES

1.1. Situación Actual de la Empresa.

1.1.1. Descripción de la Empresa.

FRENICENTRO forma parte de un grupo de empresas con muchos años de experiencia en el ramo, cuenta con una amplia red de talleres en Venezuela y más de 40 años de experiencia al servicio de sus clientes. Ecuador fue tomado como país piloto y a partir del 8 de diciembre inicia sus operaciones en la ciudad de Quito, en la calle Las Palmeras y Nogales, con una infraestructura de más de 650 metros cuadrados, para brindar una atención de calidad y calidez.

Entre los principales servicios que la empresa ofrece a sus clientes están:

- Reparación general del sistema de frenos
- Cambio de aceite
- Limpieza de inyectores por ultrasonido
- ABC de motor
- Filtros
- Cambio de amortiguadores
- Llantas
- Alineación
- Balanceo
- Baterías
- Mecánica ligera

1.1.2. Misión Empresarial:

Somos un equipo integrado, formado y constantemente capacitado, para brindar un servicio de prevención, corrección y reparación de automóviles y camiones pequeños, donde nuestros clientes, obtienen y sienten seguridad, confianza y tranquilidad por los trabajos realizados y el confort de sus vehículos.

1.1.3. Visión

Ser reconocidos en el mercado de servicios como profesionales responsables y confiables tanto en los aspectos de prevención, mantenimiento y reparación de automotores, como también, consecuentes con nuestros empleados y el medio ambiente

Se espera obtener en un corto plazo el 1% del mercado de mantenimiento automotriz en la ciudad de Quito.

1.1.4. Objetivos Empresariales.

La empresa tiene planeado cumplir los siguientes objetivos, en un período de tiempo entre los tres a cinco años.

- Obtener el 1% del mercado de mantenimiento de vehículos particulares y pesados pequeños menores a 3 toneladas en la ciudad de Quito.
- Ampliar la empresa mediante sucursales en Cumbayá y la ciudad de Guayaquil.
- Ser reconocida por brindar servicios de alta calidad y confiabilidad.

1.1.5. Estrategia Comercial.

FRENICENTRO se especializa en brindar servicios como: revisión, prevención, mantenimiento y reparación de automotores. Además de ofertar un amplio surtido de repuestos originales de marcas reconocidas a nivel mundial; cuenta con excelentes y modernas instalaciones, equipos de alta tecnología, que aseguran la calidad del servicio y la satisfacción de sus clientes.

Sus directivos y equipo de trabajo buscan formar alianzas estratégicas y corporativas con empresas prestigiosas, a fin de ofrecer a los ejecutivos y trabajadores sus servicios y beneficios como:

- Precios al alcance del presupuesto.
- Pedido de turnos vía telefónica.
- Permanente información del estado de su vehículo.
- Garantía de los trabajos efectuados.
- Trabajos supervisados y atendidos por los dueños.

1.1.6. Proceso de Venta.

FRENICENTRO, en su actividad comercial, realiza el siguiente proceso para ofertar sus diferentes servicios el mismo que se describe a continuación:

- Registro de Datos del Cliente: se toma datos del cliente como: nombre, cédula o RUC., número de teléfono, dirección domiciliaria y electrónica, modelo del auto y placa.
- Elaboración de Proforma: es un documento que maneja la empresa para dar a conocer los servicios y productos a ofertar.
- Estimación de precio: se lo estima de acuerdo a los servicios requeridos por el cliente y el número de productos a ocupar.
- Estimación de tiempo: la duración de todo el proceso de trabajo hasta la entrega del vehículo al cliente.
- Delegar el servicio al personal competente: se delega este servicio al personal con menor cantidad de trabajo.
- Entregar el vehículo al cliente: se entrega del vehículo con los servicios solicitados por el cliente.
- Emisión de Factura: se detallan todos los servicios, productos o repuestos utilizados

para realizar los servicios solicitados por el cliente.

La empresa tiene ciertos inconvenientes en seguir cada uno de los pasos anteriormente mencionados, puesto que, al no tener información exacta sobre los productos del inventario, este proceso sufre de algunos problemas sobre todo en la estimación del tiempo y la entrega del vehículo.

La emisión de facturas también es un inconveniente ya que se la realiza manualmente, lo que ocasiona poca efectividad en los cálculos.

1.1.7. Problemas comunes en el Control de Inventarios.

La gran mayoría de PYMES y sus gerentes sufren de diferentes tipos de problemas al llevar el control de sus inventarios. A continuación se detallan los más comunes:

- **Falta de registros:** En la gran mayoría de PYMES, los gerentes o dueños de las mismas no invierten mucho tiempo en realizar un control a su inventario. Los gerentes dedican más tiempo a actividades operativas, que a labores de orden estratégico.

El principal objetivo de tener un registro de inventarios no solo lo realizan las empresas grandes o por solicitud de un contador, es para tener información clara y actual para mejorar el desempeño de la empresa y minimizar costos de producción, es muy útil para tener en un nivel óptimo el inventario en bodega, con el uso de tecnología se disminuyen los gastos y se aumenta la producción.

- **Exceso de inventarios:** este es un proceso en que los empresarios tiene la idea de tener en un alto nivel sus inventarios y así asegurar la venta en cierto sector del mercado, esto da como consecuencia disminución de la calidad de los productos o caducidad de los mismos.

El exceso de inventarios en la mayoría de empresas comerciales lleva al descontrol y a una disminución de liquidez, ya que incurren en deudas con los proveedores, la venta de los productos sirve para pagar estas cuentas.

- **Insuficiencia de inventarios:** si no posee productos en stock la empresa no solo pierde la venta sino también al cliente, el mismo que puede optar por consumir los productos de la competencia.
- **Baja calidad de la materia prima dada su caducidad:** en este problema se da cuando el cliente adquiere un producto y al abrirlo se da cuenta que dicho producto no cumple con las expectativas deseadas. Este problema se da especialmente en productos perecibles o que necesitan refrigeración.
- **Robo:** al no tener un conocimiento real sobre los productos del inventario, ni un buen control la empresa puede sufrir pérdidas de sus productos y económicas.
- **Desorden:** este provoca pérdidas graves para las empresas, ya que no poseen información sobre los productos en existencia dentro de bodega y suele caer en el error

de comprar en exceso, este problema se agrava si los productos tienen fecha de caducidad.

1.1.8. Políticas de Control

Con los problemas anteriormente mencionados, los empresarios han optado por utilizar cierto tipo de políticas o reglas para tener un mejor control de sus inventarios y del personal encargado de los mismos a continuación se describen algunas de las políticas más comunes:

- Control de ingreso a la bodega: Muchas empresas utilizan sistemas de seguridad, los mismos pueden ser por el uso de tarjetas magnéticas o por identificación, por lo general se designa a una o máximo dos personas para el ingreso a la bodega para llevar a cabo un control de inventarios físico.
- Muchas empresas designan cierto espacio físico dentro de la empresa para almacenar sus productos y materia prima, lo más adecuado para almacenar productos es una bodega o cuarto frío el mismo que debe poseer cerraduras.
- En caso de pérdida el costo del material o producto faltante se repartirá entre las personas designadas para el ingreso a la bodega.
- En el caso de ingreso o salida de materia prima o productos la misma debe constar en actas firmadas por el gerente y el personal designado al área de bodegas.

Estas políticas pueden ser muy básicas pero en la actualidad por diferentes motivos no se las aplica y las empresas sufren de pérdidas o desorden al llevar a cabo un control de inventario físico, en muchos de los casos existe un alto grado de confianza con los empleados, y el costo de adquisición de un sistema de seguridad suele ser caro.

1.1.9. Problemas encontrados dentro del Taller Mecánico “Frenicentro”.

FRENICENTRO al ser una empresa dedicada a la venta de productos y servicios relacionados con el mercado de la mecánica automotriz presenta diferentes inconvenientes le momento de llevar a cabo un control respecto de sus inventarios los problemas más frecuentes son los que se detallan a continuación:

- Insuficiencia de inventarios:

La falta de conocimiento sobre el stock en bodega impide que la empresa oferte un servicio a un cliente, lo que ocurre es que al realizar una proforma sobre uno o varios de sus servicios, dicho servicio no se pueda entregar a tiempo o realizar lo que perjudica económicamente a la empresa.

Otro de los inconvenientes es que al no tener información actualizada sobre el estado de los productos en el inventario, esto ocasiona un perjuicio económico para la empresa, ya que los productos pueden estar caducados o perdidos.

- Falta de registros:

Al realizar un balance contable se debe comprobar que la información que se tiene de los inventarios sea coherente con la información que consta dentro de las facturas que se emiten a los clientes, como las facturas recibidas de los proveedores, este proceso da como resultado dos inconvenientes: el primero es la pérdida de tiempo, al tener toda la información en papeles el proceso es muy largo ya que se debe revisar producto por producto, lo que lleva varios días impidiendo la actividad que se realiza normalmente y el segundo es que existe la posibilidad de que el balance sea inexacto lo que ocasionaría que se realice nuevamente este proceso.

Analizando los problemas antes mencionados, la empresa ha optado por la adquisición de un sistema informático que le permita al gerente tener información exacta y actualizada de sus inventarios, esto le será de gran ayuda para realizar proyecciones sobre los servicios a ofertar y la compra de productos a los proveedores.

1.1.10. Problemas con el Proceso de Facturación.

FRENICENTRO para el proceso de emisión de facturas normalmente lo realiza basándose primeramente en el documento de proforma, el mismo que le permite saber que productos utilizo para realizar el servicio mecánico, en este proceso participa el vendedor y el mecánico encargado de realizar el servicio automotriz, la elaboración de la factura trae algunos inconvenientes durante y después de realizar el servicio. El principal inconveniente es que no se estima muy bien la cantidad de productos que se van a utilizar; por lo tanto dentro de la proforma el vendedor debe actualizar la información de forma manual, esto conlleva a que la factura tenga un precio elevado respecto de la proforma presentada y aceptada por el cliente.

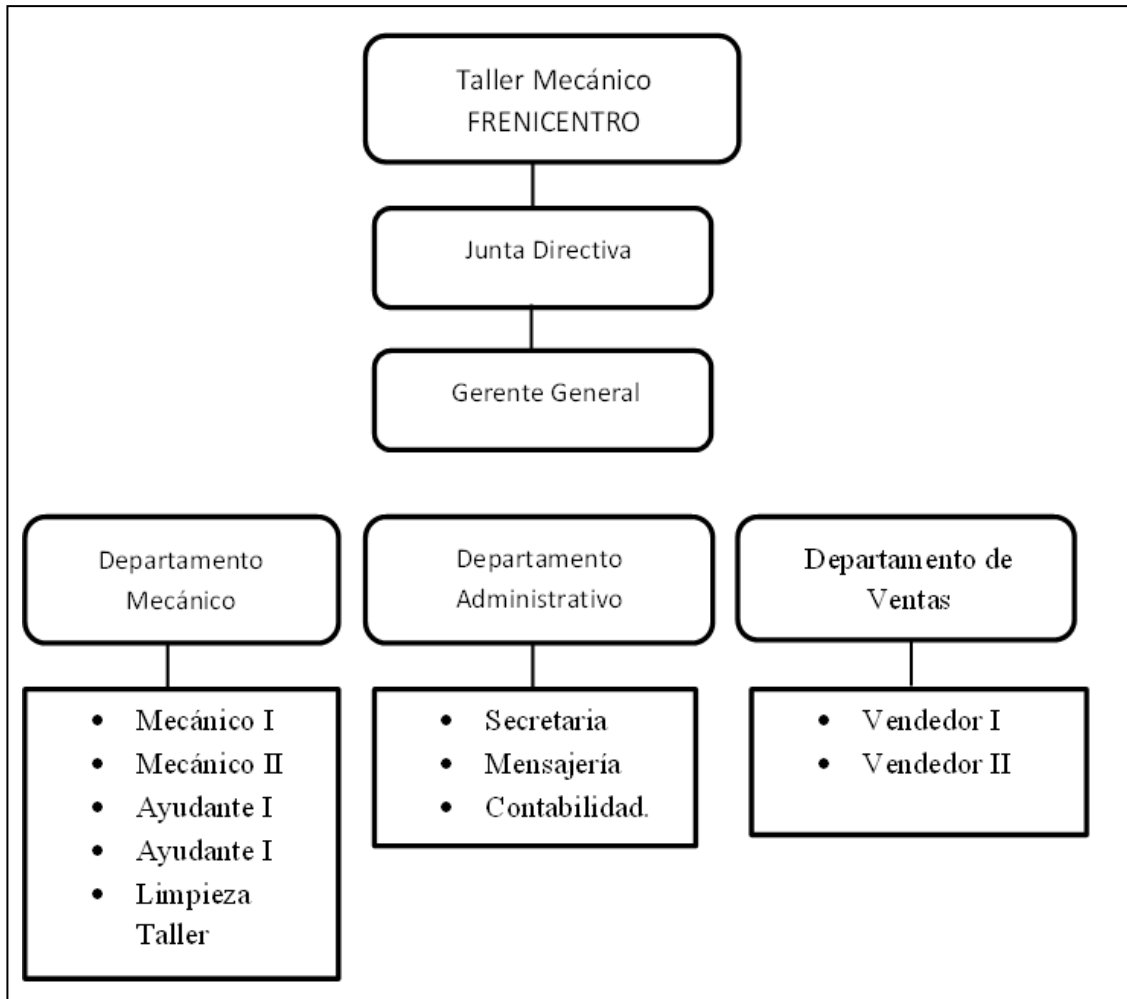
Otro de los inconvenientes que surgen de llevar este tipo de proceso de forma manual es que los balances no tenga un alto nivel de fiabilidad puesto que las facturas no coinciden con los inventarios y las proformas. Esto ocasiona que se deba analizar y realizar un cheque minucioso de cada tipo de documento, lo que ocasiona malestar a la gerencia y cierta desconfianza con los subalternos.

El proceso de facturación al estar ligado íntimamente con el control inventarios ocasiona que la empresa deba incurrir en invertir mucho tiempo en inventariar todos sus productos, recolectar, organizar y analizar cuidadosamente sus facturas para la elaboración de balances periódicos.

Además de los problemas anteriormente mencionados surge uno más que es la pérdida de competitividad en el mercado, puesto que la tecnología sigue avanzado y el próximo paso es la facturación electrónica.

En la actualidad muchas PYMES ha optado por realizar la emisión de la facturas de forma electrónica por varios motivos el primero conservar de cierta forma el medio ambiente ya que las mismas pueden ser guardadas en formato digital y la segunda es que este proceso optimiza el proceso de control de inventarios.

1.1.11. Estructura Organizativa



Cuadro 1. Estructura Organizativa del Taller Mecánico FRENICENTRO
(Frenicentro, 2015)

1.2. Procesos Actuales

1.2.1. Análisis de Gestión de Facturas.

FRENICENTRO, en la actualidad, emite facturas de forma manual, al llevar de esta manera su proceso de emisión de facturas la empresa tiene muchos inconvenientes, los mismos que se detallan a continuación:

- Pérdida de competitividad dentro del mercado: este punto es muy importante ya que en la actualidad la tecnología es utilizada para mejorar procesos como: el manejo contable de una empresa, esta actividad se la realizaba de forma manual lo que ocasionaba una gran inversión de tiempo, en la actualidad la tecnología y sistemas especializados permiten realizar dicha actividad de forma más rápida, efectiva y confiable, el uso de las TIC's y su relación con la contabilidad dentro de una empresa permite realizar varios procesos los mismo que se mencionan a continuación:
 - Elaborar balances y reportes con datos en tiempo real.
 - Tener reportes del estado de ventas.
 - Pago de impuestos por internet.
 - Envió facturas en formato electrónico.

Con los procesos que se mencionan anteriormente, la empresa podrá mejorar significativamente su competitividad dentro del mercado ofertando de mejor manera sus productos y servicios, mejorando así la atención a sus clientes.

- Demora en la elaboración de balances contables: al manejar manualmente la emisión de facturas es muy difícil elaborar un balance contable, existen varios inconvenientes que se presentan el momento de realizar el mismo entre los principales tenemos:
 - Perdida de facturas.
 - Información desactualizada.
 - Mal manejo en la emisión de Facturas.

Para la elaboración de un balance es necesario tener todas las cuentas en orden y contar con información actual, si este proceso se lo lleva de forma manual la empresa pierde tiempo valioso en ordenar y recolectar todas las facturas, esto sin tomar en cuenta que se debe realizar un control exhaustivo del inventario.

1.2.2. Análisis de Control de Inventarios.

FRENICENTRO, en la actualidad oferta a sus clientes productos y servicios mecánicos, los mismos que se encuentran dentro de su inventario, el principal problema de la empresa es el proceso que se llevan para controlar sus inventarios, actualmente la empresa lo lleva de forma manual lo que le impide tener información en tiempo real, esto ocasiona ciertos inconvenientes al ofertar sus productos y servicios.

El proceso de venta que actualmente realiza la empresa es el siguiente:

La empresa realiza un proceso de Proformas en el cual oferta sus productos y servicios a sus clientes, este proceso tiene ciertos inconvenientes como los que se mencionan a continuación:

- Desconocimiento sobre el stock de productos en su inventario: al llevar de forma manual el inventario de sus productos, la empresa no posee una información clara y actualizada sobre sus productos, lo que ocasiona cierta demora en el desarrollo del servicio automotriz y la posterior entrega del vehículo al cliente.
- Inseguridad el momento de ofertar algún servicio: al llenar una proforma el vendedor transmite cierta inseguridad al cliente respecto a la duración del servicio solicitado, ya que no posee información actualizada de los productos que se encuentran dentro del inventario para realizar dicho servicio.

La empresa, al manejar esta información de forma manual, ocasiona pérdida de competitividad dentro del mercado, ya que no puede satisfacer las necesidades del cliente respecto a retrasos en la entrega del vehículo.

Al analizar los problemas anteriormente mencionados, la empresa ha optado por el uso de herramientas informáticas para mejorar de cierta forma el manejo del inventario; pero se han encontrado con dos problemas como: el alto costo para adquirir un sistema contable y el manejo de la información en tiempo real.

Actualmente utilizan herramientas informáticas para registrar el ingreso y salida de productos; pero esta información debe ser actualizada manualmente, lo que ocasiona retraso en la actividad económica y esto, a su vez, no permite realizar una proyección en cuanto a sus ventas y una correcta administración sobre la adquisición de nuevos productos.

1.2.3. Uso de sistema para emisión de facturas a través de páginas web.

Según el Ministerio de Telecomunicaciones y de la Sociedad de la información, por medio de una publicación (2013) acerca del estudio de usabilidad y acceso a las Tecnologías de la Información y Comunicación (TIC), indica que: “en el Ecuador las micro, pequeñas y medianas empresas (MYPIMES) utilizan la red de Internet, ya sea para vender productos, servicios, realizar contactos mediante el correo electrónico o redes sociales” (citado: 12 de mayo 2016, recuperado de: <http://www.telecomunicaciones.gob.ec/el-823-de-mipymes-en-el-ecuador-utilizan-internet/>).

En esta publicación et al. (2013), destaca la proporción de empresas que utilizan Internet: las microempresas alcanzan un 48,6%, las medianas empresas un 56,9% y las pequeñas empresas un 52,9%, dando un total general de 52,8%.

Al analizar los datos de este estudio y los antecedentes anteriormente mencionados, Frenicentro, ha optado por automatizar sus procesos comerciales enfocándose principalmente en el control de inventarios y la emisión de facturas, optimizando el control de inventarios permitirá tener una mejora considerable el momento de atender y ofertar sus productos al cliente.

Por otro lado el proceso de facturación mejorará, ya que el sistema le permitirá almacenar de mejor manera las facturas evitando así los problemas que anteriormente se mencionaron, además se podrá enviarlas por mail a sus clientes de esta manera se contribuye con el medio ambiente ya que no será necesario imprimirlas. Otra de las ventajas de tener estos dos procesos automatizados es la realización de reportes en determinado periodo de tiempo estos podrán ser: estado del inventario lo que le permitirá hacer proyecciones sobre lo que necesita comprar a los proveedores y el número de productos vendidos que permitirá saber la demanda que tiene cada producto.

Cabe resaltar que la empresa realiza un proceso intermedio antes de emitir una factura este proceso es: realizar una proforma ofertando al cliente los diferentes productos y servicios que la empresa tiene a disposición, este proceso se lo hacía manualmente y al no tener un correcto control del inventario, daba como consecuencia retraso en la entrega del vehículo al cliente, este proceso también mejorara ya que la empresa tendrá información en tiempo real respecto a la disponibilidad de productos de su inventario.

Al tener almacenadas todas las facturas en formato digital se podrá realizar un balance contable con mayor fiabilidad y exactitud, lo que permitirá a la empresa tener una mejor visión de su actividad económica.

CAPÍTULO 2: METODOLOGÍAS APLICADAS PARA EL DESARROLLO DEL SISTEMA

2.1. Metodología para el análisis de requerimientos Áncora

2.1.1. Fundamentos de Áncora

Para realizar el diseño, desarrollo e implantación de presente sistema, se deberá tomar en cuenta varias metodologías; en este caso específico, se centrará en el estudio, análisis y posteriormente la aplicación de una metodología para la toma de requerimientos. En consecuencia, se ha decidido tomar Áncora como la metodología.

Conviene, entonces, definir qué es Áncora y cuál es el uso que se puede dar en el desarrollo de un nuevo software.

Según María de los Ángeles Sumano López, en su publicación: “Áncora: Metodología para el Análisis de Requerimientos de Software conducente al Reuso” (2001:7), define:

Áncora es el sinónimo de Ancla, término utilizado en el contexto del análisis de requerimientos, dentro del desarrollo de software, proporciona la idea de que para empezar a trabajar con la elaboración de un nuevo software es necesario tener buenas y sólidas bases para definir correctamente lo que se quiere, como resultado de un software nuevo y saberlo exponer claramente al usuario y que, además, permita pasar a las siguientes etapas de desarrollo de software de forma sencilla y natural.

Esta metodología maneja dos elementos, los mismos que permiten tener un amplio conocimiento acerca de lo que el sistema pretende cumplir para la satisfacción del usuario, a continuación se detallan cada uno de ellos:

El primer elemento de esta metodología es denominado “Guión”, de esta forma se puede hacer cierta semejanza entre una obra de teatro y el sistema a desarrollar; por ejemplo: el usuario será el actor principal de la obra, el mismo que podrá representar uno o más papeles dentro de la obra y finalmente la obra será escrita por los involucrados con la guía de un analista; esta forma de trabajar permite asegurar que el usuario esté muy involucrado en el desarrollo del sistema y que en el futuro, el usarlo no se ha complicado.

El segundo elemento de esta metodología es el Reuso, Áncora nos permite almacenar elementos de sistemas anteriores, esto es muy bueno, ya que se puede utilizar elementos que realizan actividades semejantes a las que se requieren en el nuevo, lo que permite ganar tiempo; ya que el nuevo sistema no empezaría desde cero. Así, literalmente, Sumano et.al.,a(2001) menciona que; “la variedad de elementos que se puede reutilizar están entre un conjunto de actividades, costo y prototipos de sistemas anteriores”.

2.1.2. Avances de Áncora

“Áncora va a apoyar al Analista, o más conocido como Ingeniero de Requerimientos, en la etapa de Análisis de Requerimientos de Software” (Sumano et.al., 2001:8). Para definir los requerimientos para el desarrollo de un nuevo software el Analista y el usuario en este caso el gerente de la empresa debe haber avanzado en las negociaciones y tener el apoyo y predisposición para la “Especificación de Requerimientos”, ya sea en forma escrita o verbal. Esto quiere decir que ya se efectuó un estudio de factibilidad y quedan pendientes de cubrir las siguientes etapas:

- Análisis de Requerimientos de Software (ARS): en esta etapa, se desarrollará la propuesta sostenida por Sumano: “...se dividen los requerimientos de software, observando sistemas existentes, así: discusión con usuarios y proveedores de información potenciales, análisis de tareas, y así sucesivamente” (Sumano, et.al.:8). Al realizar este proceso cíclico se puede tener el desarrollo de uno o varios modelos del sistema, los mismos que ayudan al analista a especificar de mejor manera el sistema.
- Definición de Requerimientos de Software: en esta etapa se recolecta y analiza toda la información obtenida en la etapa de ARS y se la transcribe a un documento, en el que se define un conjunto de requerimientos. Este documento reflejará, con claridad, lo que el cliente o usuario final desea.
Al ser un documento para el usuario final, este se debe tomar en cuenta, ya que no se podrá usar un lenguaje muy técnico y el documento deberá ser escrito de forma que sea entendible.
- Especificación de Requerimientos de Software: en esta etapa se realiza la especificación de requerimientos de software, mediante un documento detallado y la elaboración de un diseño de alto nivel, el mismo que facilita la corrección de errores el momento de definir los requerimientos, los mismos que deben ser corregidos.

La tarea principal de un analista de Software es descubrir y describir las características de un nuevo software, al terminar dicha tarea, el analista brindará la información de los elementos que el software necesita al diseñador, para que pueda especificar una arquitectura para su desarrollo.

Aunque resulte irreal el pensar que un análisis esté completo, siempre se debe considerar que los requerimientos pueden aumentar; pero se debe tener por lo menos un avance del proyecto y sus metas, esto puede facilitar el trabajo del diseñador.

El momento de entregar el avance, este debe ser claro y sin ambigüedades, además debe permitir añadir o cambiar elementos dentro del diseño. Es muy probable que el ARS deba ser negociado o discutido varias veces, lo que origina el agregar nuevas características al diseño.

2.1.3. Características Principales.

Como se menciona en los enunciados anteriores, Áncora posee varios elementos que le permiten ser una buena metodología para la toma y análisis de requerimientos. A continuación, en el Gráfico 1: “Elementos de la Metodología Áncora”, diseñado por María de los Ángeles Sumano López (et.al.:9) se especifican cada uno de los elementos, para lograr un mejor conocimiento y saber cómo aplicar cada uno de ellos y como interactúan en la metodología.

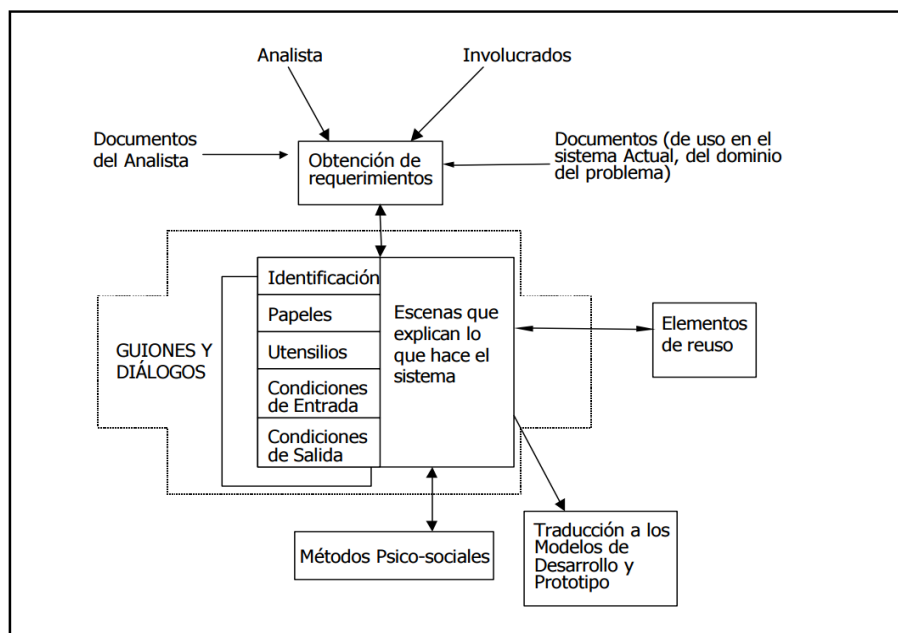


GRÁFICO 1: Elementos de la Metodología Áncora

Además, cabe indicar, textualmente, las herramientas metodológicas utilizadas por Sumano (et.al.:10): “Guiones y Diálogos, Tablas y Glosarios, Modelo de Datos Entidad-Relación, Análisis de Punto de Función”.

También, Áncora utiliza varias herramientas de Ciencias Computacionales y métodos psicológicos y de planeación como:

- Técnica de Grupo Nominal.
- Reunión de Reflexión y Diseño.

A continuación, se explicará, de forma más detallada, cada uno de los elementos y técnicas de la metodología.

2.1.3.1. Guiones y Diálogos.

Estos instrumentos son trascendentes para la metodología Áncora; así por ejemplo: el guión es indispensable y el diálogo es utilizado para aclarar los “sobrentendidos” que existen dentro de la funcionalidad necesaria para el nuevo software.

2.1.3.1.1. Guiones.

El guión, al ser un elemento muy importante dentro de la metodología, es importante saber cómo utilizarlo. Con los Guiones la forma obtener los requerimientos por parte del cliente será más clara y sistemática, ya que en la misma se tratará de evitar los tecnicismos y ambigüedades de interpretación del lenguaje natural. Cada guión posee una estructura que permite tener mejor manejo por parte del analista; estos elementos se describen a continuación:

- **Guión:** es el nombre del sistema. Cada guión tiene el límite de una hoja; sin embargo, si se necesita más de una, el guión se divide en pistas; por tanto cada hoja contiene una pista, con el nombre del guión incluido.
- **Pista:** es el nombre de un subsistema; en caso de que exista más de un subsistema; será escrito en un nuevo guión que, de hecho, tendrá el nombre del sistema.
- **Papeles:** es una lista de roles dentro del sistema, cada papel puede ser una persona, sistema o equipo, esto es muy útil para planificar las diferentes funciones dentro del equipo, ya que una persona, sistema o equipo puede tener uno o más roles, que deben estar en constante interacción con el sistema..
- **Utensilios:** se refiere a los dispositivos físicos del sistema, los mismos que pueden ser: dispositivos de entrada/salida, formas del sistema o archivos; se debe considerar a la computadora como un dispositivo siempre y cuando el sistema es del tipo cliente-servidor, si es del tipo distribuido cada computador se considera como un papel o rol dentro del sistema.
- **Condiciones de entrada:** se detallan en una lista todos los eventos o factores que se deben cumplir para empezar a trabajar en el sistema.
- **Condiciones de Salida:** se describen una lista de todos los eventos que conllevan a la finalización del sistema.
- **Escena:** Es el nombre de una función genérica, se puede pensar como un módulo dentro del sistema. Cada escena estará detallada dentro de una quinteta, si es muy larga es recomendable cambiarla a pista o subsistema y se realice un guión separado. Por lo general, cada escena corresponde a un requerimiento.
- **Quinteta:** como su nombre lo indica estará formado por cinco elementos, aunque en muchos de los casos se utilizan solo los tres primeros de la siguiente lista:

- a) Papel: es el código con que se identifica a uno de los papeles dentro del guión.
 - b) Acción: es un verbo que describe determinada actividad. Resultado es la acción del verbo, se lo puede considerar como: un archivo, informe o consulta, el mismo que debe constar en una lista de útiles.
 - c) Útil: es un elemento opcional dentro de la quinteta, si se escribe este elemento debe constar dentro de una lista de utilería del guión esto sirve para realizar una acción.
 - d) Periodicidad. Período de tiempo transcurrido entre cada una de las actividades descritas por la quinteta.
- Elementos gráficos:
 - 1) La caja que rodea y separa elementos.
 - 2) Las líneas o segmentos que marcan las secuencias de las acciones representadas en las quintetas.
 - 3) El segmento o línea continua, que indica secuencia, que puede omitirse a falta de saltos cercanos y finalmente, el segmento de línea punteada.

En el Gráfico 2: “Estructura y Elementos de un Guión” (Sumano, et.al.:11) puede observarse cada uno de los elementos anteriormente detallados:

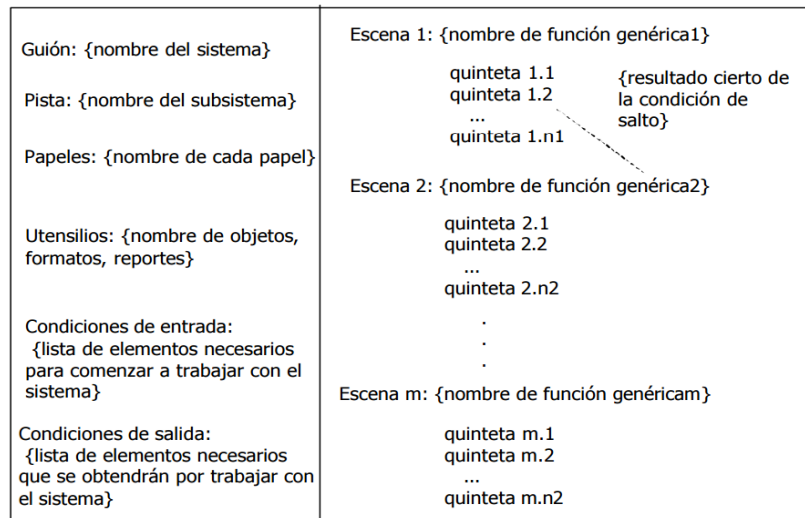


GRÁFICO 2: Estructura y Elementos de un Guión.

2.1.3.1.2. Diálogos

Son instrumentos o herramientas que la metodología Áncora utiliza para detallar o definir cada quinteta ambigua o donde se presenten actividades manuales sobreentendidas. Al respecto, Sumano, menciona literalmente: “En el caso de que existan quintetas ambiguas cada diálogo estará formado por renglones donde se expresa el papel de la quinteta con verbos que aclaran los sobreentendidos de la quinteta (et.al.:13).

2.1.3.2. Tablas y Glosarios

El uso de Tablas y Glosarios tiene como objetivo: reducir la ambigüedad que se tiene al utilizar lenguaje natural, esto ocurre generalmente por el uso de términos del área de aplicación y área de computación. Esto facilitará el momento del diseño y del modelo de datos.

2.1.3.2.1. Tablas

Es una estructura formada por filas y columnas, las mismas que permiten tener una localización de los contenidos de los elementos que la conforman. Al respecto, Sumano (et.al.:15) menciona los siguientes fines:

1. Definir cada formato a utilizar. Se entiende por formato estos ejemplos: formas impresas, oficios, memorándums, faxes y hasta mensajes vía Internet.
2. Especificar cada dato: nombre, tamaño, tipo.
3. Realizar un plan de pruebas de cada funcionalidad.
4. Cálculo de costos.

El formato de la tabla se irá definiendo conforme se vaya utilizando la tabla.

2.1.3.2.2. Glosarios

El objetivo principal es contar con un lenguaje común para todos los implicados en el nuevo sistema. Para ello, es necesario definir una lista completa con términos técnicos a utilizarse en el nuevo software e ingeniería de software, que es primordial para el mejor entendimiento del interesado con el analista.

En el contenido se debe especificar los siguientes datos: 1. Palabra, 2. Significado de la palabra en el dominio de la aplicación o de Ingeniería de Software. 3. Formas donde aparece. Se refiere a los formatos que se utilizan en la aplicación, incluyendo pantallas. 4. Lista de posibles valores. 5. Referencia (manual, libro, persona) de donde obtuvo el significado. (Sumano, et.al.:15).

2.1.4. Modelo Entidad – Relación

Áncora como herramienta adicional utiliza el “modelo entidad-relación” para el diseño del sistema, este modelo fue presentado por P.Chen (1976), el mismo que ha sufrido modificaciones y ha sido ampliado por varios autores y se lo ha utilizado para diseñar modelos semánticos de Bases de Datos, como también para el modelado de datos. Existen métodos para convertir este modelo en cualquier modelo lógico de Bases de Datos por ejemplo: relacional o jerárquico. Este modelo no ha perdido vigencia y sigue utilizándose para modelar bases de datos orientadas a objetos. Los objetos que conforman este modelo se los detallan a continuación:

- Entidad: se representa por medio del uso de sustantivos, gráficamente dentro de un rectángulo, esta puede representar a una persona, lugar, cosa o evento de interés en el sistema. Existen dos tipos de entidades que son: entidades débiles que gráficamente se

las dibuja con doble línea en rectángulo, además no pueden existir por sí solas y depende de una entidad fuerte.

- **Atributos:** son utilizados para detallar las características de un objeto o entidad. Gráficamente se representan por ovales que son unidos por medio de una línea al rectángulo de la entidad. Existen dos tipos de atributos los descriptivos son características de la entidad y los identificadores que son más conocidos como llave primaria; por lo general los poseen las entidades fuertes y permite relacionar varias entidades.
- **Relación:** gráficamente está identificado por un diamante, que contiene el nombre de la agrupación de dos o más entidades, cuya relación posee o no atributos.

A continuación se presenta un ejemplo gráfico de la estructura de un modelo Entidad-Relación que se explicó inicialmente: (Sumano, et.al.:16)

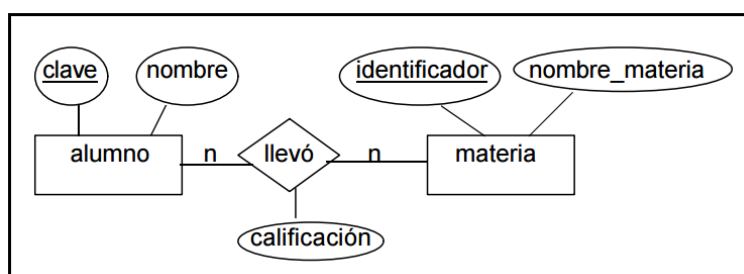


GRÁFICO 3: Estructura y Elementos de un Guión.

2.1.5. Modelo de Objetos Semánticos (MOS).

Este modelo es utilizado para entender de mejor manera los requerimientos del usuario y diseñar el modelo de datos, el mismo que se convertirá en la base de datos del Sistema. La principal característica de este modelo es que ayuda a diseñar el modelo de BD Orientada a Objetos, ya que de utilizar solo el modelo de Entidad-Relación, da como resultado un diseño de BD, en función de las entidades y sus relaciones.

2.1.6. Definición y Características de los Objetos Semánticos.

Según Sumano (et.al.:17) define los objetos semánticos como:

Un objeto semántico constituye una identidad en el medio ambiente del usuario, cada objeto debe poseer suficientes propiedades que lo identifiquen claramente.

Los objetos semánticos pueden agruparse en clases. Cada clase tiene su nombre que la distingue de otras clases y define la identidad a la que está representando. Estas clases deben estar escritas con mayúsculas.

2.1.6.1. Tipos de Objetos Semánticos.

Existen varios tipos de Objetos Semánticos los mismos que se describen a continuación:

- 1) Objetos simples. Tienen una única propiedad y el valor es único.
- 2) Objetos compuestos. Compuesto de varias propiedades simples.
- 3) Objetos combinados. Es combinación entre propiedades con valores simples y propiedades de objetos.
- 4) Objeto de asociación. Es cuando un objeto relaciona a uno o más objetos que poseen valores peculiares para dicha relación.
- 5) Objetos híbridos: es un objeto que contiene propiedades de varios objetos simples, compuesto o combinados.
- 6) Objetos de generalización y subtipos. Se utilizan para dar una estructura jerárquica a cada tipo de objetos.

2.1.6.2. Modelando con Objetos Semánticos.

Para modelar la BD Orientada a Objetos (BDOO) se puede utilizar dos métodos, los mismos que se explican a continuación:

- Top-down: en este método los usuarios y el analista trabajan conjuntamente, la función del analista es preguntar a los usuarios cuales pueden ser los objetos, de esta manera el analista se concentra en la meta general.
- Bottom-up se centra en definir las interfaces de la aplicación, entonces se analiza y se define cuáles son las propiedades que hay que tomar en cuenta, en este método se define los objetos semánticos.

En ambos casos se tiene diferentes problemas en el método top-down es necesario que el analista tenga un amplio conocimiento en el diseño de sistemas de BDOO, mientras que al aplicar el método bottom- up se invierte mucho tiempo en la definición y creación de las interfaces. Al tener este tipo de problemas una buena salida es combinarlos y seguir una serie de pasos para que el sistema resulte efectivo, estos pasos son:

- 1) Se analiza la meta de la aplicación y se elabora una lista de objetos semánticos potenciales.
- 2) Se inspeccionan los reportes, formas y vistas para determinar un posible diseño de la estructura del conjunto de objetos y sus propiedades.
- 3) Se planifica un método para relacionar los objetos, esto se puede obtener si se analiza exhaustivamente las pantallas y reportes, que se espera alcanzar del nuevo software.
- 4) Se puede sugerir al usuario el uso de nuevos reportes, formas y vistas con la aplicación de diagramas de objetos..

2.2. Cálculo de Puntos de Función.

Este modelo fue propuesto por Albertch, es utilizado por varias instituciones, “The International Function Point Users Group” es un grupo que mantiene actualizada la forma de contabilizar los puntos de función. Entre las principales ventajas de este método están: la rapidez para aplicar el cálculo de costos de un nuevo software y ayuda a limitar lo que se desea del sistema y las restricciones bajo las que funcionará.

Por estas razones Áncora utiliza este método para la primera aproximación de los costos. Más adelante se detallan la forma de contabilizar los puntos de función y como unirlos a los elementos de ARS para obtener un costo aproximado en tiempo, dinero o complejidad.

Para aplicar este método será necesario conocer y entender los siguientes indicadores:

- Archivos Lógicos Internos (ALI): es un grupo de datos lógicos, que pertenece al usuario. Su principal función es mantener datos actualizados, a través de procesos básicos de la aplicación a calcular.
- Archivo de Interfaz Externa (AIE): es un grupo de datos o de información correspondiente al usuario, cuya aplicación está dentro de los límites de otra aplicación. Su función es ayudar a contener los datos referidos mediante uno o más procesos básicos de la aplicación a evaluar.
- Entradas Externas (EE): el objetivo es almacenar uno o más ALI y/o alterar el funcionamiento del sistema.
- Salidas Externas (SE): es un proceso en el cual se envía datos o información de control fuera de la aplicación. El objetivo principal es mostrar información al usuario. Para esto se utilizan: procesamiento lógico, que debe contener, mínimo una fórmula, cálculo matemático, u otros relacionados. Además contiene una o más ALI y puede alterar el funcionamiento del sistema.
- Consultas Externas (CE): es proceso que envía datos o información fuera de los límites de la aplicación. Su objetivo es proporcionar información al usuario sobre la recuperación de información. A diferencia de SE no se requieren fórmulas, cálculos o datos derivados; además el ALI no se actualiza, ni modifica el funcionamiento del sistema.

Para contabilizar los puntos de función se debe seguir los pasos y colocar los valores como se indican en la siguiente tabla elaborada por Sumano (et.al.: 22) que se detalla a continuación:

Indicador	simple	mediano	complejo	SUMA
Archivos Lógicos Internos	7	10	15	
Archivos de Interfaz Externa	5	7	10	
Entradas Externas	3	4	6	
Salidas Externas	4	5	7	
Consultas Externas	3	4	6	
			T =	

Tabla 1: Contabilidad de Puntos de la Función

1. Se asigna un indicador a cada elemento de software.
2. Se determina un nivel de dificultad, como se explican en las tablas siguientes.
3. Se deben multiplicar por los pesos que contienen en la tabla 1.
4. La última columna contendrá los valores sumados de cada fila.
5. Se suman todos los valores de la última columna.

Para determinar los pesos que tiene cada indicador se deberá tomar en cuenta el número de iteraciones que tiene cada uno y sus características, a continuación se detallan la manera de asignar el nivel de dificultad de cada uno.

2.2.1. Asignación de pesos para ALI y AIE.

El primer paso es asignar el número al Tipo de Dato Elemental (TDE) y Tipo de Registro Elemental (TRE) que cada ALI o AIE contiene, para lo cual la autora, Sumano (et.al.:22) sugiere la aplicación de las siguientes reglas:

- 1) Para saber cuántos campos (TDE) tomar en cuenta para cada ALI o AIE.
 - a) Cuente un TDE para cada campo mantenido o recuperado de un ALI o AIE, a través de un proceso elemental..
 - b) Cuando dos aplicaciones mantienen o refieren el mismo ALI y cada diferente proceso utiliza los TDE, agrupados de diferente manera, deben contarse los TDE como se refiere. Ejemplo: una aplicación utiliza día, mes, año y otra lo usa como fecha; entonces hay 2 TDE diferentes.
 - c) Contar un TDE para cada pieza de dato requerido por el usuario para establecer una er una relación con otros ALI o AIE. Ej. Una llave externa.
- 2) Para el conteo de los TRE se aplican los siguientes criterios.
 - a) Cuente un TRE por cada subgrupo del ALI o AIE.
 - b) Si no hay subgrupos cuente al ALI o AIE como un TRE.

Una vez que se tienen contados los TDE y TRE se aplican la Tablas 2, 3 y 4, según las cuales, Sumano (et.al.: 23) determina la asignación del nivel de complejidad del ALI o AIE.

	1 -19 TDE	20 - 50 TDE	51 ó + TDE
0 – 1 TRE	simple	simple	mediano
2 – 5 TRE	simple	mediano	complejo
6 ó + TRE	mediano	complejo	complejo

Tabla 2: Asignación de nivel de dificultad para los Archivos Lógicos Internos o Archivos de Interfaz Externos.

Para el cálculo de pesos de EE, SE y CE se utilizan las siguientes tablas 3 y 4 en las cuales se hace referencia RA = Referencias de Archivos

	1-5 TDE	6-19 TDE	20 ó + TDE
0 – 1 RA	simple	simple	mediano
2 – 3 RA	simple	mediano	complejo
4 ó + RA	mediano	complejo	complejo

Tabla 3: Asignación de nivel de dificultad para las Salidas y Consultas Externas.

	1 - 4 TDE	5 -15 TDE	16 ó + TDE
0 – 1 RA	simple	simple	mediano
2 RA	Simple	mediano	complejo
3 ó + RA	Mediano	complejo	complejo

Tabla 4: Asignación de nivel de dificultad para las Salidas y Consultas Externas.

2.2.2. Ajuste de los Puntos de Función.

Al implantar un nuevo software el costo suele influir mucho para lo cual este modelo presenta catorce estimadores los mismos que se listan y detallan a continuación, a cada estimador se le asigna un grado de influencia que pueden ser 0=sin influencia, 1=accidental, 2=moderado, 3=medio, 4=significativo, 5=esencial.

✓ **Comunicación de datos.**

Para asignar los grados de asignación se utilizan las tablas Nos. 5, 6, 7, 8, 9 y 10, elaboradas por Sumano (et.al.: 24, 25, 26).

Es el grado en el que la aplicación y el procesador se comunican.

Grado	Descripción
0	La aplicación es puramente batch o para una PC aislada.
1	La aplicación es batch pero tiene entrada de datos o impresión remota.
2	La aplicación es batch pero tiene entrada de datos e impresión remota.
3	La aplicación incluye datos en línea, el procesamiento es por medio de un archivo .bat. Se realiza en un front-end de teleproceso (TP) o sistema de consulta.
4	La aplicación es más que un front-end, pero soporta un solo tipo de protocolo de comunicación de TP.
5	La aplicación es superior que un front-end, y soporta más de un tipo de Protocolo de comunicación de TP.

Tabla 5: Asignación de Grados en la Comunicación de Datos.

✓ Procesamiento Distribuido de Datos.

Es el grado en el que la aplicación transfiere datos entre sus componentes. Para asignar el grado de influencia se utiliza los datos de la siguiente tabla:

Grado	Descripción
0	La aplicación no apoya la transferencia de datos o procesado de funciones entre las componentes del sistema.
1	La aplicación prepara datos para que el usuario procese en otra componente del sistema como en una hoja de cálculo o un SMBD en PC.
2	Se preparan datos para transferir, entonces se transfieren y procesan sobre otra componente del sistema (no por el usuario final).
3	Los procesos o transferencia de datos se realizan en línea y en una sola dirección.
4	Los procesos o transferencia de datos se realizan en línea y en ambas direcciones.
5	Las funciones de procesamiento son realizadas dinámicamente sobre la mayor parte del sistema.

Tabla 6: Asignación de Grados de Procesamiento de Datos.

✓ Rendimiento.

Este grado será crítico y tendrá influencia sobre cómo se diseña, desarrolla o implementa. Como se asignarán los grados se muestran a continuación:

Grado	Descripción
0	Los requerimientos impuestos por el cliente no son especiales.
1	Se establecieron y revisaron los requerimientos de rendimiento y diseño, pero no se requirieron acciones especiales.
2	El tiempo de respuesta es crítico durante las horas pico.
3	El tiempo de respuesta crítica durante las horas de negocios. No se ha diseñado especialmente para un modelo de CPU.
4	También, si los requerimientos de rendimiento son lo suficientemente estrictos, se va a requerir realizar un análisis de rendimiento de las tareas en la fase de diseño.
5	También, se utilizarán herramientas de análisis de rendimiento en las fases de diseño, desarrollo e implantación para reconocer los requerimientos propuestos por el usuario.

Tabla 7: Asignación de Grados para el Rendimiento.

✓ Configuración Altamente Usada.

El software será implementado en un entorno existente y más utilizado. La forma de asignar su grado de influencia se asigna en la siguiente tabla:

Grado	Descripción
0	No se han incluido restricciones operacionales de forma implícita o explícitamente.
1	Existen restricciones operacionales, pero son menos restrictivas que en una aplicación típica. No se requiere hacer esfuerzos especiales para cumplir con las restricciones.
2	Se incluyen algunas restricciones de seguridad o tiempo.
3	Se incluyen requerimientos específicos para un proceso específico de la aplicación.
4	Requerimientos de operación requieren restricciones especiales sobre el procesador central o un procesador dedicado para la ejecución de la aplicación.
5	Además, existe alguna restricción especial sobre la aplicación y los componentes distribuidos del sistema.

Tabla 8: Asignación de Grados para la Configuración Altamente Usada.

✓ Promedio de Transacciones.

Este grado influye mucho en el diseño, desarrollo, implantación y soporte. Sus grados de influencia se describen a continuación:

Grado	Descripción
0	No se estima ni se anticipa un periodo pico de transacciones.
1	Se anticipa un período pico de transacciones.
2	Se anticipa un período pico de transacciones semanal.
3	Se anticipa un período pico de transacciones diario.
4	Se establece dentro de los requerimientos un período promedio de transacciones alto o los niveles de servicio son suficientemente grandes para realizar tareas de análisis de rendimiento en la fase de diseño.
5	Además, se utilizarán en las fases de diseño, desarrollo y /o implementación las herramientas de análisis de rendimiento para examinar los requerimientos de rendimiento puestos por el usuario.

Tabla 9: Asignación de Grados Promedio de Transacciones.

✓ Entrada de Datos en Línea.

El nuevo software necesitara de entradas interactivas. Los grados se detallan de la siguiente manera:

Grado	Descripción
0	Todas las transacciones se procesan en modo batch.
1	Del 1 al 7 % de las transacciones se introducen de modo interactivo.
2	Del 8 al 15 % de las transacciones se introducen de modo interactivo.
3	Del 16 al 23 % de las transacciones se introducen de modo interactivo.
4	Del 24 al 30 % de las transacciones se introducen de modo interactivo.
5	Más del 30 % de las transacciones se introducen de modo interactivo.

Tabla 10: Asignación de Grados de Entrada de Datos en Línea.

✓ Eficiencia para el Usuario Final.

Este punto es muy importante ya que Sumano (et.al.: 26) se refiere a la usabilidad que tendrá el sistema por parte del usuario final para lo cual se detalla una lista con las actividades principales:

- Ayudas para la navegación del sistema.
- Menús.
- Ayuda en línea y documentos.
- Movimiento automático del cursor.
- Scrolling (manejo de la pantalla como documento en rollo de papel).
- Impresión remota vía transacciones en línea.
- Asignación de teclas de función.
- Activación de trabajos batch desde una transacción en línea.
- Selección del cursor desde la pantalla de datos.
- Uso intenso de video inverso, colores para resaltar, luminosidad, y otros indicadores.
- Copia en documento para el usuario de las transacciones en línea.
- Interfaz de ratón.
- Ventanas pop-up.
- Tan pocas ventanas como sea posible para lograr una función de Negocio.
- Soporte bilingüe.
- Soporte multilingüe(más de dos y hasta seis lenguajes).

✓ Actualización en Línea.

“Se necesitará la actualización de archivos en forma interactiva”. Así lo menciona Sumano (et.al.:27, 28, 29,30) en las Tablas 11, 12, 13, 14, 15, 16 y 17, para asignar su grado de influencia.

Grado	Descripción
0	No existe.
1	Se incluye la actualización en línea de uno a tres archivos de control. El volumen de actualización es bajo y la recuperación es fácil.
2	Se incluye la actualización de al menos cuatro archivos de control. El volumen de actualización es bajo y la recuperación es fácil.
3	Se incluye la actualización en línea de la mayoría de los archivos lógicos internos.

4	Además, es esencial la protección de datos y tendrá que ser especialmente diseñada y programada en el sistema.
5	Además, se consideran dentro de los procesos de recuperación volúmenes altos. Se incluye la automatización de procesos de recuperación con participación mínima del usuario.

Tabla 11: Asignación de Grados de Actualización en Línea.

✓ Procesamiento Complejo.

En este grado se analizará como el procesamiento lógico influencia el desarrollo de la aplicación. Las siguientes componentes podrían ser parte del sistema: Control sensitivo (proceso de audición especial); procesamiento lógico extensivo; procesamiento matemático extensivo, una transacción incompleta provoca muchas excepciones resultantes del procesamiento al ser procesada nuevamente, el manejo de posibilidades múltiples de entrada o salida incluyen procesamiento complejo. (Sumano, et.al.:)

Para asignar los grados de influencia se utilizará la tabla que se describe a continuación: (Sumano, et.al.: 27).

Grado	Descripción
0	Ninguna de las de arriba.
1	Alguna de las de arriba.
2	Dos de las de arriba.
3	Tres de las de arriba.
4	Cuatro de las de arriba.
5	Cinco de las de arriba.

Tabla 12: Asignación de Grados de Procesamiento Complejo.

✓ Reusabilidad.

“Este es el grado en que la aplicación y el código han sido diseñados, desarrollados y soportados para que se pueda utilizar en otras aplicaciones. La forma de asignar su grado de influencia se muestra enseguida” (Sumano, et.al.: 27-28).

Grado	Descripción
0	El código no es reusable.
1	El código reusable se utiliza en la aplicación.
2	Menos del 10% de la aplicación considera más de una necesidad del usuario.
3	El 10% o más porcentaje de la aplicación consideran más de una necesidad del usuario.
4	La aplicación fue documentada para el reuso, puede ser modificada por el usuario en nivel de código fuente
5	La aplicación fue documentada para facilitar el reuso; la aplicación puede ser personalizada por el usuario mediante parámetros de mantenimiento.

Tabla 13: Asignación de Grados para la Reusabilidad.

✓ Facilidad de Instalación.

“Describe el modo en el que el medio ambiente influirá dentro del desarrollo de la aplicación. La instalación de la aplicación debe ser aprobada durante la fase de prueba. La asignación de sus grados de influencia se muestra de la siguiente forma: tabla 14” (Sumano, et.al.: 28).

Grado	Descripción
0	No existen consideraciones especiales del usuario, tampoco se requirió un set up para la instalación.
1	No existen consideraciones especiales del usuario, pero se requirió un setup para la instalación.
2	La instalación fue requerida por el usuario, fueron probadas guías para la instalación.
3	La instalación fue requerida por el usuario, fue probada la guía para la instalación.
4	Además de las dos anteriores, herramientas automáticas de instalación será probada.
5	Además de las tres anteriores, la instalación automática será probada.

Tabla 14: Asignación de Grados de Facilidad de Instalación.

✓ **Facilidad de Operación.**

Es el grado en el que cada una de las aplicaciones atienden los aspectos operacionales, por ejemplo: guardar y recuperar datos y procesos. La principal ventaja es minimizar la necesidad de actividades manuales, tales como el uso de dispositivos extraíbles o el manejo de papel. Las características pueden ser las que se listan a continuación:

- Se proporcionarán procedimientos para almacenar y recuperar datos y procesos, pero se requiere la intervención del operador.
- Se proporcionarán procedimientos para almacenar y recuperar datos y procesos, sin requerir la intervención del operador (cuenta doble).
- La aplicación minimiza el tener que conectar dispositivos periféricos.
- La aplicación minimiza el tener que manejar el papel.
- La forma de asignar su grado de influencia es como sigue:

Grado	Descripción
0	No se establecieron requerimientos especiales de operación además de procedimientos normales de salvado de datos.
1	La aplicación considera una de las características sencillas.
2	La aplicación considera dos de las características sencillas o la doble.
3	La aplicación considera tres de las características sencillas o la doble y una sencilla.
4	La aplicación considera todas las cuatro características sencillas o la doble y dos sencillas.
5	No hay intervención de un operador del sistema

Tabla 15: Asignación de Grados de Facilidad de Operación.

✓ Varios Sitios.

“Es el grado en el cual la aplicación será diseñada, desarrollada e implantada en múltiples localizaciones y organizaciones de usuarios. La forma de asignar su grado de influencia se representa en la tabla 16”: (Sumano, et.al.: 29).

Grado	Descripción
0	Cuando por parte del usuario se considera un solo sitio.
1	Las consideraciones de sitios múltiples se considerarán en el diseño y la aplicación se ejecutará en varias instalaciones de hardware y software iguales.
2	Las consideraciones de sitios múltiples se considerarán en el diseño y la aplicación se ejecutará en varias instalaciones de hardware y software similares.
3	Las consideraciones de sitios múltiples se considerarán en el diseño y la aplicación correrá bajo instalaciones de hardware y/o software diferentes.
4	La documentación y plan de soporte se proveerán y probarán para soportar la aplicación en sitios múltiples para aplicaciones del tipo descritas en 1 y 2.
5	La documentación y plan de soporte se proveerán y probarán para soportar la aplicación en sitios múltiples para aplicaciones del tipo descritas en 3.

Tabla 16: Asignación de Grados para Varios Sitios.

✓ Facilidad de Cambios.

“Es el grado en el que cada aplicación permite modificarla ya sea en su procesamiento lógico o las estructuras de datos. En la siguiente lista se muestran las principales características:” (Sumano, et.al.: 29).

- a. Consulta flexible y facilidad de reportes, mediante el manejo de requerimientos simples como el uso de los operadores lógicos o sobre un ALI.
- b. Consulta flexible y facilidad de reportes, mediante el manejo de requerimientos simples como el uso de los operadores lógicos o sobre dos ALI.
- c. Consulta flexible y facilidad de reportes, mediante el manejo de requerimientos simples como el uso combinado de los operadores lógicos o sobre tres ALI.
- d. Datos de control del negocio se guardan en tablas que son mantenidas por el usuario con procesos interactivos en línea, pero los cambios se reflejan hasta el día siguiente del negocio.
- e. Datos de control del negocio se guardan en tablas que son mantenidas por el usuario con procesos interactivos en línea, los cambios se reflejan inmediatamente.

La forma de asignar su grado de influencia se muestra en seguida:

Grado	Descripción
0	Ninguna de las de arriba.
1	Alguna de las de arriba.
2	Dos de las de arriba.
3	Tres de las de arriba.
4	Cuatro de las de arriba.

Tabla 17: Asignación de Grados de Facilidad de Cambios.

2.2.3. Técnica de Grupo Nominal.

La Técnica de Grupo Nominal (TGN) fue creada en 1960 a partir de estudios sociopsicológicos. La TGN se desarrolla bajo una reunión estructurada de grupo para la toma de decisiones organizacionales y resolución de problemas.

Los procedimientos de que se compone la TGN son: (Sumano, et.al.: 38). Generación

de ideas: en este proceso cada individuo en el grupo generará en forma silenciosa sus ideas y las escribirá. Recolección de ideas: se lo realiza mediante un recorrido de todos los participantes se escribirán las ideas a la vista de todos. Discusión de ideas: cada idea escrita se discute para clarificarla y evaluarla. Votación sobre las ideas: en esta etapa cada individuo votará por la prioridad individual de cada idea y se evaluará posteriormente para obtener su promedio. Los Beneficios que se obtienen de la TGN son: Participación balanceada de todos los participantes. Influencia balanceada sobre los individuos. Producción de ideas más creativas que las de grupos interactivos. Mayor satisfacción de los participantes sobre los resultados. Reducción del conformismo. Animar a los participantes a confrontar los temas sobre una base de resolución de problemas en lugar de basarse en agresiones personales. Conduce a una sensación de mayor acercamiento y complicidad. Las limitaciones que puede tener la TGN son: Requiere de mayor preparación que una técnica espontánea. Tiende a limitar los temas a tratar, no se puede cambiar a otro tema a la mitad de la reunión. Necesita que todos los participantes estén de acuerdo en cooperar.

2.2.4. La Reunión de Reflexión y Diseño.

Para llevar a cabo los diferentes modelos se utilizarán la adaptación del método de Reunión de Reflexión y Diseño (RRD) usado para el modelado de sistemas y que tiene como principios básicos los siguientes: Debe proporcionarse una plataforma tiempo – espacio para que los procesos de reflexión y diseño se puedan dar en la mayor libertad posible. Asegurar una amplia participación de los involucrados creando un ambiente democrático. Propiciar un clima de confianza y respeto para que las personas o representantes de cada grupo de trabajo pueda emitir sus opiniones aunque estas sean antagónicas. Motivar tanto el diseño como el esfuerzo conjunto para la consecución de metas. Promover el aprendizaje social y aumentar la habilidad para la auto-organización, la adaptación y la planeación. A continuación, se muestra un ejemplo que permitirá entender de mejor manera este modelo: (Sumano, et.al.: 39).

ETAPA	PUNTOS A TRATAR
Exploración del ambiente.	Factores externos que han afectado al sistema. Posibles factores que podrían afectar al sistema.
Análisis interno del sistema.	Situación actual del sistema. Causas de la situación actual. Situación a futuro si no se hace el software para apoyar el sistema del cliente.
Diseño del software ideal.	Definir el software ideal para los próximos 5 años. Funcionamiento de las cosas en ese futuro ideal.
Identificación de obstáculos y oportunidades.	Identificar obstáculos para alcanzar el software ideal. Identificar oportunidades para lograr el software ideal.
Selección de líneas de acción.	Acciones posibles y necesarias para alcanzar el software ideal.
Diseño de líneas de acción.	Actividades necesarias para alcanzar las líneas de acción. Establecimiento de responsabilidades y acciones de cada involucrado.

Tabla 18: Etapas y puntos a tratar en la Reunión.

2.2.5. El lugar para la RRD.

Es recomendable que las reuniones se las realice fuera del ambiente laboral ya que de esta manera el equipo que está dedicado al desarrollo del nuevo sistema, no tendrá distracciones con las actividades cotidianas de la oficina. Muchas de las empresas no lo toman en cuenta por varios factores por ejemplo el económico, ya que es muy difícil que se detengan las actividades por un largo período de tiempo.

2.2.6. Los asistentes a RRD.

“Con respecto a las personas involucradas en el desarrollo del software resulta óptimo que haya al menos un representante de cada grupo de pertenencia de los que utilizarán el software, pero si no es posible, deben reunirse los involucrados de la empresa que tengan la posibilidad de ausentarse físicamente de la empresa, los analistas y un representante del grupo de diseño de software (si son del área de computación, pueden ser las mismas personas que fungen como analistas)”. (Sumano, et.al.: 40).

2.2.7. La duración de la RRD.

Es recomendable un lapso de 5 días, ya que en este período se tratarán problemas relacionados con problemas más estructurados, la solución en este punto es que el analista lleve posibles soluciones y así disminuir el tiempo de cada reunión. Este período de tiempo depende de la empresa y en especial de las habilidades de cada grupo de trabajo por lo que se estima un período de 2 a 3 días máximo. Por otro lado la revisión de los requerimientos se la puede efectuar en sesiones de 3 horas cada una.

2.2.8. La conducción de la Reunión de Reflexión y Diseño.

Las reuniones de la RRD con los involucrados pueden ser de dos tipos: La de grupos pequeños. Un máximo de 10 participantes, incluido el facilitador. Las plenarias. Todos los involucrados. La elección de uno de los dos tipos depende de la cantidad de personas que usarán el software. Si es para pocos usuarios, todas las actividades de las reuniones se harán en un solo grupo. Para cada reunión debe existir un facilitador, el mismo que deberá seguir las siguientes normas No deben intervenir en las discusiones dando opiniones sobre el tema tratado. Mantener un ambiente de confianza y respeto para todos los miembros del grupo y sus ideas. Vigilar que participen de todos los integrantes del grupo. (Sumano, et.al.: 40).

2.2.9. Apoyos físicos para la Reunión de Reflexión y Diseño.

Para el logro de los puntos anteriores se tienen los siguientes apoyos: El conjunto de Guiones y diálogos teóricos y actuales, elaborados previamente por el analista, así como el glosario que se lleva hasta el momento. Un cartel pegado en una parte visible. Para los grupos pequeños una organización física del lugar de las reuniones en semicírculo con el facilitador y su portafolio al frente. Debe evitarse el uso de mesas. Para las sesiones plenarias un acomodo equivalente con semicírculos concéntricos o filas con el facilitador al frente. La formulación de las reglas de conducción de las participaciones por parte de los involucrados, mismas que deberán respetarse durante todo el evento. El uso del método TGN para recoger la prioridad de las escenas. La elaboración de los modelos correspondientes al final de cada sesión. (Sumano, et.al.: 40).

2.2.10. Los resultados de la Reunión de Reflexión y Diseño.

Al término de la RRD se tendrán: Un conjunto de Guiones y diálogos de grupos pequeños, en donde se establecen los papeles y responsabilidades de cada actor (involucrado). Un conjunto de Guiones y diálogos de sesiones plenarias, en donde se establecen los papeles y responsabilidades de cada actor (involucrado). Una serie de hojas de trabajo (las del portafolio). Un glosario homogéneo de términos. Una definición del contexto donde se implantará el software. El conjunto de interfaces que el nuevo software debe considerar. Un manual de operación propuesto aceptado por los involucrados. El costo tentativo del software.

Calendario para la RRD: Antes de empezar la RRD debe planearse un calendario de actividades. Debe notarse que es sólo una propuesta y que puede cambiarse al gusto y necesidades de la empresa para la que se realizará el software, pero respetándolas etapas de la RRD. En el primer día se debe entregar, en la medida de lo posible, fotocopias de la documentación a cada involucrado. Para el siguiente día, el analista modificará guión de propuesta computacional y manual de operación según las correcciones obtenidas en la sesión del día. (Sumano, et.al.: 41).

2.2.11. El calendario para la Reunión de Reflexión y Diseño.

Antes de empezar la RRD debe planearse un calendario de actividades. Debe notarse que es sólo una propuesta y que puede cambiarse al gusto y necesidades de la empresa para la que se realizará el software, pero respetándolas etapas de la RRD. En el primer día se debe entregar, en la medida de lo posible, fotocopias de la documentación a cada involucrado. Para el siguiente día, el analista modificará guión de propuesta computacional l y manual de operación según las correcciones obtenidas en la sesión del día. (Sumano, et.al.: 41).

2.2.12. Ciclo de Vida.

Las fases que componen el ciclo de vida de Áncora se muestran en el siguiente cuadro de Sumano (et. Al.: 43) .y cada una de sus componentes se explicará brevemente.

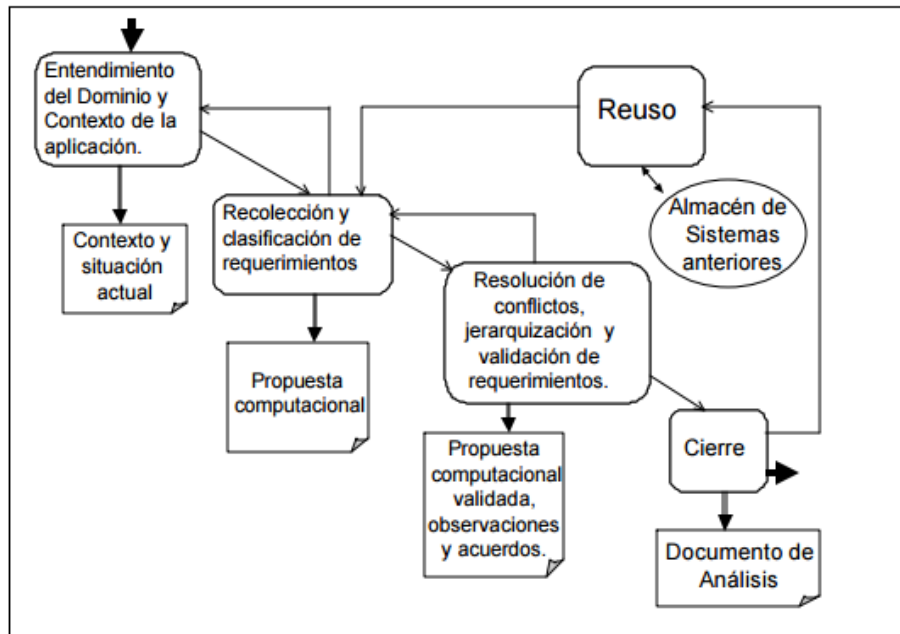


GRÁFICO 3: Ciclo de Vida Áncora.

➤ Entendimiento del Dominio y Contexto de la Aplicación.

Esta etapa es fundamental ya que se busca tener claras las metas de la empresa, el problema que la aplicación solucionará, si se tiene un software en funcionamiento saber cuál es la situación actual y qué problemas presenta para justificar el desarrollo de uno nuevo.

Para adquirir esta información se realizarán las siguientes actividades que producirán los siguientes artefactos o documentos:

- ✓ Leer el Material: permite obtener un primer glosario de la aplicación, esquema de la organización, lista de actividades de cada usuario, guiones y diálogos.
- ✓ Entrevista Informal con el cliente: con esta actividad se tendrá un conocimiento previo sobre las metas de la empresa y la influencia que tendrá el nuevo sistema.
- ✓ Observar Usuarios: esta actividad la realiza el analista, determina como se realizan actualmente las tareas en el sistema y formulará las respectivas anotaciones.

➤ Recolección, clasificación de Requerimientos.

En esta etapa se tiene como objetivo crear la propuesta de un sistema computacional el mismo que tendrá las siguientes componentes:

- ✓ La elaboración de esta propuesta estará compuesta por: Guión de propuesta computacional, prototipo rápido, modelo de datos, lista con las responsabilidades, beneficios esperados.
- ✓ Creación de la Bitácora de desarrollo del sistema: consiste en una tabla de cuatro columnas; en las que constan: funcionalidades del sistema, las pruebas por parte del usuario, tiempo estimado y tiempo de implantación.
- ✓ Puntos de función: para determinar los requerimientos no funcionales del software, que reafirme la propuesta computacional y evalúe la complejidad del sistema.

La primera actividad es la creación del guión, el mismo que nos permitirá obtener y retroalimentar al resto de artefactos.

➤ Reuso de requerimientos.

En esta etapa el principal objetivo es brindar al analista elementos con especificaciones probadas de otros sistemas de software y almacenadas dentro de una base de datos de Reuso. Al respecto, Sumano (et.al.: 44) manifiesta:

Abstraer y guardar los guiones de sistemas terminados. esta es una operación de generalización en la que, mediante transformaciones respectivas de verbos y sustantivos a primitivas y nombres genéricos y la representación de escenas y condiciones de entrada en forma clausular, se obtendrá una representación abstracta de los guiones de la propuesta computacional.

Consultar y extraer elementos reutilizables. Todos los artefactos de las especificaciones de requerimientos de sistemas semejantes que puedan reutilizarse, a saber: el guion o guiones de la propuesta computacional, manuales de operación, listas de restricciones y responsabilidades, bitácoras de desarrollo, cálculo de puntos de función y, si se usó el mismo paradigma de desarrollo elementos para la conexión con el análisis.

➤ Resolución de Conflictos, Priorización Y Validación de Requerimientos.

El objetivo de esta etapa es resolver los conflictos que pudieran existir entre los diferentes requerimientos, se puede asignar prioridades a cada uno y validarlos con la ayuda del usuario. Esta etapa tiene dos actividades que se describen a continuación:

- ✓ Preparación de la Reunión de Reflexión y Diseño (RRD): al tener la información sobre la situación actual y de la propuesta computacional se los agrupa en un documento el mismo que se entregara a cada una de las partes interesadas (clientes, usuarios, analistas, diseñadores) en el software. La razón es que antes de tener la primera reunión las partes interesadas podrán analizar dichos documentos y tendrán sus anotaciones.
- ✓ Reunión de Reflexión y Diseño. Esta actividad se la realiza después de ser entregada la documentación y se sigue la siguiente planificación: se reúne al menos un representante de cada grupo de interés y durante el transcurso de uno a dos días se realiza una reunión en la que los asistentes debatirán sus opiniones y en consenso, se aplicará la Técnica de Grupo Nominal para la categorización de requerimientos.

Al finalizar la Reunión de Reflexión y Diseño se obtendrán: guiones, restricciones, prioridades, manual, responsabilidades y plan de prueba. Entre otros documentos se añaden una lista de observaciones y acuerdos firmados.

✓ Cierre.

✓ Es la última etapa del ciclo de vida, se prevé desarrollar una lista de actividades, que permitan conectarse con otras etapas del proyecto.

Esta etapa consta de las siguientes actividades:

- ✓ Ratificar cálculo de Puntos de Función. Después de la Reunión de Reflexión y Diseño talvez sea necesario ajustar ciertos puntos de función para el siguiente ciclo.
- ✓ Envío de oficios, correo electrónico y faxes para tener un seguimiento documentado sobre los puntos pendientes.
- ✓ Acoplarse al modelo de desarrollo escogido, para lo cual se desarrollan una serie de prototipos que facilitarán el análisis de los requerimientos.

2.3. Cuadro comparativo de Metodologías.

En el siguiente cuadro, el investigador presenta un cuadro comparativo entre dos metodologías orientadas a la toma y análisis de requerimientos. Para realizar este análisis se utilizará: “Áncora y DoRCU”

Áncora	DoRCU
<ul style="list-style-type: none"> • Es una metodología más compleja ya que contiene varios artefactos como : guiones, escenas, actores, etc. • El ciclo de vida y cada actividad que se realizar permite una retroalimentación. • Permite la creación de prototipos para mejorar la especificación de requerimientos. • Maneja un modelo de Base de Datos Semánticos. • Asume que el equipo de desarrollo parte por un analista y diseñador. • La documentación es más precisa ya que no se documenta cada etapa, se realiza un documento consolidado al final de cada ciclo. • Retroalimentación es punto fuerte ya que se mide de acuerdo con los puntos de función y RRD (Reunión de Reflexión y Diseño). 	<ul style="list-style-type: none"> •Es una metodología que tiene como finalidad familiarizar términos utilizados en otras metodologías. •Posee un proceso de retroalimentación para especificación de requerimientos. •No posee puntos de función que le permitan tener una información más detallada. •Busca formar un equipo multidisciplinario. •Se documenta al finalizar cada etapa. •No posee un proceso para la Reunión de Reflexión y Diseño, esto quiere decir que el usuario no está tan involucrado con el equipo de desarrollo.

Cuadro 2: Cuadro Comparativo entre Áncora y DoRCU.

2.4. Metodología de Programación Ágil XP Extreme Programming.

Para entender que es Extreme Programming, primero se deberá tener cierto conocimiento acerca de las principales funciones que posee una metodología ágil, para lo cual conviene definir qué es una metodología ágil, sus principales características y su funcionamiento en la actualidad.

La aparición del término ágil y metodología ágil dentro del desarrollo de software se produjo como una contraposición a los métodos clásicos propuestos como CMMI.

Una metodología ágil según: Damon B. Poole, en su libro “Do it yourself agile”, define el término Ágil como el paraguas que contiene una amplia variedad de metodologías y está determinado oficialmente como los 4 valores en el Manifiesto Ágil.

Considerando la definición anterior se llega a concluir que una metodología ágil es aquella que sigue los principios del manifiesto ágil, estos principios son los que se enlistan a continuación:

- Por encima de los procesos y las herramientas se encuentran los individuos y su interacción.
- El software que funciona, frente a la documentación exhaustiva.
- La colaboración con el cliente, por encima de la negociación contractual.
- Por encima del plan se debe tener en cuenta la respuesta al cambio.

Analizando cada uno de los principios del manifiesto ágil, mucha gente tiene la idea errónea de relacionar la metodología ágil con la pérdida de control dentro de un proyecto de software y falta de documentación del mismo. El objetivo principal de una metodología ágil es aumentar la eficiencia de las partes involucradas dentro del proyecto y lograr una disminución significativa en el coste total.

Poole hace notar que se habla de Metodologías Ágiles como un solo paquete: se obtienen los beneficios, si se conectan a los principios del manifiesto ágil. Pero, hay otra forma de mirarlas y es pensar que las metodologías ágiles introducen un conjunto nuevo y completo de prácticas a las herramientas de desarrollo. Estas prácticas incluyen: product backlog, programación de a pares, cliente en el lugar, integración continua, refactorización, desarrollo dirigido por pruebas y muchas otras. Todas estas prácticas han estado asociadas con las Metodologías Ágiles o fueron creadas como resultado de las Metodologías Ágiles, los beneficios resultantes y su aplicación pueden aplicarse completamente independientes de cualquier metodología específica.

2.4.1. Historia.

Con lo expuesto anteriormente sobre el inicio, definición y uso de una metodología ágil, en esta sección se centrará en analizar y conocer una de las tantas metodologías ágiles que se encuentran en la actualidad, en este caso se tomará a Extreme Programming como la metodología a aplicar dentro del desarrollo de software. Se empezará revisando la historia de esta metodología y las siguientes secciones se profundizarán el análisis de las características y funcionalidades.

Extreme Programming o traducido al español “programación extrema” y más conocida por sus siglas XP es un enfoque de la Ingeniería de Software formulada en 1999 por Kent Beck, en su primer libro denominado: “Programming Explained: Embrace Change”. Este enfoque se diferencia del resto de metodologías tradicionales principalmente por poner mayor énfasis en la adaptabilidad que en la previsibilidad. Para muchos expertos afines a esta metodología la adaptabilidad a cambios es un punto fuerte dentro del desarrollo, puesto que durante la vida del proyecto se estima existirá cambios en los requisitos por parte del cliente; esto permite tener una mejor aproximación al intentar definir los requisitos al inicio e invertir mayor esfuerzo después para el control de los cambios.

XP se centra básicamente en potenciar las relaciones interpersonales como clave para un desarrollo de software exitoso, promueve el trabajo en equipo, haciendo énfasis en el aprendizaje de los desarrolladores, y generando un buen clima de trabajo. Otro punto fuerte de XP es la retroalimentación continua con la participación activa entre el cliente y el equipo de desarrollo, promoviendo una comunicación fluida entre las partes interesadas, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios.

XP es muy útil en proyectos que tienen requisitos bien indefinidos, muy cambiantes y alto riesgo técnico.

2.4.2. Objetivos

Entre los principales objetivos de XP tenemos los que se detallan a continuación:

- ✓ Establecer mejores prácticas de Ingeniería de Software en el desarrollo de nuevos proyectos.
- ✓ Motivar la solidaridad como espíritu de trabajo dentro del equipo de desarrollo.
- ✓ Incentivar el aprendizaje continuo entre todos los miembros del Equipo.
- ✓ Mejorar la productividad del equipo y del proyecto.
- ✓ Garantizar calidad del Software desarrollado, utilizando la retroalimentación.
- ✓ Promueve el involucramiento del cliente dentro de cada fase del desarrollo.

- ✓ Buscar la satisfacción del Cliente.
- ✓ Incentiva el respeto entre el cliente y el equipo de desarrollo.
- ✓ Disminuir costos del proyecto.
- ✓ Promueve el desarrollo incremental e iterativo.

2.4.3. Características

Esta metodología posee entre sus principales características están: los valores que se enlistan de la siguiente manera, cabe recordar que en la primera edición de esta metodología constaban 4 y para la segunda edición se añadió uno nuevo.

<p>Simplicidad</p> <ul style="list-style-type: none"> • Es la base de XP. • Agiliza el desarrollo y facilita el mantenimiento. • Se tiene reglas para la documentacion de codigo. • Facilita la comprension de todo el equipo.
<p>Comunicación</p> <ul style="list-style-type: none"> • Comunicacion entre el equipo para docuemntar el codigo. • Comunicaion con el cliente para definir prioridades del sistema y aclarar dudas.
<p>Retroalimentación (feedback)</p> <ul style="list-style-type: none"> • Al tener al cliente dentro del desarrollo puede tener informacion sobre el avance del proyecto en tiempo real. • El codigo puede ser depurado mediante pruebas unitarias. • Por medio de ciclos de desarrollo cortos se disminuye los fallos y tiempo de codificacion.
<p>Coraje o valentía</p> <ul style="list-style-type: none"> • Los programadores deben tener la valentia para desechar codigo obsoleto. • Se debe tener la mentalidad de que se desarrolla para el presente, es decir saber que en el futuro existiran cambios. • Persistencia para solucionar problemas del sistema.
<p>Respeto</p> <ul style="list-style-type: none"> • Mantener un ambiente de laboral respetuoso entre el equipo de desarrollo permite elevar el autoestima de todos permitiendo una mayor productividad.

Cuadro 3: Valores del Manifiesto Ágil.

Elaborado por: (Andres, Kent Beck with Cyntia, Extreme Programming Explained Embrace Change,2004)

Otra característica es la creación de roles, los mismos que se delegan a cada miembro del equipo en el siguiente cuadro se detallan el tipo y actividad de cada uno:

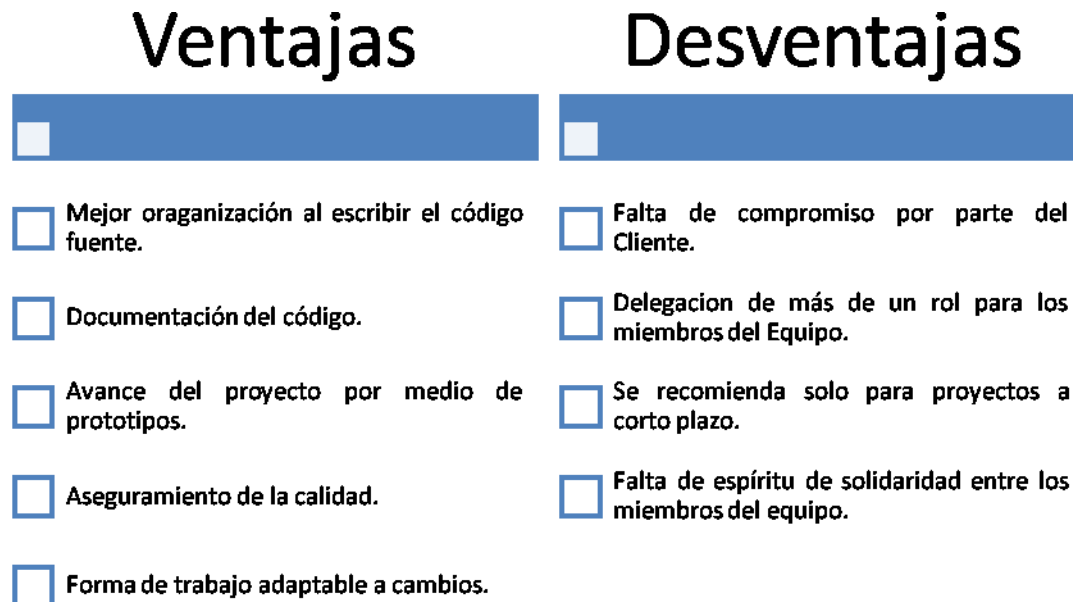
ROL	Actividad
Programador	Encargado de escribir el código del sistema y realizar las pruebas unitarias.
Cliente	Brinda información, asigna prioridades, valida las funcionalidades del sistema y ayuda a generar mayor calidad al sistema.
Tester	Ayuda al cliente con las pruebas funcionales, se encarga de realizar pruebas al sistema periódicamente y se encarga de informar sobre fallos al equipo.
Tracker	Encargado de dar seguimiento y retroalimentar al equipo y verifica el tiempo estimado contra el tiempo real.
Entrenador	Guía al equipo para seguir con el proceso.
Consultor	Es un miembro externo al equipo que posee cierto conocimiento en algún área específica y sirve de ayuda para la solución de problemas.
Gestor	Es el intermediario entre el cliente y el equipo de desarrollo, su labor es la coordinación.

Tabla 19: Roles de XP Extreme Programming.

Elaborado por: (Andres, Kent Beck with Cynthia, Extreme Programming Explained Embrace Change, 2004)

2.4.4. Ventajas Y Desventajas De XP.

En el siguiente diagrama se muestra un análisis crítico de las ventajas y desventajas del uso de esta metodología:



Cuadro 4: Ventajas y Desventajas de XP Extreme Programming.

Elaborado por: (Andres, Kent Beck with Cyntia, Extreme Programming Explained Embrace Change,2004)

2.4.5. Comparativa Con Otras Metodologías.

En el siguiente gráfico se muestra un análisis comparativo entre XP y SCRUM dos metodologías de desarrollo ágil.

XP	SCRUM
<input type="checkbox"/> Las iteraciones para cada entrega de 1 a 3 semanas.	<input type="checkbox"/> Las iteraciones para cada entrega de 2 a 4 semanas.
<input type="checkbox"/> Las funcionalidades son susceptibles a constante cambio por parte del cliente.	<input type="checkbox"/> Las funcionalidades que son validadas no se vuelven a tocar.
<input type="checkbox"/> El avance del proyecto sigue un orden estricto de prioridades descritas por el cliente.	<input type="checkbox"/> El equipo puede cambiar el orde de prioridades descrito por parte del cliente.
<input type="checkbox"/> Programacion en parejas.	<input type="checkbox"/> Progrmación individual.

Cuadro 5: Diferencias entre XP Extreme Programming y SCRUM.

Elaborado por: Galvin Chaves, SlideShare. Obtenido de diferencias entre scrum y xp: <http://es.slideshare.net/deborahgal/diferencias-entre-scrum-y-xp-12219336,2012>)

CAPÍTULO 3: HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO DEL SISTEMA

3.1. Lenguaje de Programación del lado del Servidor.

3.1.1. Características Principales.

EL lenguaje de programación del lado del servidor consiste en el procesamiento de una petición del usuario, mediante la lectura y ejecución de un script en el mismo servidor, lo que ayuda a generar paginas HTML como resultado. Los procesos que se ejecuten en este lado, son respuestas a todas las peticiones de los clientes, el tiempo que se demora en bajar la información solicitada a los equipos de los clientes depende principalmente del tipo de información.

Los lenguajes de lado del servidor se los conoce como aquellos que se ejecutan desde el servidor. En la actualidad se conocen varios lenguajes entre los más conocidos están: JSP, PHP, .NET, Python, etc. Estos lenguajes usualmente crean paginas HTML dinámicas.

Para poder utilizarlos es necesario un servidor de aplicaciones y un software instalado dentro del servidor Web, este software cumplirá dos funciones que son: interpretar o ejecutar el código compilado, además de gestionar las peticiones de los navegadores Web.

Para entender cómo se procesa la información en el lado del servidor es necesario considerar que tipo de solicitud que se va responder, es decir, el tipo de información solicitada y el tipo de dato, el tipo e dato puede ser numérico, alfabético o el resultado de alguna operación matemática.

Por lo general la información que se consigue del lado del servidor es para realizar lo que en términos informáticos se conoce con las siglas CRUD, que significan, crear o generar, actualizar y consultar datos almacenados en una Base de Datos, en este punto se debe considerar todos los tipos de datos que se pueden almacenar, ya que no siempre se almacenan datos del tipo numérico o alfabético, existen casos en los que se almacenan datos del tipo multimedia ya sean estos música, imágenes o videos.

Cuando se realiza una llamada al servidor, dicha llamada se gestiona a través de un servidor Web. El servidor Web transfiere, si es necesario, la petición al servidor de aplicaciones. Entre los servidores aplicaciones conocidos tenemos: Apache, Tomcat, JBoss y IIS.

Estos servidores necesitan de filtros que los ayuden a redirigir las peticiones, por ejemplo el IIS, utiliza filtros ISAPI. Por lo general estos filtros se instalan automáticamente.

Anteriormente los primeros servidores web brindaban la opción de visualizar la información de forma estática, esto presentó una limitación; principalmente en el campo comercial y publicitario ya que empezaron a utilizar el internet.

Con estos antecedentes se implementó la primera solución técnica esta consistía en tener

un servidor web que ejecutará programas residentes en la máquina de servicio. Esta tecnología es conocida como Common Gateway Interface (CGI). Esta tecnología resolvía en gran parte el problema de presentación de información estática; pero tenía dos limitaciones importantes: la primera fue la ejecución de programas indeseados en el servidor y la segunda la carga del servidor al ejecutar una página que era llamada por varios clientes a la vez.

Hoy en día se han desarrollado soluciones que resuelven estos problemas, se desarrolló una tecnología que permite que en único proceso del servidor se ejecute el código sin importar el número de clientes que se conecten. Es así como surgen los famosos Servlets, desarrollados por Sun Microsystem y ISAPI desarrollado por Microsoft.

La aplicación del lenguaje de lado del servidor, se la realiza normalmente, por ejemplo, el momento de acceder a nuestro correo electrónico enviamos una petición al servidor por medio del internet, la velocidad en cuanto al tiempo de respuesta depende mucho de la distancia que existe entre el servidor y el cliente, por lo tanto si el cliente está muy alejado del servidor la página a la que está accediendo se demorara en mostrar la información.

Hay que tomar en consideración el tener actualizados los plugins del navegador ya que estos facilitan la correcta codificación y despliegue de información de las páginas web como por ejemplo los elementos multimedia.

Para comprender como funciona el lenguaje de lado del servidor, a continuación se muestra dos gráficos en los que se presentan la arquitectura de este lenguaje y como se ejecutan las peticiones al servidor:

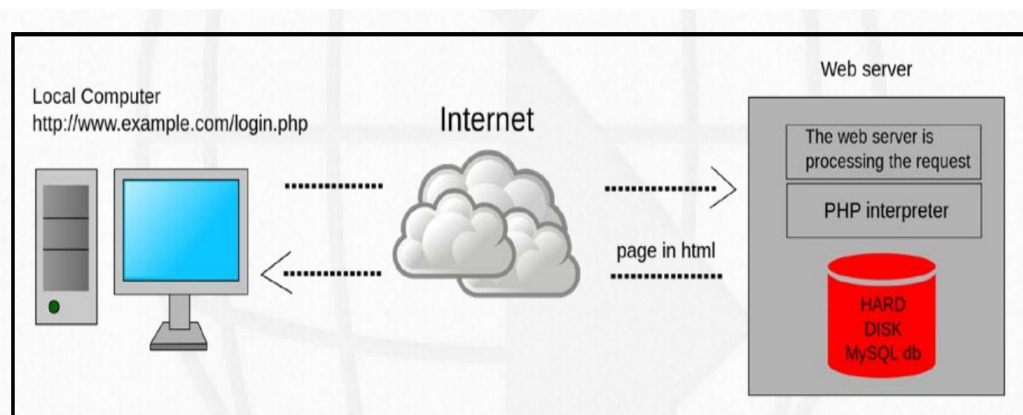


GRÁFICO 4: Arquitectura Lenguaje de Lado del Servidor.

Elaborado por: (Universidad Interamericana para el Desarrollo, Lenguajes De Programación Web, 2012)

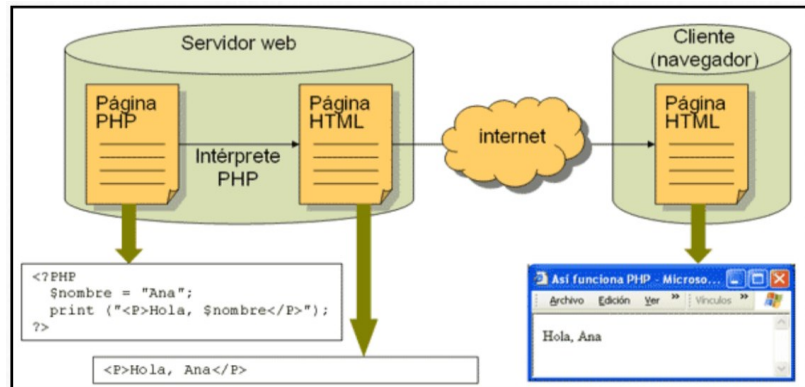
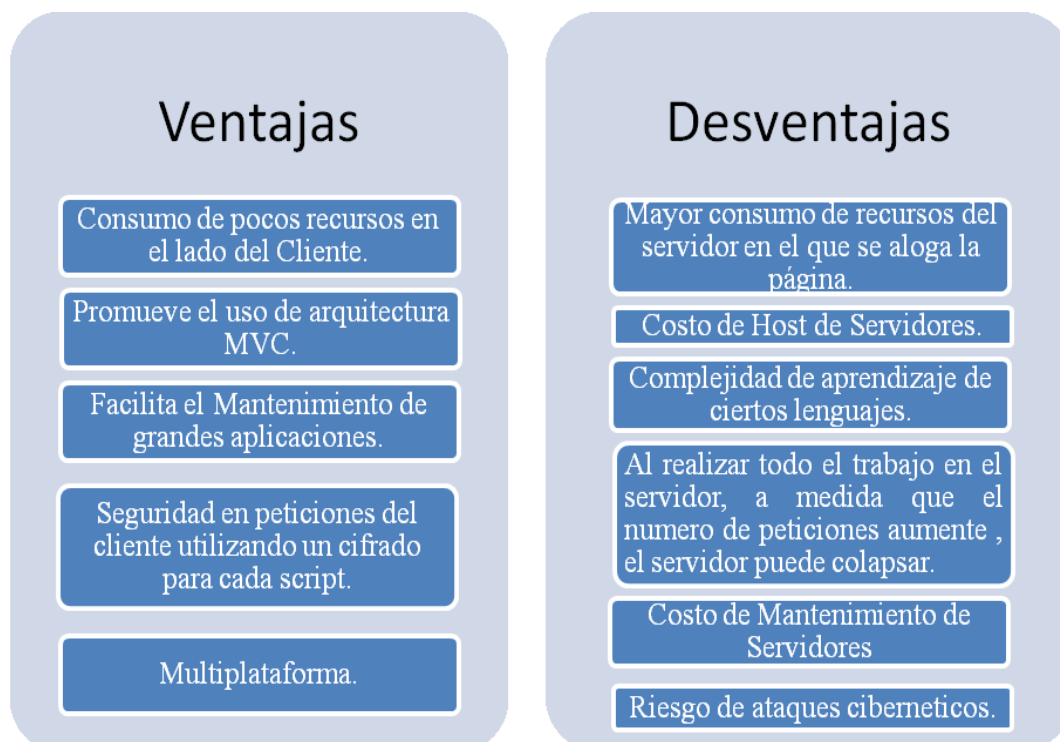


GRÁFICO 5: Ejecución de Peticiones al Servidor.

Elaborado por: (Universidad Interamericana para el Desarrollo, Lenguajes De Programación Web, 2012)

3.1.2. Ventajas y Desventajas de la Programación del Lado del Servidor.

En el siguiente cuadro se resumen ciertas ventajas y desventajas que tiene el implementar un lenguaje de Servidor.



Cuadro 6: Ventajas y Desventajas de la Programación del Lado del Servidor.

Elaborado por: (Universidad Interamericana para el Desarrollo, Lenguajes De Programación Web, 2012)

3.1.3. Curva de Aprendizaje de Lenguaje del Lado del Servidor.

Para aprender e implementar cualquier lenguaje de lado del servidor de la amplia gama que existe en la actualidad se debe tomar en cuenta varios factores como el tipo de página web, el tiempo estimado para la creación de dicha página, los recursos económicos del cliente, recursos técnicos que se posee en la actualidad, recursos bibliográficos ya sean estos: libros, manuales o la web.

Entre los principales lenguajes encontramos los que se describen en el siguiente diagrama estadístico realizado por GitHub:

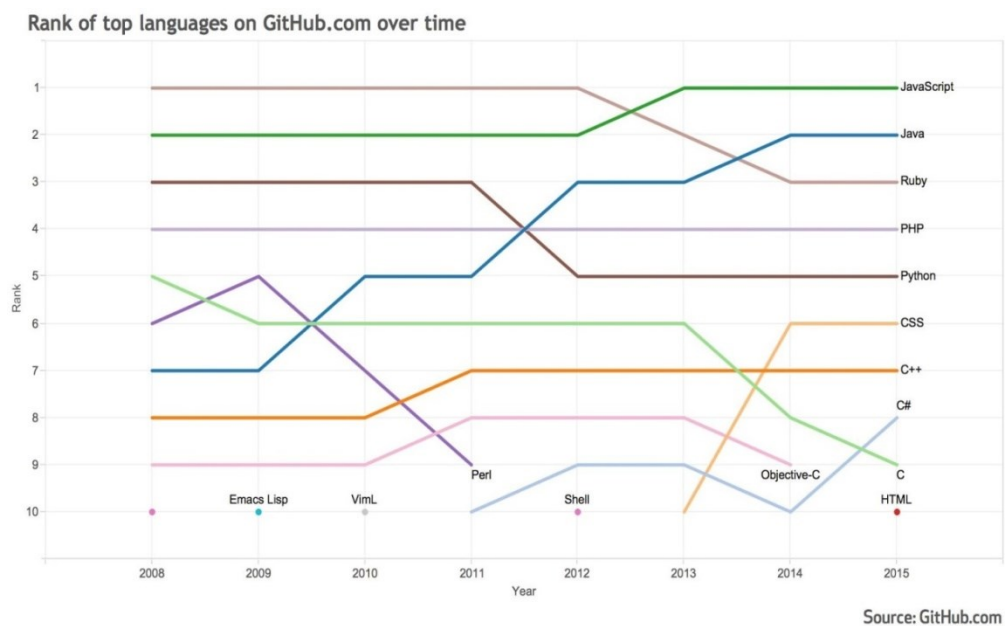


GRÁFICO 6 : Estadísticas sobre los Lenguajes de lado del Servidor más usados hasta el 2015.
Elaborado por: (GitHub, Language Trends on GitHub, 2015)

Al analizar el diagrama anterior se puede observar como en el transcurso de los últimos años se ha incrementado el uso de los dispositivos móviles y el internet.

Por esta razón se observa claramente el despunte y declive de algunos de los lenguajes de lado del servidor como en el caso de Python y Ruby, la aparición de nuevos lenguajes como: CSS y C#, también se debe tomar en cuenta que PHP no ha perdido popularidad en este lapso de tiempo.

Para finalizar el análisis se tomara como referencia tres lenguajes que se encuentra en los primeros lugares como: JavaScript, Java y PHP, estos lenguajes han venido haciendo grandes mejoras como la creación de frameworks, manuales y mantenido actualizaciones constantes en su código fuente, esto les ha permitido mantenerse en la vanguardia dentro del mercado informático. Ayudando al aprendizaje de muchos programadores a nivel mundial.

3.2. Framework usado para el Desarrollo

3.2.1. Características.

En esta sección se realiza un breve estudio del lenguaje de programación PHP, tomando como punto de partida: la reseña histórica del lenguaje, su utilización en la actualidad y el framework que se utilizará para el desarrollo del sistema.

En el desarrollo del sistema se usara un framework denominado Yii que permite realizar el desarrollar páginas Web de forma ágil, poniendo énfasis en la fase de diseño.

PHP es un lenguaje de programación desarrollado en 1994 por Rasmus Lerdorf, en su primera versión PHP empezó como un simple conjunto de archivos binarios Common Gateway Interface(CGI) de lenguaje C.

PHP se conocía como “Personal Home Page Tools” y referenciado como “PHP Tools”, su principal uso fue el rastrear vistas online.

En la segunda versión Rasmus reescribió el código PHP, añadiéndole nuevas funcionalidades y consolidando así un lenguaje con una implementación más grande y rica. En esta nueva versión PHP fue capaz de interactuar con las Bases de Datos, lo que proporcionó un entorno más completo para el desarrollo de páginas web dinámicas y sencillas por ejemplo un libro de visitas.

- En junio de 1995, Rasmus publicó el código fuente de PHP Tools, esto permitió usarlo con libertad y mejorarlo por parte de la comunidad de desarrolladores.
- En septiembre de 1995, Rasmus amplió PHP y se dejó el nombre de PHP renombrándolo como FI, cuyas siglas significan “Forms Interpreter”, esta nueva implementación incluía algunas de las funciones básicas de PHP las mismas que se conocen hasta el día de hoy.

Esta nueva versión tenía algunas características de Perl como la declaración e interpretación de variables, y sintaxis alojada en páginas HTML.

- En octubre de 1995, Rasmus publicó una nueva versión de PHP, regresando a definirlo como PHP, esta versión era conocida con el nombre “Personal Home Page Construction Kit”, en ese momento la versión era conocida como una interfaz de scripts avanzada.
- En 1996, el código de PHP fue totalmente rehecho, combinando los nombres de versiones anteriores, denominándolo como PHP/FL. Esta nueva implementación promociono a PHP como un lenguaje de programación de derecho propio, el mismo que contenía un conjunto de herramientas.

PHP en esta versión incluía soporte las diferentes tecnologías como: DBM, mySQL, y bases de datos Postgres95, cookies, soporte para funciones definidas por el usuario, y mucho más.

- En junio de 1996, PHP/FL lanzó al mercado la versión 2.0 un hecho curioso de este lanzamiento fue que esta versión era la única versión completa del lenguaje.
- En noviembre de 1997, se lanza la versión beta de PHP. Esta versión tuvo un corto tiempo de vida de desarrollo, a pesar de su crecimiento de popularidad dentro del aun joven mundo del desarrollo.
- En 1998 tenía una aceptación del 1% de todos los dominios de Internet en ese momento. A pesar de este hecho PHP tenía limitaciones debido a que seguía siendo un proyecto de un solo individuo con colaboración de contribuidores menores.
- Hoy en día PHP posee algunas herramientas y una amplia gama de Framework que permiten desarrollar de manera ágil páginas web dinámicas, entre las principales herramientas tenemos:
 - PHPUnit.
 - phpDocumentor.
 - Scavenger.
 - PHP_Debug.

Mientras que entre los principales framework están:

- Laravel
- CodeIgniter
- Yii
- Cakephp
- Zend

En el siguiente grafico se muestra la curva de popularidad de los frameworks anteriormente mencionados, en este grafico se refleja cómo han aparecido a lo largo de los últimos 10 años y su uso por parte de los desarrolladores en el mercado del software.

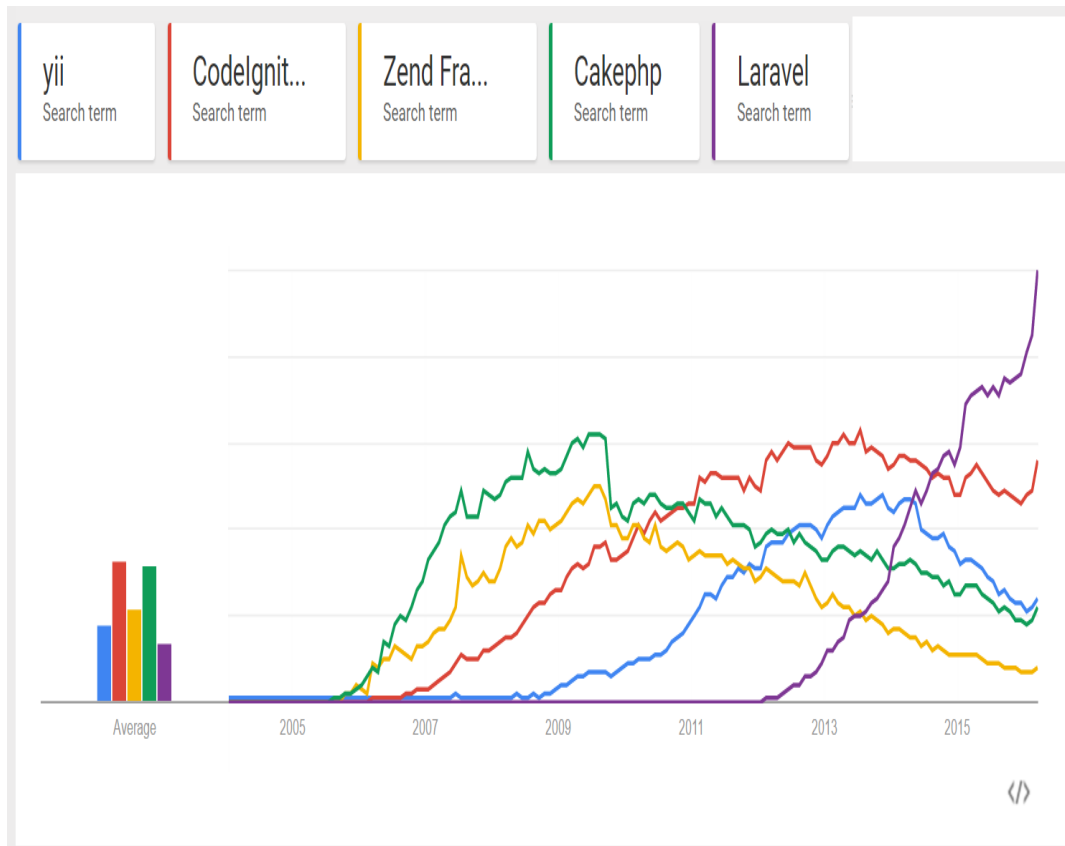


GRÁFICO 7: Estadísticas sobre Frameworks PHP más utilizados en la última década.
Elaborado por: (Long, www.omegaweb.com/best-php-frameworkomegaweb, 2013)

Para el desarrollo del presente sistema se utilizará el siguiente framework denominado Yii este framework es usado para el desarrollo de aplicaciones web de gran escala, una de sus principales ventajas es la reutilización de código lo que permite acelerar el proceso de desarrollo. El nombre de Yii está dado por las palabras “easy, efficient and extensible”. Su acrónimo significa “Yes It is”

Entre las principales características que presenta Yii Framework están:

- Patrón de diseño MVC.
- Conexión de Base de Datos utilizando Database Access Objects (DAO), query builder y Active Record.
- Posee integración con jquery.
- Contiene Widgets de Ajax.
- Autenticación Incorporado.
- Generación compleja de WSDL, administración de peticiones a Web Services.

- Soporte de traducciones en campos de fecha y hora, formato de números y localización de la vista.
- Soporta el cache de: datos, páginas, fragmentos y contenido dinámico.
- Manejo de errores o logs.
- Posee medidas de Seguridad como: cross-site scripting (XSS), cross-site request forgery (CSRF) y manipulación de cookies.
- Posee herramientas para realizar pruebas Unitarias y funcionales.
- Generación automática de código para el esqueleto de la aplicación.
- Diseñado para trabajar con el código escrito en otros frameworks.
- Este framework es utilizado para aplicaciones de alto tráfico como portales, foros, sistemas de administración de contenidos gracias al uso sofisticado de cache.

3.2.2. Curva de Aprendizaje de Yii Framework

Yii presenta una gran ventaja respecto de framework anteriores posee una amplia gama de fuentes de consulta en su página web lo que permite a desarrolladores que desean iniciarse en este nuevo framework, la parte más complicada para empezar a trabajar es la instalación del framework, en muchos de los tutoriales se explica de una forma muy complicada ya que la realizan por línea de comandos, lo interesante de este framework es su compatibilidad con Netbeans lo que permite que sea más fácil su manipulación.

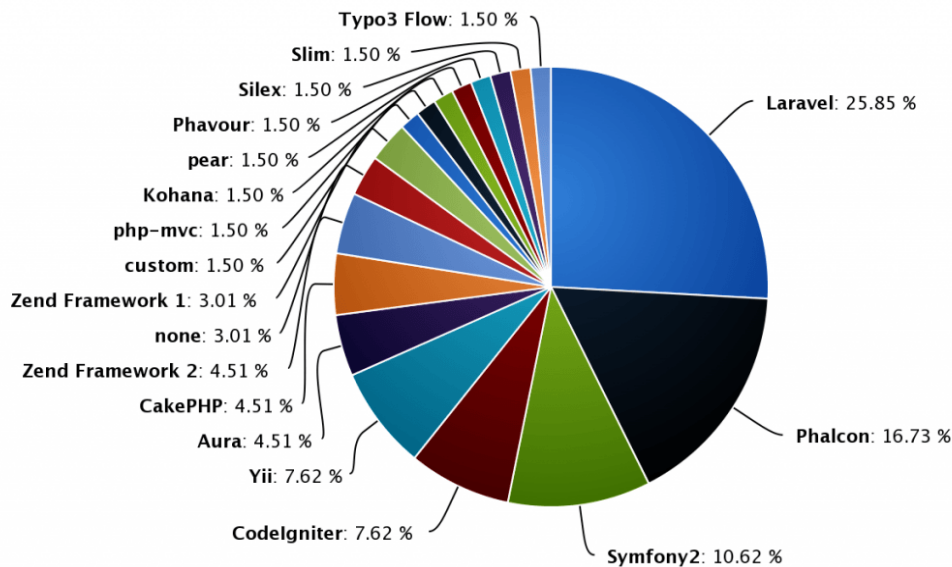
Una vez instalado el framework permite conectarse a la base de datos ya sea MySQL o Postgres, al tener esta configuración el framework nos permite crear la estructura MVC de forma gráfica utilizando la interfaz que el mismo framework tiene incorporada. Es recomendable realizar un correcto análisis de requerimientos para que el modelo de base de datos sea consistente, en la opinión de muchos desarrolladores esta herramienta es muy útil para el desarrollo siempre y cuando el modelo este bien concebido.

Una de las posibles limitaciones que presenta este framework es que muchos de los manuales se encuentran en Ingles lo cual dificulta de cierto modo el aprendizaje, en la opinión de muchos usuarios en algunos blogs sobre este framework, se alienta al uso de este framework ya que es muy eficiente y rápido respecto a otros frameworks, al utilizar este framework no solo que se utiliza el lenguaje PHP, también se utiliza Ajax y JavaScript para las consultas a la base de datos, analizando todos estos aspectos la curva de aprendizaje si es extensa; pero no imposible de seguirla.

Para muchos de los desarrolladores al utilizar este framework ha aprendido no solo a diseñar una página web tradicional, han adquirido diferentes conocimientos y ha logrado comprender la funcionalidad del Modelo MVC dentro de una aplicación web.

En la actualidad Yii no es muy conocido a pesar de todas sus características en el siguiente grafico se refleja esta realidad, cabe recalcar que los framework con mayor índice son utilizados por empresa grandes.

Framework popularity, end of 2013; SitePoint



Highcharts.com

GRÁFICO 8: Estadísticas sobre Frameworks PHP más utilizados.

Elaborado por:(mycyberacademy,mycyberacademy.com/frameworks-php-tener-en-cuenta/,2014)



3.2.3. Ventajas del Framework.

Entre las principales ventajas que nos brinda este framework están:

- Patrón MVC.
- Facilidad para entender el código.
- Código mejor organizado.
- Posee amplia documentación.
- Posee ORM que permite generar vistas, modelo, controlador de forma visual.
- Multiplataforma.
- Acelera el desarrollo de la aplicación permitiendo dedicar mayor tiempo al diseño.
- Permite implementar código de otros frameworks.
- Conectividad con otros frameworks.
- Consumo de pocos recursos de hardware.

3.2.4. Comparativa con otros Framework.

En el siguiente cuadro se detallaran varias diferencias entre Yii y Zend frameworks de PHP.

	
<ul style="list-style-type: none"> -Memoria ram recomendada 256 MB. -Contiene lenguaje PHP y Javascript. -Posee: <ul style="list-style-type: none"> Active-Record Model-View Controller DAO HMVC Observer Dependency injection RESTfull -Fecha de creacion 2008 - Mas facil de usar. -Consume menos recursos del procesador. 	<ul style="list-style-type: none"> -Memoria ram recomendada 512 MB. -Contiene lenguaje PHP. -Posee <ul style="list-style-type: none"> DependencyinjectionModel, View, ControllerEvent, DrivenData MapperSingleton, Active Record RESTfull. -Fecha de Creacion 5 de Septiembre 2012

Cuadro 7: Comparación entre Yii Framework y Zend.

Elaborado por: (Yii vs. Zend Framework 2, www.vschart.com/compare/yii/vs/zend-framework, 2015)

3.3. Lenguaje de programación de Lado del Cliente

3.3.1. Características de programación de Lado del Cliente.

La programación de lado del cliente tiene como principal funcionalidad la ejecución de una aplicación web en el cliente, esto evita que el servidor se recargue de trabajo. El servidor solo envía el código y el encargado de interpretar el código es el browser.

El Browser prácticamente es un programa que se encarga de interpretar órdenes recibidas en forma de código HTML y convertirlas en páginas que son el resultado de cada orden.

Al utilizar un browser por lo general se utiliza HTML un lenguaje estándar que puede ser interpretado por la mayoría de exploradores, esta es una ventaja ya que este lenguaje no necesita ser compilado lo que permite que las páginas escritas se muestren con mayor rapidez.

Entre los lenguajes más conocidos se encuentran: HTML y CSS.

En la actualidad la programación de lado del cliente ha evolucionado para que el diseño de una página web sea agradable o amigable con el cliente. Se han desarrollado lenguajes de programación que se especializan en el diseño interactivo de páginas web entre los principales tenemos: Java Script, HTML5, CSS y CSS3.

En el caso de JavaScript, este lenguaje posee algunos framework que permiten insertar dentro de una página web contenida dinámico, este código por lo general esta embebido dentro del código HTML de la página o puede estar en un script y ser llamado desde la misma.

A continuación se muestra un gráfico de la estructura de una página HTML y un gráfico en que se puede observar el funcionamiento de un navegador Web.

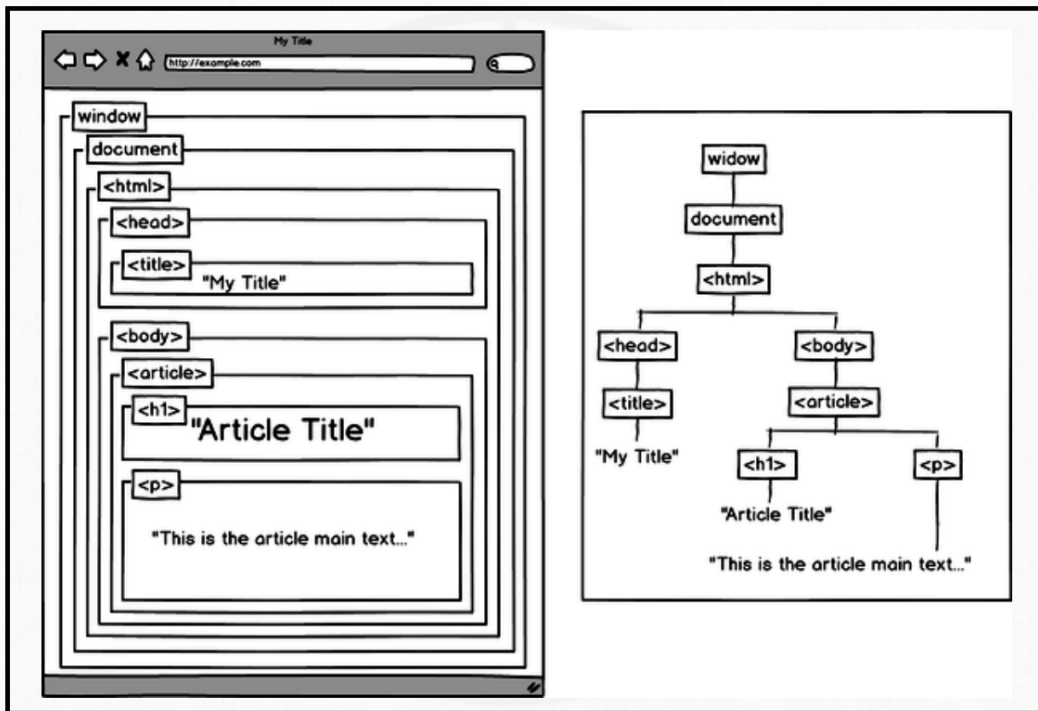


GRÁFICO 9: Estructura de Página Web en HTML.

Elaborado por: (Universidad Interamericana para el Desarrollo, Lenguajes De Programación Web, 2012)

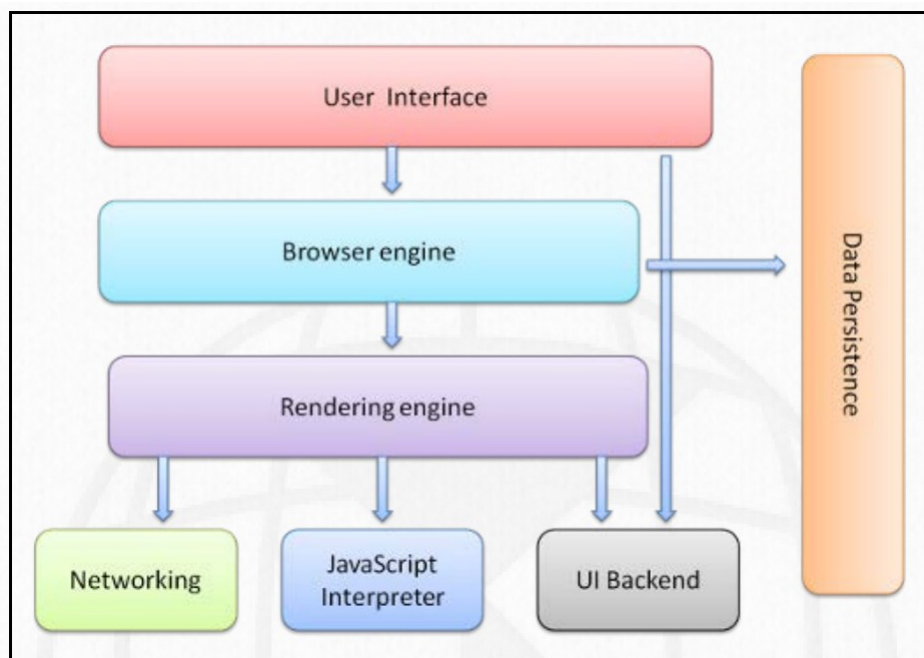


GRÁFICO 10: Funcionamiento de un Browser.

Elaborado por: (Universidad Interamericana para el Desarrollo, Lenguajes De Programación Web, 2012)

3.3.2. Ventajas y Desventajas del Lenguaje del Lado del Servidor.

Ventajas	Desventajas
<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> No es necesario un compilador.	<input type="checkbox"/> No posee un buen nivel de Seguridad.
<input type="checkbox"/> Es independiente de un servidor.	<input type="checkbox"/> El código puede ser objeto de plagio.
<input type="checkbox"/> No se necesita de un editor especial.	<input type="checkbox"/> El usuario puede modificar el código y la página deja de funcionar.
<input type="checkbox"/> Cualquier pc puede albergar cualquier pagina.	
<input type="checkbox"/> El mantenimiento es fácil y rápido, por parte del autor o de otro programador.	

Cuadro 8: Ventajas y Desventajas de Lenguajes de lado del Cliente.

Elaborado por: (Universidad Interamericana para el Desarrollo, Lenguajes De Programación Web, 2012)

3.3.3. Curva de Aprendizaje del Lenguaje de Lado del Cliente.

En la actualidad existe gran información sobre los principales lenguajes de lado del servidor, se puede encontrar una amplia gama de editores de texto que permiten diseñar una página web de forma visual si necesidad de escribir el código como se lo realizaba tradicionalmente, el aprendizaje de cualquier lenguaje de lado del cliente tiene un grado de dificultad menor comparándolo con los lenguajes de lado del servidor.

HTML es el principal lenguaje de lado del servidor lo único que se necesita para poder programar en este lenguaje es tener cierto conocimiento acerca de cada una de las etiquetas que posee, entre las ventajas que tiene respecto a los lenguajes de lado del servidor es el uso de un servidor web y no es necesario compilarlos ya que el encargado de esto es el Browser el mismo que interpreta el código escrito y lo despliega.

La principal utilidad de este tipo de lenguajes es desplegar información en páginas web del tipo portales o foros.

El tiempo que toma aprender a programar en este tipo de lenguaje es menor respecto a los lenguajes de lado del servidor, en HTML no se puede realizar una conexión a la base de datos es necesario incluir en el código HTML código JavaScript o PHP para poder realizar dicha acción.

Otra de las grandes ventajas de HTML es la interpretación por otros desarrolladores lo que permite realizar tareas de mantenimiento y modificación en menor tiempo.

3.4. Servidor de plataforma Xampp.

3.4.1. Características.

XAMPP es un servidor de software libre y multiplataforma, entre sus principales funcionalidades están: la gestión de Base de Datos MySQL, servidor web Apache y los intérpretes para lenguajes de script: PHP y Perl.

XAMPP es una herramienta de desarrollo que permite probar el código escrito en php o perl sin la necesidad de utilizar un servidor en la web.

Su instalación es muy fácil ya que esta versión funciona para cualquier versión de sistema operativo. En un principio XAMPP fue creado para probar páginas web en el computador de cada programador, pero en la actualidad se lo ocupa para alojar diferentes páginas web, XAMPP puede realizar esta actividad pero es recomendable tener cuidado en algunas configuraciones de la seguridad que este servidor no posee.

3.4.2. Ventajas y Desventajas.

Ventajas	Desventajas
<ul style="list-style-type: none">• Facilidad de Instalación y configuración• El paquete de instalación viene con MySQL y Apache.• Contiene buen material de información.	<ul style="list-style-type: none">• No soporta MySQL desde la consola.• No se pueden actualizar individualmente las versiones de los programas que instala.• Dificultad para configurar aplicaciones de terceros.

Cuadro 9: Ventajas y Desventajas de XAMP.

Elaborado por: (www.pnfiservidores.blogspot.com/2013/01/ventajas-y-desventajas-de-xampp.html, 2012)

3.4.3. Plataforma de gestión de bases de datos MySQL.

3.4.3.1. Introducción.

MySQL es un sistema de gestión de bases de datos relacional, multi hilo y multiusuario. Es uno de los gestores más importantes en el campo de diseño y programación de base de datos de tipo relacional. En la actualidad es la más utilizada en el mundo informático.

En cuanto a la historia de MySQL se empieza definiendo el significado de las siglas en ingles es My Structured Query Language o traducida al español Lenguaje de Consulta Estructurado.

MySQL se desarrolló a principios de la década de 1980 por programadores de IBM, quienes vieron la necesidad de contar con un código de programación que pudiera generar múltiples y extendidas bases de datos para empresas y organizaciones de diferente tipo.

En la actualidad MySQL es desarrollado y actualizado por Microsystem quienes han optado por poner a disposición una licencia para desarrolladores GPL y para las empresas una licencia Oracle.

3.4.3.2. Características de MySQL.

MySQL en sus primeras versiones carecía de elementos principales como: integridad referencial y transacciones. Esto no fue inconveniente ya que gracias al contenido dinámico y su simplicidad tuvo gran acogida entre los desarrolladores.

Al ser en un principio software libre y hoy en día pertenecer a Oracle muchos de estos elementos se fueron añadiendo a las versiones actuales, estos elementos fueron desarrollados tanto por desarrolladores internos de Oracle como los desarrolladores de software libre. Entre las principales características que se encuentran en las presentes versiones están:

- Amplio subconjunto del lenguaje SQL.
- Multiplataforma
- Selección de mecanismos de almacenamiento que proporcionan diferentes velocidades de operación, soporte físico, capacidad, distribución geográfica y transacciones.
- Uso de transacciones y claves foráneas.
- Conectividad segura.
- Replicación.
- Búsqueda e indexación de campos de texto.

CAPÍTULO 4. CASO DE ESTUDIO.

4.1. Fase de Planificación del Proyecto.

4.1.1. Toma y Análisis de Requerimientos utilizando la Metodología Áncora.

En el presente capítulo se utilizará la Metodología Áncora y sus herramientas anteriormente detalladas, cabe señalar para el desarrollo del presente sistema se acoplarán varias de las características de la metodología XP con Áncora.

En esta primera parte se realizará la toma, análisis y diseño con los requerimientos provistos por el cliente.

Para realizar esta actividad se seguirá paso a paso cada una de las características de Áncora como: el uso de Guiones, diálogos, pistas, escenas y utensilios.

El método utilizado para la toma de requerimientos fue una entrevista con el cliente, esta entrevista proporciona información valiosa para el análisis de factibilidad del presente sistema, esta información consta de antecedentes actuales sobre la manera en que la empresa y el cliente realizan los procesos a automatizar, además proporciona información sobre las necesidades que el cliente tiene y como satisfacer dichas necesidades con el sistema a desarrollar.

Siguiendo los pasos propuestos por la metodología Áncora y su ciclo de vida, lo primero que se definirá son los roles que detalla en la siguiente tabla:

Roles	Actividades
Analista de Requerimientos	Recolección y desarrollo de reporte de Requerimientos, tiene comunicación constante con el Cliente y Diseñador.
Involucrados (Cliente).	Provee de Información sobre la situación actual al analista de requerimientos.
Diseñador	Encargado de Diseñar una versión computacional de los Requerimientos.

Tabla 20: Roles y Actividades de Áncora.
Elaborado por: Carranza, J. Roles y Actividades, (2016)

Definidos los Roles y Actividades de cada uno es necesario definir los artefactos resultantes del análisis de requerimientos, los mismos que son: guiones y diálogos, al seguir el ciclo de vida de Áncora se deberá clasificar, ordenar jerárquicamente cada uno de los requerimientos y validarlos, finalmente se define que funcionalidades pueden ser objeto de reusó.

En la siguiente tabla Sumano (et.al.: 43) define y detalla cada uno de los artefactos que se obtendrán al aplicar el ciclo de vida de Áncora.

Fase	Artefacto
Entendimiento del Dominio y Contexto de la Aplicación.	Contexto y situación Actual.
Recolección y clasificación de Requerimientos	Propuesta Computacional
Resolución de conflictos, jerarquización y validación de requerimientos.	Propuesta computacional validada, observaciones y acuerdos.
Cierre	Documento de Análisis.
Reusó	Almacén de Sistemas Anteriores.

Tabla 21: Fase y Artefactos de Áncora.

4.1.2. Elaboración de Reporte de Requerimientos.

Para la elaboración del reporte de requerimientos se utilizarán guiones, a continuación, se muestra el guion de la situación actual de empresa.

Guion:	Escena: Ofertar Producto al Cliente.
Venta de Productos	El Vendedor oferta al cliente los servicios El gerente revisa si el producto existe en bodega.
Papeles: Vendedor Gerente Mecánico.	
Utensilios:	
Venta de Servicios Uso de productos Condiciones de Entrada	Escena: Realización de Servicio El gerente delega dicho servicio a un mecánico. El mecánico detalla los productos a utilizarse.
El vendedor oferta servicios al cliente sin tener información actualizada. El gerente tiene problemas al actualizar los datos del inventario. El mecánico tiene problemas al realizar servicio por falta de productos. Condiciones de Salida	Escena: Actualizar el Inventario. El gerente entrega productos al Mecánico.
El gerente actualiza el inventario. Servicios retrasados. Cliente insatisfecho.	El gerente actualiza datos del inventario manualmente.

Tabla 22: Guión de Gestión de Venta Actual.
Elaborado por: (Frenicentro,Situacion Actual, 2015)

En la siguiente tabla se diseña un guión sobre el control de inventarios que lleva en la actualidad la empresa. Ambos guiones facilitan el entendimiento del problema que el software deberá solucionar.

Guión:	Escena: Emisión de productos al Mecánico.
Control de Inventario	El Gerente controla si el producto se encuentra en Bodega. Si el producto existe, el Mecánico realiza el Servicio.
Papeles:	
Gerente. Mecánico.	
Utensilios:	Escena: Realización de Servicio
Venta de Servicios Uso de productos	El mecánico detalla los productos a utilizarse. El gerente actualiza manualmente los productos utilizados del inventario.
Condiciones de Entrada	Escena: Actualizar el Inventario. El gerente entrega productos solicitados por el Mecánico.
El vendedor oferta servicios al cliente sin tener información actualizada. El gerente tiene problemas al actualizar los datos del inventario. El mecánico tiene problemas al realizar servicio por falta de productos.	
Condiciones de Salida	
El gerente actualiza el inventario. Servicios retrasados. Cliente insatisfecho.	El gerente actualiza datos del inventario en forma manual.

Tabla 23: Guion de Gestión Actual de Control de Inventarios.
Elaborado por: (Carranza, J.,Gestion Actual, 2016)

Al analizar ambos guiones y después de haber tenido la entrevista con el cliente se obtiene el siguiente reporte con las potenciales funciones del sistema a desarrollar, las mismas que se detallan a continuación:

1. Ingreso de Clientes deberá cumplir con lo siguiente:

- Datos Personales.
- Datos del Automóvil.

2. Registro del Mantenimiento deberá constar con lo siguiente:

- Mano de Obra.
- Repuestos.
- Materiales.

3. Elaboración de Orden de Servicio.

4. Elaboración y emisión de Facturas en este deberán constar con los siguientes datos:

- Mano de Obra.
- Materiales.
- Tercerización (en caso de que el repuesto se obtenga de otra empresa).

5. Repuestos.

- Inventario el mismo que tendrá que controlar el stock y adquisición de nuevos repuestos.

6. Manejo de Inventario.

- Repuestos.
- Materiales.
- Insumos.
- Materiales de oficina.

7. Guardar Factura.

- Guardar factura por Cliente.
- Guardar registro de Factura.

8. Tipo de Pago

- Cheque.
- Tarjeta.
- Efectivo.

9. Factura a Crédito se deberá controlar el estado.

- Tiempo de Cancelación (fechas límite de cancelación).
- Cancelada.

4.1.3. Documento de Factibilidad.

De acuerdo con el ciclo de vida de Áncora, la Metodología XP y el análisis de los requerimientos que se detallaron anteriormente, en esta sección se clasificaran los requerimientos agrupándolos en potenciales funcionalidades del sistema, las mismas que se detallan en la siguiente Tabla continuación:

Funcionalidad	Actividades
Gestión de Clientes.	CRUD de Clientes.
Gestión de Proveedores.	CRUD de Proveedores.
Gestión de Automóvil.	CRUD de Automóvil.
Gestión de Inventarios.	Estado de stock de productos.
Gestión de Productos	CRUD de Productos.
Gestión de Empleados.	CRUD de Empleados.
Control de Inventarios.	Reporte sobre los productos más vendidos.
Gestión de Facturas.	Consulta de la Tabla Mecánico. Consulta de Inventario. Llenar información de la Factura. Consulta a la tabla Proveedores.

Tabla 24: Funcionalidades y Actividades a Desarrollar.

Elaborado por: (Carranza, J.,Funcionalidades a Desarrollar, 2016)

Con las funcionalidades anteriormente mencionadas queda por definir las prioridades de cada una para lo cual utilizaremos dos tipos que son: Prioridad Alta y Prioridad Baja. Para definir estas prioridades se tomará en consideración la actividad que cada una realiza dentro del sistema, la facilidad de codificarla y la prioridad que representa para el cliente, en la siguiente tabla se muestra la funcionalidad y su prioridad.

La prioridad Alta se dará a las funcionalidades principales y Baja para las funcionalidades como los reportes de datos estadísticos.

Funcionalidad	Prioridad
Gestión de Clientes.	ALTA
Gestión de Proveedores.	ALTA
Gestión de Automóvil.	ALTA
Gestión de Mecánicos.	ALTA
Gestión de Inventarios.	ALTA
Control de Inventarios.	BAJA
Gestión de Facturas.	ALTA

Tabla 25: Funcionalidades y Prioridades.

Elaborado por: (Carranza, J.,Prioridades, 2016)

4.2. Fase de Diseño.

4.2.1. Diagramas UML.

En el siguiente diagrama se muestra los requerimientos principales del sistema y se los clasifica por: prioridad, riesgo y estado.

La prioridad se clasifica de la siguiente manera 1 prioridad baja hasta 5 la más alta, el riesgo se clasifica en tres niveles: Alto, Medio y Bajo, finalmente el campo de estado corresponde si el requerimiento es aprobado para su implementación o requiere una mejor definición.

Para realizar esta clasificación, se tomó en cuenta la necesidad y prioridad del cliente y el análisis realizado en los puntos anteriores, también fue necesario un análisis sobre la curva de aprendizaje de las herramientas y metodologías a utilizar en el desarrollo del sistema.

	Descripción	Código	Prioridad	Workload	Riesgo	Estado
1	Ingreso de Clientes	REQ_0001	5		Medium	Aprobado
2	Ingreso de Empleados	REQ_0002			Medium	Aprobado
3	Ingreso de Proveedores	REQ_0003	5		Medium	Aprobado
4	Ingreso de Servicios	REQ_0004	5		Medium	Aprobado
5	Ingreso de Automóviles	REQ_0005	5		Medium	Aprobado
6	Ingreso de Productos	REQ_0006	5		Medium	Aprobado
7	Elaboración de Facturas	REQ_0007	3		Medium	Definido
8	Seguimiento de Facturas a Cobrar	REQ_0008	4		High	Definido
9	Control de Usuarios	REQ_0008	4		Medium	Definido

Diagrama 1: Diagrama UML de Requerimientos.

Elaborado por: (Carranza, J., Prioridades, 2016)

4.2.2. Diagramas de Caso de Uso.

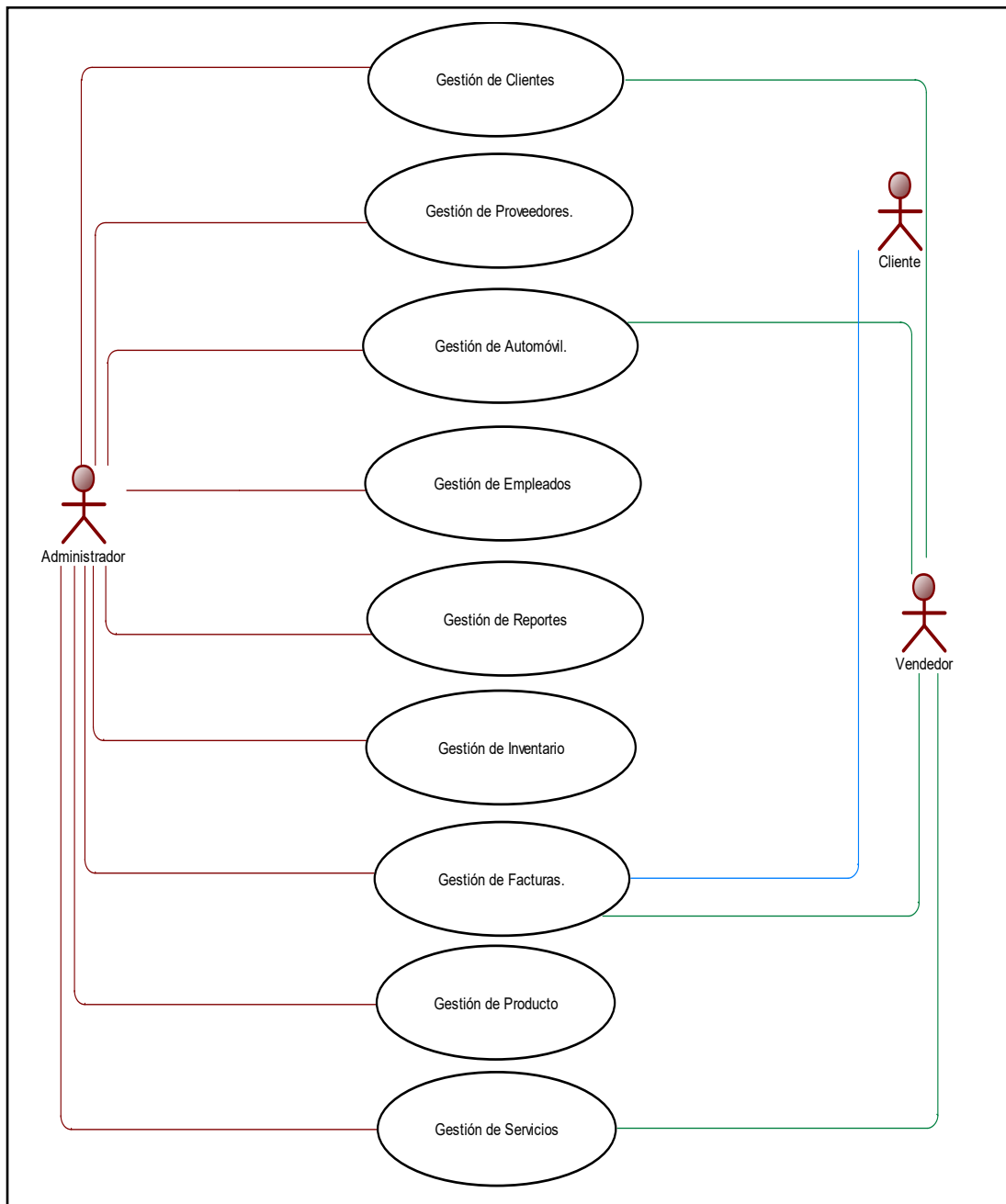
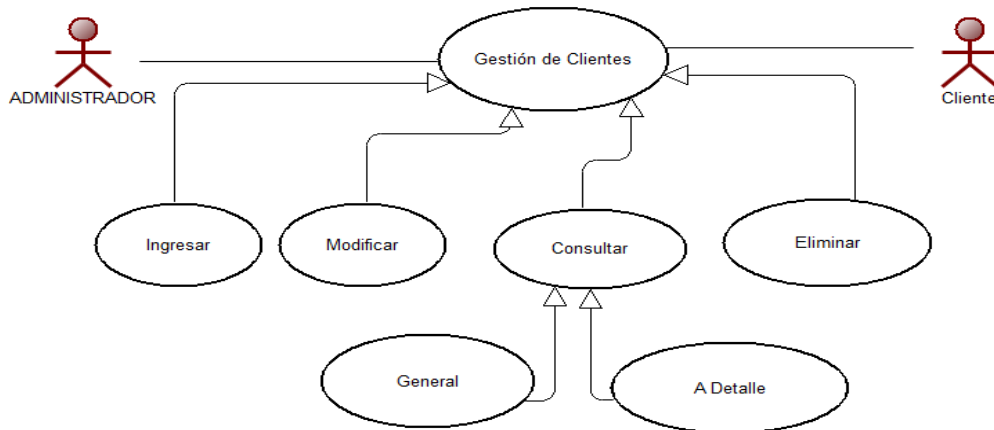


Diagrama 2: Caso de Uso.

Elaborado por: (Carranza, J., Casos de Uso, 2016)

Para la fase de diseño se han realizado los diseños de todas las funcionalidades pero presentarlo todo sería muy extenso por esta razón se presentaran las funcionalidades más representativas y el restos de funcionalidades serán agregadas en el anexo 1 el mismo que será agregado en un CD.

4.2.2.1. Nivel de Abstracción Gestión de Clientes.



➤ **A Detalle: Interrelación entre el Usuario y el sistema.**

✓ **Ingresar Clientes.**



Actores: Administrador

1. Flujo Principal.

- a) El sistema muestra la pantalla principal.
- b) El actor pulsa sobre el Menú Clientes.
- c) El actor selecciona la opción ingresar
- d) El sistema muestra una pantalla de ingreso.
- e) El actor ingresa Cedula del Cliente.
- f) El sistema verifica la Cedula (E1)
- g) El actor llena los demás campos.
- h) El actor presiona guardar
- i) El sistema guarda los datos.

2. Flujo Alterno.

- a) Ingresar al Caso Consultar o Modificar.

3. Excepciones.

E1: Causa: El Cliente ya ha sido registrado con anterioridad.

Solución: Ingresar un nuevo cliente con otra cedula.

✓ **Modificar Clientes.**



Actores: Administrador

1. Flujo Principal.

- a) El sistema muestra la página principal.
- b) El actor pulsa sobre el menú Clientes.
- c) El actor escoge la opción Administrar.
- d) El sistema muestra la pantalla de Administración.
- e) El actor pulsa sobre el botón Actualizar.
- f) El actor modifica los campos necesarios.
- g) El actor pulsa sobre el botón actualizar. (E1)
- h) El sistema actualiza la información del cliente.

2. Excepciones.

E1: Causa: La cedula fue modificada, o algún campo obligatorio no fue ingresado.

Solución: Ingresar verificar campos obligatorios.

✓ **Consultar Clientes.**



Actores: Administrador

1. Flujo Principal.

- a) El sistema muestra la página principal
- b) El actor pulsa sobre la opción Clientes.
- c) El actor escoge la opción Administrar.

- d) El sistema le muestra la página de Administración.
- e) El actor selecciona un cliente
- f) El actor pulsa sobre la opción ver.
- g) El sistema le muestra los datos del Cliente seleccionado.

✓ **Eliminar Clientes.**



Actores: Administrador

1. Flujo Principal.

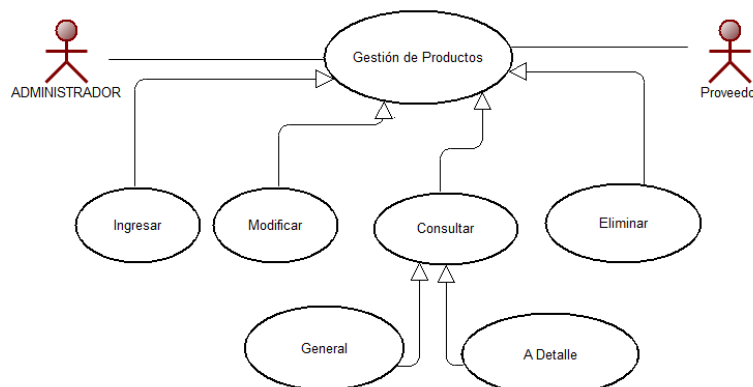
- a) El sistema muestra la página principal.
- b) El actor pulsa sobre el Menú Clientes.
- c) El actor selecciona la Opción Administrar.
- d) El sistema muestra la página de Administración.
- e) El actor selecciona un cliente.
- f) El actor pulsa sobre el botón Eliminar.
- g) El sistema verifica que el cliente no tenga Facturas (E1).
- h) El sistema elimina la información del Cliente.

2. Excepciones.

E1: Causa: El Cliente tiene generada una factura, o tiene una factura en proceso.

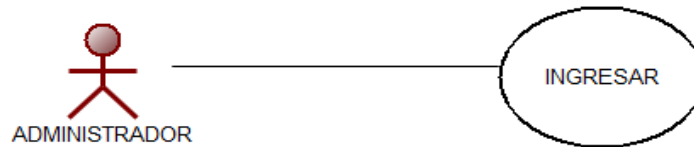
Solución: El cliente no se puede eliminar y solo se le puede cambiar de estado Activo a Inactivo.

4.2.2.2. Nivel de Abstracción Gestión de Productos.



➤ **A Detalle: Interrelación entre el Usuario y el sistema.**

✓ **Ingresar Productos.**



Actores: Administrador

1. Flujo Principal.

- a) El sistema muestra la pantalla principal.
- b) El actor pulsa sobre el Menú Productos.
- c) El actor selecciona la opción ingresar.
- d) El sistema muestra una pantalla de ingreso.
- e) El actor ingresa el código del proveedor.
- f) El sistema verifica el código (E1).
- g) El actor selecciona proveedor del producto.
- h) El actor ingresa cantidad del producto.
- i) El actor presiona guardar
- j) El sistema guarda los datos del producto y del inventario.

2. Flujo Alterno.

- a) Ingresar al Caso Consultar o Modificar.

3. Excepciones.

E1: Causa: El producto ya ha sido registrado con anterioridad.

Solución: Consultar si el producto está en estado activo.

✓ **Modificar Productos.**



Actores: Administrador

1. Flujo Principal.

- a) El sistema muestra la página principal.
- b) El actor pulsa sobre el menú Productos.
- c) El actor escoge la opción Administrar.
- d) El sistema muestra la pantalla de Administración.
- e) El actor selecciona un producto.
- f) El actor pulsa sobre el botón Actualizar.
- g) El sistema le muestra una página con la información del producto.
- h) El actor modifica los campos necesarios.
- i) El actor pulsa sobre el botón actualizar. (E1)
- j) El sistema actualiza la información del producto.

2. Excepciones.

E1: Causa: Los campos obligatorios no han sido completados.

Solución: No modificar el RUC y llenar todos los campos.

✓ **Consultar Productos.**



Actores: Administrador

1. Flujo Principal.

- a) El sistema muestra la página principal
- b) El actor pulsa sobre la opción productos.
- c) El actor escoge la opción Administrar.
- d) El sistema le muestra la página de Administración.
- e) El actor selecciona un producto.
- f) El actor pulsa sobre la opción ver.
- g) El sistema le muestra los datos del producto seleccionado.

✓ **Eliminar Productos.**

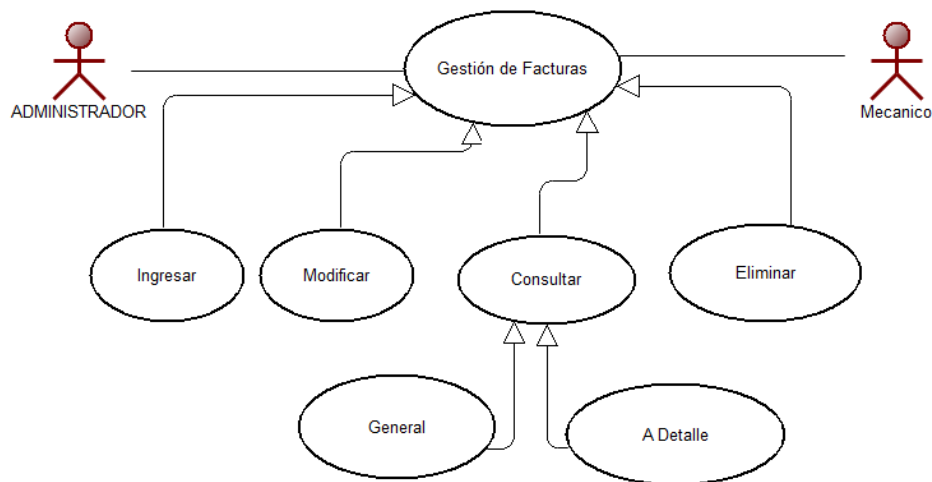


Actores: Administrador

1. Flujo Principal.

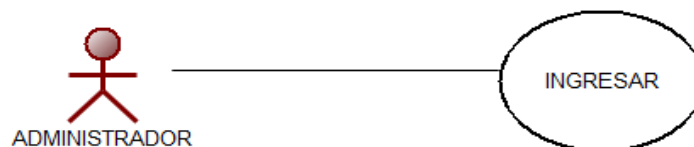
- a) El sistema muestra la página principal.
- b) El actor pulsa sobre el Menú Productos.
- c) El actor selecciona la Opción Administrar.
- d) El sistema muestra la página de Administración.
- e) El actor selecciona un producto.
- f) El actor pulsa sobre el botón Eliminar.
- g) El sistema elimina la información del producto.

4.2.2.3. Nivel de Abstracción Gestión de Facturas.



➤ A Detalle: Interrelación entre el Usuario y el sistema.

✓ Ingresar Factura.



Actores: Administrador

1. Flujo Principal.

- a) El sistema muestra la pantalla principal.
- b) El actor pulsa sobre el Menú Facturas.
- c) El actor selecciona la opción ingresar.
- d) El sistema muestra una pantalla de ingreso.

- e) El actor ingresa el código de la Factura.
- f) El sistema verifica el código (E1).
- g) El actor selecciona un cliente.
- h) El actor selecciona la fecha de Emisión.
- i) El actor presiona guardar.
- j) El sistema guarda los datos de la factura.
- k) El sistema le muestra una pantalla de ingreso de Detalles de la Factura.
- l) El actor selecciona los servicios.
- m) El actor pulsa en agregar.
- n) El sistema agrega los productos al servicio.
- o) El actor selecciona el servicio agregado pulsa sobre modificar.
- p) El Sistema le muestra una pantalla con los productos que tiene ese servicio.
- q) El actor selecciona cada uno de los productos y pulsa sobre modificar cantidad.
- r) El sistema le muestra una pantalla para el ingreso de cantidad.
- s) El actor ingresa la cantidad.
- t) El sistema verifica la cantidad. (E2).
- u) El sistema pulsa agregar.
- v) El sistema almacena la información.
- w) El actor pulsa sobre Agregar más Servicios.
- x) El sistema le muestra la página con los servicios ingresados.
- y) El actor pulsa sobre el botón Excel.
- z) El sistema exporta la factura en formato Excel.

2. Flujo Alterno.

- a) Ingresar al Caso Consultar o Modificar.

3. Excepciones.

E1: Causa: La factura ya ha sido ingresado al sistema.

Solución: Consultar si la factura se encuentra en estado activo o facturado.

E2: Causa: La cantidad de producto no puede ser ingresada porque es mayor a la cantidad de stock.

Solución: Consultar el nivel de inventario.

✓ **Modificar Factura.**



Actores: Administrador

1. **Flujo Principal.**

- a) El sistema muestra la página principal.
- b) El actor pulsa sobre el menú Facturas.
- c) El actor escoge la opción Administrar.
- d) El sistema muestra la pantalla de Administración.
- e) El actor selecciona una Factura.
- f) Si la factura ya ha sido generada el sistema le mostrará la información y no podrá ser modificada.
- g) Si la factura está en estado Activo.
- h) El sistema le muestra una página con la información de la factura.
- i) El actor modifica los campos de la Factura.
- j) El actor pulsa sobre el botón actualizar. (E1)
- k) El sistema le muestra una pantalla de ingreso de Detalles de la Factura.
- l) El actor selecciona los servicios.
- m) El actor pulsa en agregar.
- n) El sistema agrega los productos al servicio.
- o) El actor selecciona el servicio agregado pulsa sobre modificar.
- p) El Sistema le muestra una pantalla con los productos que tiene ese servicio.
- q) El actor selecciona cada uno de los productos y pulsa sobre modificar cantidad.
- r) El sistema le muestra una pantalla para el ingreso de cantidad.
- s) El actor ingresa la cantidad.
- t) El sistema verifica la cantidad. (E2).
- u) El sistema pulsa agregar.
- v) El sistema almacena la información.
- w) El actor pulsa sobre Agregar más Servicios.
- x) El sistema le muestra la página con los servicios ingresados.
- y) El actor pulsa sobre el botón Excel.
- z) El sistema exporta la factura en formato Excel.

2. Excepciones.

E1: Causa: Los campos obligatorios no han sido completados.

Solución: No se debe modificar el código de la Factura.

E2: Causa: La cantidad de producto no puede ser ingresada porque es mayor a la cantidad de stock.

Solución: Consultar el nivel de inventario.

✓ Consultar Factura.



Actores: Administrador

1. Flujo Principal.

- El sistema muestra la página principal
- El actor pulsa sobre la opción factura.
- El actor escoge la opción Administrar.
- El sistema le muestra la página de Administración.
- El actor selecciona una factura.
- El actor pulsa sobre la opción ver.
- El sistema le muestra los datos de la factura.

✓ Eliminar Factura.



Actores: Administrador

1. Flujo Principal.

- El sistema muestra la página principal.
- El actor pulsa sobre el Menú Facturas.
- El actor selecciona la Opción Administrar.
- El sistema muestra la página de Administración.
- El actor selecciona una factura.
- El actor pulsa sobre el botón Eliminar.(E1)
- El sistema actualiza el estado de la Factura.

2. Excepciones.

E1: Causa: La factura está en estado Facturado.

Solución: Si la factura está en estado Activo puede ser eliminada, caso contrario el sistema le envía mensajes de error.

4.2.2.4. Puntos a tomar en cuenta

Para el desarrollo del presente sistema se tomó en cuenta los siguientes aspectos, la empresa como tal no tiene una cantidad de productos determinada para cada servicio, además cada servicio tiene comportamiento dinámico para la utilización de productos, esto quiere decir que cada servicio puede ocupar n productos y cada producto puede utilizar n cantidad. Para el diseño del sistema se utilizarán tablas débiles para relacionar los servicios-productos y facturas-Servicios.

En el diseño no se muestra el funcionamiento del Inventario ya que esto se actualizará automáticamente cada vez que se registre o se actualice un producto.

Al igual que el inventario el ingreso de usuarios al sistema se lo realiza cada vez que un empleado sea registrado.

El uso de estados para cada servicio y factura se aplicará para no perder información. Dentro de los estados tenemos:

<i>Estado</i>	Descripción.
<i>Activo</i>	Se aplica una vez que se ingresa un producto, servicio, proveedor, empleado o factura.
<i>Inactivo</i>	Se aplica cuando se ha eliminado un producto, servicio, proveedor, empleado o factura.
<i>Facturado</i>	Este estado se modifica una vez que se ha generado la factura y se lo aplica a la factura y a los servicios que en ella se encuentran.

Tabla 26: Tabla de Estados del Sistema
Elaborado por: (Carranza, J., Estados del Sistema, 2016)

4.2.3. Diagrama de Secuencia.

Para el presente sistema se presentan los diagramas de secuencia para los siguientes casos de uso: Gestión de Clientes, Gestión de Empleados, Gestión de Factura y Gestión de Productos.

4.2.3.1. Gestión de Clientes.

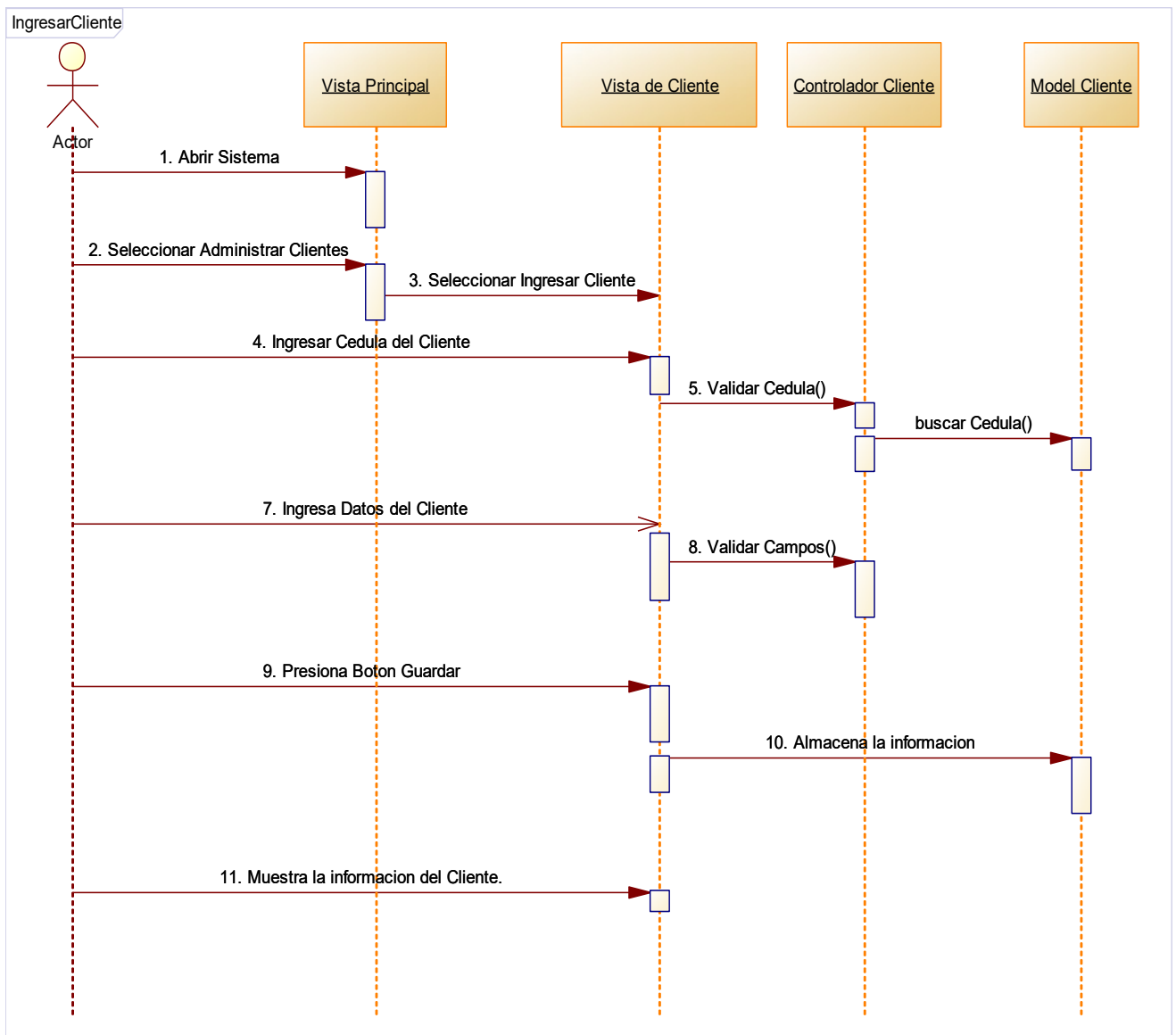


Diagrama 3: Diagrama de Secuencia para el Ingreso de Clientes.

Elaborado por: (Carranza, J.,Diagrama de Secuencia Gestion de Clientes, 2016)

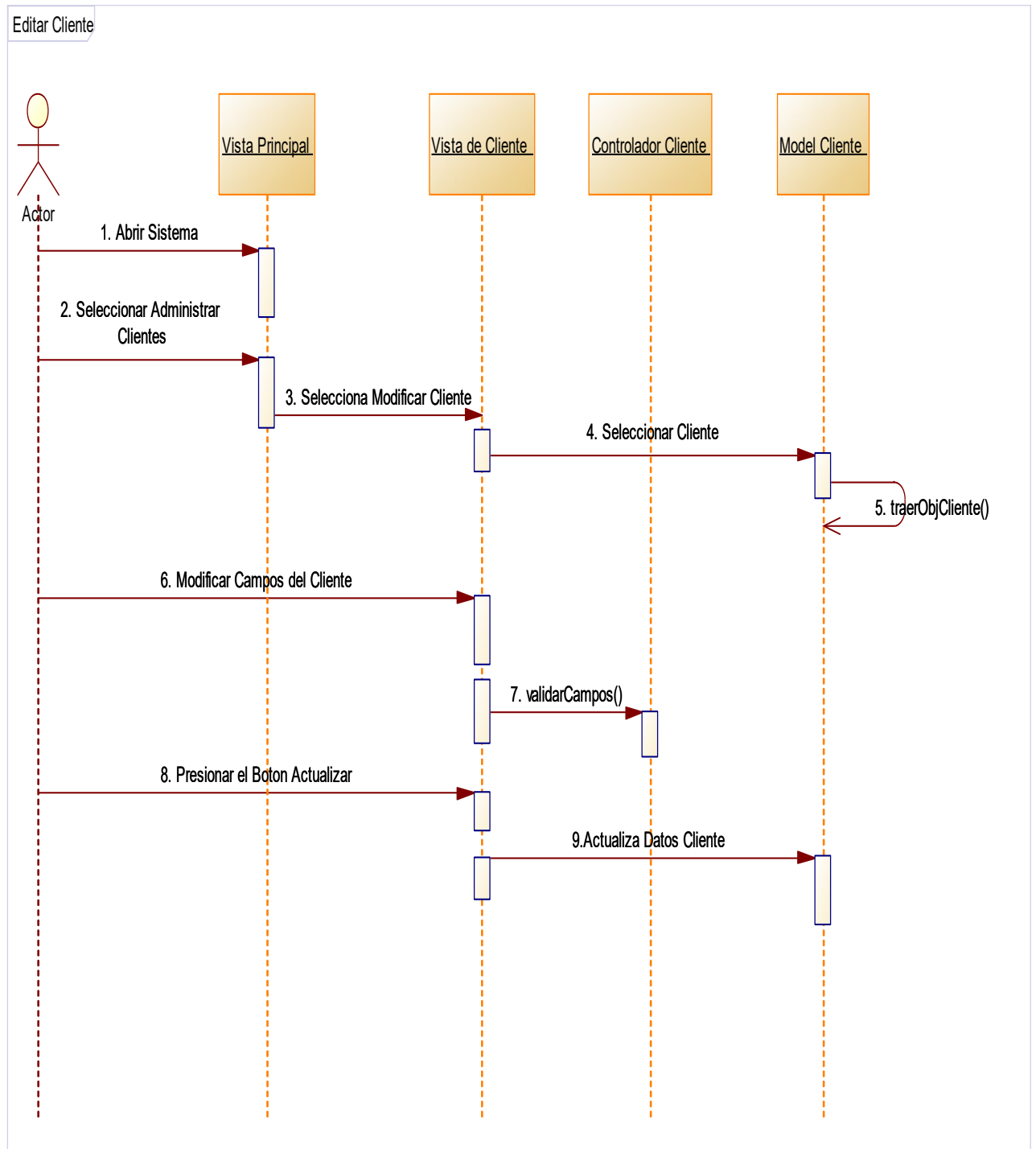


Diagrama 3: Diagrama de Secuencia para Editar un Cliente.
 Elaborado por: (Carranza, J., Diagrama de Secuencia Gestion de Clientes, 2016)

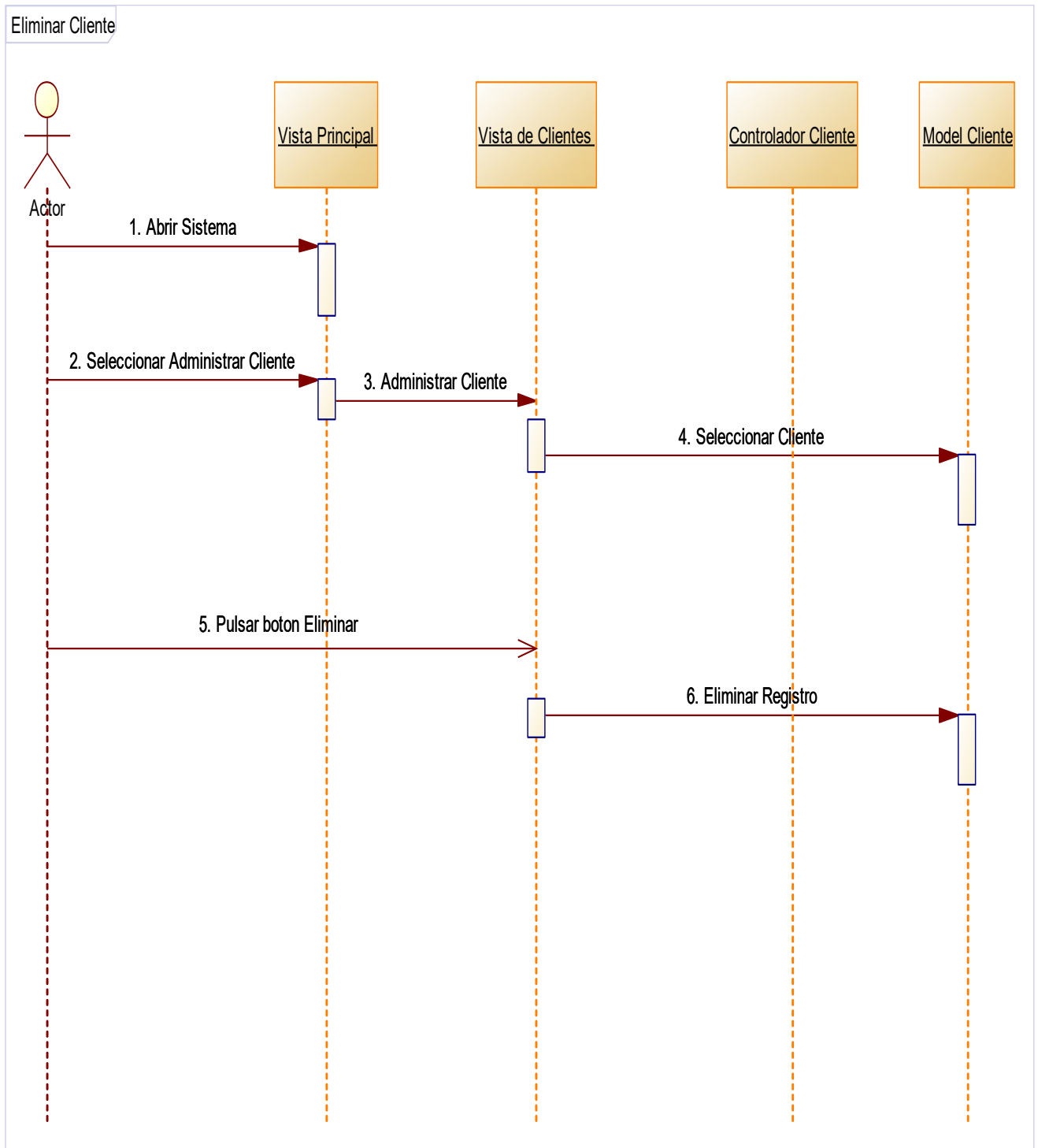


Diagrama 4: Diagrama de Secuencia para Eliminar un Cliente.
 Elaborado por: (Carranza, J., Diagrama de Secuencia Gestion de Clientes, 2016)

4.2.3.2. Gestión de Productos.

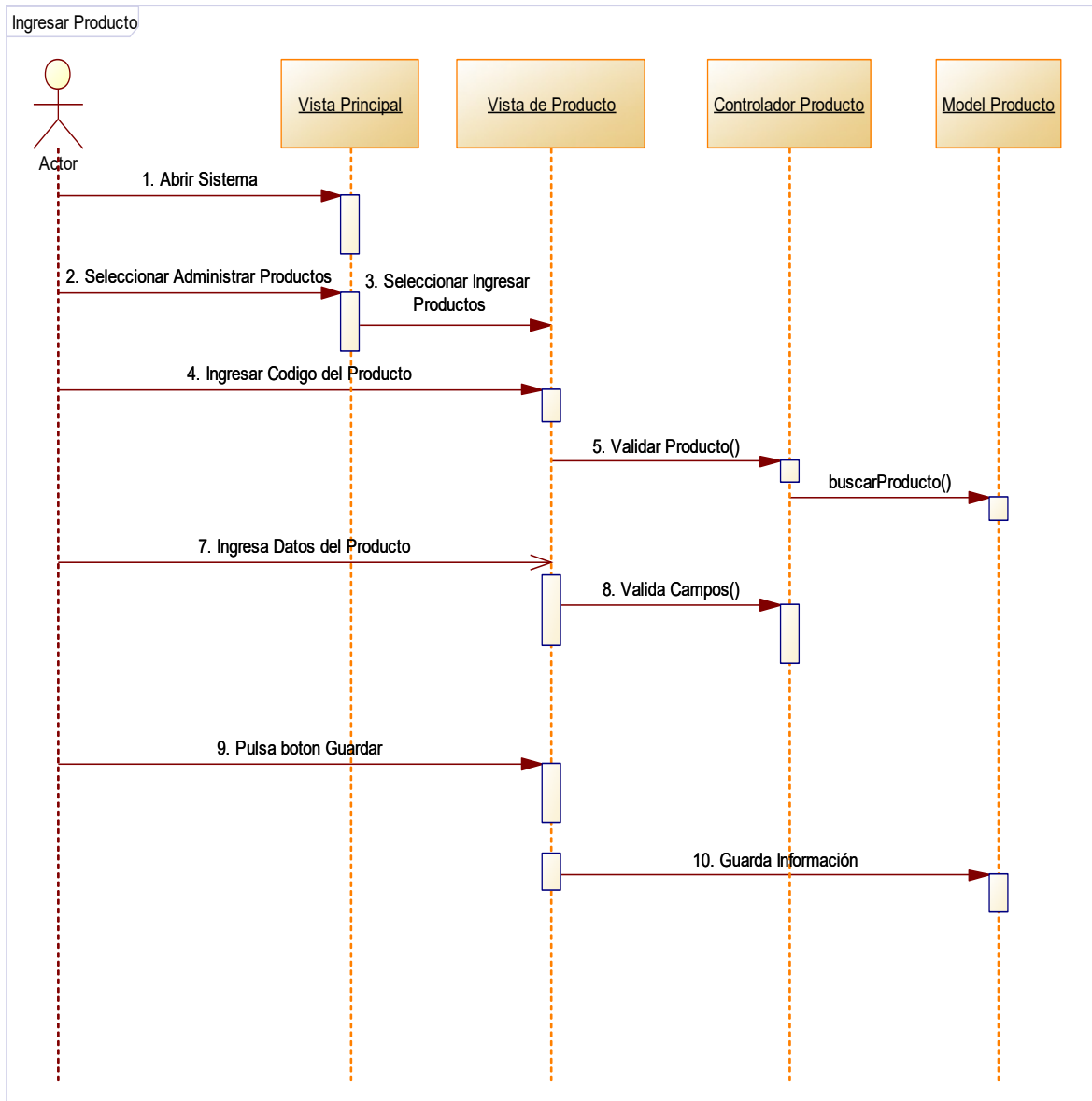


Diagrama 5: Diagrama de Secuencia para el Ingreso de Productos.
 Elaborado por: (Carranza, J., Diagrama de Secuencia Gestion de Productos, 2016)

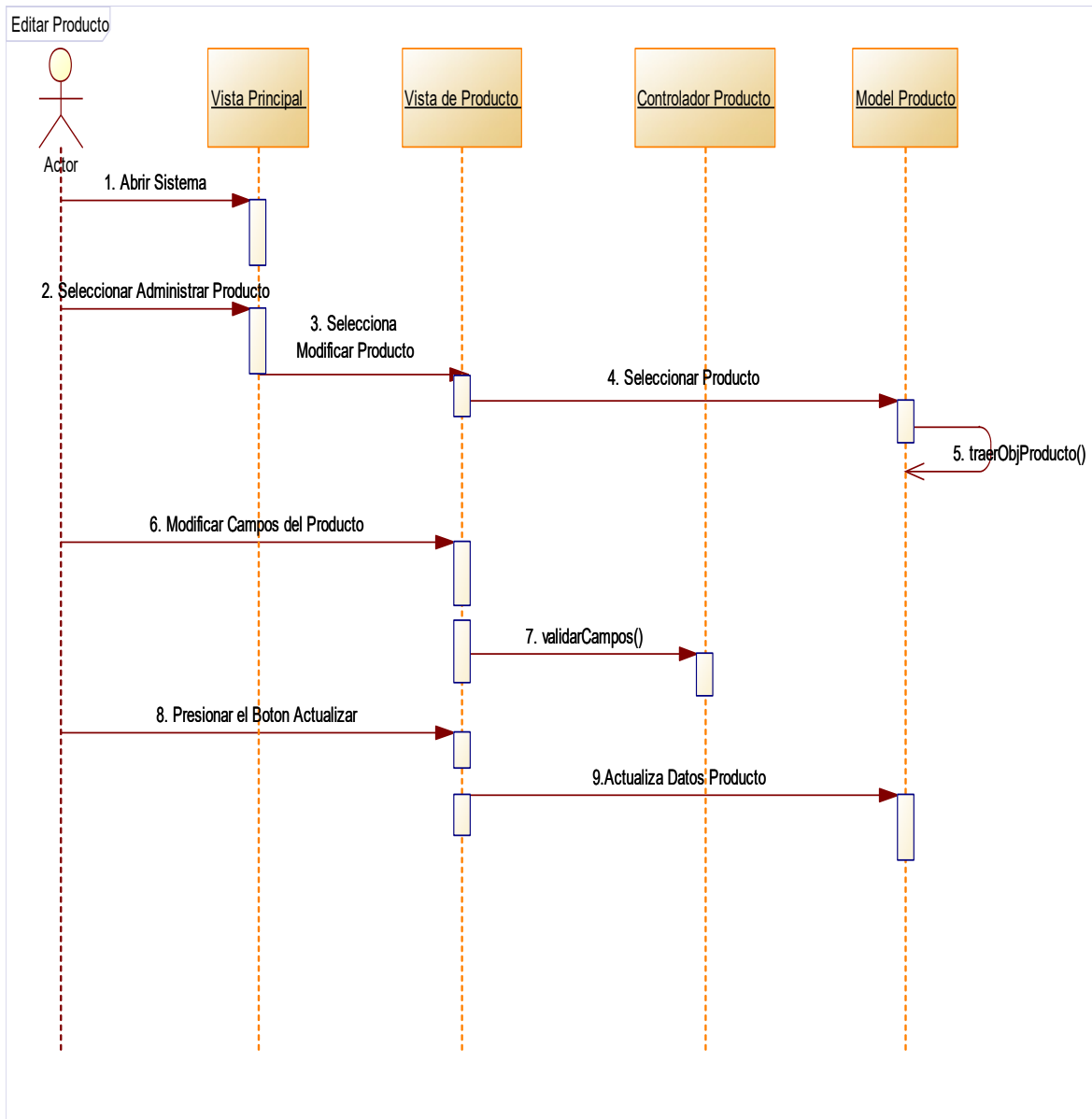


Diagrama 6: Diagrama de Secuencia para Modificar datos del Producto.
 Elaborado por: (Carranza, J., Diagrama de Secuencia Gestion de Productos, 2016)

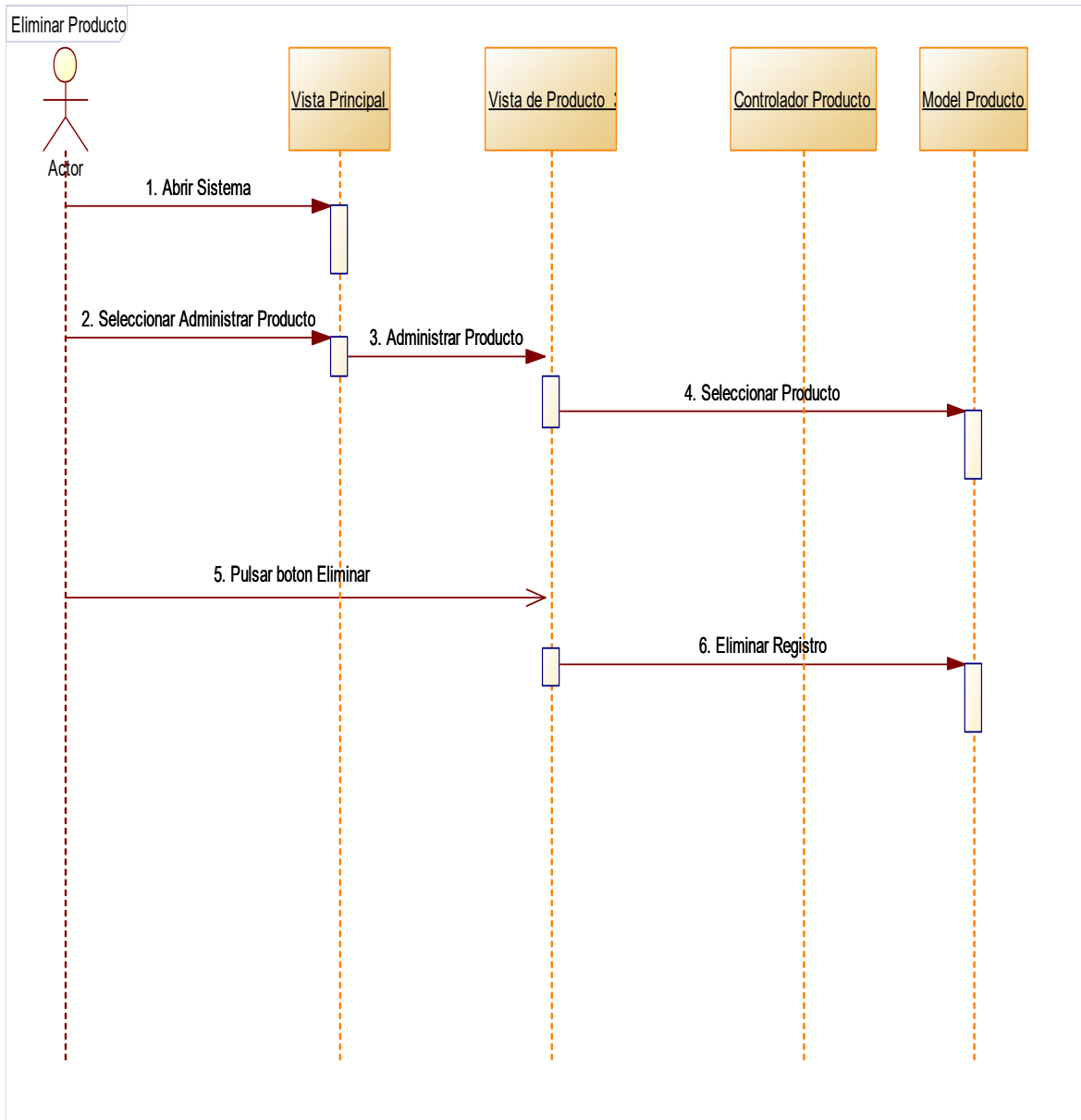


Diagrama 7: Diagrama de Secuencia para Eliminar Productos.
 Elaborado por: (Carranza, J., Diagrama de Secuencia Gestion de Productos, 2016)

4.2.3.3. Gestión de Facturas

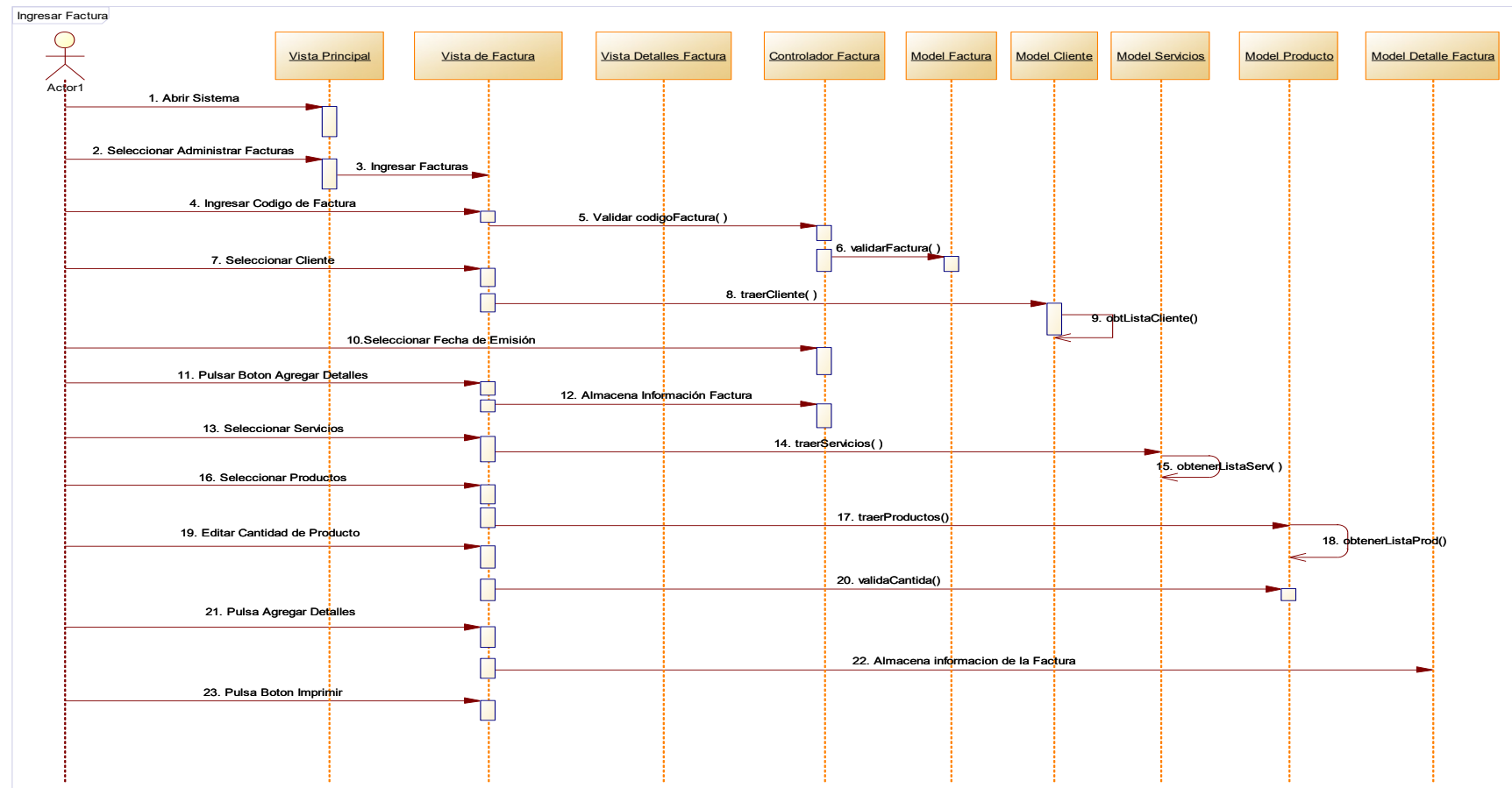


Diagrama 8: Diagrama de Secuencia para el Ingreso de Factura.
 Elaborado por: (Carranza, J., Diagrama de Secuencia Gestion de Facturas, 2016)

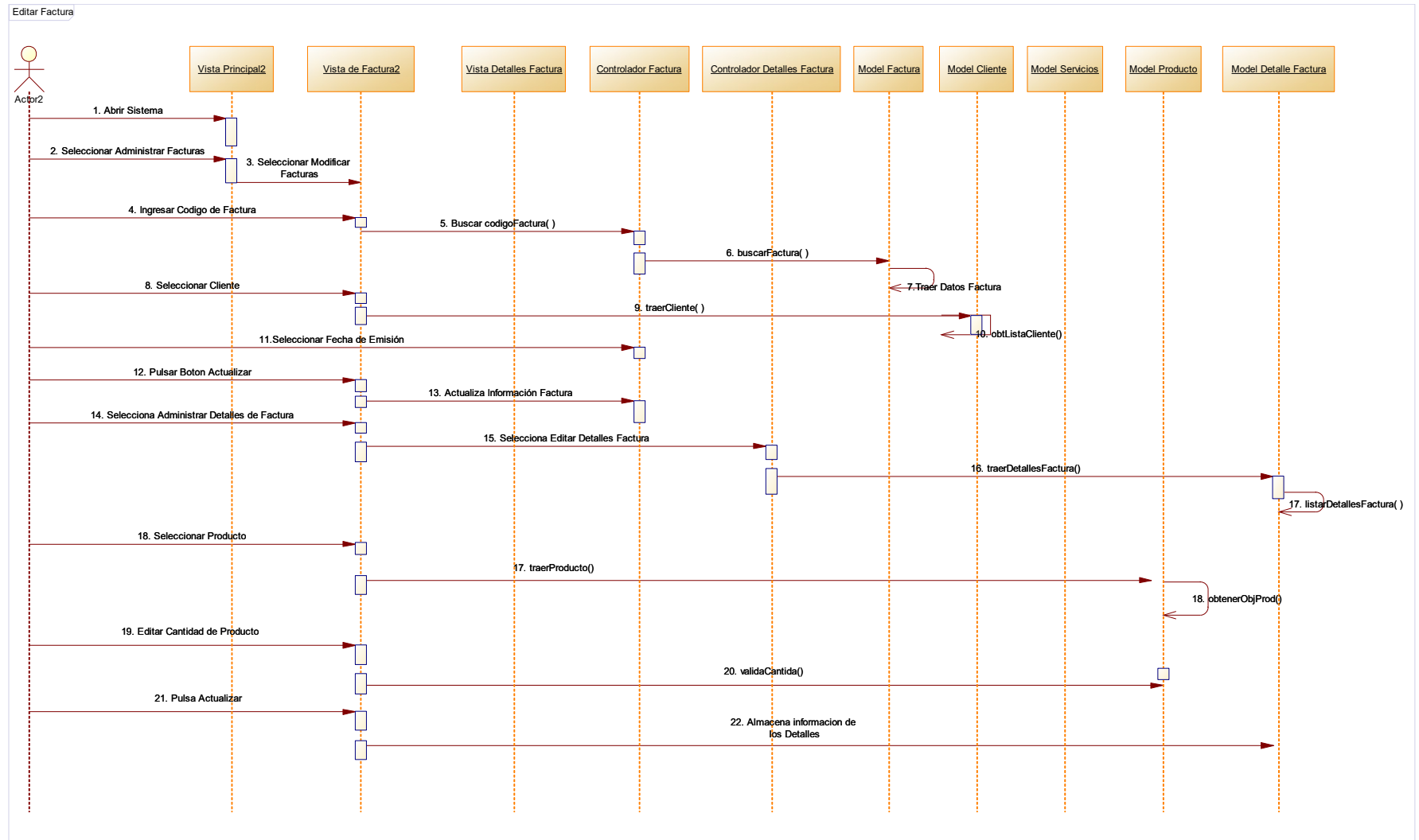


Diagrama 9: Diagrama de Secuencia para Modificar datos de la Factura.
 Elaborado por: (Carranza, J., Diagrama de Secuencia Gestion de Facturas, 2016)

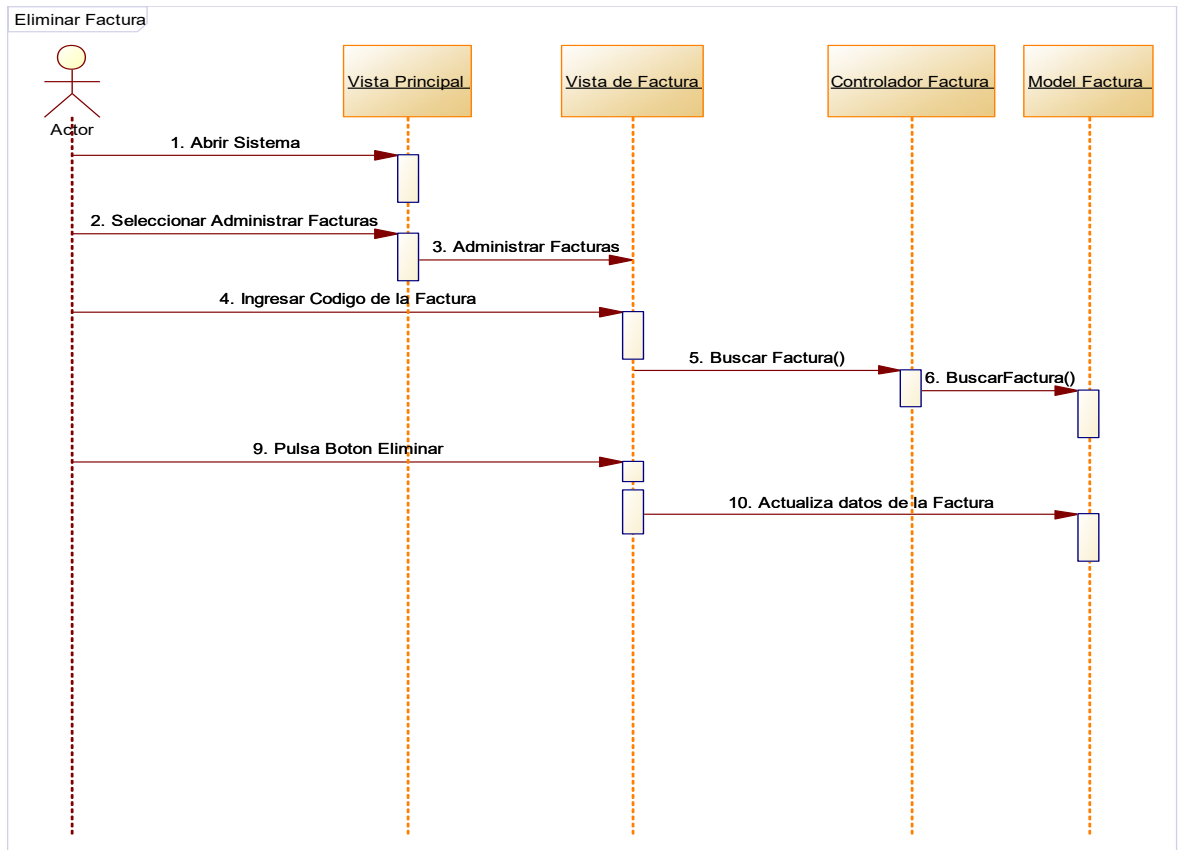


Diagrama 10: Diagrama de Secuencia para Eliminar Facturas.
 Elaborado por: (Carranza, J., Diagrama de Secuencia Gestion de Facturas, 2016)

4.2.4. Diagrama de Clases

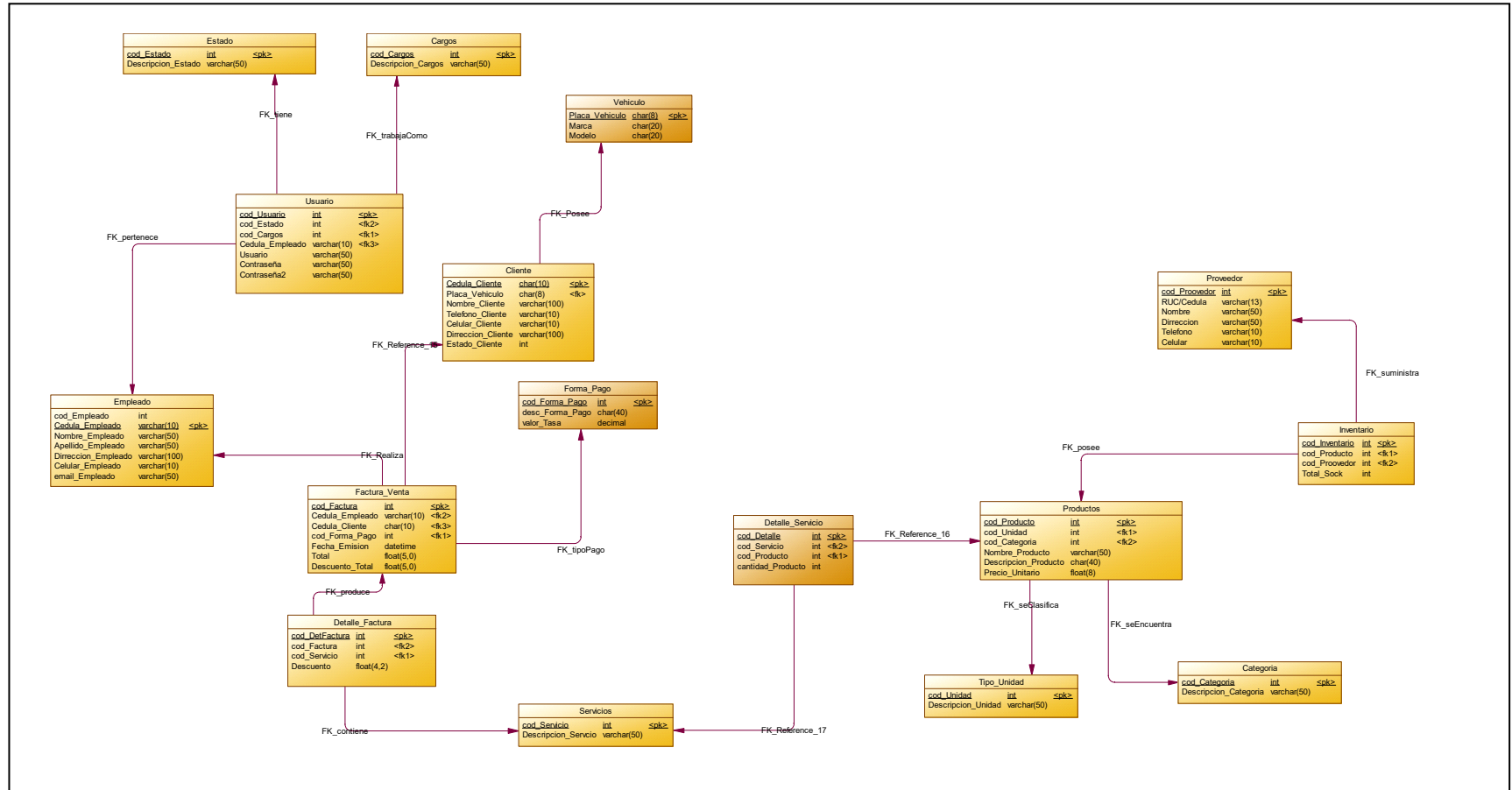


Diagrama 11: Diagrama de Clases.
Elaborado por: (Carranza, J.,Diagrama de Clases, 2016)

4.2.5. Diagrama de Base de Datos.

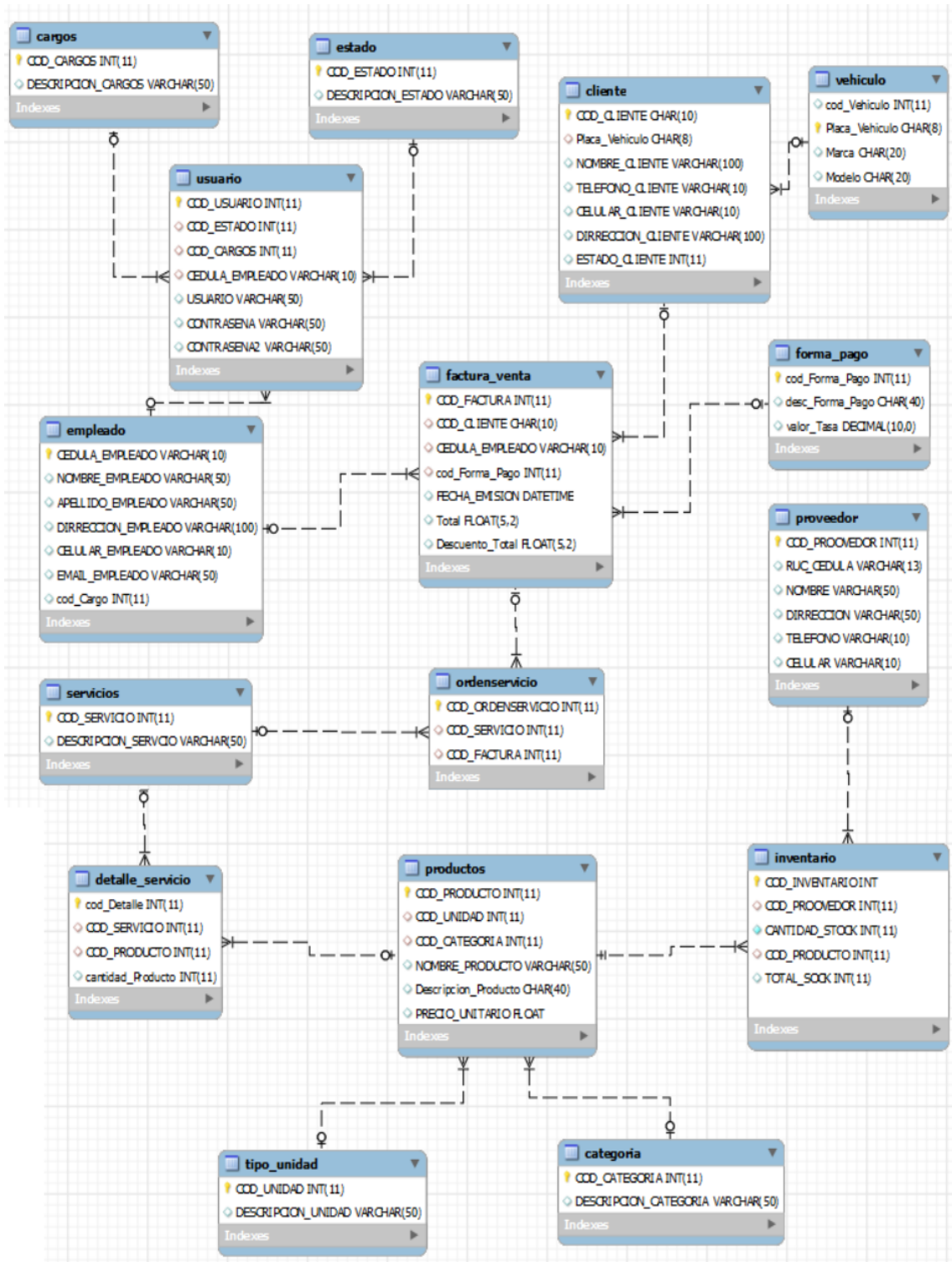


Diagrama 12: Diagrama de Base de Datos.
Elaborado por: (Carranza, J., Diagrama de Base de Datos, 2016)

4.2.6. Diccionario de Datos.

Para la elaboración del Diccionario de datos se utilizó la herramienta de Reportes contenida en el Case de desarrollo Power Designer.

4.2.6.1. Lista de Tablas.

Nombre	Codigo
Cargos	CARGOS
Categoria	CATEGORIA
Cliente	CLIENTE
Detalle_Factura	ORDENSERVICIO
Detalle_Servicio	Detalle_Servicio
Empleado	EMPLEADO
Estado	ESTADO
Factura_Venta	FACTURA_VENTA
Forma_Pago	Forma_Pago
Inventario	INVENTARIO
Productos	PRODUCTOS
Proveedor	PROVEEDOR
Servicios	SERVICIOS
Tipo_Unidad	TIPO_UNIDAD
Usuario	USUARIO
Vehículo	Vehículo

Tabla 27: Lista de tablas de Base de Datos.

Elaborado por: (Carranza, J., Lista de Tablas de Base de Datos, 2016)

4.2.6.2. Lista de Columnas

Nombre	Codigo
cod_Estado	COD_ESTADO
Descripción_Estado	DESCRIPCIÓN_ESTADO
cod_Cargos	COD_CARGOS
Descripción_Cargos	DESCRIPCIÓN_CARGOS
cod_Proveedor	COD_PROVEEDOR
RUC/Cédula	RUC_CÉDULA
Nombre	NOMBRE
Dirreccion	DIRRECCIÓN
Teléfono	TELEFONO
Celular	CELULAR
cod_Usuario	COD_USUARIO
cod_Estado	COD_ESTADO
cod_Cargos	COD_CARGOS
Cédula_Empleado	CÉDULA_EMPLEADO
Usuario	USUARIO
Contraseña	CONTRASENA
Contraseña2	CONTRASENA2
Cédula_Cliente	COD_CLIENTE
Placa_Vehiculo	Placa_Vehiculo
Nombre_Cliente	NOMBRE_CLIENTE
Teléfono_Cliente	TELÉFONO_CLIENTE
Celular_Cliente	CELULAR_CLIENTE
Dirrección_Cliente	DIRRECCIÓN_CLIENTE
Estado_Cliente	ESTADO_CLIENTE
cod_Empleado	COD_EMPLEADO
Cedula_Empleado	CÉDULA_EMPLEADO
Nombre_Empleado	NOMBRE_EMPLEADO
Apellido_Empleado	APELLIDO_EMPLEADO
Dirreccion_Empleado	DIRRECCIÓN_EMPLEADO
Celular_Empleado	CELULAR_EMPLEADO
email_Empleado	EMAIL_EMPLEADO
cod_Factura	COD_FACTURA
Cedula_Cliente	COD_CLIENTE
Cedula_Empleado	CÉDULA_EMPLEADO
cod_Forma_Pago	cod_Forma_Pago
Fecha Emisión	FECHA EMISIÓN
Total	Total
Descuento_Total	Descuento_Total
cod_DetFactura	COD_ORDENSERVICIO
cod_Servicio	COD_SERVICIO
cod_Factura	COD_FACTURA
cod_Servicio	COD_SERVICIO
Descripción_Servicio	DESCRIPCIÓN_SERVICIO
cod_Categoría	COD_CATEGORÍA

Descripcion_Categoria	DESCRIPCION_CATEGORIA
cod_Producto	COD_PRODUCTO
cod_Unidad	COD_UNIDAD
cod_Categoria	COD_CATEGORIA
Nombre_Producto	NOMBRE_PRODUCTO
Descripcion_Producto	Descripcion_Producto
Precio_Unitario	PRECIO_UNITARIO
cod_Inventario	CANTIDAD_STOCK
cod_Proovedor	COD_PROOVEDOR
cod_Producto	COD_PRODUCTO
Total_Sock	TOTAL SOCK
cod_Unidad	COD_UNIDAD
Descripcion_Unidad	DESCRIPCION UNIDAD
cod_Forma_Pago	cod_Forma_Pago
desc_Forma_Pago	desc_Forma_Pago
valor_Tasa	valor_Tasa
cod_Vehiculo	cod_Vehiculo
Placa_Vehiculo	Placa_Vehiculo
Marca	Marca
Modelo	Modelo
cod_Detalle	cod_Detalle
cod_Servicio	COD_SERVICIO
cod_Producto	COD_PRODUCTO
cantidad_Producto	cantidad_Producto

Tabla 28: Lista de Datos de la Base de Datos.
Elaborado por: (Carranza, J., Lista de columnas , 2016)

4.2.6.3. Lista de Referencias

Nombre	Código	Tabla Padre	Tabla Hijo
Contiene	contiene	Servicios	Detalle_Factura
Pertenece	pertenece	Empleado	Usuario
Posee	posee	Productos	Inventario
Posee	Posee	Vehiculo	Cliente
Produce	produce	Factura_Venta	Detalle_Factura
Realiza	Realiza	Empleado	Factura_Venta
Reference_15	Reference_15	Cliente	Factura_Venta
Reference_16	Reference_16	Productos	Detalle_Servicio
Reference_17	Reference_17	Servicios	Detalle_Servicio
Se Clasifica	seClasifica	Tipo_Unidad	Productos
Se encuentra	seEncuentra	Categoria	Productos
Suministra	suministra	Proveedor	Inventario
Tiene	tiene	Estado	Usuario
tipoPago	tipoPago	Forma_Pago	Factura_Venta
Trabaja como	trabajaComo	Cargos	Usuario

Tabla 29: Lista de relaciones de Base de Datos.
 Elaborado por: (Carranza, J., Lista de Referencias , 2016)

4.3. Fase de Codificación.

Para la fase de codificación del presente sistema se aplicó ciertos puntos que propone la metodología como seguir estándares de programación, integración del cliente al proyecto e integración continua del código.

El desarrollo del sistema se lo hizo en dos ciclos ya que el primer ciclo sirvió para Desarrollar pequeños prototipos los mismos que fueron presentados al cliente en reuniones cortas como propone la metodología Ancora, dando como resultado la integración del cliente al proyecto y la mejora de la definición de los requerimientos del sistema, el sistema en el primer ciclo de desarrollo tenía funcionalidad modular esto quiere decir que las funcionalidades se ejecutaban independientemente.

Para el segundo ciclo la principal meta será integrar las diferentes funcionalidades dando como resultado un sistema integrado que cumpla con los requerimientos del cliente, para esto se realizará una integración por módulos ya probados, con la del framework la integración no presentará mayores problemas. Por ejemplo, para la elaboración de la factura se tendrá que relacionar dos o más tablas.

La meta al final del segundo ciclo será tener un sistema funcional e integrado, y aplicando lo que nos dice la metodología XP una vez terminado este ciclo se buscara optimizar el código y realizar una prueba total del sistema revisando y corrigiendo errores que puedan ocurrir.

El sistema se desarrolló en Yii Framework de php esta herramienta proporciona una programación que sigue el MVC, la misma que permite tener control absoluto en términos de orden y estándares de programación permitiendo que cualquier desarrollador pueda actualizar o modificar el código, cumpliendo así con otra norma que plantea la Metodología XP.

El código fuente, el diseño de la base de datos y la base de datos se encuentra como anexo en el CD.

Además del código fuente en el CD se anexa un acta de Requerimientos que se plantearon antes de desarrollar el sistema y los motivos por los cuales no se desarrollaron esta acta se encuentra en el Anexo 2 y la solicitud de cambio de requerimientos también se encuentran detalladas en el anexo 3.

A continuación, se detallaran de mejor forma como se desarrolló el sistema, los errores que pudieron presentarse durante la integración y los cambios que se han realizado.

4.3.1. Integración de Sistema.

Como se explicó en la parte anterior para la fase de integración se explicara el proceso de integración de las funcionalidades anteriormente explicadas, para esta fase el sistema tiene todas las partes probadas y en correcto funcionamiento individual.

A continuación, se detallan las principales funciones integradas y como se llegó a integrar a nivel de base de datos y programación, para no hacerlo muy extenso en esta sección se centrará en las funcionalidades como: Gestión de Productos, Gestión de Servicios y Gestión de Facturas.

En esta fase dentro del desarrollo del sistema se explicará los problemas que se presentaron al integrar funcionalidades y que soluciones se dieron para solucionarlos.

Aplicando la metodología XP y sus principales propiedades antes de integrar una funcionalidad se tuvo que hacer una prueba del sistema comprobando que la versión antes de la integración estaba en perfecto funcionamiento una vez realizada la integración se debe realizar que el sistema no se vea afectado, aprobando así la nueva versión.

Para el presente proyecto cabe mencionar que no se aplicó totalmente la metodología ya que XP exige la programación en parejas y la integración continua, esto quiere decir que una pareja es la encargada de integrar continuamente el código validado por el resto de miembros del equipo.

4.3.2. Desarrollo de Funcionalidades del Sistema.

4.3.2.1. Integración de la Funcionalidad Gestión de Productos.

a) Proceso.

Para el desarrollo de esta funcionalidad se tomó en cuenta el diseño de la base de datos y las relaciones que intervienen.

Primero se desarrolló por separado, la función de Gestión de Productos consta de: ingreso de productos, proveedor del producto y cantidad en inventarios.

Por esta razón se implementó por separado cada una de las funcionalidades que intervienen, una vez probada cada una de ellas se integraron dando como resultado la actualización de datos en la Base de datos.

La relación entre productos e inventario fue un punto en el que se puso mucha atención ya

que la actualización del stock de inventario se lo debe hacer en tiempo real, por esta razón se tuvo que rediseñar la base de datos y realizar ciertas validaciones a nivel de programación y base de datos.

Al empezar a codificar nos facilitó mucho la herramienta del Framework ya que por medio de esta se obtuvo un código robusto y ordenado, la principal limitante fue el aprendizaje de cómo realizar la actualización de datos en dos tablas.

b) Defectos Encontrados.

Una de las principales limitaciones que se tuvieron que solventar fue la inserción al mismo tiempo en dos o más tablas.

La inserción de datos repetidos o claves primarias repetidas este problema se presentaba siempre que se insertaba un nuevo registro.

La inserción de datos vacíos el sistema mostraba mensaje error y el sistema finalmente se interrumpía.

El control de las diferentes acciones que cada usuario puede o no ejecutar dentro del sistema.

c) Soluciones.

La principal solución fue consultar el manual del Framework, como aplicar las validaciones de campos, el mostrar mensajes por pantalla y la inserción de datos en dos tablas diferentes.

Yii framework proporciona dentro de su código la opción de validar campos por tipo de dato o datos repetidos, esto facilitó en gran manera la forma de programar ahorrando en gran medida el tiempo de codificación de estas validaciones y optimizando las pruebas después de cada integración.

4.3.2.2. Integración de la Funcionalidad Gestión de Servicios.

a) Proceso.

Para el desarrollo de esta funcionalidad se tuvo varios inconvenientes principalmente en la fase de diseño, codificación e integración ya que se tomó en cuenta como realiza este proceso el Taller Mecánico Frenicentro, esto ocasiono que se rediseñe varias veces y la codificación se realizó como más demoras que las que se esperaban aumentando así el tiempo planificado.

El proceso de crear un servicio es dinámico ya que el Taller Mecánico Frenicentro no tiene una lista de servicios y la cantidad de productos que ocupan para cada uno de ellos no se estima, esto ocasionó que el sistema tenga la opción de añadir de manera dinámica la inserción de productos en los diferentes servicios y la modificación de la cantidad de cada uno.

Como se puede ver en el diseño de la base de datos para esta funcionalidad se utilizó tablas débiles para poder hacer la modificación de cada servicio.

En la parte grafica que se detallara más adelante se tuvo que realizar un proceso recursivo.

b) Defectos Encontrados.

El desarrollo de esta funcionalidad requirió una gran inversión de tiempo de diseño, codificación e integración ya que se probó varias formas de implementarlo, y al final se desarrolló una forma más larga pero eficiente.

Al rediseñar a nivel de diseño el sistema sufrió daños a nivel de código ya que las funcionalidades que anteriormente se probaron tuvieron que ser probadas nuevamente y por lo tanto se extendió el tiempo estimado para esta fase.

c) Soluciones.

Para evitar nuevas modificaciones del sistema se realizaron varias reuniones con el cliente para llegar a un acuerdo para esto se realizaron prototipos funcionales de esta parte del sistema.

4.3.2.3. Integración de la Funcionalidad Gestión de Facturas.

a) Proceso.

Esta funcionalidad es la más importante del sistema ya que en esta intervienen las siguientes tablas: cliente, servicio, detalle servicio y detalle factura. A parte de estas tablas esta funcionalidad tiene gran incidencia en el inventario ya que, al generar la factura, el sistema resta del inventario la cantidad de productos utilizados.

Se debe tomar en cuenta que el proceso de factura a pedido del cliente la factura será exportada en dos formatos: Pdf y Excel para posteriormente ser impreso.

Para generar una factura se tomó en cuenta el procesos de crear un servicio ya que básicamente este proceso es parecido, al igual que en la creación de servicios en este proceso se utilizara una tabla débil y la inserción tendrá que seguir el siguiente proceso:

1. Ingresar los datos de la Factura.
2. Ingresar los datos del Cliente.
3. Ingresar los detalles que tendrá la factura: en este proceso se deberá ingresar los servicios a facturar.
4. Modificar la cantidad de los productos que se encuentran dentro de cada producto.
5. Exportar la factura en pdf o Excel.
6. El sistema modifica la cantidad de cada producto en el inventario.
7. El sistema modifica el estado de la factura: en la fase de codificación se optó por utilizar diferentes estados para la factura entre los cuales se encuentran:

- Activo: este estado se le da a la factura que va hacer generada.
- Inactivo: este estado se modifica cada vez que una factura que ha estado en estado Activo ha sido eliminada.
- Facturado: este estado se aplica a la factura que ya ha sido generada y exportada.

b) Defectos Encontrados.

El principal defecto que se encontró fue la actualización del inventario, ya que se tomó en cuenta una validación respecto a la cantidad del producto en stock.

La función de exportar a pdf causo varios problemas ya que el momento cuadrar los datos exportados con la factura física los datos no coincidían

Existen ciertos problemas el momento de imprimir ya que el Taller Mecánico ocupa una impresora matricial y se tuvo que modificar los bordes del documento pdf.

La función de exportar a Excel causó varios problemas ya que el momento de implementar se tuvo que utilizar librerías de php el momento de utilizarlas el sistema sufrió cierta inconsistencia y se tuvo que configurar dichas librerías, al cuadrar los datos exportados con la factura física los datos no coincidían por lo que se tuvo que utilizar una plantilla proporcionada por el Taller mecánico Frenicentro y basarse en esa para que los datos finalmente se cuadren con la factura.

c) Soluciones.

Para solucionar el problema de la impresión de la factura se tuvo que calcular el número de pixeles utilizado en el Excel e implementarlos en el código para que finalmente la impresión coincidiera con la factura física.

Para la parte de validar la cantidad de productos utilizados en cada servicio se realizó una validación el momento de ingreso de la cantidad.

4.4. Validación de la Versión del Sistema.

Para la validación del sistema se realizaron pruebas comprobando que el ingreso y procesamiento de los datos sea el correcto.

Siguiendo la metodología XP una vez desarrollada la integración se realizó una prueba funcional, para la parte de la factura se realizó una prueba en la cual se probó desde el ingreso de datos, modificación del inventario, validación de cantidad de productos e impresión de factura.

Este proceso se lo hizo con el acompañamiento del usuario final como lo señala la metodología XP, esto permitió que el cliente se familiarice y acepte los cambios que se realizaron y además de acuerdo a las pruebas que se realizaron se mejoró en gran medida la consistencia del sistema.

4.5. Desarrollo de la Interfaz de Usuario.

Herramientas Utilizadas.

Para el diseño de la interfaz de usuario se realizaron varios prototipos que se presentaron al cliente, y de acuerdo al desarrollo se fue cambiando la interfaz a preferencia del usuario.

Para diseñar con efectos y detalles web se utilizaron las siguientes herramientas y estándares:

- JQuery.
- Hojas de Estilos CSS.
- Tema Bootstrap.
- Responsive Design.
- HTML.
- Algunos efectos proporcionados por Yii Framework.

Pasos a seguir para el diseño de una interfaz de usuario.

• Según el libro “Pensar Primero” de Daniel Mordecki, se define 6 pasos para la creación de una interfaz de usuario amigable estos pasos se detallan a continuación:

Diseño de la Interacción.

En el libro se describen los 5 objetivos que se persiguen para tener un diseño de la interacción los mismos que se listan a continuación:

- Definir el producto final: en esta etapa se debe tener una visión sobre el alcance del producto definiendo así cuando estará completo.
- Acotar y minimizar los costos: en este punto se debe tomar en cuenta que el costo en muchos de los casos no se dispara por una falta en la estimación, sino porque se desarrolló una solución muy distinta a la que se planeó en un comienzo.
- Tener en cuenta al usuario en todo el desarrollo del sistema.
- No presionar al equipo con el tema de diseñar bien y más bien centrarse en el funcionamiento de la aplicación.
- Hacer cronogramas que se puedan cumplir tomando en cuenta todos los aspectos que

pueden afectar el cumplimiento del mismo.

Personajes y Reparto.

En este punto el autor menciona como ejemplo las reuniones que se tiene durante el desarrollo de un sistema en la cual participan: diseñadores, programadores, gerentes, etc.

También propone la eliminación del término usuario, dando como resultado la aparición de personajes principales y secundarios.

Los personajes principales son definidos como usuarios finales del sistema mientras que los personajes secundarios son personas que no son usuarios finales, pero tiene su relevancia dentro del desarrollo.

Se debe tener que, si existen más de 5 o 6 personajes, entre principales y secundarios lo más probable es que el sistema tenga fallas o se estén realizando mal ciertas cosas.

Objetivos

Como objetivo el autor define que se debe dar el paso entre el concepto de software que hace cosas al software que ayuda a quien lo usa para conseguir sus objetivos.

También inserta un concepto sobre el termino diseñar, lo define como:³“encontrar dentro del espectro de objetivos de nuestros personajes el nivel justo de detalle que los comprenda y pueda ser descrito de forma útil para desarrollar la aplicación”.

También menciona que el objetivo principal de los usuarios es jugar el rol de equilibrar y amalgamar las funcionalidades del software en un todo único.

Escenarios

Como se describen en el libro los escenarios cumplen la función de vincular los objetivos de los usuarios con las acciones concretas que desarrollan al utilizar la aplicación.

Además, propone clasificar cada una de las tareas por nivel de importancia puntuado cada una de ellas del 1 al 5 y dividiéndolas de la siguiente manera:

Nivel de Importancia: el nivel de importancia que tiene la tare para alcanzar el objetivo.

Nivel de Dificultad de implementación: pondera el esfuerzo de diseño y programación que hay que poner en una determinada tarea.

Mordecki, Daniel, pág. 98

Frecuencia: representa el número de veces que se repite una tarea determinada esto tiene una importancia vital para el diseño.

4.6. Especificar el diseño de un sitio web

Una vez que se tiene definidos los objetivos, personajes y escenarios se puede pasar al diseño, para esto se plantean dos alternativas que son el esqueleto estático o el storyBoard.

Esqueleto Estático: no es más que una combinación de páginas que simulan el posible comportamiento que tendrá el sistema, es importante el no confundir este esqueleto con un prototipo, ya que un prototipo no es completo y un esqueleto si tiene el 100% de funcionalidades y contiene el 100% de páginas web.

Este esqueleto servirá de maqueta para los diseñadores gráficos los mismos que al especificar la interacción y pondrán pasar dicha maqueta a los programadores.

Story Board es la captura en imágenes de cada una de las páginas del sitio web, este paso suele ser más fácil de crear o modificar pero su explicación hacia el usuario suele tener ciertas complicaciones, ya que no se puede ver el funcionamiento.

Usabilidad

Es entender como los usuarios ven al software y si los mensajes que este arroja pueden guiar en las decisiones que el usuario toma, el objetivo que tiene la usabilidad es analizar el nivel de coincidencia que un sistema o sitio web tiene con la representación mental del usuario.

También se plante realizar pruebas o test a persona ajenas al sistema lo que permite conocer si el sistema tiene fallas respecto a su navegación con este tipo de pruebas se puede determinar errores que no se tenían en frente y no se tomaron en cuenta durante el tiempo de desarrollo.

Prototipos.

Para mejorar la experiencia y familiaridad del usuario con el sistema se realizaron varios prototipos del sistema mismos que se muestran y detallan a continuación:

En la siguiente imagen se puede observar el primer prototipo del sistema el mismo que contienen una pantalla de inicio de sesión y un menú principal despegable en la parte izquierda de la pantalla que permite al usuario manipular las distintas funcionalidades.



Ingreso al Sistema

*Ingrese sus Credenciales

*Usuario: josca1221@hotmail.com

*Contraseña:

Mantener mi Sesión Activa

Iniciar Sesión

GRÁFICO 11: Pantalla de Ingreso del Prototipo.

Elaborado por: (Carranza, J., Pantalla Ingreso Prototipo 1 , 2016)



GRÁFICO 12: Menú desplegable del Prototipo 1.

Elaborado por: (Carranza, J., Menu Prototipo 1 , 2016)

El objetivo de este primer prototipo es lograr una funcionalidad óptima de cada una de las funcionalidades del sistema.

Para el prototipo final y siguiendo los ciclos de desarrollo que propone la metodología XP se desarrolló un prototipo más amigable al usuario, se mantuvieron ciertas características y otras se cambiaron como se muestra en las siguientes imágenes:



GRÁFICO 13: Pantalla de Ingreso del Prototipo 2.

Elaborado por: (Carranza, J., Menu Prototipo 1 , 2016)



GRÁFICO 14: Barra de menú superior del Prototipo 2.

Elaborado por: (Carranza, J., Menu Prototipo 2 , 2016)

La pantalla de inicio de sesión o login se mantuvo, la pantalla principal del sistema se modificó el menú despegable ubicado en la parte izquierda de la pantalla por un menú estático en la parte superior.

Al realizar esta modificación se mejoró en gran medida la usabilidad por parte del usuario ya que se evitó el mostrar tantos menús como en el primer prototipo, se agruparon en sub menús dentro de cada funcionalidad.

➤ Problemas encontrados.

Los principales problemas que se encontraron fueron:

Menú despegable y recarga de la página: al utilizar un menú tipo acordeón, se presentaba un inconveniente ya que el usuario seleccionaba una opción y el sistema le mostraba la pantalla correspondiente a esa opción, el menú acordeón regresaba a su estado original. Esto producía en el usuario cierta frustración ya que no sabía que acción había realizado.

Exceso de opciones en la barra del menú: El menú tipo de acordeón almacenaba muchos menús que en ciertos casos podrían ser agrupados, al tener un exceso de menús, la barra de menú no podía ser visualizada correctamente, esto ocasionaba que el usuario tenga que recurrir a desplazarse hacia arriba o abajo para poder utilizar el sistema.

Mensajes de alerta sin funcionalidad: El sistema no mostraba mensajes de error sobre las acciones que se ejecutaron.

➤ **Soluciones.**

En el prototipo final del sistema se solucionaron la mayoría de defectos que se tuvieron en el primer prototipo, esto ayudo a que el usuario se familiarice con el sistema.

En esta versión se aplicaron ciertos estándares como la creación de una barra de menús en la parte superior, además del envío de mensajes de acción al ejecutar una acción.

➤ **Versión Final.**

Con los cambios realizados en el primer prototipo, este segundo prototipo presentado al cliente para su uso y aprobación, como este era el último ciclo que se iba a realizar, los errores que se encontraron fueron corregidos en ese instante.

Una vez que el cliente se encontró satisfecho se dio por concluido el ciclo, esto dio como resultado el sistema actual mismo que se encuentra como anexo en el CD.

4.7. Fase de Pruebas e Implantación.

Para realizar las pruebas del sistema se utilizar un software especializado denominado sonar que y sonar scanner, además se debe tomar en cuenta que para realizar la fase de pruebas al igual que la fase de diseño se tomara los casos de uso principales.

4.7.1. Pruebas Unitarias.

Para aplicar las pruebas unitarias se debe tomar en cuenta que el sistema se lo desarrollo en las siguientes capas: el modelo, vista y controlador.

A continuación, se mostrarán los resultados de que nos da el software para cada capa.

Los parámetros principales que nos muestra SonarQube son: Bugs y Vulnerabilidades del Código, porcentaje de Código duplicado, archivos analizados, líneas de código totales, seguridad y mantenibilidad.

Si todo funciona correctamente el software SonarQube nos presenta una calificación de Passed caso contrario nos presenta los errores a corregir o modificar.

✓ Modelo



GRÁFICO 15: Resultado de Bugs y Vulnerabilidades en el Modelo.
Elaborado por: (SonarQube, Resultados de Pruebas del Modelo , 2016)

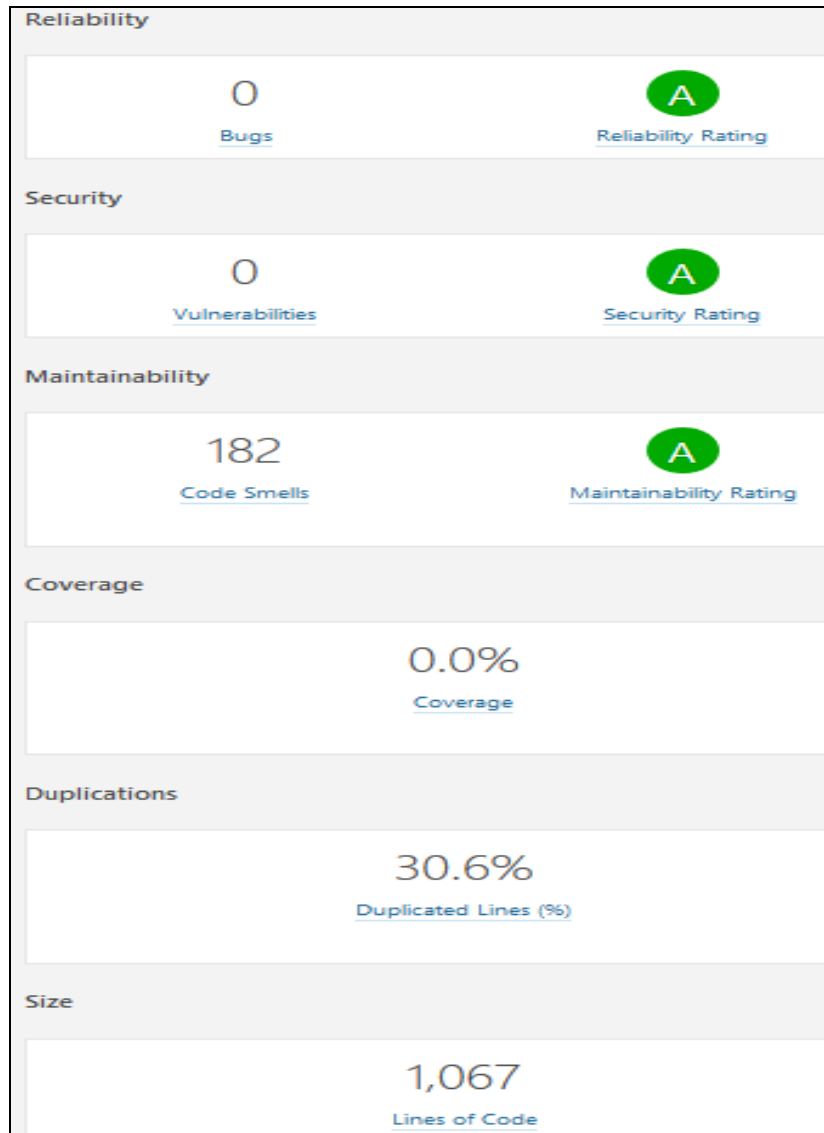


GRÁFICO 16: Resultados a detalle de la capa de Modelo.
 Elaborado por: (SonarQube, Resultados a Detalle del Modelo, 2016)

	Lines of Code	Bugs	Vulnerabilities	Code Smells	Coverage	Duplications
pruebaModelos	1.1k	0	0	182	0.0%	30.6%
Cargos.php	44	0	0	7	0.0%	66.7%
Categoria.php	44	0	0	7	0.0%	66.7%
Ciente.php	68	0	0	12	0.0%	0.0%
ContactForm.php	22	0	0	1	0.0%	0.0%
DetalleFactura.php	64	0	0	9	0.0%	0.0%
Detalleservicio.php	80	0	0	13	0.0%	0.0%
Empleado.php	94	0	0	19	0.0%	66.3%
Estado.php	42	0	0	6	0.0%	0.0%
Facturaventa.php	66	0	0	15	0.0%	38.5%
FormaPago.php	48	0	0	9	0.0%	66.4%
Inventario.php	55	0	0	10	0.0%	0.0%
Iva.php	48	0	0	5	0.0%	0.0%
LoginForm.php	46	0	0	3	0.0%	0.0%
Productos.php	74	0	0	17	0.0%	37.1%
Proveedor.php	68	0	0	15	0.0%	62.6%
Servicios.php	45	0	0	6	0.0%	0.0%
TipoUnidad.php	43	0	0	7	0.0%	66.3%
Usuario.php	65	0	0	12	0.0%	41.0%
Vehiculo.php	51	0	0	9	0.0%	45.5%

GRÁFICO 17: Modelos probados por Sonar Qube.
Elaborado por: (SonarQube,Lista de Modelos a Prueba, 2016)

✓ Vistas

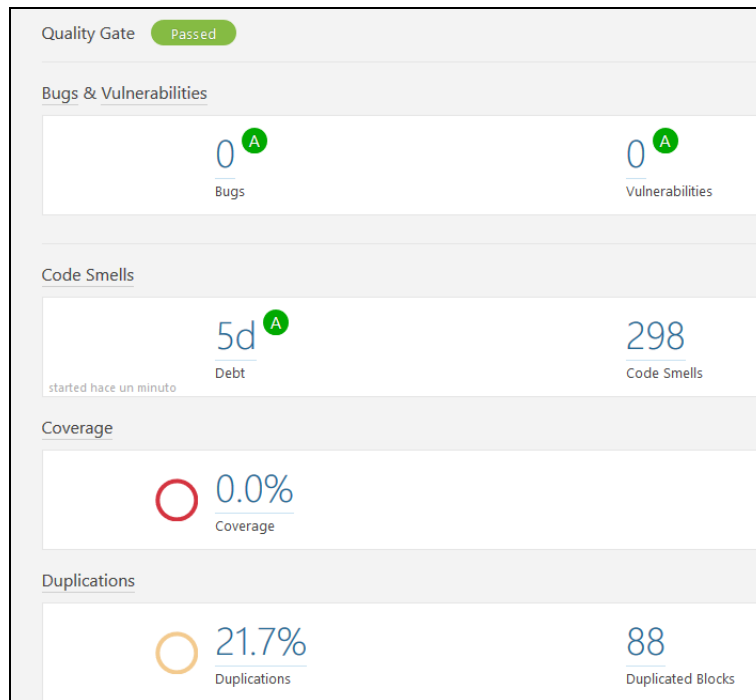


GRÁFICO 18: Modelos probados por Sonar Qube.
 Elaborado por: (SonarQube,Resultados a Detalle, 2016)

	Lines of Code	Bugs	Vulnerabilities	Code Smells	Coverage	Duplications
pruebaVistas	3.2k	2	0	298	0.0%	21.7%
cargos	123	0	0	12	0.0%	20.7%
categoria	122	0	0	12	0.0%	22.6%
cliente	178	0	0	15	0.0%	26.4%
detalleFactura	338	2	0	20	0.0%	40.8%
detalleServicio	290	0	0	35	0.0%	25.2%
empleado	190	0	0	15	0.0%	16.2%
estado	114	0	0	13	0.0%	12.2%
facturaventa	324	0	0	33	0.0%	19.0%
formaPago	127	0	0	11	0.0%	13.0%
inventario	152	0	0	14	0.0%	20.3%
iva	188	0	0	15	0.0%	13.0%
layouts	33	0	0	4	0.0%	0.0%
productos	204	0	0	16	0.0%	19.9%
proveedor	168	0	0	15	0.0%	30.8%
servicios	175	0	0	13	0.0%	13.6%
site	63	0	0	9	0.0%	16.1%
site/pages	5	0	0	1	0.0%	0.0%
tipoUnidad	122	0	0	12	0.0%	22.6%
usuario	173	0	0	18	0.0%	16.6%
vehiculo	152	0	0	15	0.0%	19.7%

GRÁFICO 19: Archivos contenidos dentro de la Capa Vista.
Elaborado por: (SonarQube, Resultados a Detalle, 2016)

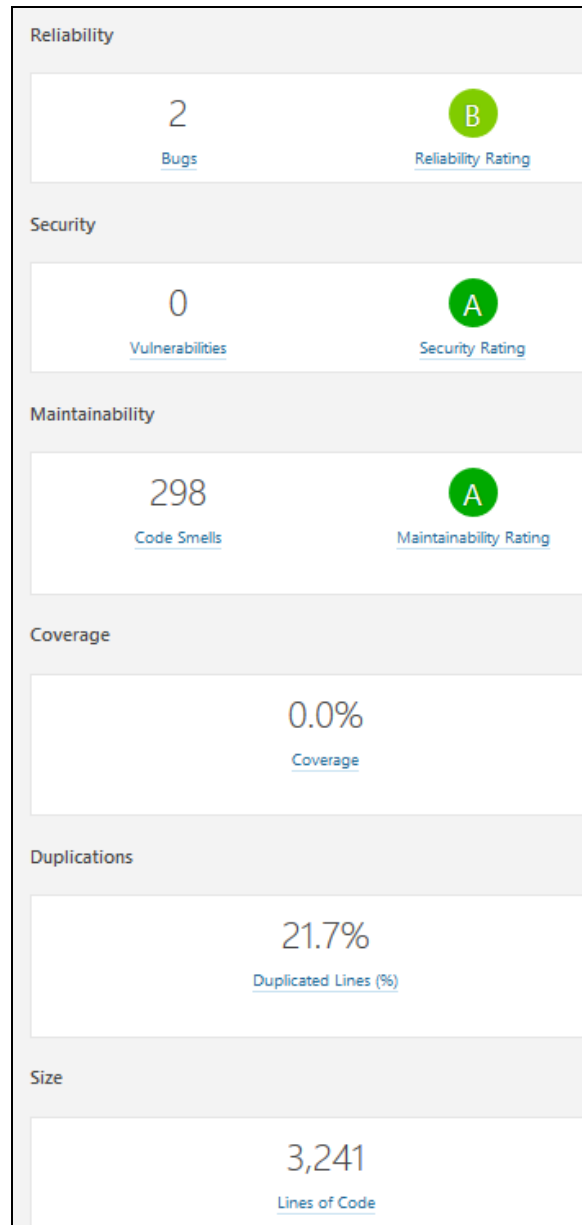


GRÁFICO 20: Resultados a Detalle de la Capa Vista.
Elaborado por: (SonarQube, Resultados a Detalle, 2016)

✓ **Controlador.**

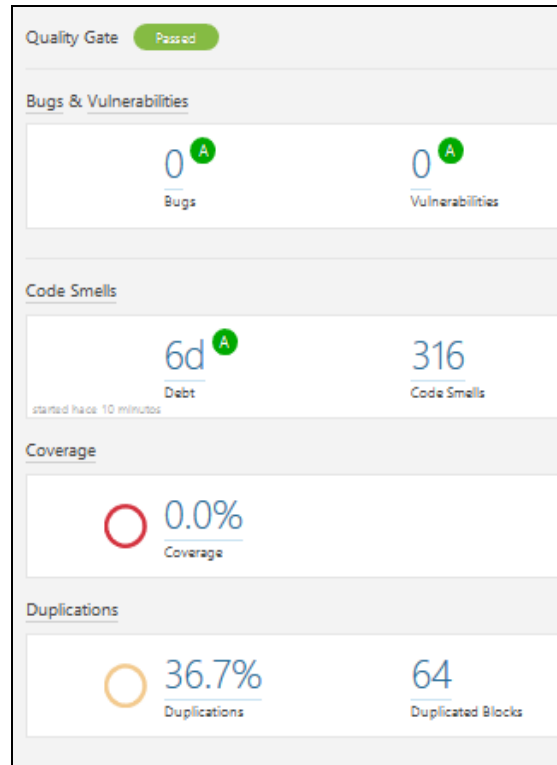


GRÁFICO 21: Resultados de la prueba en la Capa Controlador.
 Elaborado por: (SonarQube, Resultados de Pruebas en el Controlador, 2016)







































	Lines of Code	Bugs	Vulnerabilities	Code Smells	Coverage	Duplications
  pruebaControlador	2.5k	2	0	316	0.0%	36.7%
  CargosController.php	104	0	0	16	0.0%	73.6%
  CategoriaController.php	104	0	0	16	0.0%	57.5%
  ClienteController.php	119	0	0	13	0.0%	19.8%
  DetalleFacturaController.php	350	2	0	30	0.0%	15.7%
  DetalleservicioController.php	154	0	0	19	0.0%	10.0%
  EmpleadoController.php	122	0	0	15	0.0%	0.0%
  EstadoController.php	105	0	0	16	0.0%	57.3%
  FacturaventaController.php	178	0	0	29	0.0%	28.7%
  FormaPagoController.php	107	0	0	15	0.0%	63.0%
  InventarioController.php	120	0	0	16	0.0%	42.0%
  IvaController.php	107	0	0	14	0.0%	74.2%
  ProductosController.php	141	0	0	16	0.0%	41.6%
  ProveedorController.php	110	0	0	16	0.0%	37.0%
  ServiciosController.php	161	0	0	17	0.0%	13.8%
  SiteController.php	102	0	0	11	0.0%	24.7%
  TipoUnidadController.php	105	0	0	16	0.0%	73.6%
  UsuarioController.php	134	0	0	22	0.0%	31.5%
  VehiculoController.php	134	0	0	19	0.0%	48.4%

GRÁFICO 22: Archivos contenidos dentro de la Capa Controlador.
 Elaborado por: (SonarQube,Lista de Archivos de la Capa Controlador, 2016)

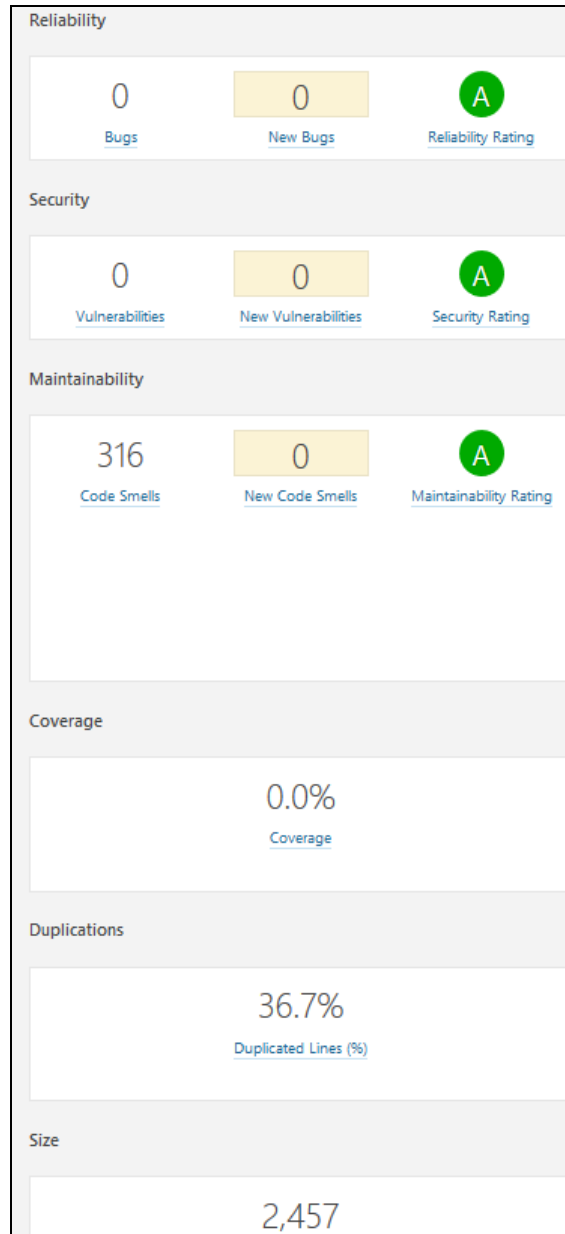


GRÁFICO 23: Resultados a Detalle de la Capa Controlador.

Elaborado por: (SonarQube, Resultado a Detalle de la Prueba a la Capa Controlador, 2016)

4.7.2. Pruebas del Sistema.

Para realizar las pruebas del sistema se tiene en cuenta que el sistema ya se encuentra integrado y como se mostró en la parte de pruebas unitarias el sistema no tiene Bugs ni fallos, en este tipo de pruebas se realizara una lista con todas las funcionalidades que se desarrollaron, estas funcionalidades estarán detalladas en la siguiente tabla, además en el anexo 4 se pueden observar los requerimientos y funcionalidades que se desarrollaron.

Requerimiento	Descripción
Gestión de Clientes.	Ingreso, modificación y eliminación de Datos del Cliente
Gestión de Proveedores.	Ingreso, modificación y eliminación de Datos del Proveedor.
Gestión de Automóvil.	Ingreso, modificación y eliminación de Datos del vehículo del Cliente
Gestión de Inventarios.	Mostrar el estado actual del inventario de los productos.
Gestión de Productos	Ingreso, modificación y eliminación de Productos.
Gestión de Empleados.	Ingreso, modificación y eliminación de Datos de los Empleados.
Control de Usuarios	Ingreso, modificación y eliminación de los Usuarios del Sistema.
Control de Inventarios.	Actualización en tiempo real del stock de los inventarios.
Emisión de Reportes.	Mostrar Reportes del inventario, ventas, productos vendidos y servicios realizados.
Gestión de Facturas.	Creación, modificación, eliminación e impresión de Facturas.

Tabla 30: Descripción de Requerimientos realizados.
Elaborado por: (Carranza, J.,Requerimientos Realizados, 2016)

Para verificar que cada uno de los requerimientos se cumplieron se elaborara una lista detallada con las funcionalidades que se desarrollaron y se señalaran si se cumplieron o no.

Requerimiento	Actividades	A	R
Gestión de Clientes.	Ingreso de Datos. Modificación de Datos. Eliminar Datos.	X X X	
Gestión de Proveedores.	Ingreso de Datos. Modificación de Datos. Eliminar Datos.	X X X	
Gestión de Automóvil.	Ingreso de Datos de Cliente Modificación de Datos Cliente. Eliminar Clientes.	X X X	
Gestión de Inventarios.	Ingreso de Datos. Modificación de Datos. Eliminar Datos.	X X X	
Gestión de Productos	Ingreso de Datos. Modificación de Datos. Eliminar Datos.	X X X	
Gestión de Empleados.	Ingreso de Datos. Modificación de Datos. Eliminar Datos.	X X X	
Control de Usuarios	Ingreso de Datos. Modificación de Datos. Eliminar Datos.	X X X	
Control de Inventarios.	Ingreso de Datos. Modificación de Datos. Eliminar Datos.	X X X	
Emisión de Reportes.	Mostrar reportes estadísticos.	X	
Gestión de Facturas.	Ingreso de Datos. Ingreso de Servicios. Modificación de Datos. Eliminar Datos.	X X X X	

Gestión de Servicios	Ingreso de Datos.	X	
	Ingreso de Productos.	X	
	Modificación.	X	
	Eliminación.	X	

Tabla 31: Lista de Verificación de requerimientos desarrollados.
Elaborado por: (Carranza, J., Verificación de Requerimientos, 2016)

4.7.3. Pruebas de Aceptación.

Como rigüe la metodología de desarrollo ágil XP en cada una de las fases el cliente o usuario final estuvo inmerso, lo que permitió que en esta etapa se afinara el sistema corrigiendo los problemas que podría haber ocurrido en la prueba con el cliente, además se capacitó al personal de la empresa, para que el uso del sistema sea el correcto y se pueda explotar de mejor manera las funcionalidades del mismo.

Como es normal durante el desarrollo de un sistema el cliente solicitaba ciertas modificaciones en el sistema, estas modificaciones se evaluaron y si eran posibles de aplicarlas se lo hizo, caso contrario se llegó a un acuerdo con el cliente explicando los motivos para no aplicarles o guiarle en el correcto uso del sistema.

La aceptación del presente sistema se refleja en la carta de aceptación presentada por el principal Representante de la empresa, este documento se encuentra como anexo dentro del CD.

4.7.4. Implantación del Sistema.

Para la implantación del presente sistema se tomó en cuenta ciertas restricciones en infraestructura tecnología que presenta la empresa, además se debe poner énfasis que esta aplicación pudo haber sido alojada en un servidor externo, a pedido del cliente y por motivos de costo económico no se lo hizo y al contrario se lo instaló de forma local.

Para la implantación del presente sistema se utilizó las siguientes herramientas y software que se detallan a continuación:

- ✓ Servidor de Aplicaciones XAMPP (Indispensable).
- ✓ Workbench 6.2 (no indispensable).
- ✓ Conexión de Internet.

Para la implantación se realizaron los siguientes pasos:

1. Instalar el programa XAMPP.
2. Instalar el programa Workbench.
3. Configurar los puertos de Apache y MySQL
4. Copiar la aplicación en la Carpeta htdocs del XAMMP.
5. Configurar una conexión nueva en el Workbench.
6. Restaurar la Base de Datos del Sistema.
7. Configurar la cadena de conexión dentro de la Aplicación.
8. Ingresar la siguiente url dentro del navegador:

[http://localhost:\(puerto configurado\)/Tesis/index.php](http://localhost:(puerto configurado)/Tesis/index.php)

4.8. Documentación

4.8.1. Manual Técnico.

El manual técnico muestra a detalle cómo se desarrolló el sistema, desde la fase de diseño hasta la fase de implementación, este documento se encuentra como anexo dentro del CD.

El uso de este manual tiene como finalidad el entender como está construido y como se lo puede modificar si en caso se requiriera.

4.8.2. Manual de Usuario.

El manual de usuario es un documento que sirve de guía para el correcto uso del presente sistema, este manual fue entregado al cliente el momento de haber sido implementado el sistema.

El presente manual se encuentra como anexo del presente trabajo de disertación.

CAPÍTULO 5. CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones:

- La metodología XP y Áncora facilitaron el desarrollo del sistema principalmente en las fases de codificación y pruebas, lo que dio como resultado el cumplimiento con todos los requerimientos solicitados por el usuario.
- El Framework utilizado para el desarrollo disminuyó el tiempo de codificación y diseño web ya que posee una serie de herramientas muy útiles.
- Las interfaces de usuario se las diseñó siguiendo ciertos estándares y utilizando herramientas como javascript y css con la finalidad de que el sistema sea amigables para el usuario.
- El presente sistema tuvo ciertos problemas al automatizar el proceso de facturación y control de inventario de la mecánica “Frenicentro”, ya que dicho proceso fue muy complejo.
- Al realizar el presente sistema de facturación y control de inventarios se pudo aplicar todos los conocimientos adquiridos sobre: Programación, Ingeniería de Software y Bases de Datos.
- La curva de aprendizaje de un nuevo lenguaje y su framework causaron ciertos retrasos con respecto a la planificación que se planteo en un inicio.
- La especificación de requerimientos es muy importante para el desarrollo de un sistema esto depende del interés de las partes interesadas.
- La presentación de prototipos en cada reunión es muy útil para especificar y satisfacer de mejor manera los requerimientos del cliente.

5.2. Recomendaciones:

- Previo al desarrollo de un sistema es recomendable conocer las propiedades y ventajas de cada una de las metodologías a utilizar.
- Para el desarrollo del sistema es recomendable tener una planificación, centrándose en la fase de codificación, que a menudo es cuando más se demoran en salir los proyectos, por esta razón es recomendable el uso de framework que faciliten el desarrollo.
- Para que un sistema cumpla con los requerimientos del usuario es recomendable seguir metodologías como UX o user experience para que la aplicación sea amigable.
- En muchos de los casos, el desarrollador no tiene muy claro cómo se ejecuta el proceso a automatizar y por esta razón es recomendable tener una buena comunicación con el cliente.
- Para el manejo y modificación del presente sistema se recomienda que las personas interesadas se instruyan con los manuales que se anexan en el presente trabajo y tengan conocimientos en las siguientes áreas: Programación y Bases de Datos.
- Se recomienda tener muy en cuenta que el tiempo que se tome en aprender el uso de un nuevo framework y lenguaje de programación incidirá directamente en el tiempo de desarrollo y finalización de un proyecto.
- Es recomendable que las partes interesadas dentro de un proyecto, estén 100% involucradas y comprometidas; esto quiere decir, que se mantengan reuniones semanales para definir o detallar de mejor manera los requerimientos.
- Se recomienda hacer prototipos y presentarlos en cada reunión; así se facilitará su comprensión de mejor manera; a fin de optimizar los requerimientos del cliente.

BIBLIOGRAFÍA

Andres, K. B. (2004). *Extreme Programming Explained Embrace Change*. Boston: Addison-Wesley.

<http://www.telecomunicaciones.gob.ec/>. (31 de 12 de 2015). *Ministerio de Telecomunicaciones y de la Sociedad de la información*. Obtenido de <http://www.telecomunicaciones.gob.ec/el-823-de-mipymes-en-el-ecuador-utilizan-internet/>

López, M. d. (2001). *Áncora: Metodología para el Análisis de Requerimientos de Software conducente al Reuso*. En M. d. López, *Áncora: Metodología para el Análisis de Requerimientos de Software conducente al Reuso* (pág. 7). Mexico.

M. Griselda Báez, S. I. (2003). <http://www.inf.puc-rio.br/wer01/Mod-Req-1.pdf>. Obtenido de <http://www.inf.puc-rio.br/wer01/Mod-Req-1.pdf>.

Mordecki, Daniel. (s.f.). *Pensar Primero*.

Poole, D. B. (2009). *Do It Yourself Agile*.

ANEXOS

Los anexos de la presente Tesis, se encuentran en la carpeta “Anexos”, que se incluyen en el C.D.

MANUAL TÉCNICO Y MANUAL DE USUARIO

Estos documentos constan, también, en el C.D., en la carpeta “Anexos”.

GLOSARIO DE TÉRMINOS

- Actualización en Línea, 41
- Alcances de Áncora, 22
- Análisis de Requerimientos de Software, 22
- Áncora, 21
- Archivo de Interfaz
- Externa, 30
- Archivos Lógicos Internos, 30
- Atributos, 28
- BD Orientada a Objetos, 29
- Bottom-up, 29
- Calendario para la RRD, 52
- Características
- Extreme Programming, 59
- Cierre, 55
- Cliente, 60
- Comunicación de datos, 34
- Condiciones, 25
- Condiciones de entrada, 24
- Condiciones de Salida, 24
- Configuración Altamente
- Usada, 37
- Consultas Externas, 31
- Consultor, 60
- Definición de Requerimientos de Software, 22
- Diálogos, 26
- Eficiencia para el Usuario Final, 40
- El lugar para la RRD, 50
- Elementos gráficos, 25
- Entidad, 27
- Entrada de Datos en Línea, 39
- Entradas Externas, 31
- Entrenador, 60
- Escena, 25
- Especificación de
- Requerimientos de Software, 22
- Extreme Programing, 57
- Facilidad de Cambios, 47
- Facilidad de Instalación, 44
- Facilidad de Operación, 45
- Gestor, 60
- Glosarios, 27
- Grupo Nominal, 48
- Guión, 24
- Guiones y Diálogos., 24
- Guiones., 24
- Historia
- Extreme Programing, 58
- La conducción de la RRD, 50
- La duración de la RRD, 50
- lenguajes de lado del servidor, 79
- Los asistentes a RRD, 50
- Modelo de Objetos Semánticos, 28
- Modelo Entidad – Relación, 27
- MySQL, 81
- Objetivos Extreme Programing, 58
- Objeto de asociación, 29

- Objetos combinados, 29
- Objetos compuestos, 29
- Objetos de generalización y subtipos., 29
- Objetos híbridos, 29
- Objetos simples, 29
- Papeles, 24
- PHP, 69
- Pista, 24
- Preparación de la Reunión de Reflexión y, 55
- Procesamiento Complejo, 42
- Procesamiento Distribuido de Datos, 35
- programación de lado del cliente, 76
- Programador, 60
- Promedio de Transacciones, 38
- Puntos de Función, 30
- Quinteta, 25
- Relación, 28
- Rendimiento, 36
- Reunión de Reflexión y
Diseño, 49, 55
- Reusabilidad, 43
- Salidas Externas, 31
- Tablas, 26
- Tablas y Glosarios, 26
- TDE, 32
- Tester, 60
- Top-down, 29
- Tracker, 60
- TRE, 32
- Utensilios, 24
- Varios Sitios, 46
- XAMPP, 80
- Yii, 72