

# **ESCUELA POLITÉCNICA DEL EJÉRCITO**

## **DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN**

### **CARRERA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA**

Análisis, Diseño, Desarrollo e Implementación de  
una Aplicación Web para la Automatización de  
Administración de Clientes, Vehículos, Facturación,  
Inventario y Campañas para Auto Servicios RBS

Previo a la obtención del Título de:

### **INGENIERÍA EN SISTEMAS E INFORMÁTICA**

**POR:**

**Camilo González**

**Sangolquí, Octubre de 2012**

## ÍNDICE DE CONTENIDOS

CAPÍTULO 1: INTRODUCCIÓN .....	8
1.1 Antecedentes.....	9
1.2. Planteamiento del Problema .....	6
1.3. Justificación e Importancia.....	7
1.4. Objetivos .....	8
1.4.1 <i>Objetivo General</i> .....	12
1.4.2 <i>Objetivos Específicos</i> .....	12
1.5. Alcance.....	12
1.6. Definición del entorno del Problema .....	10
1.6.1 Resultados y Analisis de la Encuesta realizada para la implementacion del proyecto en el taller de servicios RBC .....	11
1.7. Factibilidad.....	18
1.7.1 <i>Factibilidad Operativa</i> .....	22
1.7.2 <i>Factibilidad Técnica</i> .....	22
1.7.2.2 <i>Recursos Humanos</i> .....	23
1.7.2.3 <i>Recursos de Entorno</i> .....	19
1.7.3 <i>Factibilidad Económica</i> .....	24
1.7.3.1 <i>Tiempo Previsto</i> .....	24
1.7.3.2 <i>Gastos que asumirá la Unidad de Transporte de la ESPE – Sangolquí</i> .....	24
1.7.3.3 <i>COSTO TOTAL DEL PROYECTO</i> .....	21

CAPÍTULO 2: MARCO TEÓRICO .....	26
2.1 Metodología .....	22
2.1.1 <i>Análisis de Metodologías Ágiles</i> .....	22
2.1.1.1 <i>Metodología SCRUM</i> .....	30
2.1.1.2 <i>Metodología XP</i> .....	31
2.1.1.3 <i>Otras Metodologías</i> .....	33
2.1.1.4 <i>Cuadro Comparativo</i> .....	36
2.1.1.5 <i>Justificación</i> .....	37
2.1.2 <i>Aplicación de Metodología a Implantar</i> .....	38
2.1.2.1 <i>Metodología de Desarrollo de Software XP (Extreme Programming)</i> .....	38
2.1.2.2 <i>Definición</i> .....	38
2.1.2.3 <i>Características</i> .....	38
2.1.2.4 <i>Utilidad</i> .....	39
2.1.2.5 <i>Esquema de la metodología XP (Extreme Programming)</i> .....	39
2.1.2.6 <i>Ciclo de vida XP (Extreme Programming)</i> .....	41
2.1.2.7 <i>Fases</i> .....	41
2.1.2.7.1 <i>Fase 1ª: Planificación del proyecto</i> .....	41
2.1.2.7.2 <i>Fase 2ª: Diseño</i> .....	44
2.1.2.7.3 <i>Fase 3ª: Codificación</i> .....	45
2.1.2.7.4 <i>Fase 4ª: Pruebas</i> .....	46
2.2 <i>Plataforma de Desarrollo</i> .....	47
2.2.1 <i>Aplicaciones Web (web application, webapp)</i> .....	47
2.2.1.1 <i>Introducción</i> .....	47
2.2.1.2 <i>Definición</i> .....	47
2.2.1.3 <i>Características</i> .....	48
2.2.1.4 <i>Clasificación</i> .....	559
2.2.2 <i>Herramientas de Desarrollo</i> .....	593
2.2.2.1 <i>AJAX</i> .....	593

2.2.2.2	<i>Términos de Web 2.0</i> .....	625
2.2.2.3	<i>JSF 2.0</i> .....	58
2.2.2.4	<i>Rich Faces 4</i> .....	59
2.2.2.5	<i>ActionScript 3.0</i> .....	60
2.2.3	<i>Motores de Base de Datos</i> .....	682
2.2.3.1	<i>MySQL</i> .....	682
2.2.4	<b>Interfaz de la aplicación del Programa (API)</b> .....	704
2.2.4.1	<i>EJB 3.1</i> .....	64
2.2.4.2	<i>JPA 2.0</i> .....	71
2.2.5	<i>Lenguaje de Programación</i> .....	782
	<b>CAPÍTULO 3: Construcción del Sistema</b> .....	77
3.1	<b>Contenido</b> .....	77
3.1.1	<b>Análisis de Requerimientos</b> .....	77
3.1.2	<b>Diseño</b> .....	94
3.1.3	<b>Desarrollo e Implementación</b> .....	125
	<b>CAPÍTULO 4 : Conclusiones y Recomendaciones</b> .....	136
4.1	<b>Contenido</b> .....	136
4.1.1	<b>Conclusiones</b> .....	136
4.1.2	<b>Recomendaciones</b> .....	137
4.2.	<b>Bibliografía</b> .....	139
	<b>Anexo</b> .....	140
	<b>Manual de Usuario</b> .....	140
	<b>Manual de Instalación</b> .....	195
	<b>CARTAS</b> .....	195
	<b>BIOGRAFÍA</b> .....	195

## LISTA DE TABLAS

	<b>Pg.</b>
Tabla 1: Recursos humanos	22
Tabla 2: Recursos de entorno	23
Tabla 3: Factibilidad económica	23
Tabla 4: Tiempo previsto	23
Tabla 5: Gastos que asumirá el taller RBS	23
Tabla 6: Costo total del proyecto	24
Tabla 7: Cuadro comparativo de metodologías ágiles y no ágiles	32
Tabla 8: Cuadro de consideraciones a analizar para escoger una metodología	40
Tabla 9: Historia de usuario 1 en el diseño	81
Tabla 10: Historia de usuario 2 en el diseño	83
Tabla 11: Historia de usuario 3 en el diseño	89
Tabla 12: Historia de usuario 4 en el diseño	92
Tabla 13: Historia de usuario 5 en el diseño	94
Tabla 14: Historia de usuario 6 en el diseño	96
Tabla 15: Atributos inputText - inputSecret	99
Tabla 16: Atributo inputTextarea	100
Tabla 17: Atributos outputText – graphicImage	100
Tabla 18: Atributo commandButton	100
Tabla 19: Atributo commandLink	100
Tabla 20: Datos usuario	115
Tabla 21: Datos persona	116
Tabla 22: Datos cliente	117
Tabla 23: Datos ingresa vehículo	118
Tabla 24: Datos toma de orden	118
Tabla 25: Datos proveedor	119
Tabla 26: Datos repuesto proveedor	119
Tabla 27: Datos inventario	120
Tabla 28: Datos repuesto inventario	120
Tabla 29: Datos facturación	121
Tabla 30: Datos repuesto	122
Tabla 31: Datos servicio parametrización	123
Tabla 32: Datos número de visitas en campaña	123
Tabla 33: Tarjeta CRC usuario	124
Tabla 34: Tarjeta CRC persona	124
Tabla 35: Tarjeta CRC vehículo	125
Tabla 36: Tarjeta CRC cliente	125
Tabla 37: Tarjeta CRC detalle orden	125
Tabla 38: Tarjeta CRC orden	126
Tabla 39: Tarjeta CRC detalle proveedor	126
Tabla 40: Tarjeta CRC proveedor	127
Tabla 41: Tarjeta CRC detalle inventario	127

Tabla 42:	Tarjeta CRC inventario	128
Tabla 43:	Tarjeta CRC detalle facturación	128
Tabla 44:	Tarjeta CRC facturación	128
Tabla 45:	Tarjeta CRC producto	129
Tabla 46:	Tarjeta CRC parametrización	129
Tabla 47:	Tarjeta CRC campañas	129
Tabla 48:	Pruebas automatización de caja negra	138
Tabla 49:	Pruebas automatización de caja blanca	140

## LISTA DE GRÁFICOS

	<b>Pg.</b>
Gráfico 1: Metodología con mayor presencia en el internet	40
Gráfico 2: Metodología mejor documentada	41
Gráfico 3: Faces XP	46
Gráfico 4: Tiempos en XP	51
Gráfico 5: Aplicaciones web	53
Gráfico 6: Capas en las aplicaciones web	56
Gráfico 7: Servicios web	57
Gráfico 8: AJAX	59
Gráfico 9: Modelo de aplicación – AJAX y Clásico	60
Gráfico 10: Diferencias Web 1 y Web 2	61
Gráfico 11: Arquitectura Richfaces 4	65
Gráfico 12: Anotaciones EJB	75
Gráfico 13: Composición de un Bean	75
Gráfico 14: Arquitectura	98
Gráfico 15: Login	101
Gráfico 16: Página principal	102
Gráfico 17: Usuario	103
Gráfico 18: Personas	103
Gráfico 19: Clientes	104
Gráfico 20: Productos	104
Gráfico 21: Proveedor maestro	105
Gráfico 22: Proveedor detalle	105
Gráfico 23: Inventario maestro	106
Gráfico 24: Inventario detalle	106
Gráfico 25: Orden maestro	107
Gráfico 26: Orden detalle	108
Gráfico 27: Parametrización	109
Gráfico 28: Campañas	109
Gráfico 29: Facturación maestro	110
Gráfico 30: Facturación detalle	110
Gráfico 31: Red del sistema	131
Gráfico 32: Proceso de pruebas	132

## RESUMEN

El taller de Auto Servicio RBS es un taller de reparación y mantenimiento de vehículos y afán de mejorar los procesos de Automatización de administración de Clientes, Facturación, Inventarios y realizar promociones temporales llamadas campañas, busca un soporte informático de calidad para el manejo de su información.

El presente estudio se enfocó en el análisis de la metodología ágil XP (Extreme Programming) para la implementación de una aplicación web para la facturación, control de clientes, inventario, proveedores y control de promociones a clientes.

La ejecución y culminación del proyecto permitió establecer una metodología basada en XP. El resultado es un producto de software funcional, en cuyo desarrollo se pudo demostrar que la validez de XP aplicada a proyectos de software de mediano tamaño, en entornos cambiantes con grupo de trabajos pequeños que involucran permanentemente al dueño del producto.

# **CAPÍTULO 1: INTRODUCCIÓN**

## **1.1 Antecedentes**

El taller de Auto Servicio RBS es un taller de reparación y mantenimiento de vehículos.

El Taller de Auto Servicios RBS en el afán de mejorar los procesos de Automatización de Clientes, Facturación e Inventarios busca un soporte informático de calidad para el manejo de su información.

El taller realiza procesos de control de clientes, vehículos, facturación e inventario diariamente; mediante el ingreso de datos en una hoja de cálculos en el programa Microsoft Office Excel respaldado por el almacenamiento de documentos impresos clasificados en distintas carpetas.

Para la Integración del Control de ingreso de clientes y vehículos, el taller, en la presente fecha realiza las siguientes actividades.

1. Ingreso datos del cliente.
2. Ingreso datos del vehículo.
3. Ingreso de actividades que se va a resolver para la reparación del vehículo.

Para la integración del control de Facturación el taller en la presente fecha realiza las siguientes actividades.

1. Ingreso datos cliente.
2. Ingreso datos vehículos.
3. Actividades realizadas para la reparación del vehículo.
4. Costos.
5. Verificación de productos usados en bodega.

Para la integración del control de inventarios el taller en la presente fecha realiza las siguientes actividades:

1. Ingreso de repuestos y productos.
2. Asignar códigos a los repuestos y productos según categoría.
3. Asignación de precios y porcentaje de ganancia por producto.

El taller busca implantar un sistema que agilite y mejore este proceso, contando con un manejo sistematizado para su información, sin riesgos de pérdida de información o de confiabilidad de la misma.

## **1.2 Planteamiento del Problema**

Actualmente para la integración del control de clientes, vehículos, facturación e inventarios en el Taller de Auto Servicios RBS, se desarrollan actividades administrativas que llevan a la falta de confiabilidad de la información, pérdida de la misma y desperdicio de tiempo

Se generan problemas como:

- Falta de autenticación y seguridad para el manejo de la información almacenada.
- Pérdida y desorganización de documentos.
- Falta de respaldo de información en una base de datos.
- Falta de actualización de datos.
- Redundancia en los procesos.
- Confusión en búsqueda de facturas.
- Falta de control de históricos de clientes y reparación de vehículos.
- No concordancia en inventarios.
- La falta de campañas publicitarias por medio de correos para publicar las promociones vigentes.

Y como consecuencia de ello:

- Demora en la entrega de informes cuando lo requiere el propietario.
- No existen informes generales de mantenimiento de vehículos.
- Demora en el correcto mantenimiento de los vehículos.
- Demora en declaración de impuestos por el mal manejo de facturación
- Datos no reales en inventario
- Desconocimiento de las promociones vigentes para los clientes frecuentes al taller.

Para evitar los problemas antes mencionados se desarrollará una aplicación web que permita automatizar el proceso de Integración del Control de Clientes, Vehículos, Facturación e Inventario. Y, de esta manera; generar automáticamente reportes, controles e información confiable para el buen funcionamiento del Taller de Auto Servicios RBS

### **1.3 Justificación e Importancia**

Dentro de los procesos que se llevan a cabo en el taller de Auto Servicio RBS, se ha considerado que la Integración del Control de Clientes, Vehículos, Facturación e Inventarios, requiere automatización y optimización en sus actividades; a fin de mejorar los servicios que presta y maximizar la calidad que ofrece el taller, minimizando debilidades que se presentan dentro de dicho proceso.

Por lo que se plantea el desarrollo de esta aplicación Web para el mejoramiento de los procesos del control de Clientes, Vehículos, Facturación e Inventarios, consiguiendo de esta manera un mejor y más claro manejo de los requerimientos del proceso antes expuesto, tanto en el corto como en el mediano plazo.

Gracias a las mejoras a realizarse en el proceso; se conseguirá entregar mayor seguridad y confiabilidad en la información y en el servicio que el taller de Auto

Servicio RBS está al momento, mejorando el servicio a los usuarios.

Se decidió hacer un aplicativo Web porque la empresa está pensando expandirse a corto plazo a un taller en Quito

## **1.4 Objetivos**

### **1.4.1 Objetivo General**

Desarrollar e Implementar una Aplicación Web para la integración y administración de los Procesos de Control de Clientes, Vehículos, Facturación e Inventario para Autoservicios RBS; y así proveer información confiable y oportuna a los usuarios de dicho taller, para que puedan utilizarla en posteriores toma de decisiones, del mantenimiento preventivo y reparación de fallas de los vehículos del taller; de manera muy sencilla y rápida.

### **1.4.2 Objetivos Específicos**

- Revisar los módulos y conceptos, relacionados a las metodologías ágiles.
- Revisar las fases de la metodología XP (Extreme Programming).
- Revisar los conceptos relacionados con las Web App's con java.
- Realizar el análisis y el diseño del sistema, aplicando la metodología XP.
- Realizar el desarrollo y la implementación del sistema, aplicando la metodología XP.

## **1.5 Alcance**

Se desarrollará una Aplicación Web parametrizable para la Integración del Control de Clientes, Vehículos, Facturación e Inventarios del taller de Auto Servicio RBS.

El desarrollo del proyecto manejará distintas fases, las cuales se encargarán de generar varios productos para conseguir el mejor resultado que justifique la

solución real que traerá la aplicación a desarrollarse. Las mismas que se encuentran detalladas a continuación:

Como fase inicial, gracias al libre acceso a la información se tiene; la entera y completa recolección de datos y requerimientos que serán de gran utilidad para el posterior análisis de los mismos para continuar con la fase siguiente de diseño de la aplicación.

Como segunda fase se cuenta con el diseño propio del sistema; mediante el análisis anterior y previamente realizado un análisis comparativo óptimo a la aplicación de una metodología correspondiente con el tema, para que sea más ágil el desarrollo del sistema. Definiendo las partes a manejar en el aplicativo.

Como son:

- ✓ Autenticación del sistema
- ✓ Definición de clientes y vehículos.
- ✓ Toma de la orden para reparación o mantenimiento de vehículo.
- ✓ Validación de repuestos o productos que se encuentran en inventario.
- ✓ Facturación y despacho.
- ✓ Campañas orientadas a promociones para los clientes.

En la tercera fase, utilizando el diseño preestablecido en la fase anterior, se continuó con la parte de la construcción y desarrollo del aplicativo propiamente.

Llevando un Control y Manejo de Versiones propio para cada una de las partes de la aplicación, conforme éstas vayan avanzando en el proceso de creación de la misma.

Para la cuarta fase, se realizarán las pruebas críticas correspondientes a la aplicación. Llevando a la misma, a las distintas pruebas de estrés para control de errores y alcances del aplicativo; verificando concordancia entre los procesos y el manejo de base de datos en su totalidad.

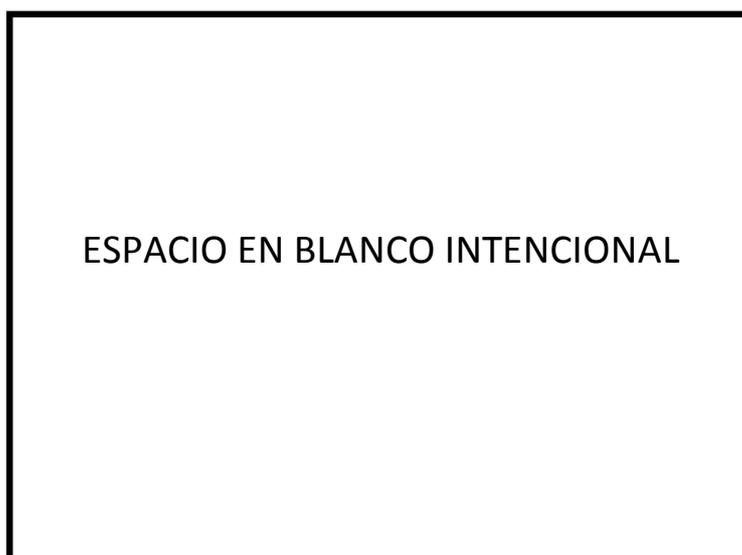
Finalmente, después de la total depuración de errores; se llevará a cabo la implementación del aplicativo en el taller. Y por último se realiza la capacitación al personal encargado del manejo de los procesos del taller de Auto Servicio RBS.

Adicional a estos requerimientos el taller buscando ser una empresa competitiva con el mercado desea realizar un módulo de campañas, este módulo se lo realizará de la siguiente manera:

- Promoción por número de visitas: Consiste en la contabilización de visitas de los clientes y el cliente que más visitas tenga al taller tendrá acceso a alguna promoción o descuento en servicios o productos.

## **1.6 Definición del entorno del problema**

Para obtener información más directa de los hechos que surgen día a día con el servicio se recurre a métodos de investigación de campo, como es el caso de una encuesta. Se elabora un cuestionario con preguntas realizadas tanto al personal que labora en el Taller como a los usuarios y posibles usuarios, quienes son los directamente afectados y por consecuencia; también involucrados en el desarrollo de esta tesis en estos primeros pasos del proceso. Para de esta manera, conseguir aclarar mucho más los requerimientos del servicio que presta la unidad.



## 1.6.1 Resultados y análisis de la encuesta realizada para la implementación del proyecto en el taller de repuestos RBC.

### PREGUNTAS

#### 1. ¿Usted ha requerido o necesitado los servicios del Taller Mecánico RBS?

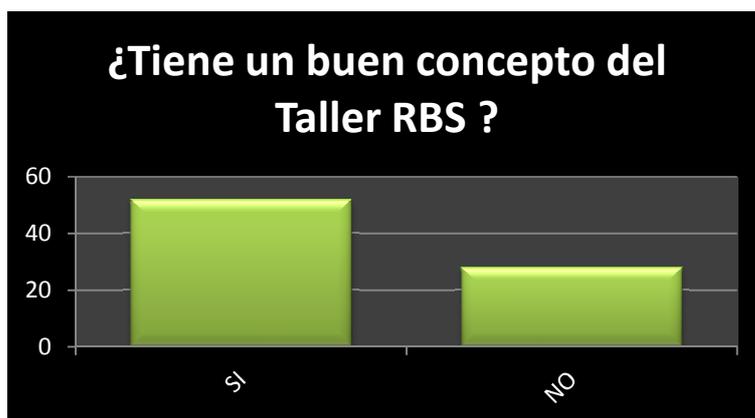
SI	69	86%
NO	11	14%
	80	



Análisis técnico		Conclusiones destacadas
Media	1.138	La opción más elegida fue "SI". La opción menos elegida fue "NO".
Intervalo de confianza (95%)	[1,062 - 1,213]	
Tamaño de la muestra	80	
Desviación típica	0.347	
Error estándar	0.039	

#### 2. ¿Tiene un buen concepto del Taller RBS?

SI	52	65%
NO	28	35%
	80	



Análisis técnico		Conclusiones destacadas
Media	1.350	La opción más elegida fue "SI".  La opción menos elegida fue "NO".
Intervalo de confianza (95%)	[1,245 - 1,455]	
Tamaño de la muestra	80	
Desviación típica	0.480	
Error estándar	0.054	

### 3. ¿Conoce usted que servicios brinda el Taller RBS a los usuarios ?

SI	38	48%
NO	42	53%
	80	



Análisis técnico		Conclusiones destacadas
Media	1.525	La opción más elegida fue "NO".  La opción menos elegida fue "SI".
Intervalo de confianza (95%)	[1,415 - 1,635]	
Tamaño de la muestra	80	
Desviación típica	0.503	
Error estándar	0.056	

4. Del 1 al 10. ¿Qué calificación le daría al servicio del Taller ?

1	2	3	4	5	6	7	8	9	10	TOTAL POBLACIÓN
5	6	3	2	17	5	14	19	6	2	79



Análisis técnico - Población: 79		Conclusiones destacadas
Media	5.987	El "45,57%" eligieron:
Intervalo de confianza (95%)	[5,458 - 6,516]	
Tamaño de la muestra	79	8
Desviación típica	2.399	5
Error estándar	0.270	La opción menos elegida representa el "2,53%":
		4

5. ¿Cree usted que el Taller RBS brinda una buena atención a los clientes ?

SI	42	55%
NO	35	45%
	77	



Análisis técnico		Conclusiones destacadas
Media	1.455	La opción más elegida fue "SI".  La opción menos elegida fue "NO".
Intervalo de confianza (95%)	[1,343 - 1,566]	
Tamaño de la muestra	77	
Desviación típica	0.501	
Error estándar	0.057	

6. ¿Cree usted que existe un minucioso control al momento de entregarle su vehículo ?

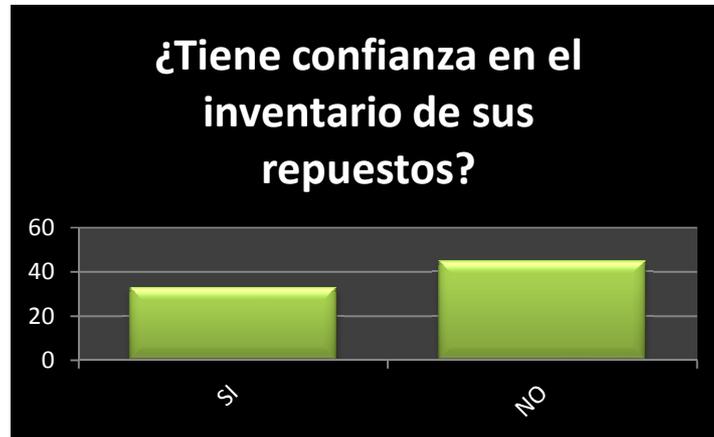
SI	36	46%
NO	42	54%
	78	



Análisis técnico		Conclusiones destacadas
Media	1.538	La opción más elegida fue "NO".  La opción menos elegida fue "SI".
Intervalo de confianza (95%)	[1,427 - 1,650]	
Tamaño de la muestra	78	
Desviación típica	0.502	
Error estándar	0.057	

7. ¿Tiene confianza en el inventario de sus repuestos ?

SI	33	42%
NO	45	58%
	78	



Análisis técnico		Conclusiones destacadas
Media	1.577	La opción más elegida fue "NO".
Intervalo de confianza (95%)	[1,467 - 1,687]	
Tamaño de la muestra	78	La opción menos elegida fue "SI".
Desviación típica	0.497	
Error estándar	0.056	

8. ¿Cree usted que es necesario otra forma de publicidad para el taller ?

SI	74	94%
NO	5	6%
	79	



Análisis técnico		Conclusiones destacadas
Media	1.063	La opción más elegida fue "SI".  La opción menos elegida fue "NO".
Intervalo de confianza (95%)	[1,009 - 1,117]	
Tamaño de la muestra	79	
Desviación típica	0.245	
Error estándar	0.028	

9. ¿Cree usted que en el Taller existe organización con respecto al control de documentos y tramitación de servicios?

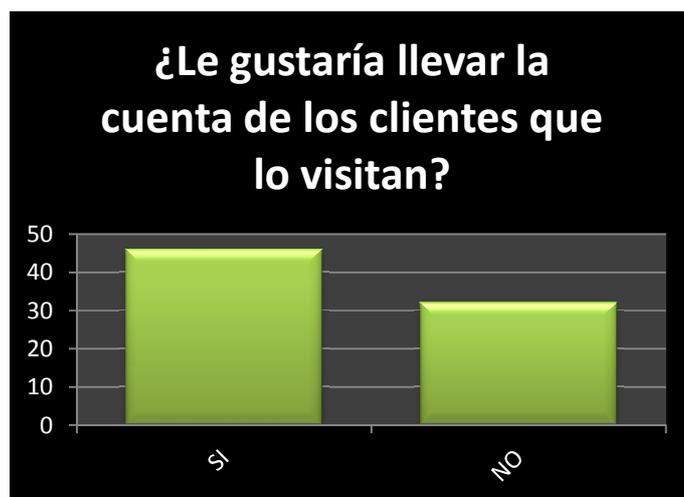
SI	33	42%
NO	45	58%
	78	



Análisis técnico		Conclusiones destacadas
Media	1.577	La opción más elegida fue "NO".  La opción menos elegida fue "SI".
Intervalo de confianza (95%)	[1,467 - 1,687]	
Tamaño de la muestra	78	
Desviación típica	0.497	
Error estándar	0.056	

10. ¿Le gustaría llevar la cuenta de los clientes que lo visitan ?

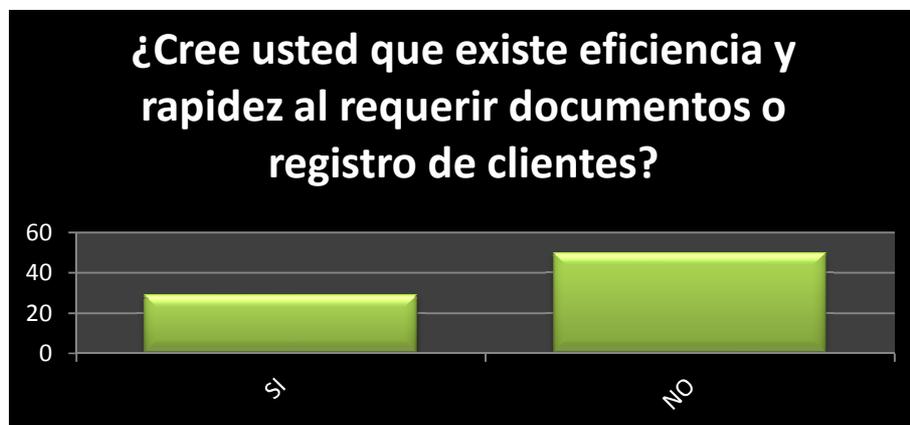
SI	46	59%
NO	32	41%
	78	



Análisis técnico		Conclusiones destacadas
Media	1.410	La opción más elegida fue "SI".
Intervalo de confianza (95%)	[1,300 - 1,520]	
Tamaño de la muestra	78	La opción menos elegida fue "NO".
Desviación típica	0.495	
Error estándar	0.056	

**11. ¿Cree usted que existe eficiencia y rapidez al requerir documentos o registro de clientes?**

SI	29	37%
NO	50	63%
	79	



Análisis técnico		Conclusiones destacadas
Media	1.633	La opción más elegida fue "NO".
Intervalo de confianza (95%)	[1,526 - 1,740]	
Tamaño de la muestra	79	La opción menos elegida fue "SI".
Desviación típica	0.485	
Error estándar	0.055	

## **CONCLUSIÓN:**

Bajo los resultados obtenidos en la encuesta y cuestionario, se puede anotar cada una de las necesidades específicas con los que el taller se encuentra con menor aceptación por parte de los usuarios y con ello; se puede armar un mejor esquema de los requerimientos de este proyecto.

Según la encuesta realizada se concluyó que se debe agilizar los procesos de facturación al cliente así como también al momento de realizar inventario verificando el stock máximo

## **1.7 Factibilidad**

### **1.7.1 Factibilidad Operativa**

Para desarrollar la aplicación web se tiene el apoyo del taller de Auto Servicio RBS:

✓ Ing. Robert Borja Sáenz

Quien proporcionará lo necesario tanto en equipos como en información para la implementación del sistema.

Existe gran interés por parte de los propietarios del Taller de Servicios RBS y del estudiante programador para llevar a cabo la culminación exitosa del sistema.

### **1.7.2 Factibilidad Técnica**

#### **1.7.2.1 Hardware:**

Se piensa a futuro en la adquisición de un servidor, pero por el momento el aplicativo será implantado en una máquina con características normales para su funcionamiento.

Los computadores del taller cuentan con programas informáticos para ejecutar la aplicación web.

### 1.7.2.2 Software:

Para el desarrollo del sistema se va a utilizar Software Libre, como JAVA que es una herramienta muy poderosa para el desarrollo de aplicaciones web.

La base de datos con la que trabajará el sistema de seguridad será My SQL.

*(Mandato del Gobierno con el decreto Nº. 1014 del 10 de abril del 2008, promueve a la utilización de software libre en las Entidades de la Administración Pública. )*

### 1.7.2.3 Recursos Humanos

Rol	Duración	Observaciones
Programadores de Alto nivel	4 mes	Encargados del análisis de requerimientos y datos recogidos. Conjuntamente con el diseño del sistema.
	2 mes	Encargados del desarrollo de cada módulo del sistema
Jefe de Proyecto	6 mes	Supervisa los avances de cada uno de los programadores.

Tabla 1: Recursos Humanos

### 1.7.2.4 Recursos de Entorno

<b>Hardware</b>		
Recurso	Duración	Descripción
Dos Equipos	6 mes	Para cada uno de los programadores y para la implantación del aplicativo web.
Cables de Red	2 mes	Para mantener en red a las computadoras de los programadores y para la conexión a internet dado que es un aplicativo web.
<b>Software</b>		
Recurso	Duración	Descripción
Java	2 mes	El entorno de desarrollo OpenSource que se utilizará será Eclipse Índigo
My SQL	1 mes	Se utilizará la versión MySQL 5.0 para la realización del proyecto.

Mozilla Firefox o Internet Explorer	2 mes	Como explorador web, en la versión 3.0 para Mozilla y para Internet Explorer la versión 9.
-------------------------------------	-------	--

Tabla 2: Recursos de Entorno

### 1.7.3 Factibilidad Económica

Parámetros	Gasto mensual por alumno
Aporte Intelectual	\$700,00
Alimentación	\$30,00
Transporte	\$30,00
Uso equipo computacional	\$50,00
Suministros de oficina	\$10,00
Internet	\$0,00
<b>Total por mes</b>	<b>\$720,00</b>

Tabla 3: Recursos de Entorno

#### 1.7.3.1 Tiempo Previsto

El tiempo previsto para el desarrollo del proyecto es de 6 meses.

	Gasto mensual por alumno	Número de alumnos	Tiempo de Implantación (meses)	Total
Aporte de Estudiantes	\$720,00	1	6	\$4320,00

Tabla 4: Tiempo Previsto

#### 1.7.3.2 Gastos que asumirá el Taller de Servicios RBS

ADQUISICIÓN DE LICENCIAS		
Tanto Java como MySQL son de distribución libre y dado que el Gobierno decretó software libre para instituciones públicas, no hay costos.		
	<b>Total</b>	<b>\$ 0,00</b>
ADQUISICIÓN DE SERVIDORES		
El proyecto será implementado sobre el hardware existente en el Taller de Servicios RBS, para lo cual no se deberá incurrir en ningún gasto adicional.		
	<b>Total</b>	<b>\$ 0,00</b>

Tabla 5: Gastos que asumirá el taller de servicios RBS

### 1.7.3.3 COSTO TOTAL DEL PROYECTO

<b>RESUMEN</b>	
GASTOS ASUMIDOS POR LOS ESTUDIANTE	<b>\$4320,00</b>
GASTOS ASUMIDOS POR LA INSTITUCIÓN	<b>\$ 0,00</b>
<b>COSTO TOTAL DEL PROYECTO</b>	<b>\$4320,00</b>

*Tabla 6: Costo Total del Proyecto*

ESPACIO EN BLANCO INTENCIONAL

## **CAPÍTULO 2: MARCO TEÓRICO**

El desarrollo de software es un proceso de ingeniería que incluye diversas tareas, y que necesita de conocimientos específicos de temas como tecnologías, lenguajes de programación, herramientas de desarrollo, diseño de software, etc. Sin embargo, estos conceptos no son siempre entendibles para los clientes o para los usuarios, por lo que es necesario brindar definiciones precisas para la comprensión del presente proyecto.

Por otro lado, es de igual importancia introducir al lector al marco de trabajo del proyecto que coordinará e integrará las diferentes cadenas de trabajo y facetas del desarrollo, esto es la *Programación Extrema* (XP Extrem Programming en inglés) y el *Lenguaje Unificado de Modelado* (UML **U**nified **M**odeling **L**anguage, en inglés) que en combinación proporcionarán una guía clara y práctica para ordenar, dirigir y fundamentar las actividades del desarrollo.

### **2.1 Metodología**

#### **2.1.1 Análisis de Metodologías Ágiles**

Hoy en día las aplicaciones que se desarrollan son muy diferentes a las que solían ser hace unos años. Las comunicaciones y la gran difusión de Internet han creado nuevas necesidades dentro de los sistemas software.

En este sentido, muchas definiciones de sistemas se están dando dentro del mundo de la ingeniería del software: sistemas multimedia, sistemas hipermedia, aplicaciones web, sistemas de información global, etc.

Para el desarrollo se tomará muy en cuenta las metodologías ágiles orientadas al desarrollo de Aplicaciones Web. Sin embargo, antes de comenzar a plantear el problema que va a ocupar el contenido de este trabajo, es necesario enmarcar el entorno de trabajo en el que estará. Para ello, es necesario hacer una serie de definiciones previas.

Como una reacción a estas metodologías, un nuevo grupo de metodologías ha surgido en los últimos años. Durante algún tiempo se conocían como metodologías ligeras, pero el término aceptado ahora es metodologías ágiles. Para mucha gente el encanto de estas metodologías ágiles es su reacción ante la burocracia de las metodologías monumentales. Estos nuevos métodos buscan un justo medio entre ningún proceso y demasiado proceso, proporcionando simplemente suficiente proceso para que el esfuerzo valga la pena.

El resultado de todo esto es que los métodos ágiles cambian significativamente algunos de los énfasis de los métodos ingenieriles. La diferencia inmediata es que son menos orientados al documento, exigiendo una cantidad más pequeña de documentación para una tarea dada. De muchas maneras son más bien orientados al código: siguiendo un camino que dice que la parte importante de la documentación es el código fuente.

Sin embargo, éste no es el punto importante sobre los métodos ágiles. La falta de documentación es un síntoma de diferencias mucho más profundas:

- Los métodos ágiles son adaptables en lugar de predictivos. Los métodos ingenieriles tienden a intentar planear una parte grande del proceso del software en gran detalle para un plazo largo de tiempo, esto funciona bien hasta que las cosas cambian. Así que su naturaleza es resistirse al cambio. Para los métodos ágiles, no obstante, el cambio es bienvenido. Intentan ser procesos que se adaptan y crecen en el cambio, incluso al punto de cambiarse ellos mismos.
- Los métodos ágiles son orientados a la gente y no orientados al proceso. La meta de los métodos ingenieriles es definir un proceso que funcionará bien con cualquiera que lo use. Los métodos ágiles afirman que ningún proceso podrá nunca maquillar las habilidades del equipo de desarrollo, de modo que el papel del proceso es apoyar al equipo de desarrollo en su trabajo. Explícitamente puntualizan el trabajar a favor de la naturaleza humana en

lugar de en su contra y enfatizan que el desarrollo de software debe ser una actividad agradable.

En las secciones siguientes se verán estas diferencias más en detalle, para discernir lo que es un proceso adaptable y centrado en la gente, sus beneficios y desventajas, y qué se debería usar: sea como desarrollador o como cliente de software.

## **¿Qué es una Metodología Ágil?**

Las Metodologías Ágiles o “ligeras” constituyen un nuevo enfoque en el desarrollo de software, mejor aceptado por los desarrolladores de e-projects que las metodologías convencionales (ISO-9000, CMM, etc.) debido a la simplicidad de sus reglas y prácticas, su orientación a equipos de desarrollo de pequeño tamaño, su flexibilidad ante los cambios y su ideología de colaboración.

## **Predictivo versus Adaptable**

### **Separación de Diseño y Construcción**

Aquí se enfatiza que hay que planear antes de construir. Los ingenieros trabajan sobre una serie de esquemas que indican precisamente qué hay que construir y cómo deben juntarse estas cosas.

Existen dos actividades fundamentalmente diferentes. El diseño, que es difícil de predecir y requiere personal caro y creativo, y la construcción que es más fácil de predecir. Después de adquirido el diseño, se planea la construcción. Una vez que se tiene el plan de construcción se desarrollará de una manera más predecible.

Así el acercamiento de muchas metodologías es: obtener un plan de trabajo predecible que pueda usar gente del más bajo nivel. Para hacerlo se requiere separar el plan de la construcción. Por consiguiente es necesario entender cómo hacer el diseño de software de modo que la construcción pueda ser sencilla una vez que el plan esté hecho.

¿Qué forma toma este plan? Para muchos, éste es el papel de notaciones de diseño como el UML. Se puede hacer todas las decisiones significativas usando UML, armar un plan de construcción y entonces dar planes a los programadores como una actividad de construcción.

Todo esto trae a la mente más preguntas. La primera es la cuestión de cuán difícil es conseguir un diseño UML en un estado que pueda entregarse a los programadores. El problema con un diseño tipo UML es que puede parecer muy bueno en el papel, pero resultar seriamente fallido a la hora de la programación. La única verificación que se hace con los diagramas UML es la revisión cuidadosa. Mientras esto es útil trae errores al diseño que sólo se descubren durante la codificación y pruebas. Incluso los diseñadores experimentados, se sorprenden a menudo cuando convierten dichos diseños en software.

### **La Impredecibilidad de los requisitos**

En el negocio de construcción de software los cambios en los requisitos son la norma, la pregunta es qué hacer al respecto.

Un problema con esto es que simplemente tratar de entender las opciones para los requisitos es duro. Es aún más duro porque la organización del desarrollo normalmente no proporciona la información del costo en los requisitos. El cliente termina solicitando algo que el vendedor no puede cotizar con exactitud. Sin una buena idea del costo.

Casi todo en el desarrollo de software depende de los requisitos. Si no se pueden obtener requisitos estables no se puede obtener un plan predecible.

### **Manejando un proceso orientado a la gente**

Uno de los elementos clave es la aceptación de un proceso en lugar de la imposición de un proceso. A menudo los procesos de software se imponen desde la gerencia. Como tales se les resiste a menudo, particularmente cuando la gerencia ha estado fuera del desarrollo activo un buen tiempo. Aceptar un proceso

requiere compromiso, y como tal se necesita el involucramiento activo de todo el equipo.

Esto termina con el resultado interesante de que sólo los desarrolladores pueden escoger seguir un proceso adaptable. Esto es particularmente cierto para la XP, que requiere mucha disciplina para ejecutarse.

Otro punto es que los desarrolladores deben poder tomar todas las decisiones técnicas. XP llega al corazón de esto cuando en su proceso de planeación establece que sólo los desarrolladores pueden estimar cuánto tiempo tomará hacer un trabajo.

Tal liderazgo técnico es un gran cambio para muchas personas en posiciones gerenciales. Tal acercamiento requiere compartir una responsabilidad donde desarrolladores y gerencia tienen un mismo lugar en la dirección del proyecto. La gerencia aun juega un papel, pero reconoce la pericia de los desarrolladores.

### **El papel del liderazgo de negocio**

Pero los técnicos no pueden hacer el proceso entero ellos. Necesitan una guía en las necesidades del negocio. Esto lleva a otro aspecto importante de los procesos adaptables: necesitan un contacto muy íntimo con los expertos del negocio.

Esto va más allá del compromiso de negocios en la mayoría de los proyectos. Los equipos ágiles no pueden existir con una comunicación ocasional. Necesitan un acceso continuo a los expertos del negocio. Además este acceso no es algo que se maneje a nivel gerencial, es algo que está presente para cada desarrollador. Como los desarrolladores son profesionales capaces en su propia disciplina, pueden trabajar como iguales con otros profesionales de otras disciplinas.

## **Las Metodologías Ágiles valoran:**

1. Al individuo y las interacciones en el equipo de desarrollo más que a las actividades y las herramientas.
2. Desarrollar software que funciona más que conseguir una buena documentación, implica minimalismo respecto del modelado y la documentación del sistema.
3. La colaboración con el cliente más que la negociación de un contrato.
4. Responder a los cambios más que seguir estrictamente una planificación.

## **¿Por qué surgen las Metodologías Ágiles?**

1. Dificultad para implantar metodologías tradicionales. Sofisticadas herramientas CASE y notaciones (UML).
2. Una solución a medida para un segmento importante de proyectos de desarrollo de software.
3. Pugna entre comunidades / gurúes.
4. Aceptar el cambio.

## **Los principios que se establecieron en el manifiesto de las MA son:**

1. La prioridad principal es satisfacer al cliente mediante tempranas y continuas entregas de software que le reporte un valor.
2. Dar la bienvenida a los cambios. Los MA capturan los cambios para que el cliente tenga una ventaja competitiva.

3. Entregar frecuentemente software que funcione, desde un par de semanas a un par de meses, con el menor intervalo de tiempo posible entre una entrega y la siguiente.
4. La gente del negocio y los desarrolladores deben trabajar juntos a lo largo del proyecto.
5. Construir el proyecto entorno a individuos motivados. Darles el entorno y el apoyo que necesitan y confiar en ellos para conseguir el trabajo.
6. El diálogo cara a cara es el método más eficiente y efectivo para comunicar información dentro de un equipo de desarrollo.
7. El software que funciona es la medida principal de progreso.
8. Los procesos ágiles promueven un desarrollo sostenible. Los promotores, desarrolladores y usuarios deberían ser capaces de mantener una paz constante.
9. La atención continua a la calidad técnica y al buen diseño mejora la agilidad.
10. La simplicidad es esencial.
11. Las mejores arquitecturas, requisitos y diseños surgen de los equipos organizados por sí mismos.
12. En intervalos regulares, el equipo reflexiona respecto de cómo llegar a ser más efectivo, y según esto ajusta su comportamiento.

## Cuadro comparativo de Metodologías Ágiles y no Ágiles

<b>Metodología Ágil</b>	<b>Metodología No Ágil (Tradicional)</b>
Pocos artefactos.	Más artefactos.
Pocos roles.	Más roles.
No existe un contrato tradicional o al menos es bastante flexible.	Existe un contrato prefijado.
El cliente es parte del equipo de desarrollo.	El cliente interactúa con el equipo de desarrollo mediante reuniones.
Grupos pequeños (< 10 integrantes) y trabajando en el mismo sitio.	Grupos grandes.
Menos énfasis en la arquitectura.	La arquitectura es esencial.
Énfasis en la comunicación del grupo.	Énfasis en los roles.
Proceso menos controlado, con pocos principios.	Proceso mucho más controlado, con numerosas políticas/normas.
Ofrecen una buena solución para entornos cambiantes.	Cierta resistencia a los cambios.

*Tabla7: Cuadro comparativo de metodologías ágiles y no ágiles*

## Principales Metodologías Ágiles

- Crystal Methodologies, Alistair Cockburn,  
[www.crystallmethodologies.org](http://www.crystallmethodologies.org)
- SCRUM, Ken Schwaber & Jeff Sutherland,  
[www.controlchaos.com](http://www.controlchaos.com)
- DSDM (Dynamic Systems Development Method),  
[www.dsdm.org](http://www.dsdm.org)
- Lean Programming, Mary Poppendieck,  
[www.poppendieck.com](http://www.poppendieck.com)
- FDD (Feature-Driven Development), Peter Coad & Jeff De Luca,  
[www.nebulon.com/fdd](http://www.nebulon.com/fdd), [www.coad.com/peter/#fdd](http://www.coad.com/peter/#fdd)
- Extreme Programming(XP), Kent Beck  
[www.extremeprogramming.org](http://www.extremeprogramming.org), [www.xprogramming.com](http://www.xprogramming.com)
- Adaptative Software Development, Jim Highsmith  
[www.adaptivesd.com](http://www.adaptivesd.com)

### 2.1.1.1 Metodología SCRUM

Scrum se enfoca en el hecho de que procesos definidos y repetibles sólo funcionan para atacar problemas definidos y repetibles con gente definida y repetible en ambientes definidos y repetibles.

Scrum divide un proyecto en iteraciones (que ellos llaman carreras cortas) de 30 días. Antes de que comience una carrera se define la funcionalidad requerida para esa carrera y entonces se deja al equipo para que la entregue. El punto es estabilizar los requisitos durante la carrera.

Sin embargo, la gerencia no se desentiende durante la carrera corta. Todos los días el equipo sostiene una reunión corta (quince minutos), llamada scrum, donde el equipo discute lo que hará al día siguiente. En particular muestran a los bloques de la gerencia: los impedimentos para progresar que se atraviesan y que la gerencia debe resolver. También informan lo que se ha hecho para que la gerencia tenga una actualización diaria de dónde va el proyecto.

La literatura de Scrum se enfoca principalmente en la planeación iterativa y el seguimiento del proceso. Es muy cercana a las otras metodologías ágiles en muchos aspectos y debe funcionar bien con las prácticas de código de XP.

### **Ventajas**

- Incentivar la formación de equipos de trabajos autosuficientes y disciplinados.
- Tener un alcance acotado y viable.
- Se puede visualizar el proyecto día a día.
- Es posible el ajuste de funcionalidades en base a las necesidades de negocio del cliente.
- Permite la entrega de un producto funcional al finalizar cada Sprint (iteración).

### **Desventajas**

- Tal vez sea necesario complementarlo con otras metodologías ágiles, como puede ser XP en ciertos casos de ser necesario.
- No es muy recomendable para todos los proyectos.
- No genera mucha documentación de otras metodologías

#### **2.1.1.2 Metodología XP**

De todas las metodologías ágiles, ésta es la que ha recibido más atención. Esto se debe en parte a la notable habilidad de los líderes XP. También se debe a

la habilidad de atraer a las personas a este acercamiento, y tomar un papel principal en él.

De algunas maneras, la popularidad de XP se ha vuelto un problema, pues ha acaparado la atención fuera de las otras metodologías y sus valiosas ideas.

La metodología XP empieza con cuatro valores:

- Comunicación,
- Retroalimentación,
- Simplicidad, y
- Coraje.

Construye sobre ellos una docena de prácticas que los proyectos XP deben seguir. Muchas de estas prácticas son técnicas antiguas, tratadas y probadas, aunque a menudo olvidadas por muchos, incluyendo la mayoría de los procesos planeados.

Una de las más llamativas, así como inicialmente atractiva, es su fuerte énfasis en las pruebas. XP pone la comprobación como el fundamento del desarrollo, con cada programador escribiendo pruebas cuando escriben su código de producción. Las pruebas se integran en el proceso de integración continua y construcción lo que rinde una plataforma altamente estable para el desarrollo futuro.

En esta plataforma XP construye un proceso de diseño evolutivo que se basa en refactorizar un sistema simple en cada iteración. Todo el diseño se centra en la iteración actual y no se hace nada anticipadamente para necesidades futuras. El resultado es un proceso de diseño disciplinado, es más, combina la disciplina con la adaptabilidad de una manera que indiscutiblemente la hace la más desarrollada de entre todas las metodologías adaptables.

## **Ventajas**

- Los equipos de trabajo en esta metodología pueden ser numerosos.
- Los desarrollos serán de calidad ya que son probados frecuentemente en todos los procesos.
- Garantiza la satisfacción del cliente ya que el mismo se encuentra tan involucrado como los desarrolladores en las constantes validaciones que se hacen con los release.
- Funciona muy bien cuando los requisitos son inciertos y volátiles.

## **Desventajas**

- En caso de contrataciones en costos fijos, no existe la interacción que se necesita con el cliente.
- Si los requisitos están bien definidos no es necesario implementar una metodología ágil.
- No existe confianza en las capacidades del equipo de trabajo
- Un equipo de trabajo menor a 6 personas.

### **2.1.1.3 Otras Metodologías**

#### **Desarrollo Manejado por Rasgos (FDD)**

El FDD tiene cinco procesos. Los primeros tres se hacen al principio del proyecto.

- Desarrollar un Modelo Global.
- Construir una Lista de los Rasgos.
- Planear por Rasgo.
- Diseñar por Rasgo.
- Construir por Rasgo.

Los últimos dos se hacen en cada iteración. Cada proceso se divide en tareas y se dá un criterio de comprobación.

Los desarrolladores entran en dos tipos:

- dueños de clases, y
- programadores jefe.

Los programadores jefe son los desarrolladores más experimentados. A ellos se les asignan rasgos a construir. Sin embargo, ellos no los construyen solos. Sólo identifican qué clases se involucran en la implantación de un rasgo y juntan a los dueños de dichas clases para que formen un equipo para desarrollar ese rasgo. El programador jefe actúa como el coordinador, diseñador líder y mentor, mientras los dueños de clases hacen gran parte de la codificación del rasgo.

### **Ventajas**

- Es un método de desarrollo de ciclos cortos que se concentra en la fase de diseño y construcción.

### **Crystal Methodologies**

Se trata de un conjunto de metodologías para el desarrollo de software caracterizadas por estar centradas en las personas que componen el equipo (de ellas depende el éxito del proyecto) y la reducción al máximo del número de artefactos producidos. El desarrollo de software se considera un juego cooperativo de invención y comunicación, limitado por los recursos a utilizar. El equipo de desarrollo es un factor clave, por lo que se deben invertir esfuerzos en mejorar sus habilidades y destrezas, así como tener políticas de trabajo en equipo definidas.

Crystal aconseja que el tamaño del equipo sea reducido (Pocos componentes).

La mejora de la comunicación entre los miembros del equipo del proyecto:

*Mismo lugar de trabajo* → Disminuye el coste de la comunicación.

*Mejora individual* → Mejora global del equipo.

Crystal da vital importancia a las personas que componen el equipo de un proyecto, y por tanto sus puntos de estudio son:

- Aspecto humano del equipo.
- Tamaño de un equipo (número de componentes).
- Comunicación entre los componentes.
- Distintas políticas a seguir.
- Espacio físico de trabajo.

Los siete valores o propiedades de Crystal Clear son:

- Entrega frecuente.
- Comunicación osmótica. (Todos juntos en el mismo cuarto.)
- Mejora reflexiva. (Tomarse tiempo para reanalizar lo que se está haciendo)
- Seguridad personal. ( Hablar cuando algo molesta)
- Foco. (Saber lo que se está haciendo y tener la tranquilidad y el tiempo para hacerlo.)
- Fácil acceso a usuarios expertos.
- Ambiente técnico con prueba automatizada, management de configuración e integración frecuente.

En conclusión se podría decir que:

- Cuantas más personas estén implicadas, más grande debe ser la metodología.

- Si el proyecto tiene mucha densidad, un error no detectado puede ser crítico.
- El aumento de tamaño o densidad añade un coste considerable al proyecto.
- La forma más eficaz de comunicación es la interactiva (cara a cara).

### **Ventajas**

- Dispone de un código de colores para poder identificar la complejidad de la metodología mientras más oscuro el color más complejidad se va a presentar.
- Puede combinarse con otras metodologías.
- Total seguridad de lo que se entrega.
- La comunicación cumple el principal rol.
- Determina el tamaño del sistema por el número de personas empleadas en su desarrollo.

### **Desventajas**

- Es altamente recomendable para equipos pequeños.
- Si el proyecto tiene un error no detectado puede ser crítico.

#### **2.1.1.4 Cuadro Comparativo**

A continuación se muestra un cuadro comparativo de las metodologías ágiles que se analizó para ver cuál es la que más convendría tomar para el aplicativo.

En el siguiente cuadro se muestra algunas consideraciones que se han tenido a la hora de realizar estas clasificaciones.

	<b>FDD</b>	<b>SCRUM</b>	<b>XP</b>	<b>CRISTAL</b>
<b># Papers</b>	3	43	+100	0
<b>Google</b>	31200	3420000	1190000	244000
<b>Yahoo</b>	177000	5120000	4470000	2930000
<b>Live</b>	68000	1970000	1470000	724000
<b>Libros en Español</b>	0	2	2	0
<b>Libros en Otro Idioma</b>	1	4	+20	1
<b>Certificación/Training</b>	FDD Certified Training	Scrum Certified Training	Training	TRAINING
<b>Comunidades</b>	Agile Alliance	Agile Alliance	-	Agile Alliance
<b>Presencia empresarial</b>	-	Yahoo, Google, etc...	Chrysler, Sabre, Airlines, CSEE Transport, etc ...	-
<b>Proyectos realizados</b>	-	Desarrollos Internos Principalmente	Control Automatizado de trenes	Proyecto Winifred

Tabla 8: Cuadro de consideraciones a analizar para escoger una metodología

1. La metodología con mayor presencia en Internet.

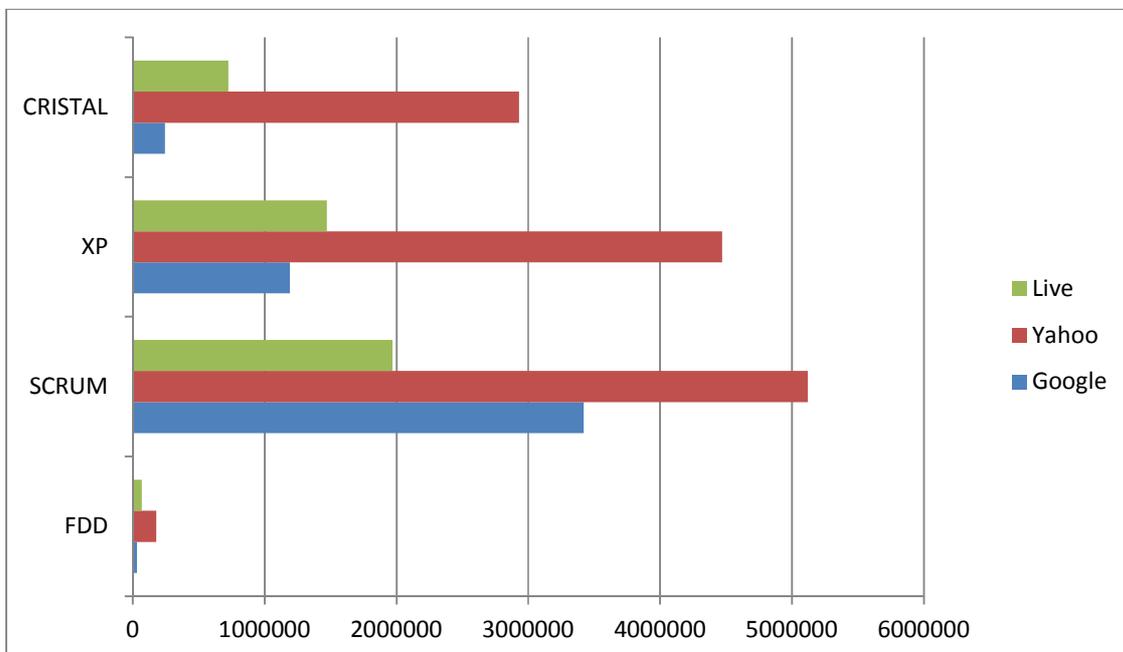


Gráfico 1: La metodología con mayor presencia en Internet.

Según el número de resultados obtenidos en las búsquedas por Yahoo, Google y Microsoft Live, las metodologías con mayor presencia en la red y en este orden son:

1. Scrum
2. Extreme Programming (XP)
3. Crystal Methods
4. Feature Driven Development (FDD)

### 2. La metodología mejor documentada.

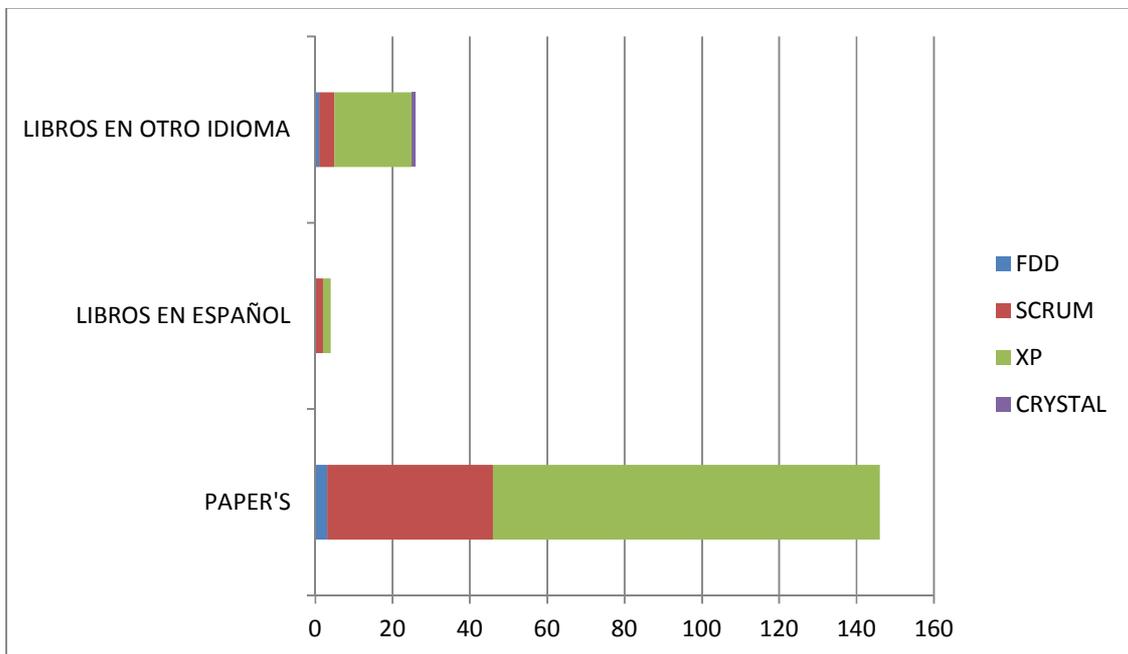


Gráfico 2: La metodología mejor documentada.

Las metodologías mejor documentadas son:

1. Extreme Programming
2. Scrum
3. Crystal
4. Feature Driven Development (FDD)

### 3. Metodologías certificadas y con training.

Metodologías con certificación:

Feature Driven Development tiene FDD Certified.

Scrum dispone de Scrum Certified.

Metodologías con training:

Crystal methods.

Feature Driven Development.

Scrum.

Extreme Programing.

Con certificación y training:

Todas las metodologías certificadas ofrecen training.

#### *4. Metodologías con comunidades.*

La mayoría pertenecen a la Agile Alliance, pero algunas han montado auténticas comunidades y alianzas a su alrededor.

Metodologías asociadas a la Agile Alliance:

Crystal methods.

Feature Driven Development.

Scrum.

Extreme Programing.

Metodologías con comunidades o alianzas diferentes:

Scrum, con Scrum alliance.

#### *5. Metodología más utilizada por empresas. Presencia empresarial.*

Como se ha podido observar es realmente complicado encontrar ejemplos de proyectos realizados en una empresa privada y con una metodología en concreto. Por lo que los resultados obtenidos no se tienen en cuenta en la selección inicial de las metodologías.

## *6. Metodología más utilizada en proyectos software.*

Exactamente igual que el punto anterior.

### **2.1.1.5 Justificación**

Las metodologías ágiles, son procesos de desarrollo de software que intentan evitar los tortuosos y burocráticos caminos de las metodologías tradicionales, enfocados en la gente y en el resultado.

Se orientó en este tipo de metodologías ya que evitará la extensa documentación y un proceso demasiado complejo que conlleva mucho tiempo. En cambio, se podría realizar el software en ciclos cortos que muestren un verdadero avance.

Entre las diferentes metodologías ágiles existentes se ha escogido XP ya que plantea de forma más clara los procesos metodológicos a seguir para la construcción del software, siendo además sin exceso de documentación la mejor documentada y la de mayor uso en la actualidad.

Se eligió ésta metodología debido al tipo proyecto, en este caso es un software mediano el cual no necesita utilizar las tradicionales metodologías que a pesar de ser muy conocidas y difundidas no ameritan su uso. Se le considera a XP como una metodología ágil ya que permite cambios rápidos de requerimientos de usuario.

### **2.1.2 Aplicación de metodología a implantar**

#### **2.1.2.1 Metodología de Desarrollo de Software XP (Extreme Programming)**

#### **2.1.2.2 Definición**

Es una de las metodologías de desarrollo de software más exitosas en la actualidad utilizadas para proyectos de corto plazo, corto equipo, La metodología consiste en una programación rápida, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto.

Es una metodología ligera para el desarrollo de software que simplifica la comunicación con el cliente y ayuda a la reutilización de código desarrollado.

Los objetivos de XP en realidad profundizan en la satisfacción del cliente, proporcionándole el software que necesita y cuando lo requiera, por lo tanto se debe responder muy rápido a las necesidades del cliente incluyendo cambios al final del ciclo de la programación. Esto es posible lograrlo, potenciando al máximo el trabajo en grupo

### **2.1.2.3 Características**

- Énfasis en el desarrollo del software más que una buena documentación.
- Empieza en pequeñas fases que a su vez se autoalimentan según las necesidades del usuario.
- Se maneja con pequeños cambios que a su vez se convierten en sustitutos de un anterior.
- El costo del cambio no depende de la fase o etapa.
- No se agregan funcionalidades sin ser necesarias.
- Corrección de errores antes de agregar nuevas funcionalidades.
- Entregas frecuentes.
- El usuario o cliente es bien definido para formar parte del equipo de trabajo.

### **2.1.2.4 Utilidad**

- *Simplicidad en las soluciones implementadas*

En esta parte se debe hacer siempre la pregunta:

¿Qué es lo más simple que puede funcionar?

En realidad es muy importante tratar de hacer las cosas lo más simple y conciso que se pueda, para ahorrar tiempo ya que de eso se trata esta metodología.

- *Comunicación*

La comunicación en esta metodología es la base del sistema para lograr un sistema de calidad y poder satisfacer los requerimientos solicitados por el cliente.

- *Retroalimentación*

Mientras más retroalimentación exista más fiable será el sistema, la retroalimentación actúa junto con la sencillez y la comunicación. Cuanto más simple un sistema más fácil de probar será. Escribir pruebas, orienta como simplificar un sistema, hasta que las pruebas funcionen.

- *Coraje (si funciona ... mejóralo)*

Asumir retos, ser valientes antes los problemas y afrontarlos.

### **2.1.2.5 Esquema de la metodología XP (Extreme Programming)**

El ciclo de desarrollo consiste (a grandes rasgos) en los siguientes pasos:

1. El cliente define el valor de negocio a implementar.
2. El programador estima el esfuerzo necesario para su implementación.
3. El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo.
4. El programador construye ese valor de negocio.
5. Vuelve al paso 1.

En todas las iteraciones de este ciclo tanto el cliente como el programador aprenden. No se debe presionar al programador a realizar más trabajo que el estimado, ya que se perderá calidad en el software o no se cumplirán los plazos.

De la misma forma, el cliente tiene la obligación de manejar el ámbito de entrega del producto, para asegurarse que el sistema tenga el mayor valor de negocio posible con cada iteración.

El ciclo de vida ideal de XP consiste de seis fases:

1. Exploración,
2. Planificación de la entrega (Release),
3. Iteraciones,
4. Producción,
5. Mantenimiento y
6. Muerte del proyecto.

### 2.1.2.6 Ciclo de vida XP (Extreme Programming)

#### 2.1.2.7 Fases

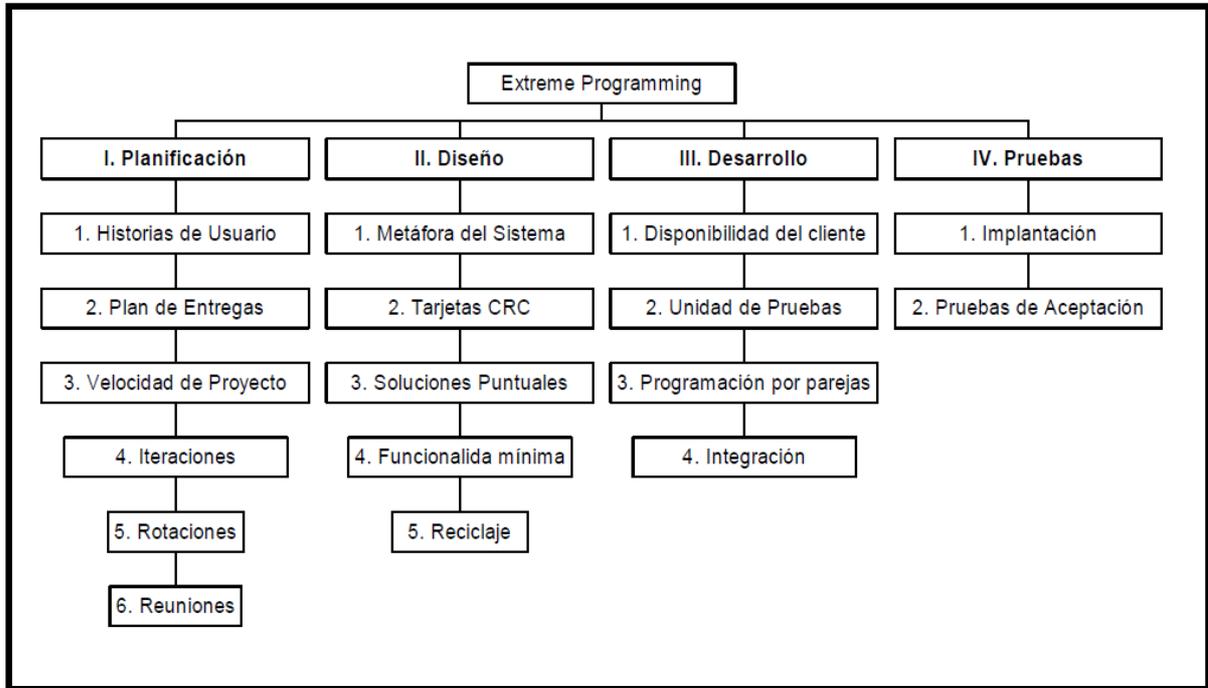


Gráfico 3: Facetas XP

##### 2.1.2.7.1 Fase 1ª: Planificación del proyecto

- Historia de Usuario

En esta metodología de XP, el primer paso es definir la historia del usuario con el cliente. Es similar a los casos de uso pero con algunas variantes: Consta de 3 o 4 líneas escritas por el cliente en lenguaje no técnico y sin profundizar en detalles, son usados para la estimación de tiempo de desarrollo de la parte de aplicación que se describe. También son usadas en la fase de pruebas, para la verificación de cumplimiento del programa.

Al implementar una Historia de Usuario se reúne el cliente y el desarrollador para concretar y detallar el cumplimiento de dicha historia; tomando en cuenta que el tiempo estimado para una Historia de Usuario es de 1 a 3 semana.

- Roles XP

Aunque en otras fuentes de información aparecen algunas variaciones y extensiones de roles XP, en este apartado se describe los roles de acuerdo con la propuesta original de Beck.

*Programador;* escribe las pruebas unitarias y produce el código del sistema. Debe existir una comunicación y coordinación adecuada entre los programadores y otros miembros del equipo.

*Cliente;* escribe las historias de usuario y las pruebas funcionales para validar su implementación. Además, asigna la prioridad a las historias de usuario y decide cuáles se implementan en cada iteración centrándose en aportar mayor valor al negocio. El cliente es sólo uno dentro del proyecto pero puede corresponder a un interlocutor que está representando a varias personas que se verán afectadas por el sistema.

*Encargado de pruebas (Tester);* ayuda al cliente a escribir las pruebas funcionales. Ejecuta las pruebas regularmente, difunde los resultados en el equipo y es responsable de las herramientas de soporte para pruebas.

*Encargado de seguimiento (Tracker);* proporciona realimentación al equipo en el proceso XP. Su responsabilidad es verificar el grado de acierto entre las estimaciones realizadas y el tiempo real dedicado, comunicando los resultados para mejorar futuras estimaciones.

También realiza el seguimiento del progreso de cada iteración y evalúa si los objetivos son alcanzables con las restricciones de tiempo y recursos presentes.

Determina cuándo es necesario realizar algún cambio para lograr los objetivos de cada iteración.

*Entrenador (Coach);* Es responsable del proceso global. Es necesario que conozca a fondo el proceso XP para proveer guías a los miembros del equipo de forma que se apliquen las prácticas XP y se siga el proceso correctamente.

*Consultor;* Es un miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto. Guía al equipo para resolver un problema específico.

*Gestor (Big boss)*; Es el vínculo entre clientes y programadores, ayuda a que el equipo trabaje efectivamente creando las condiciones adecuadas. Su labor esencial es de coordinación.

- *Plan de Entrega*

En esta fase se indican las Historias de Usuario que se crearán para cada versión del programa y las fechas en que serán publicadas dichas versiones, aquí el cliente y el desarrollador establecerán los tiempos de implementación de dichas historias, la prioridad de implementación y las historias que serán implementadas en cada versión del programa.

Los factores más importantes que se deben tener en cuentas son:

- ✓ Objetivos a cumplir.
- ✓ Tiempo que tardará en desarrollarse y publicarse las versiones del programa.
- ✓ Número de personas implicadas en el desarrollo.
- ✓ Cómo se evaluará la calidad del trabajo realizado.

- *Iteraciones*

Para esta metodología se debe dividir en iteraciones de 3 semanas de duración aproximadamente. Al comienzo de cada iteración los clientes deben seleccionar las historias de usuario definidas en el "Plan de Entrega" que serán implementadas. Estas historias de usuario son divididas en tareas de entre 1 y 3 días de duración que se asignarán a los programadores.

- *Velocidad del Proyecto*

Es una medida donde se representa la velocidad de desarrollo del proyecto, para esto se debe contar el número de Historia de Usuarios que se puede implementar en una iteración; con esto se conocerá el número de historias que se pueden desarrollar en las distintas iteraciones.

Usando la Velocidad del Proyecto se controlará que todas las tareas se puedan desarrollar en el tiempo que se dispone la iteración, para esto se debe reevaluar esta medida cada 3 o 4 iteraciones y en el caso de no ser adecuada se debe renegociar con el cliente.

- Programación en Pareja (Rotaciones)

Esta metodología X.P. aconseja la programación en pareja ya que incrementa la productividad y la calidad del software desarrollado, involucra a dos programadores ya que mientras uno codifica el otro analiza si el método o función es la adecuada o está bien diseñada.

- Reuniones Diarias

Es necesario que se lo haga de ser posible diariamente para exponer inquietudes y aportar con ideas de forma conjunta, en estas reuniones se debe tomar en cuenta que todos deben tener voz y voto para mejoras.

#### **2.1.2.7.2 Fase 2ª: Diseño**

- Diseño Simple

Conseguir un diseño simple y sencillo. Hacer lo menos complicado posible para que sea entendible y aplicable, para poder obtenerlo en el menor tiempo posible y con menor esfuerzo al desarrollarlo.

- Glosario de Términos

Realizar un glosario de términos y especificación de nombres de métodos y clase.

- Riesgos

Al existir problemas, se sugiere una pareja de desarrolladores, para reducir el riesgo que significará ese problema.

- Funcionalidad Extra

Nunca debe añadirse esta funcionalidad, ya que no es óptima para esta metodología.

- Refactorizar

Revisar de nuevo el código para optimizar su funcionamiento.

- Tarjetas C.R.C. (Class, Responsibilities and Collaboration)

Ayudan a olvidar la programación procedural orientando a la programación orientada a objetos.

Donde se describe: La clase donde pertenece el objeto, Responsabilidad u objetivos que debe cumplir el objeto y las clases que colaboran con cada responsabilidad

### **2.1.2.7.3 Fase 3ª: Codificación**

Como se dijo en la introducción de esta metodología el cliente cumple un papel muy importante y mucho más en esta tercera fase de codificación, ya que ellos son los que crean las historias de usuario y negocian el tiempo en que estas historias serán implementadas.

El usuario deberá especificar detalladamente lo que cada historia de usuario hará y también deberá estar presente al momento de realizar los test de verificación donde comprobará su correcto funcionamiento.

La codificación deberá ser realizada desacuerdo a estándares ya establecidos, para facilitar su comprensión.

Es muy importante el llevar test que comprueben el correcto funcionamiento; esto ayuda en el desarrollo de dicho código; crear estos tests ayudan a saber qué es exactamente lo que tiene que hacer el código a implementar y saber que una vez implementado pasará dicho test sin problema, se deberá realizar pequeños test para cada unidad desarrollada así poco a poco se consigue un desarrollo que cumpla todos los requisitos especificados.

XP sugiere un modelo de trabajo usando repositorios de código dónde las parejas de programadores publican cada pocas horas sus códigos implementados y corregidos junto a los test que deben pasar. De esta forma el resto de programadores trabajan con código ajeno utilizando siempre las últimas versiones, adquiridas de un repositorio de código.

Se propone un desarrollo colectivo en donde todos los programadores están implicados en todas las tareas; cualquiera puede modificar cualquier clase y subirla al repositorio de código.

El hecho de que cualquier programador puede modificar cualquier código no implica ningún riesgo ya que para que el código pueda ser publicado en el

repositorio debe pasar el test de funcionamiento.

La optimización del código siempre se debe dejar para el final, primero hay que hacer que funcione correctamente para después optimizar.

En el caso que se necesite más tiempo para la culminación del proyecto, X.P., propone realizar un nuevo “Release Plan” para concretar los nuevos tiempos de publicación y de velocidad del proyecto.

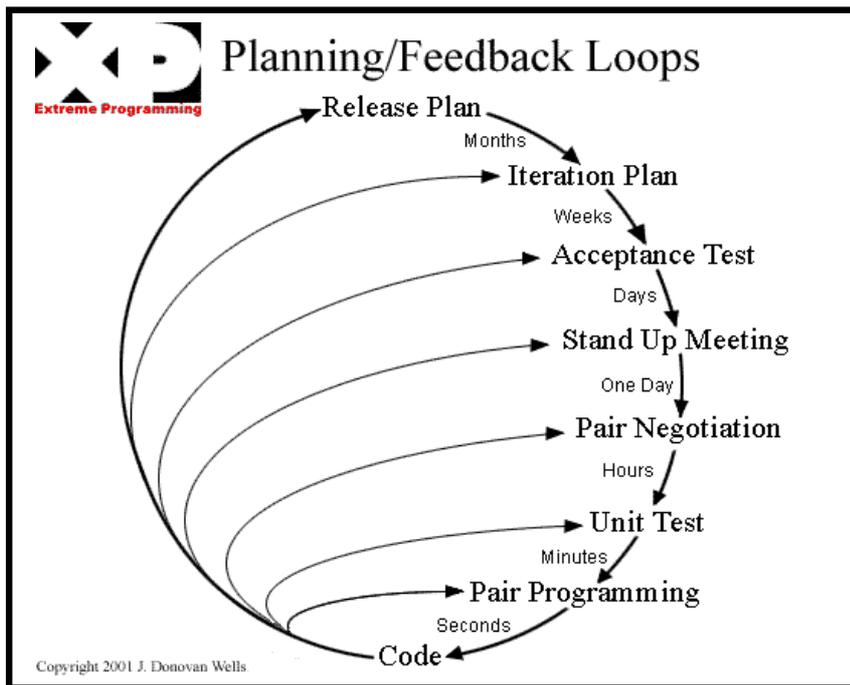


Gráfico 4: Tiempos en XP

#### 2.1.2.7.4 Fase 4ª: Pruebas

Se deben crear las aplicaciones que realizarán los test con un entorno de desarrollo específico para test.

Los distintos test se deben subir al repositorio de código acompañados del código de verificación, ningún código puede ser publicado en el repositorio sin haber pasado primero por su test de funcionamiento respectivamente.

Se debe tomar en cuenta que se realizan dos tipos de test, uno que es para la verificación de cada versión, llamado test de funcionamiento, es decir para el

control de cada historia y su garantía de funcionamiento; el otro llamado test de aceptación que es el último que se lo realiza y con verificación del usuario o cliente para comprobar el correcto funcionamiento del mismo y posteriormente implementarlo.

## **2.2 *Plataforma de Desarrollo***

### **2.2.1 Applications Web (web application, webapp)**

#### **2.2.1.1 Introducción**

La WWW o World Wide Web o, de forma coloquial, la web, se ha convertido, con el tiempo y junto con el correo electrónico, en el principal servicio de Internet.

Ha dejado de ser en el transcurso de los tiempos, solamente un servicio de información donde contenía información estática, como una biblioteca; para ser ahora en la actualidad un servicio en donde se puede acceder a muchas prestaciones, funciones e infinidad de servicios.

El éxito de la web se basa en tres factores fundamentales: el protocolo HTTP, el lenguaje HTML y CGI. El primero permite implantar un sistema de comunicación sencillo que facilita enviar cualquier fichero; el segundo, el lenguaje HTML, permite la creación de páginas enlazadas o relacionadas.

El siguiente paso es actualmente la creación de páginas dinámicas, que permite que lo mostrado tenga una funcionalidad dinámica, este método es conocido como CGI ("Common Gateway Interface"), donde define un mecanismo de paso de información entre el servidor y ciertos sistemas externos, los servidores web permiten su uso debido a su sencillez, ya que dan total libertad para el uso de cualquier lenguaje de programación.

#### **2.2.1.2 Definición**

En la Ingeniería de Software las aplicaciones web son aplicaciones que se acceden vía web por una red mediante internet o una intranet; es decir, es una aplicación software que se codifica en un lenguaje soportado por los navegadores web en la que se confía la ejecución al navegador, generando dinámicamente una serie de páginas en un formato estándar, como HTML o XHTML, soportados por

los navegadores web comunes; las aplicaciones web son populares debido a la facilidad para actualizar y mantenerlas sin tener que distribuir e instalar software a miles de usuarios.

Las aplicaciones Web pueden utilizar lenguajes diferentes interpretados en el lado del cliente, tales como JavaScript, para añadir elementos dinámicos a la interfaz de usuario, además esta aplicación permite al usuario acceder a los datos de modo interactivo, gracias a que la página responderá a cada una de sus acciones, como por ejemplo rellenar y enviar formularios, participar en blogs diversos y acceder a gestores de base de datos de todo tipo.

Las aplicaciones Web son de tres capas: como se muestra en la figura:

1. El navegador Web es la primera capa de comunicación con el usuario.
2. La tecnología Web dinámica es la capa dos donde se usan: PHP, Java Servlets o ASP, etc.
3. Tercera capa: Base de Datos.

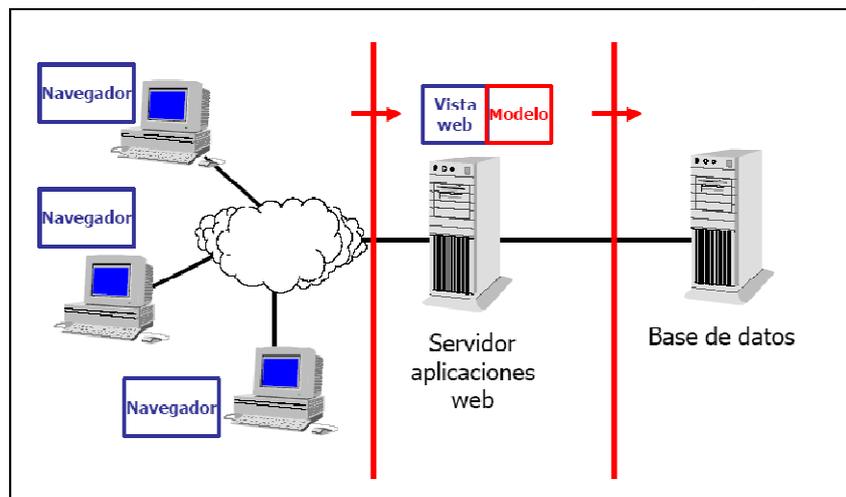


Gráfico 5: Aplicaciones Web

### 2.2.1.3 Características

- El cliente puede acceder fácilmente a estas aplicaciones empleando un navegador web o similar.
- Si es por internet, el cliente puede entrar desde cualquier lugar del mundo donde tenga un acceso a internet.

- Pueden existir miles de clientes pero una única aplicación instalada en un servidor, por lo tanto se puede actualizar y mantener una única aplicación y todos sus clientes.
- Emplean tecnologías como Java, JavaFX, JavaScript, DHTML, Flash, Ajax y otras más que dan gran potencia a la interfaz de usuario.

#### **2.2.1.4 Clasificación**

##### Internet

El Internet es una gran red descentralizada que funciona a nivel mundial brindando información, productos y servicios a cualquiera que tenga acceso a él. Para esto el Internet está compuesto de páginas, sitios y portales web, los cuales a su medida ofrecen distintos servicios.

- Página web

Una página web es un documento HTML (*Lenguaje de Mercado de Hipertexto*) /XHTML (*Lenguaje de Mercado de Hipertexto Extensible*) accesible generalmente mediante el protocolo HTTP (*Protocolo de Tránsito de Hipertexto*) de Internet, que constituye una fuente de información accesible mediante un navegador de Internet.

- Sitio web

Un sitio web es un conjunto de páginas web relacionadas entre sí y comunes a un dominio de Internet, (tanto por la temática: afines, como físicamente: hiperenlaces) albergadas generalmente en un mismo servidor. Puede incluir texto, gráficos, ficheros de audio y vídeo, entre otros. Cada sitio web tiene una página de inicio (*home page*) que es el primer documento que ve el usuario cuando entra en el sitio web poniendo el nombre del dominio (dirección URL *Localizador Uniforme de Recurso*) de ese sitio web en un navegador.

- Aplicaciones Web 2.0

La Web 2.0 es la representación de la evolución de las aplicaciones tradicionales hacia aplicaciones web enfocadas al usuario final, la Web 2.0 no

precisamente una tecnología. Se trata de aplicaciones que generen colaboración y de servicios que reemplacen las aplicaciones de escritorio.

Se utiliza Aplicaciones Web 2.0 en:

- ✓ Aplicaciones para expresarse/crear y publicar/difundir: blog.
- ✓ Aplicaciones para publicar/difundir y buscar información: podcast, YouTube.
- ✓ Aplicaciones para buscar/acceder a información de la que interesa estar siempre bien actualizados: RSS, XML, Atom, Bloglines, GoogleReader, buscadores especializados.
- ✓ Redes sociales: BSCW, Ning, Second Life, Twitter.

- Aplicaciones Multinivel

- ✓ Los sistemas típicos cliente/servidor pertenecen a la categoría de las aplicaciones de dos niveles.
- ✓ La aplicación reside en el cliente mientras que la base de datos se encuentra en el servidor.
- ✓ La capa intermedia es el código que el usuario invoca para recuperar los datos deseados.
- ✓ La capa de presentación recibe los datos y los formatea para mostrarlos adecuadamente.
- ✓ Esta división entre la capa de presentación y la de la lógica permite una gran flexibilidad a la hora de construir aplicaciones, ya que se pueden tener múltiples interfaces sin cambiar la lógica de la aplicación.
- ✓ La tercera capa consiste en los datos que gestiona la aplicación. Estos datos pueden ser cualquier fuente de información como una base de datos o documentos XML.

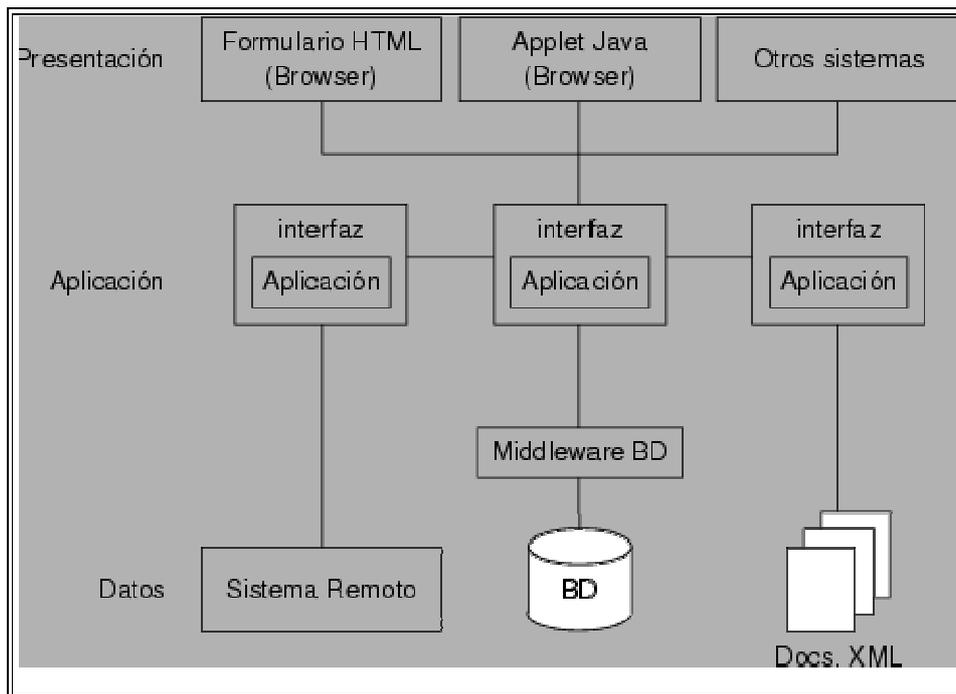


Gráfico 6: Capas de las Aplicaciones Web

- Servidores Web

Un servidor web es un programa que implementa el protocolo HTTP . Este protocolo pertenece a la capa de aplicación del modelo OSI y está diseñado para transferir lo que se llama hipertextos, páginas web o páginas HTML: textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de música.

El Servidor web se ejecuta continuamente en un ordenador, manteniéndose a la espera de peticiones por parte de un cliente (un navegador web) y que responde a estas peticiones adecuadamente, mediante una página web que se exhibirá en el navegador o mostrando el respectivo mensaje si se detectó algún error. A modo de ejemplo, al teclear `www.ejemplo.com` en el navegador, éste realiza una petición HTTP al servidor de dicha dirección. El servidor responde al cliente enviando el código HTML de la página; el cliente, una vez recibido el código, lo interpreta y lo exhibe en pantalla.

Instalar un servidor web en la PC permitirá, entre otras cosas, poder montar una propia página web sin necesidad de contratar hosting, probar los desarrollos

vía local, acceder a los archivos de los equipos desde un PC remoto (aunque para esto existen otras opciones, como utilizar un servidor FTP) o utilizar alguno de los programas basados en web tan interesantes que están viendo la luz últimamente.

El hecho de que HTTP y HTML estén íntimamente ligados no debe dar lugar a confundir ambos términos. HTML es un lenguaje de marcas y HTTP es un protocolo.

Algunos conceptos relacionados con las aplicaciones web son:

- ✓ PHP
- ✓ ASP
- ✓ Perl
- ✓ Python/WSGI
- ✓ CGI
- ✓ .NET
- ✓ JSP (Tecnología Java )
- ✓ CF (Adobe Coldfusion)
- ✓ Algunos servidores web importantes son:
- ✓ Apache
- ✓ IIS
- ✓ Cherokee

La funcionalidad de las aplicaciones Web como se muestra en la figura

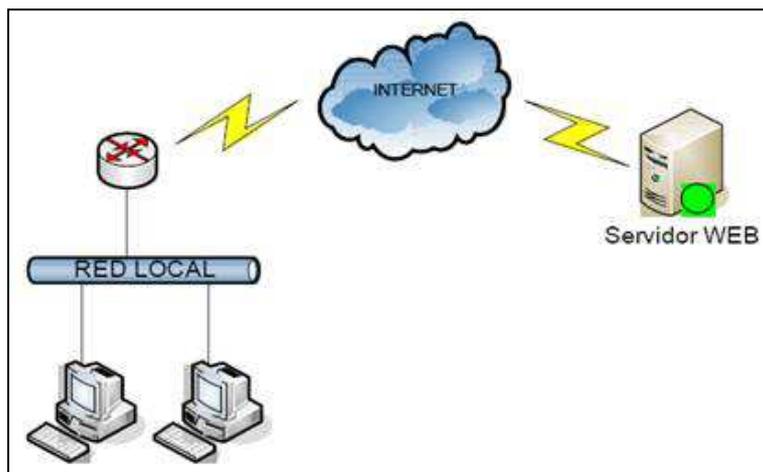


Gráfico 7: Servicios WEB

## 2.2.2 Herramientas de Desarrollo

### 2.2.2.1 AJAX

Ajax, acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano.

De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones.

Es un conjunto de tecnologías que incorpora:

- ✓ Presentación basada en estándares usando XHTML y CSS.
- ✓ Exhibición e interacción dinámicas usando el Document Object Model.
- ✓ Intercambio y manipulación de datos usando XML y XSLT.
- ✓ Recuperación de datos asíncrona usando XMLHttpRequest.
- ✓ JavaScript.

#### ***Diferencias entre aplicaciones Web normales y aplicaciones Web Ajax***

Interfaces diferentes de usuario: Uso de Java script mediante un motor de Ajax para la respuesta de las peticiones realizadas.

ESPACIO EN BLANCO INTENCIONAL

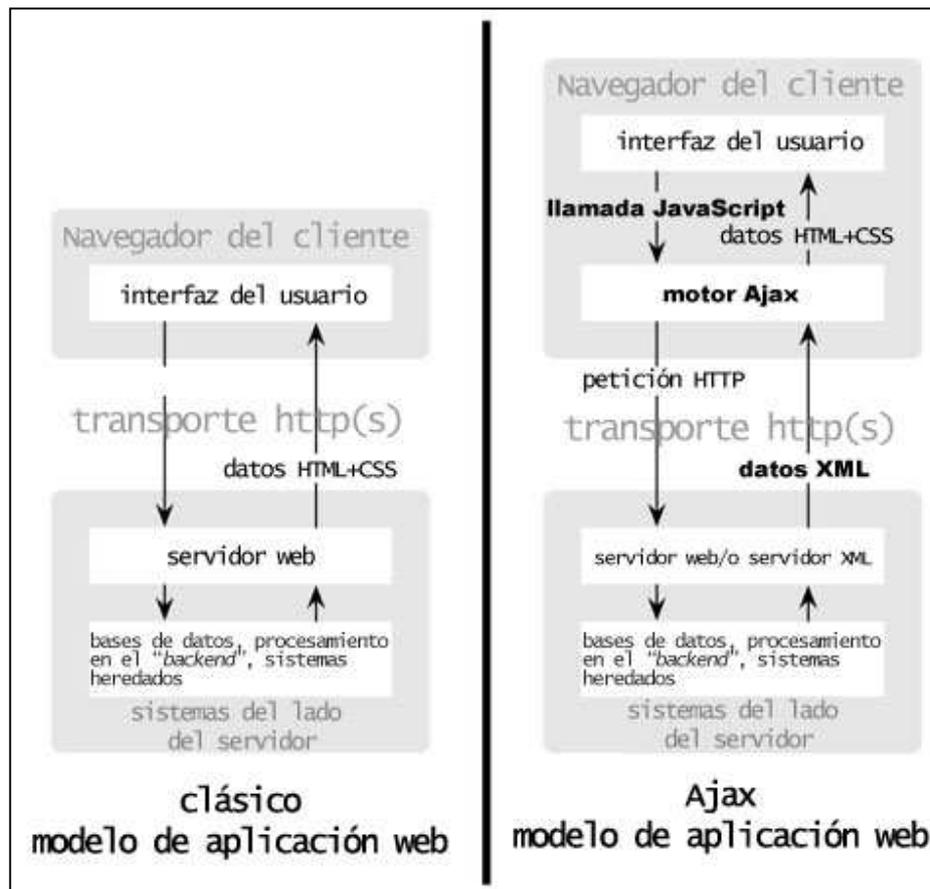


Gráfico 8: AJAX

### Como es diferente AJAX

- ✓ Un motor AJAX elimina el actualizado de las páginas Web entre el usuario y el servidor.
- ✓ En vez de cargar un página Web, al inicio de la sesión, el navegador carga al motor AJAX (escrito en JavaScript y usualmente escrito en un frame oculto).
- ✓ Este motor es el responsable de renderizar la interfaz que el usuario ve y de comunicarse con el servidor en nombre del usuario.
- ✓ El motor AJAX permite que la interacción del usuario con la aplicación suceda asincrónicamente (independientemente de la comunicación con el servidor). Así el usuario nunca estará mirando una ventana en blanco del navegador y un icono de reloj de arena esperando a que el servidor haga algo. Como se muestra en la figura.

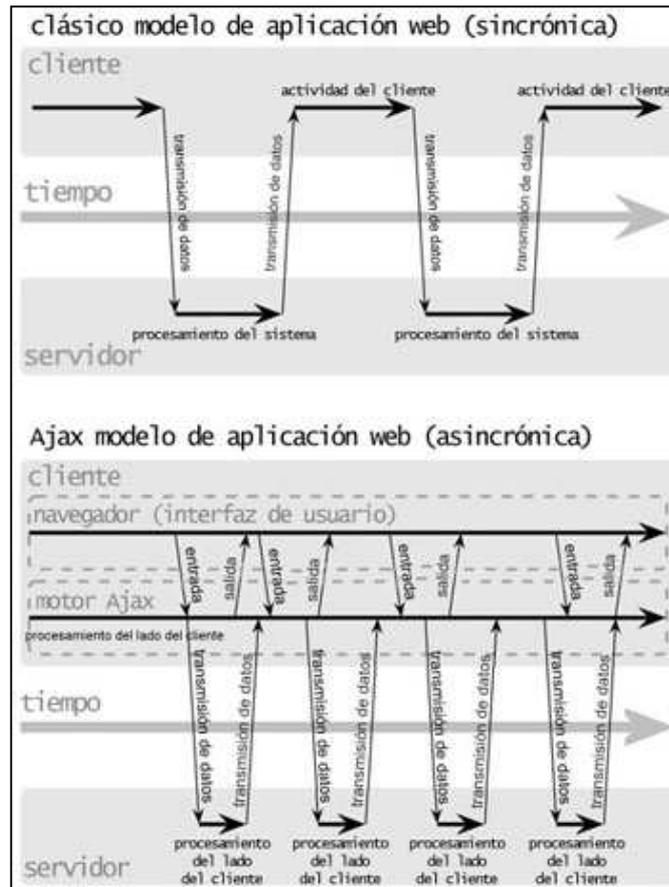


Gráfico 9: Modelos de aplicación Web – AJAX y Clásico

- ✓ Cada acción de un usuario que normalmente generaría un requerimiento HTTP toma la forma de un llamado JavaScript al motor AJAX en vez de ese requerimiento.
- ✓ Cualquier respuesta a una acción del usuario que no requiera un viaje de vuelta al servidor (como una simple validación de datos, edición de datos en memoria, incluso algo de navegación) es manejado por su cuenta.
- ✓ Si el motor necesita algo del servidor para responder (sea enviando datos para procesar, cargar código adicional, o recuperando nuevos datos) hace esos pedidos asincrónicamente, usualmente usando XML, sin frenar la interacción del usuario con la aplicación.

### 2.2.2.2 Términos de Web 2.0

A continuación en la figura ilustrada muestra las diferencias entre Web 1.0 y Web 2.0, aumentando el contenido generado por los usuarios y la inteligencia colectiva permitiendo a más usuarios acceder a la información.

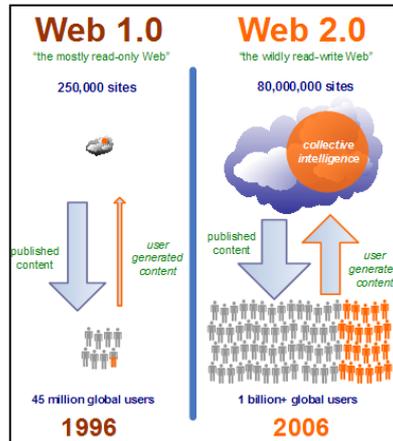


Gráfico 10: Diferencia Web1 y Web 2

La web 1.0 fue una de las primeras en salir con la aparición del internet hace 20 años. La web 2.0 salió aproximadamente hace 7 y 5 años. Las diferencias entre estas 2 webs son notables:

- Web 1.0
  - ✓ Pocos productores de contenido y muchos lectores de estos.
  - ✓ Páginas estáticas.
  - ✓ La actualización de los sitios web no se hacía periódicamente.
  - ✓ Su tecnología estaba asociada con HTML, gif, Java Script 1.0.
- Web 2.0
  - ✓ Los usuarios se transforman en productores de contenido.
  - ✓ Páginas con facilidad de renovar fácilmente su contenido.
  - ✓ Su tecnología está relacionada con Java Script 2.0, XHTML, Google, etc...
  - ✓ Todos los cibernautas tienen la facilidad de acceder a la página web que prefiera.
- AJAX

Nueva tecnología de desarrollo Web con la que se crean aplicaciones interactivas en la red. Estos sitios actúan de manera similar a una aplicación de escritorio clásica, con la diferencia de que se accede a través de la red.

- Beta

Etiqueta común en servicios Web 2.0, significa que tras ser liberado, se encuentra en constante etapa de pruebas y mejoras.

- Java Script

Lenguaje orientado a páginas Web, que es interpretado por la aplicación cliente, normalmente un navegador (browser). Tiene una sintaxis semejante a la del lenguaje.

- Servidor

Computadora, dispositivo o programa que distribuye los recursos dentro de una red, proveyendo la información requerida por los usuarios de esa red. Existen diversos tipos de servidores, según el servicio que proporcionen, por ejemplo: servidor de archivos, servidor Web, servidor de correo, servidor FTP, etcétera.

- Web 2.0

Término usualmente utilizado para la transición del web, que agrupa una colección de sitios hacia una plataforma en línea que brinda aplicaciones web a usuarios finales. Los servicios de la Web 2.0, supuestamente, reemplazarán a las aplicaciones de escritorio para diversos propósitos.

- XML

Siglas en inglés de Extensible Markup Language (lenguaje de marcado ampliable o extensible) desarrollado por el World Wide Web Consortium (W3C).

### **Ventajas**

- ✓ Proporcionan movilidad, dado que puedes ejecutarlas desde cualquier ordenador con conexión a Internet.
- ✓ La información que manejan se accede a través de Internet, motivo por el cual son especialmente interesantes para desarrollar aplicaciones multiusuario basadas en la división de información.
- ✓ El cliente o usuario que utiliza la aplicación no necesita tener un ordenador de grandes prestaciones para trabajar con ella.

### **Desventajas**

- ✓ La comunicación constante con el servidor que ejecuta la aplicación establece

una dependencia con una buena conexión a Internet.

- ✓ El servidor debe tener las prestaciones necesarias para ejecutar la aplicación de manera fluida, no sólo para un usuario sino para todos los que la utilicen de forma concurrente.
- ✓ Emplean tecnologías como Java, JavaScript, DHTML, Flash, Ajax, etc, que dan gran potencia a la interfaz de usuario.

### 2.2.2.3 JSF2.0

**JavaServerFaces (JSF)** es una tecnología y framework para aplicaciones Java basadas en web que simplifica el desarrollo de interfaces de usuario en aplicaciones Java EE. **JSF** usa JavaServer Pages (JSP) como la tecnología que permite hacer el despliegue de las páginas, pero también se puede acomodar a otras tecnologías como XUL.

JSF incluye:

- Un conjunto de APIs para representar componentes de una interfaz de usuario y administrar su estado, manejar eventos, validar entrada, definir un esquema de navegación de las páginas y dar soporte para internacionalización y accesibilidad.
- Un conjunto por defecto de componentes para la interfaz de usuario.
- Dos bibliotecas de etiquetas personalizadas para JavaServer Pages que permiten expresar una interfaz JavaServer Faces dentro de una página JSP.
- Un modelo de eventos en el lado del servidor.
- Administración de estados.
- Beans administrados.

Estos objetivos de diseño representan el foco de desarrollo de JSF:

1. Definir un conjunto simple de clases base de Java para componentes de la interfaz de usuario, estado de los componentes y eventos de entrada. Estas

clases tratarán los aspectos del ciclo de vida de la interfaz de usuario, controlando el estado de un componente durante el ciclo de vida de su página.

2. Proporcionar un conjunto de componentes para la interfaz de usuario, incluyendo los elementos estándares de HTML para representar un formulario. Estos componentes se obtendrán de un conjunto básico de clases base que se pueden utilizar para definir componentes nuevos.
3. Proporcionar un modelo de JavaBeans para enviar eventos desde los controles de la interfaz de usuario del lado del cliente a la aplicación del servidor.
4. Definir APIs para la validación de entrada, incluyendo soporte para la validación en el lado del cliente.
5. Especificar un modelo para la internacionalización y localización de la interfaz de usuario.
6. Automatizar la generación de salidas apropiadas para el objetivo del cliente, teniendo en cuenta todos los datos de configuración disponibles del cliente, como versión del navegador.

Versiones existentes:

- JSF 1.0 (11-03-2004) - lanzamiento inicial de las especificaciones de JSF.
- JSF 1.1 (27-05-2004) - lanzamiento que solucionaba errores. Sin cambios en las especificaciones ni en el renderkit de HTML.
- JSF 1.2 (11-05-2006) - lanzamiento con mejoras y corrección de errores.
- JSF 2.0 (12-08-2009) - último lanzamiento.

#### **2.2.2.4 Richfaces 4**

RichFaces es una librería de componentes visuales para JSF, escrita en su origen por Exadel y adquirida por Jboss. Además, RichFaces posee un framework avanzado para la integración de funcionalidades Ajax en dichos componentes visuales, mediante el soporte de la librería Ajax4JSF.

Son características de RichFaces las siguientes:

- Se integra perfectamente en el ciclo de vida de JSF.
- Incluye funcionalidades Ajax, de modo que nunca se ve el JavaScript y tiene un contenedor Ajax propio.
- Contiene un set de componentes visuales, los más comunes para el desarrollo de una aplicación web rica (Rich Internet Application), con un número bastante amplio que cubren casi todas las necesidades.
- Soporta facelets.
- Soporta css themes o skins.
- Es un proyecto open source, activo y con una comunidad también activa.

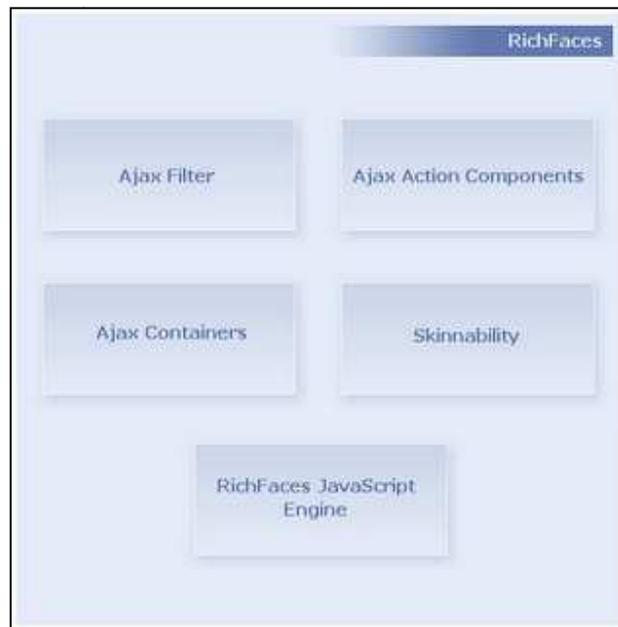


Gráfico 11: Arquitectura Richfaces 4

### 2.2.2.5 ActionScript 3.0

ActionScript es un lenguaje de programación orientado a objetos (OOP), utilizado en especial en aplicaciones web animadas realizadas en el entorno

Adobe Flash, la tecnología de Adobe para añadir dinamismo al panorama web.

ActionScript es un lenguaje de script, esto es, no requiere la creación de un programa completo para que la aplicación alcance los objetivos. El lenguaje está basado en especificaciones de estándar de industria ECMA-262, un estándar para Javascript, de ahí que ActionScript se parezca tanto a Javascript.

### ***Ventajas de ActioScript 3.0***

ActionScript 3.0 aumenta las posibilidades de creación de scripts de las versiones anteriores de ActionScript.

Se ha diseñado para facilitar la creación de aplicaciones muy complejas con conjuntos de datos voluminosos y bases de código reutilizables y orientadas a objetos. Aunque no se requiere para el contenido que se ejecuta en Adobe Flash Player 9, ActionScript 3.0 permite introducir unas mejoras de rendimiento que sólo están disponibles con AVM2, la nueva máquina virtual. El código ActionScript 3.0 puede ejecutarse con una velocidad diez veces mayor que el código ActionScript heredado.

La versión anterior de la máquina virtual ActionScript (AVM1) ejecuta código ActionScript 1.0 y ActionScript 2.0. Flash Player 9 admite AVM1 por compatibilidad con contenido existente y heredado de versiones anteriores.

### ***Evolución de ActionScript***

- *Excepciones en Tiempo de Ejecución*

ActionScript 3.0 notifica más situaciones de error que las versiones anteriores de ActionScript. Las excepciones de tiempo de ejecución se utilizan en situaciones de error frecuentes y permiten mejorar la depuración y desarrollar aplicaciones para gestionar errores de forma robusta. Los errores de tiempo de ejecución pueden proporcionar trazas de pila con la información del archivo de código fuente y el número de línea. Esto permite identificar rápidamente los errores.

- *Clases Cerradas*

ActionScript 3.0 introduce el concepto de clases cerradas. Una clase cerrada posee únicamente el conjunto fijo de propiedades y métodos definidos durante la compilación; no es posible añadir propiedades y métodos adicionales;

aunque sí se puede usándolas dinámicas. Cerrarlas permite realizar una comprobación más estricta en tiempo de compilación, lo que aporta una mayor solidez a los programas.

También mejora el uso de memoria, pues no requiere una tabla hash interna para cada instancia de objeto.

Además, es posible utilizar clases dinámicas mediante la palabra clave `dynamic`. Todas las clases de ActionScript 3.0 están cerradas de forma predeterminada, pero pueden declararse como dinámicas con la palabra clave `dynamic`.

- Cierres de Métodos

ActionScript 3.0 permite que un cierre de método recuerde automáticamente su instancia de objeto original. Esta función resulta útil en la gestión de eventos. En ActionScript 2.0, los cierres de métodos no recordaban la instancia de objeto de la que se habían extraído, lo que provocaba comportamientos inesperados cuando se llamaba al cierre de método.

## **2.2.3 Motores de Base de Datos**

### **2.2.3.1 MySQL**

MySQL es un sistema de gestión de bases de datos relacional, licenciado bajo la GPL de la GNU. Su diseño multihilo y multiusuario le permite soportar una gran carga de forma muy eficiente. MySQL fue creada por la empresa sueca MySQL AB, que mantiene el copyright del código fuente del servidor SQL, así como también de la marca.

MySQL es una herramienta un gestor de base de datos calificado como Software Libre. Este gestor de bases de datos es, probablemente, el gestor más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso.

Esta gran aceptación es debida, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración.

### ***Características de MySQL***

- ✓ Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
- ✓ Soporta gran cantidad de tipos de datos para las columnas.
- ✓ Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP, etc).
- ✓ Gran portabilidad entre sistemas.
- ✓ Soporta hasta 32 índices por tabla.
- ✓ Gestión de usuarios y passwords, manteniendo un muy buen nivel de seguridad en los datos.
- ✓ MySQL está escrito en C y C++
- ✓ Emplea el lenguaje SQL para consultas a la base de datos.
- ✓ MySQL Server está disponible como freeware bajo licencia GPL.
- ✓ MySQL Enterprise es la versión por suscripción para empresas, con soporte las 24 horas.
- ✓ Trabaja en las siguientes plataformas: AIX, BSDi, FreeBSD, HP-UX, GNU/Linux, Mac OS X, NetBSD, Novell NetWare, OpenBSD, OS/2 Warp, QNX, SGI IRIX, Solaris, SunOS, SCO OpenServer, SCO UnixWare, Tru64, Microsoft Windows (95, 98, ME, NT, 2000, XP, Vista, 7).
- ✓ Completo soporte para operadores y funciones en cláusulas select y where.
- ✓ Completo soporte para cláusulas group by y order by, soporte de funciones de agrupación
- ✓ Seguridad: ofrece un sistema de contraseñas y privilegios seguro mediante verificación basada en el host y el tráfico de contraseñas está cifrado al conectarse a un servidor.
- ✓ Soporta gran cantidad de datos. MySQL Server tiene bases de datos de hasta 50 millones de registros.
- ✓ Se permiten hasta 64 índices por tabla (32 antes de MySQL 4.1.2). Cada índice puede consistir desde 1 hasta 16 columnas o partes de columnas. El máximo ancho de límite son 1000 bytes (500 antes de MySQL 4.1.2).

- ✓ Los clientes se conectan al servidor MySQL usando sockets TCP/IP en cualquier plataforma. En sistemas Windows se pueden conectar usando named pipes y en sistemas Unix usando ficheros socket Unix.
- ✓ En MySQL 5.0, los clientes y servidores Windows se pueden conectar usando memoria compartida.
- ✓ MySQL contiene su propio paquete de pruebas de rendimiento proporcionado con el código fuente de la distribución de MySQL.

## **2.2.4 Interfaz de la aplicación del Programa (API)**

### **2.2.4.1 EJB 3.1**

Es una plataforma para construir aplicaciones de negocio portables, reusables y escalables usando el lenguaje de programación Java cuyo objetivo es dotar al programador de un modelo que le permita abstraerse de los problemas generales de una aplicación empresarial (conurrencia, transacciones, persistencia, seguridad, etc.) para centrarse en el desarrollo de la lógica de negocio en sí. Permitiendo que éstos sean flexibles y sobre todo reutilizables.

Son una de las API que forman parte del Estándar de construcción de aplicaciones empresariales J2EE de Sun Microsystems.

Su especificación detalla cómo los servidores de aplicaciones proveen objetos desde el lado del servidor que son, precisamente, los EJB:

- ✓ Comunicación remota utilizando CORBA.
- ✓ Transacciones.
- ✓ Control de la concurrencia.
- ✓ Eventos utilizando JMS (Java messaging service).
- ✓ Servicios de nombres y de directorio.
- ✓ Seguridad.
- ✓ Ubicación de componentes en un servidor de aplicaciones.

### ***Tipos de EJB's***

- *EJB de Entidad (Entity EJBs)*

Su objetivo es encapsular los objetos del lado del servidor que almacena los datos. Los EJB de entidad presentan la característica fundamental de la persistencia.

- *EJB de Sesión (Session EJBs)*

Gestionan el flujo de la información en el servidor. Generalmente sirven a los clientes como una fachada de los servicios proporcionados por otros componentes disponibles en el servidor.

- *EJB dirigidos por mensajes (Message-driven EJBs)*

Son los únicos beans con funcionamiento asíncrono. Usando el Java Messaging System (JMS), se suscriben a un tema (topic) o a una cola (queue) y se activan al recibir un mensaje dirigido a dicho tema o cola. No requieren de su instanciación por parte del cliente.

### **Lo nuevo de EJB 3.1**

EJB 3.1 trae un nuevo conjunto de características que mejoran el potencial de esta tecnología. Es una versión muy importante que traerá nuevas capacidades faltantes a estos componentes.

#### **Vista sin interfaz**

EJB 3.1 introduce el concepto de vista sin interfaz, que consiste en una variación de la vista Local, la cual expone todos los métodos públicos de un bean. Los Session Beans no están obligados a implementar una interfaz. El contenedor de EJB provee una implementación a la referencia a una vista sin interfaz, lo que le permite al cliente invocar cualquier método público de un bean, y por supuesto contar con transacciones, seguridad y comportamiento con interceptores.

Todos los métodos públicos del bean (incluyendo aquellos definidos en superclases) están disponibles en la vista sin interfaz. Un cliente puede

obtener una referencia a esta vista inyectando la dependencia, tal cual se hace con las vistas locales o remotas.

A diferencia de las vistas locales o remotas, en donde la referencia consiste de la interfaz local/remota respectivamente, la referencia a una vista sin interfaz se declara como la clase del bean.

## **Singleton**

La mayoría de las aplicaciones tuvieron alguna vez un bean singleton, es decir, un bean que se instancia una única vez por aplicación. Varios proveedores brindan soporte para esto, permitiendo especificar la cantidad máxima de instancias en un descriptor propietario. Estos "arreglos" rompen el principio de "escribir una vez, desplegar en cualquier lado", y por lo tanto era responsabilidad de la especificación estandarizar este aspecto. EJB 3.1 finalmente introduce los Singleton Session Beans.

Ahora hay tres tipos de Session Beans: stateless, stateful y singleton. Los Singleton Session Beans se identifican por la anotación Singleton y se instancian una única vez por aplicación. La instancia existente se comparte entre todos los clientes, y permite accesos concurrentes.

La vida de un Singleton Bean comienza cuando el contenedor realiza la instanciación (por instanciación se refiere a la instanciación, inyección de dependencias y ejecución de los callback *PostConstruct*). El contenedor es responsable de decidir cuándo se construye el bean, aunque el desarrollador puede decirle al contenedor que inicialice el bean durante el inicio de la aplicación, usando la anotación Startup. Más aún, la anotación Startup permite definir dependencias hacia otros Singleton Beans. El contenedor debe inicializar todos los beans marcados con Startup antes de comenzar a responder peticiones de los clientes.

## **Invocaciones asincrónicas**

Las invocaciones asincrónicas de métodos de un Session Bean es una de las características más interesantes de esta nueva versión. Puede utilizarse en todos los tipos de beans, en todas las vistas. La especificación define que una invocación asincrónica tiene que retornar al cliente antes de que el contenedor entregue la invocación a la instancia del bean. Esto eleva la utilización de los Session Bean a un nuevo nivel, permitiendo que los desarrolladores se beneficien de invocaciones asincrónicas a los beans, lanzando así flujos de procesos en paralelo.

## **Nombres JNDI globales**

Esta es una característica que se venía esperando hace mucho. La asignación de nombres JNDI globales a los EJB siempre se implementó de manera propietaria para cada proveedor, lo que ocasionaba varios problemas. La misma aplicación que se desplegaba en contenedores de distintos proveedores podían quedar con los EJB con nombres JNDI distintos, provocando problemas en los clientes. Además, algunos proveedores hacían que las interfaces locales se pudieran obtener via JNDI, mientras que otros no.

## **Timer-Service**

Existen varias aplicaciones que tienen requerimientos guiados por el tiempo. Por mucho tiempo se ignoraron estas necesidades, forzando a los desarrolladores a utilizar soluciones no estándar como Quartz o Flux. EJB 2.1 introdujo el concepto de Timer Service, pero tenía varias limitaciones.

Con EJB 3.1 la situación cambia para mejor. Hay dos formas de crear timers:

- Programáticamente, usando la interfaz `TimerService` ya existente. La interfaz fue mejorado desde la versión 2.1 para brindar más flexibilidad al crear timers.
- Declarativamente, usando anotaciones o en el descriptor de despliegue. De esta manera se puede definir un timer de manera estática para que sea creado automáticamente durante el inicio de la aplicación.

### **Servidor y Contenedor EJB**

El servidor EJB provee un ambiente que soporta la ejecución de aplicaciones desarrolladas usando componentes EJB. Este maneja y coordina la asignación de recursos a las aplicaciones. El servidor EJB debe proveer uno o más contenedores EJB.

Un contenedor de EJB proporciona un entorno de ejecución para beans, maneja todos los aspectos de las operaciones del bean dentro del servidor de aplicaciones y actúa como intermediario entre la lógica empresarial escrita en el bean y el resto del entorno del servidor de aplicaciones. Para cada bean, el contenedor es responsable de registrar el objeto, proveer una interfaz remota para el objeto, crear y destruir las instancias de los objetos, verificar la seguridad del objeto, gestionar el estado activo del objeto y coordinar transacciones distribuidas.

### **Persistencia en EJB**

Persistencia es la habilidad de tener datos contenidos en objetos Java almacenados en una base de datos relacional como Oracle, SQL Server, MySQL, y DB2. La persistencia en EJB es manejada por el JPA (API de persistencia Java). Este automáticamente persiste los objetos Java usando una técnica llamada mapeo objeto-relacional (ORM). ORM es esencialmente el proceso de mapeo de datos mantenidos en objetos Java hasta tablas de base de datos. Este proceso

libera al programador de la tarea de escribir código JDBC a bajo nivel muy complejo para persistir los objetos en la base de datos.

El framework que provee la capacidad de ORM para desempeñar persistencia automática es conocido como framework ORM. Un framework ORM desempeña una persistencia transparente haciendo uso de un mapeo objeto-relacional, algunos de los frameworks ORM son: Oracle Toplink, y de código abierto JBoss Hibernate. En términos de EJB 3, un proveedor de persistencia es esencialmente un framework ORM que soporta la API de persistencia Java (JPA) de EJB 3. El JPA define un estándar para:

- La creación de configuración de metadatos ORM para mapear entidades en tablas relacionales.
- La API del gestor de entidad (EntityManager), un estándar para desempeñar las operaciones de persistencia para las entidades (crear, leer, actualizar y borrar).
- El lenguaje de consultas de persistencia Java (*Java Persistence Query Language*, JPQL) para búsqueda y recuperación de datos persistentes de las aplicaciones.

Desde la estandarización de los frameworks ORM para la plataforma Java, se pueden escoger los productos Jboss Hibernate, Oracle Toplink o BAEA Kodo como proveedores de persistencia (JPA) para el desarrollo de aplicaciones usando EJB.

### **Anotaciones EJB**

Como framework, el contenedor EJB provee servicios que los componentes EJB pueden usar, lo que permite que las aplicaciones sean de alta calidad y más rápidas que con otras tecnologías. Los servicios para los componentes EJB se acceden por medio de anotaciones, estas pre configuran el EJB; son una forma

declarativa de programación en la que el desarrollador especifica que se debe hacer y el sistema agrega el código para hacerlo, esto se explica en la Figura: ?.

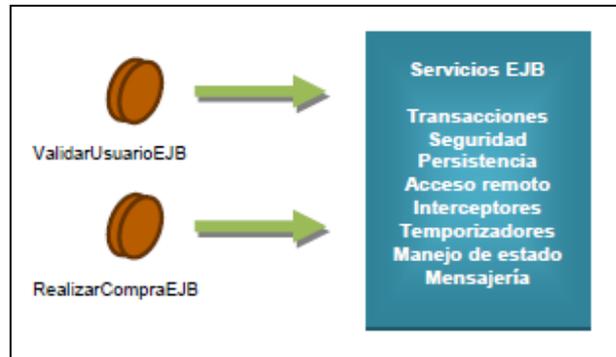


Gráfico 12: Anotaciones EJB

### Las anotaciones permiten acceder a servicios

Las anotaciones simplifican el desarrollo de las aplicaciones sin tener que depender de un archivo de configuración XML. Las anotaciones también son usadas para especificar el tipo de componente EJB.

El gráfico 13 muestra como una anotación transforma un objeto plano de Java en un EJB.



Gráfico 13: Composición de un Bean

### EJBs son objetos Java configurados usando Anotaciones

#### Componentes EJB

Un EJB es un componente del lado del servidor que se usa para construir partes de una aplicación, como la lógica de negocio o el código de persistencia. Un componente EJB es un POJO (*Plain Old Java Object*, una clase simple que no depende de un framework) con algunas particularidades que encapsulan el

comportamiento de la aplicación, por lo tanto un usuario del componente EJB no necesita saber cómo trabaja, simplemente necesita saber que enviarle y que recibirá de él.

Hay tres tipos de componentes EJB: beans de sesión (*sesión beans*), beans dirigidos por mensajes (*message-driven beans*) y entidades (*entities*). Los beans de sesión y los beans dirigidos por mensajes son usados para implementar la lógica de negocio en una aplicación EJB, se alojan en el contenedor, quien los gestiona y les provee de servicios. Por otro lado están las entidades que son gestionadas por el proveedor de persistencia.

**Beans de sesión (*Session beans en inglés*):** Un bean de sesión es invocado por un cliente para desempeñar una operación de negocio específica. El nombre sesión implica que la instancia del bean estará disponible mientras dure una “unidad de trabajo” y no sobrevivirá al apagado o caída del servidor.

Los bean de sesión modelan cualquier lógica de funcionalidad de una aplicación. Los bean de sesión no son persistentes, ellos reciben su estado del cliente y viven mientras el cliente los necesite. Los bean de sesión tienen interfaces locales y remotas que define los métodos públicos del mismo.

Hay dos tipos de bean de sesión:

**Bean de sesión con estado (*Stateful Session bean*):** Un bean de este tipo, guarda automáticamente el estado entre las invocaciones del cliente sin tener que escribir ningún código adicional. Un ejemplo de un proceso con estado es el de un carrito de compras.

**Bean de sesión sin estado (*Stateless Session bean*):** Un bean de este tipo no mantiene ningún estado y la solicitud de modelo de servicios puede ser completada en una simple invocación del cliente. Se puede implementar beans de sesión sin estado para implementar procesos de negocio como cargar una tarjeta de crédito.

#### 2.2.4.2 JPA 2.0

Este es el comienzo de una serie de artículos sobre JPA (Java Persistence API). JPA es el estándar de Java para el mapeo de objetos Java a modelos relacionales de base de datos. La unidad básica de persistencia en JPA es la **entidad**, que es una clase Java, que debe cumplir una serie de requisitos, con metadatos (anotaciones o fichero XML), y que generalmente representa una tabla de la base de datos donde cada instancia de la entidad corresponde con una fila de la tabla. Los metadatos pueden ser o bien anotaciones sobre la entidad y sus atributos o métodos, o bien un fichero XML, de hecho, cuando se utilizan a la vez ambas técnicas, los ficheros XML sobrescriben los valores establecidos por las anotaciones.

JPA es un API, una serie de interfaces, y por lo tanto se trabaja con un proveedor de persistencia que implemente dicha interfaz. Actualmente, en el mercado existen varios proveedores de persistencia JPA como Hibernate, iBatis, TopLink, etc.

Estas sentencias son métodos del EntityManager que se aplican sobre la entidad y son:

- **select** : equivale al select de SQL pero se aplica sobre "entidades"
- **find** : equivale a la búsqueda de una entidad basado en su PK
- **persist** : equivale a un insert.
- **merge**: equivale a un update.
- **remove**: equivale a un delete.
- **refresh**: equivale a una re-lectura de la entidad ( refresco desde la base de datos ).

#### 2.2.5 Lenguaje de Programación

##### *Java*

Java es una plataforma virtual de software desarrollada por Sun Microsystems, de tal manera que los programas creados en ella puedan

ejecutarse sin cambios en diferentes tipos de arquitecturas y dispositivos computacionales ("Diferentes plataformas").

La plataforma Java consta de las siguientes partes:

- El lenguaje de programación, mismo.
- La máquina virtual de Java o JRE, que permite la portabilidad en ejecución.
- El API Java, una biblioteca estándar para el lenguaje.

A fines del siglo XX, Java llegó a ser el lenguaje de mayor acogida para programas de servidor. Utilizando una tecnología llamada JSP (similar a otras tecnologías del lado del servidor como ASP de Microsoft o PHP), se hizo muy fácil escribir páginas dinámicas para sitios de Internet. Sumado a JSP la tecnología de JavaBeans, permitía adaptar al mundo web el patrón MVC (modelo-vista-controlador) que ya se había aplicado con éxito a interfaces gráficas.

Además de programas del servidor, Java permite escribir programas de interfaz gráfica o textual. También se pueden correr programas de manera incorporada o incrustada en los navegadores web de Internet en forma de Java applets, aunque no llegó a popularizarse como se esperaba en un principio.

Los programas en Java generalmente son compilados a un lenguaje intermedio llamado bytecode, que luego son interpretados por una máquina virtual (JVM). Esta última sirve como una plataforma de abstracción entre la máquina y el lenguaje permitiendo que se pueda "escribir el programa una vez, y correrlo en cualquier lado". También existen compiladores nativos de Java, tanto software libre como no libre. El compilador GCC de GNU compila Java a código de máquina con algunas limitaciones al año 2002.

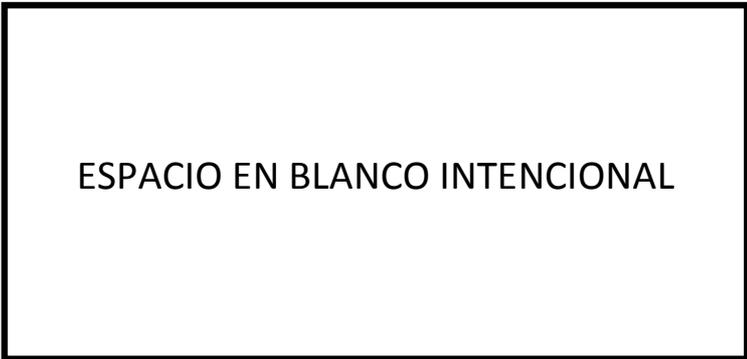
Con la evolución de las diferentes versiones, no sólo se han producido cambios en el lenguaje, sino que se han producido cambios mucho más importantes en sus bibliotecas asociadas, que han pasado de unos pocos cientos en Java 1.0, a más de tres mil en Java 5.0. En particular, se han añadido APIs completamente nuevas, tales como Swing y Java2D.

Una de las principales características por las que Java se ha hecho muy

famoso es que es un lenguaje independiente de la plataforma. Eso quiere decir que si realiza un programa en Java podrá funcionar en cualquier ordenador del mercado. Es una ventaja significativa para los desarrolladores de software, pues antes tenían que hacer un programa para cada sistema operativo, por ejemplo Windows, Linux, Apple, etc. Esto lo consigue porque se ha creado una Máquina de Java para cada sistema que hace de puente entre el sistema operativo y el programa de Java y posibilita que este último se entienda perfectamente.

La independencia de plataforma es una de las razones por las que Java es interesante para Internet, ya que muchas personas deben tener acceso con ordenadores distintos. Pero no se queda ahí, Java está desarrollándose incluso para distintos tipos de dispositivos además del ordenador como móviles, agendas y en general para cualquier cosa que se le ocurra a la industria.

Actualmente Java se utiliza en un amplio abanico de posibilidades y casi cualquier cosa que se puede hacer en cualquier lenguaje se puede hacer también en Java y muchas veces con grandes ventajas. Con Java se puede programar páginas web dinámicas, con accesos a bases de datos, utilizando XML, con cualquier tipo de conexión de red entre cualquier sistema. En general, cualquier aplicación que se desea hacer con acceso a través web se puede hacer utilizando Java.



ESPACIO EN BLANCO INTENCIONAL

## CAPÍTULO 3: Construcción del Sistema

### 3.1 Contenido

#### 3.1.1 Análisis de requerimientos

A diferencia de la metodología tradicional, XP utiliza las historias de usuario para la especificación de requisitos, permitiendo disminuir la documentación.

XP presenta 4 valores que al seguirlos y utilizarlos facilita la especificación de requerimientos:

**La comunicación:** permite que el cliente y el programador lleguen a un acuerdo en la especificación de requerimientos evitando los malos entendidos.

**La sencillez:** es lo que diferencia a XP con las demás metodologías tradicionales las cuales utilizan estándares para la especificación de requerimientos que hacen del sistema muy complejo. La sencillez evita la documentación extensa centrándose en lo básico, en lo que se utiliza en este momento y no en lo que se podrá utilizar.

**La realimentación:** permite que la especificación de requerimientos se comprenda mejor con el pasar del tiempo, permitiendo que los usuarios aprendan a describir mejor las Historias.

**Las Historias de usuario:** es una pequeña descripción del programa con el fin de estimar tiempos y costos. Para obtener mayor detalle de las historias de usuario en el momento de la implementación, el programador preguntará al cliente, aumentando el detalle de cada historia.

##### 3.1.1.1 Historias de Usuarios

Las historias de usuario permiten obtener los requerimientos del sistema a implementar.

La Historia de usuario 1 presenta los primeros requerimientos por parte del usuario, es importante no detallar las historias de usuario porque son utilizadas solo para dar una pequeña visión de lo que se quiere obtener.

Cuando se inicio la fase de desarrollo el investigador con la ayuda del usuario detallarán las historias de usuario.

- **Historia de Usuario 1 en el Diseño**

INGRESO AL SISTEMA Y MANTENIMIENTO Y CREACIÓN DE PERFILES DE USUARIOS							
FECHA	08/12/2011	TIPO DE ACTIVIDAD	Nuevo	X	PRIORIDAD TÉCNICA	Normal	X
			Corrección			Medio	
			Mejora			Alto	
REFERENCIA		RIESGO	Ninguno	X	PRIORIDAD USUARIO	Normal	X
			Medio			Medio	
			Alto			Alto	
USUARIO		CARGO			CLASIFICACIÓN	Seguridad	
DESCRIPCIÓN							
Controla el ingreso al sistema utilizando una código y clave Se ingresa a perfiles de usuarios para determinar permisos para cada uno Validaciones para el perfil de Usuario							
NOTAS							
SEGUIMIENTO							
ESTADO	Atendido		COMENTARIOS			ÚLTIMO SEGUIMIENTO	
	En Proceso	X					
	Finalizado						

Tabla 9: Historia de Usuario 1 en el Diseño

- **Historia de Usuario 1 en el Desarrollo**

**Controla el ingreso al sistema utilizando una cuenta y clave:**

Se creará una ventana para que el usuario pueda ingresar al sistema introduciendo una cuenta de usuario y una clave única y personal.

Si la clave es correcta podrá ingresar al sistema

Caso contrario se presentará un mensaje indicando que la cuenta o clave es errónea.

En el caso de tratar de ingresar al sistema con una cuenta que ya está caducada, el sistema mandará a una ventana emergente en donde se mostrará las opciones para actualizar los datos del código y clave de usuario al momento de guardar estos nuevos datos del usuario, el sistema automáticamente dará una vigencia de 3 meses de permiso de ingreso al sistema.

***Se ingresa a perfiles de usuarios para determinar permisos para cada usuario:***

Se crea una ventana en donde obtendrá la lista de todos los usuarios que pueden ingresar al sistema.

Esta ventana permitirá hacer búsquedas de usuarios según el nombre del mismo, adicional permitirá crear, editar y eliminar usuarios.

Al crear un nuevo usuario permitirá ingresar la siguiente información:

Cédula del usuario

Nombre del usuario

Código del perfil de Usuario

Clave del perfil de Usuario

Confirmación de clave de usuario

Al modificar el usuario el sistema podrá modificar solo los siguientes campos:

Nombre del usuario, fecha que se registró al usuario, fecha que expira el permiso del usuario

### Validaciones para el perfil de Usuario:

El sistema validará que el código del perfil de usuario no se repita.

El sistema validará que todos los campos obligatorios se hayan ingresado.

Al guardar el usuario el sistema lo crea y asigna estatus del usuario y fecha de vigencia de ingreso al sistema para el usuario.

Para diferenciar que los campos son o no obligatorios se pondrá al comienzo un asterisco.

### Historia de Usuario 2 en el Diseño

INGRESO DE DATOS AL SISTEMA PARA TODOS LOS MÓDULOS							
FECHA	08/12/2011	TIPO DE ACTIVIDAD	Nuevo	X	PRIORIDAD TÉCNICA	Normal	X
			Corrección			Medio	
			Mejora			Alto	
REFERENCIA		RIESGO	Ninguno	X	PRIORIDAD USUARIO	Normal	X
			Medio			Medio	
			Alto			Alto	
USUARIO		CARGO			CLASIFICACIÓN	Ingreso de Datos	
DESCRIPCIÓN							
Ingreso, actualización, eliminación y búsqueda en las tablas: Producto, Proveedor, Usuario, Clientes, Vehículo, Persona, Parametrización Cada una de las tablas presentará su interface sencilla y fácil de utilizar. Las búsquedas serán personalizadas de acuerdo a las necesidades del usuario							
NOTAS							
SEGUIMIENTO							
ESTADO	Atendido		COMENTARIOS			ÚLTIMO SEGUIMIENTO	
	En Proceso	X					
	Finalizado						

Tabla 10: Historia de Usuario 2 en el Diseño

- **Historia de Usuario 2 en el Desarrollo**

#### ***Creación de la ventana para el manejo de Productos:***

Para ingresar el producto se creará una ventana en la cual se listará primeramente todos los productos (repuestos), que se tiene disponible en el sistema, estos podrán ser creados, modificados y eliminados respectivamente.

Al proceder a crear o modificar, presentará una ventana nueva en donde se ingresará el código, descripción, cantidad, precio de costo, precio de venta, stock mínimo, y proveedor.

El código del producto es un identificador único del mismo, y permite ubicarlo de una forma rápida.

#### ***Validación del formulario Producto:***

El sistema validará que el código y la descripción del producto no se repitan.

El sistema validará que la cantidad sea de tipo numérico.

El sistema validará que el precio costo y el precio venta sean de tipo flotante.

El sistema validará que el stock sea de tipo entero.

El sistema validará que se ingrese todos los campos obligatorios. Para diferenciar los campos obligatorios se colocará un asterisco seguido del título de cada campo respectivamente.

#### ***Creación de la ventana para el manejo de Proveedores:***

Para ingresar el producto se creará una ventana en la cual se listarán primeramente todos los proveedores, que se tiene disponibles en el sistema, estos podrán ser creados, modificados y eliminados respectivamente.

Se creará un formulario para el ingreso de proveedores esta ventana contiene los siguientes campos: RUC, nombre, nombre de la empresa, nombre de la empresa, descripción.

También se presentará un multiregistro en cual listará los repuestos que ofrece el proveedor que se seleccionó, estos repuestos se los puede agregar, modificar o eliminar.

Para el ingreso de los repuestos que dispone cada proveedor, se deberá ingresar los datos de:

Código del repuesto

Nombre del repuesto

Precio del repuesto

Descripción del repuesto

***Validación del formulario Proveedor:***

El sistema validará que el ruc no se repita.

El sistema validará que el ruc cumpla los 10 o 13 caracteres

El sistema validará que se ingrese todos los campos obligatorios. Para diferenciar los campos obligatorios se colocará un asterisco seguido del título de cada campo respectivamente.

### ***Creación de la ventana para el manejo de Clientes, Personas, Vehículos:***

En el menú se crearán dos ítems los cuales serán:

#### **Listado de Personas:**

Aquí primeramente al ingresar listará todas la personas que se tiene en el sistema pero estas personas pueden o no pueden ser clientes para ello se deberá ir al segundo ítems de creación de clientes.

Este listado permitirá encontrar a la persona por el número de cédula, el mismo que se puede modificar o eliminar, tomando en cuenta que esta persona no puede ser eliminada si ya consta como cliente.

En esta pantalla también permite crear nuevas personas para posteriormente hacerlas clientes del taller.

Para ingresar una nueva persona o modificarla se deberán ingresar los siguientes datos:

Cédula, nombre, apellido, dirección, teléfono (celular y particular), mail, fecha de nacimiento.

#### **Listado de Clientes:**

En este segundo ítems de listado de clientes permitirá asignarle el vehículo a cada persona, esta persona puede tener uno o varios vehículos, después de hacerlos se procederá a guardarlo y en ese momento la persona se convierte en cliente del taller.

Para esto primeramente se procederá a buscar a la persona que va a asignar el vehículo ingresando el número de cédula de la persona, al existir esa persona puede asignarle el vehículo respectivamente.

Al ingresar el vehículo o modificarlo presentará los siguientes datos:

Placa, kilometraje, tipo, marca, año, número de chasis, color.

### ***Validación del formulario para:***

#### ***Persona***

El sistema validará que la Cédula, nombre, apellido, dirección, teléfono (celular y particular), mail, fecha de nacimiento estén correctamente ingresados.

El sistema validará que la cédula no se repita

El sistema validará que se ingrese todos los campos obligatorios. Para diferenciar los campos obligatorios se colocará un asterisco seguido del título de cada campo respectivamente.

#### ***Clientes***

El sistema validará que la placa no se repita.

El sistema validará que la Placa, kilometraje, tipo, marca, año, número de chasis, color estén correctamente ingresados

El sistema validará que se ingrese todos los campos obligatorios. Para diferenciar los campos obligatorios se colocará un asterisco seguido del título de cada campo respectivamente.

### ***Creación de la ventana para el manejo de Usuarios:***

Se crea una ventana en donde obtiene la lista de todos los usuarios que pueden ingresar al sistema.

Esta ventana permitirá hacer búsquedas de usuarios según el nombre del mismo, adicional permitirá crear, editar y eliminar usuarios.

Al crear un nuevo usuario permitirá ingresar la siguiente información:

Cédula del usuario

Nombre del usuario

Código del perfil de Usuario

Clave del perfil de Usuario

Confirmación de clave de usuario

Al modificar el usuario el sistema podrá modificar solo los siguientes campos:

Nombre del usuario

Fecha que se registró al usuario

Fecha que expira el permiso del usuario

***Validación del formulario Usuario:***

El sistema validará que la cédula no se repita.

El sistema validará que la cédula sea correcta.

El sistema validará que se ingrese todos los campos obligatorios. Para diferenciar los campos obligatorios se colocará un asterisco seguido del título de cada campo respectivamente.

***Creación de la ventana para el manejo de Parametrización:***

Para este módulo de parametrización se creará una ventana en donde se listará todos los servicios que tiene en el sistema, esta parametrización permitirá ingresar, modificar y eliminar los servicios que presta el taller.

Para buscar un servicio determinado se deberá ingresar el nombre del servicio para que el sistema retorne una lista con todos los servicios posibles al nombre ingresado.

Al querer editar o modificar se presentará una ventana emergente en donde se ingresa los siguientes campos:

Código del servicio

Nombre del servicio

Descripción del servicio

**Validación del formulario de Parametrización:**

Se validará que ingrese todos los campos obligatorios marcador con asterisco en cada etiqueta.

El sistema validará que los códigos de repuesto no se repitan.

• **Historia de Usuario 3 en el Diseño**

CONTROL DE INVENTARIO							
FECHA	08/12/2011	TIPO DE ACTIVIDAD	Nuevo	X	PRIORIDAD TÉCNICA	Normal	X
			Corrección			Medio	
			Mejora			Alto	
REFERENCIA		RIESGO	Ninguno	X	PRIORIDAD USUARIO	Normal	X
			Medio			Medio	
			Alto			Alto	
USUARIO		CARGO		CLASIFICACIÓN	Inventario		
DESCRIPCIÓN							
Creación, modificación y eliminación de inventarios. Buscar inventario por fecha que se inició el inventario. Inventario por cada repuesto existente en el sistema. Modificación solo si el repuesto inventariado consta como aun no aplicado. Modificar inventario general solo si consta como no aplicado. La cantidad existente no debe ser menor a la cantidad del stock mínimo.							
NOTAS							
SEGUIMIENTO							
ESTADO	Atendido		COMENTARIOS		ÚLTIMO SEGUIMIENTO		
	En Proceso	X					
	Finalizado						

Tabla 11: Historia de Usuario 3 en el Diseño

## Historia de Usuario 3 en el Desarrollo

### ***Creación de ventanas para Inventarios:***

Se creará las ventanas pertinentes para el desarrollo del módulo de inventarios en donde lista todos los inventarios realizados, estos se podrán crear, modificar y eliminar siempre y cuando no consten como APLICADOS y haya caducado la fecha del último inventario realizado.

Los inventarios se los realizará cada mes.

Para realizar el inventario se abre un inventario general donde se ingresará los siguientes datos:

Cédula del responsable del inventario

Nombre del responsable del inventario

Fecha que inicia el inventario

Fecha que expira el inventario

Comentario general del inventario

Se presentará un multiregistro en donde listará o ingresa los inventarios individuales según cada repuesto o producto, al escoger el producto para hacer el inventario, se debe presentar la información del producto (repuesto), mostrando la cantidad existente y el stock de cada repuesto.

Los campos que se ingresa por cada producto son:

Nueva cantidad existente del producto

Comentarios del inventario

Campo para seleccionar si se desea cerrar el inventario para el producto, si se da como aplicado ya no se podrá modificar el inventario en otra ocasión.

***Validación del formulario de Inventarios:***

El sistema validará que se ingrese todos los campos obligatorios. Para diferenciar los campos obligatorios se colocará un asterisco seguido del título de cada campo respectivamente.

La nueva cantidad que se ingresa para el inventario del producto seleccionado debe ser mayor que el stock mínimo y que la cantidad existente actualmente.

El sistema debe validar que en el caso de haber cerrado el inventario tanto por producto como por inventario general no se podrá modificar el inventario.

El sistema debe validar que solo se ingrese un inventario por mes.

El sistema debe listar todos los productos existentes en el sistema y según se haga el inventario por cada uno de ellos se lo restará de esta lista.



ESPACIO EN BLANCO INTENCIONAL

- **Historia de Usuario 4 en el Diseño**

CONTROL DE ORDENES							
FECHA	08/12/2011	TIPO DE ACTIVIDAD	Nuevo	X	PRIORIDAD TÉCNICA	Normal	X
			Corrección			Medio	
			Mejora			Alto	
REFERENCIA		RIESGO	Ninguno	X	PRIORIDAD USUARIO	Normal	X
			Medio			Medio	
			Alto			Alto	
USUARIO		CARGO			CLASIFICACIÓN	Facturación	
DESCRIPCIÓN							
Creación, modificación y eliminación de órdenes. Validación entre el kilometraje actual del vehículo comparándolo con el kilometraje nuevo al momento de hacer la orden. Información del cliente a quien se le va a asignar la orden. Responsable de la orden. Mensajes de alerta de mantenimiento. Ingreso de todos los servicios que se van a realizar al vehículo por orden. Fecha de recepción y fecha de entrega del vehículo. Registro de artículos con las que ingreso el vehículo. Registro de nivel de combustible en el que se encuentra el vehículo							
NOTAS							
SEGUIMIENTO							
ESTADO	Atendido		COMENTARIOS			ÚLTIMO SEGUIMIENTO	
	En Proceso	X					
	Finalizado						

Tabla 12: Historia de Usuario 4 en el Diseño

**Creación de ventanas para Órdenes:**

Se creará la ventanas pertinentes para la creación del módulo de ordenes en donde lista todos las ordenes realizadas, estas se podrán crear, modificar y eliminar.

Para realizar búsqueda de órdenes se la realizará por medio del número de orden único en el sistema.

Al crear una nueva orden se buscará la información del cliente buscando por el número de placa.

Para ingresar los datos de la orden se debe seleccionar con que artículos ingreso el vehículo.

Los datos que se ingresa para cargar una nueva orden serán:

Responsable

Kilometraje actual

Fecha de recepción

Fecha de entrega

Nivel de combustible con el que ingreso el vehículo.

Se presentará un multiregistro en donde se irá ingresando todos los servicios y detalles del mismo, realizados a determinado vehículo y que serán parte de la orden.

***Validación del formulario de Órdenes:***

El sistema validará que se ingrese todos los campos obligatorios. Para diferenciar los campos obligatorios se colocará un asterisco seguido del título de cada campo respectivamente.

El sistema al ingresar el nuevo kilometraje, hará la comparación con el último kilometraje registrado en el sistema y validará que tipo de mantenimiento se debería hacer el vehículo, este puede ser cada cinco mil, diez mil, quince mil o veinte mil kilómetros de recorrido.

El sistema deberá mandar un mensaje de cuál será el mantenimiento periódico que debería hacerse y según el mensaje obtenido, el usuario registrará el servicio correspondiente, en el sistema.

- **Historia de Usuario 5 en el Diseño**

CONTROL DE FACTURACIÓN							
FECHA	08/12/2011	TIPO DE ACTIVIDAD	Nuevo	X	PRIORIDAD TÉCNICA	Normal	X
			Corrección			Medio	
			Mejora			Alto	
REFERENCIA		RIESGO	Ninguno	X	PRIORIDAD USUARIO	Normal	X
			Medio			Medio	
			Alto			Alto	
USUARIO		CARGO			CLASIFICACIÓN	Inventario	
DESCRIPCIÓN							
<p>Creación y anulación de facturas.            Búsqueda por número de factura.            Generación secuencial de número de factura.            Lista de servicios que se hicieron para la orden y que se facturarán.            Listado de repuestos que se asignarán a un servicio.            Detalle de facturación incluyendo cantidad, descripción de servicio y repuesto asignado, costo unitario y total, IVA, y total a pagar.            La factura no se puede eliminar solo cancelar.</p>							
NOTAS							
Tomar en cuenta que de este módulo se sacará información según el número de visitas del cliente para sacar la lista de los clientes que tienen derecho a promociones.							
SEGUIMIENTO							
ESTADO	Atendido		COMENTARIOS		ÚLTIMO SEGUIMIENTO		
	En Proceso	X					
	Finalizado						

Tabla 13: Historia de Usuario 5 en el Diseño

**Creación de ventanas para Facturación:**

Se creará las ventanas pertinentes para la creación del módulo de facturación en donde lista todas las facturas realizadas, estas se podrán crear y anular.

Para realizar una nueva factura se deberá hacer una búsqueda por el número de orden, esto permitirá cargar los datos ya generados de la orden que son:

Nombre del cliente.

Ruc del cliente.

Teléfono del cliente.

Dirección del cliente.

Fecha en la que se realizará la facturación.

Descripción del producto.

Nombre de los servicios.

Cantidad de los repuestos que se van a usar.

Al escoger el servicio y el repuesto que se asignó a ese servicio se lo agregará al detalle y este listará automáticamente el precio unitario y total por producto.

También deberá sacar un total general incluyendo el IVA para posteriormente proceder a guardar la factura.

#### ***Validación del formulario de Facturación:***

El sistema validará que se ingrese todos los campos obligatorios. Para diferenciar los campos obligatorios se colocará un asterisco seguido del título de cada campo respectivamente.

No se podrán eliminar las facturas ya que se debe tener un registro de todas las facturas realizadas, para esto se podrá simplemente anular la factura y después de anularla se podrá asignar un nuevo número de factura para la misma orden.

El sistema debe validar que no se ingresen ordenes repetidas si existiese ordenes repetidas el sistema deberá decir a que número de factura pertenece esa orden.

- **Historia de Usuario 6 en el Diseño**

CONTROL DE CAMPAÑAS							
FECHA	08/12/2011	TIPO DE ACTIVIDAD	Nuevo	X	PRIORIDAD TÉCNICA	Normal	X
			Corrección			Medio	
			Mejora			Alto	
REFERENCIA		RIESGO	Ninguno	X	PRIORIDAD USUARIO	Normal	X
			Medio			Medio	
			Alto			Alto	
USUARIO		CARGO			CLASIFICACIÓN	Inventario	
DESCRIPCIÓN							
Listado de clientes frecuentes parametrizando el número de visitas.							
NOTAS							
SEGUIMIENTO							
ESTADO	Atendido		COMENTARIOS			ÚLTIMO SEGUIMIENTO	
	En Proceso	X					
	Finalizado						

Tabla 14: Historia de Usuario 6 en el Diseño

**Creación de ventanas para Campañas:**

Se creará las ventanas pertinentes para la creación del módulo de campañas en donde lista todos los clientes que pueden acceder a la promoción vigente según el número de visitas que han realizado al taller.

Se ingresará el número de visitas en el sistema y según el número de visitas listará los clientes que tienen acceso a esa promoción.

### ***Validación del formulario de Campañas:***

El sistema contabilizará el número de visitas que ha realizado el cliente y que han sido facturadas.

El sistema validará que solo tome en cuenta las facturas validas no las anuladas.

### **3.1.2 Diseño**

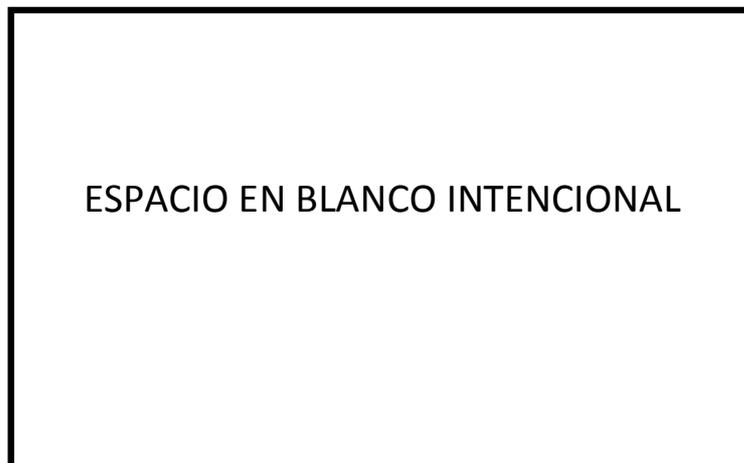
En el diseño el investigador utilizará algunas estrategias XP como: “La cosa más sencilla podría funcionar” para lo cual se necesita utilizar las 4 variables antes mencionadas por el autor (comunicación, sencillez, realimentación y valentía) asegurando la buena aplicación de la estrategia.

Se evita con esto la duplicidad de código, se aprende de la recodificación ya que al ser sencillo no costaría mucho arreglarlo y mejorarlo.

Otra estrategia importante es contar con una arquitectura del sistema como columna vertebral del mismo.

#### **3.1.2.1 Arquitectura**

La arquitectura que se utilizará es de aplicaciones web de 3 capas, debido a que permite la independencia de cada una.



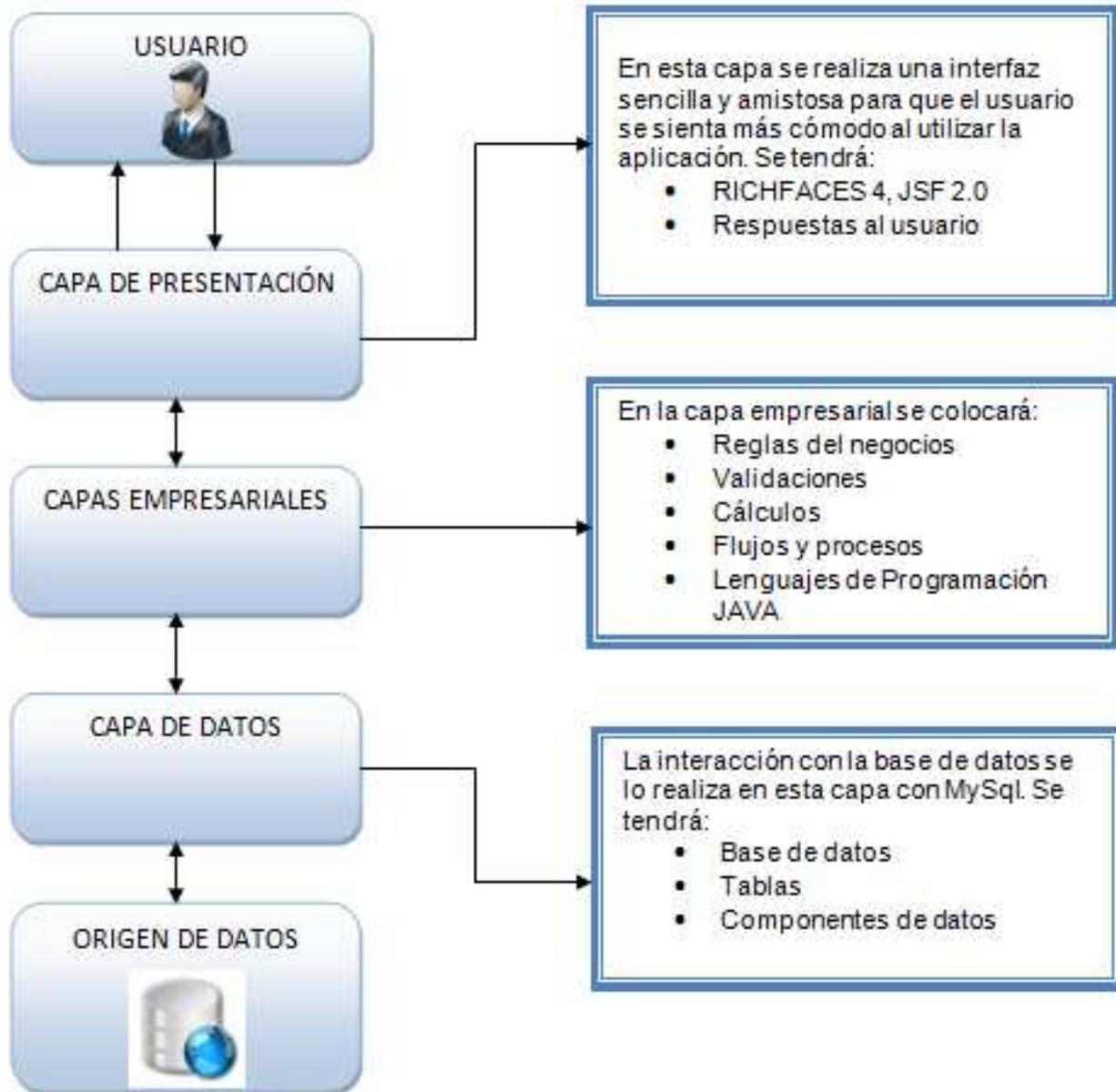


Gráfico 14: Arquitectura

### 3.1.2.2 Interfaz de la aplicación

En el diseño de la interfaz de usuario se explicará el conjunto de elementos que forman y darán cuerpo a la aplicación. Para esto se utiliza JSF 2.0 Y RICHFACES 4.

- **Nomenclatura de objeto para objetos estándar.**

La utilización de nomenclaturas reconocidas permite tener un estándar para la colocación de nombres a los objetos.

Para esto utiliza los siguientes tag's más conocidos en JSF:

`h:inputText, h:inputSecret`

Example	Result
<code>&lt;h:inputText value="#{form.testString}" readonly="true" /&gt;</code>	<input type="text" value="12345678901234567890"/>
<code>&lt;h:inputSecret value="#{form.passwd}" redisplay="true" /&gt;</code>	<input type="password" value="*****"/>
<code>&lt;h:inputSecret value="#{form.passwd}" redisplay="false" /&gt;</code>	<input type="password"/>
<code>&lt;h:inputText value="inputText" style="color: Yellow; background: Teal;" /&gt;</code>	<input type="text" value="input_text"/>
<code>&lt;h:inputText value="1234567" size="5" /&gt;</code>	<input type="text" value="123456"/>
<code>&lt;h:inputText value="1234567890" maxlength="6" size="10" /&gt;</code>	<input type="text" value="123456"/>

Tabla 15: Atributos `inputText` y `inputSecret`

ESPACIO EN BLANCO INTENCIONAL

h:inputTextarea

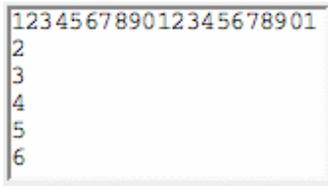
Example	Result
<pre>&lt;h:inputTextarea rows="5"/&gt;</pre>	

Tabla 16: Atributos inputTextarea

h:outputText **and** h:graphicImage **Examples**

Example	Result
<pre>&lt;h:outputText value="#{form.testString}"/&gt;</pre>	12345678901234567890
<pre>&lt;h:graphicImage value="/tjefferson.jpg"/&gt;</pre>	

Tabla 17: Atributos outputText, graphicImage

h:commandButton

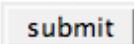
Example	Result
<pre>&lt;h:commandButton value="submit" type="submit"/&gt;</pre>	

Tabla 18: Atributos commandButton

h:commandLink

Example	Result
<pre>&lt;h:commandLink value="welcome" actionListener="#{form.useLinkValue}" action="#{form.followLink}"/&gt;</pre>	

Tabla 19: Atributos commandLink

### 3.1.2.3 Interfaces del Sistema

Para el ingreso al sistema se presenta la ventana de **Entrada al Sistema (Login)**, este formulario contendrá 2 cajas de texto para que el usuario ingrese su cuenta y clave.

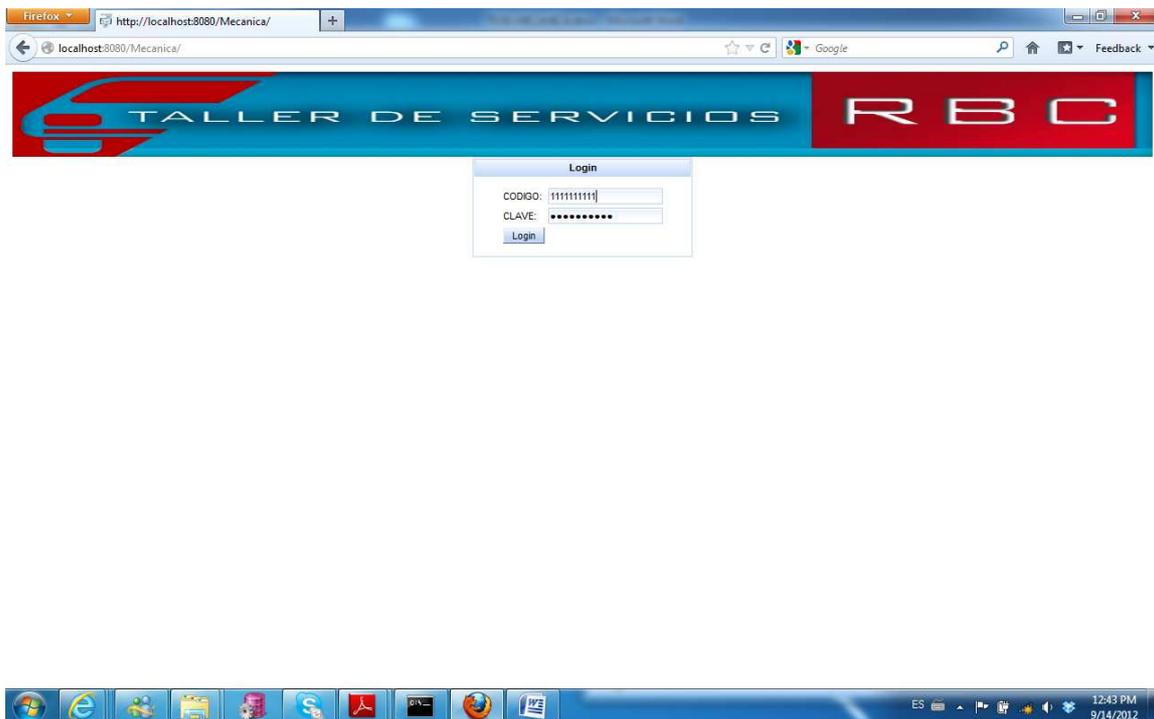
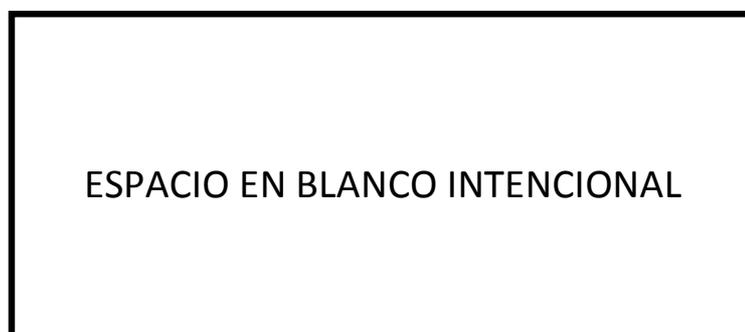


Gráfico 15: Login



El **Menú Principal** permite un manejo adecuado de la aplicación, los item's son: **REPUESTOS, FACTURACIÓN, INVENTARIOS, CLIENTES, CAMPAÑAS, USUARIOS, ORDEN, PROVEEDORES Y PERSONALIZACIÓN.**



*Gráfico 16: Pagina Principal*



En el formulario **Perfil de Usuario** se realiza el listado, ingreso, actualización, búsqueda y eliminación de los registros.

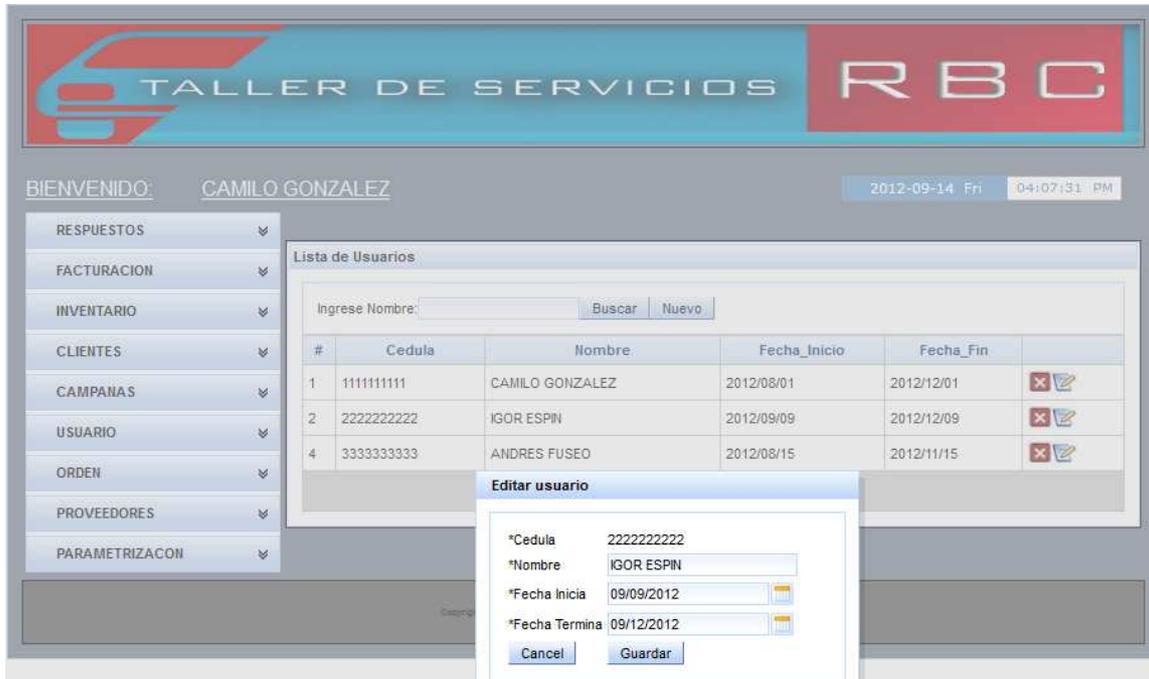


Gráfico 17: Usuarios

En el formulario **Personas** se realiza el listado, ingreso, actualización, búsqueda y eliminación de los registros.



Gráfico 18: Personas

En el formulario **Cientes** se realiza el listado, ingreso, actualización, búsqueda y eliminación de los registros.



Gráfico 19: Clientes

En el formulario **Producto** se realiza el listado, ingreso, actualización, búsqueda y eliminación de los registros.



Gráfico 20: Productos

En el formulario **Proveedor** se realiza el listado, ingreso, actualización, búsqueda y eliminación de los registros.



Gráfico 21: Proveedor Maestro



Gráfico 22: Proveedor Detalle

En el formulario **Inventario** se realiza el listado, ingreso, actualización, búsqueda y eliminación de los registros.

BIENVENIDO: CAMILO GONZALEZ 2012-09-14 Fri 04:26:46 PM

RESPUESTOS  
FACTURACION  
INVENTARIO  
CLIENTES  
CAMPANAS  
USUARIO  
ORDEN  
PROVEEDORES  
PARAMETRIZACION

**Lista de Inventarios**

Ingrese la Fecha de Inicio de Inventario:

#	Fecha Inicio	Fecha Expira	Responsable	Cedula	Aplicada	
1	01/08/2012	28/08/2012	JUAN PIO MONTUFAR	1111111111	<input checked="" type="checkbox"/>	<input type="button" value="X"/> <input type="button" value="E"/>
2	29/08/2012	31/08/2012	ERNESTO CORREA	2222222222	<input checked="" type="checkbox"/>	<input type="button" value="X"/> <input type="button" value="E"/>
3	03/09/2012	11/09/2012	FREILES S.A.	6666666666	<input type="checkbox"/>	<input type="button" value="X"/> <input type="button" value="E"/>

<<<< << < > >>>>

Copyright © 2012 Camilo González. Todos los derechos reservados.

Gráfico 23: Inventario Maestro

BIENVENIDO: CAMILO GONZALEZ 2012-09-14 Fri 04:28:34 PM

Lista de Repuestos para Inventariar

CodRepuesto	NombreRepuesto	
REP_3	REPUESTO 3	<input type="button" value="Inventariar"/>
REP_4	REPUESTO 4	<input type="button" value="Inventariar"/>
REP_5	REPUESTOS	<input type="button" value="Inventariar"/>
REP_6	REPUESTO 6	<input type="button" value="Inventariar"/>
REP_7	REPUESTO 7	<input type="button" value="Inventariar"/>
REP_8	REPUESTO 8	<input type="button" value="Inventariar"/>
PRO_9	PRODUCTO 9	<input type="button" value="Inventariar"/>

Regresar

CAMPANAS  
USUARIO  
ORDEN  
PROVEEDORES  
PARAMETRIZACION

**Inventario**

Regresar

Datos Inventario

\* Cedula: 6666666666 \* Nombre Responsable: FREILES S.A.  
\* Fecha Inicia: 03/09/2012 \* Fecha Expira: 11/09/2012  
\* Comentario General:  
INVENTARIO CORRESPONDIENTE AL MES DE SEPTIEMBRE

Detalle Inventario

Cod.Producto	Nom.Producto	Cantidad	Stock	Aplicado	
REP_1	REPESTO 1	48	5	<input type="checkbox"/>	<input type="button" value="X"/> <input type="button" value="E"/>
REP_2	REPESTO 2	48	5	<input type="checkbox"/>	<input type="button" value="X"/> <input type="button" value="E"/>

Inventariar Producto

Cerrar Inventario?

Aplicado?

Guardar Inventario

**Inventario por Producto**

Información del Repuesto

Cod. Repuesto: REP\_5 Descr. Repuesto: REPUESTOS  
Compra: \$45.34 Venta: \$69.98  
Cantidad Existente: 21 Stock: 5

Información del Inventario

\*Nueva Cantidad: 0

Comentarios:

Aplicarlo?

Regresar Aceptar

Gráfico 24: Inventario Detalle

En el formulario **órdenes** se realiza el listado, ingreso, actualización, búsqueda y eliminación de los registros.

BIENVENIDO: CAMILO GONZALEZ 2012-09-14 Fri 04:34:20 PM

Toma de Orden

Ingrese Numero de Orden: 0

#	Nombre	Apellido	Fecha Recepcion	Fecha Entrega	Responsable	
1	GALO MARCO	GONZALEZ CASTILLO	2012-08-29	2012-08-30	JUAN TOAPANTA	<input type="checkbox"/> <input type="checkbox"/>
2	JUAN CARLOS	BAEZ PONCE	2012-08-29	2012-08-31	PABLO MONTOYA	<input type="checkbox"/> <input type="checkbox"/>
3	GALO MARCO	GONZALEZ CASTILLO	2012-09-04	2012-09-13	JUAN TOAPANTA	<input type="checkbox"/> <input type="checkbox"/>
4	GALO MARCO	GONZALEZ CASTILLO	2012-09-04	2012-09-13	JUAN TOAPANTA	<input type="checkbox"/> <input type="checkbox"/>
5	JUAN CARLOS	BAEZ PONCE	2012-09-07	2012-09-09	JUAN CANO	<input type="checkbox"/> <input type="checkbox"/>
6	JUAN CARLOS	BAEZ PONCE	2012-09-07	2012-09-10	JUAN TORRES	<input type="checkbox"/> <input type="checkbox"/>
7	JUAN CARLOS	BAEZ PONCE	2012-09-13	2012-09-18	JUAN TORRES CANO	<input type="checkbox"/> <input type="checkbox"/>

Copyright © 2012 Camilo Gonzalez. Todos los derechos reservados.

Gráfico 25: Orden Maestro



**Orden**

Regresar

\* Ingrese Placa del Vehiculo:

**Orden No:2**

**Datos Persona**

Cedula:

Nombre:  Apellido:

Telefono:  Celular:

Mail:  Fecha Nacimiento:

Direccion:

**Datos Vehiculo**

Tipo:  Marca:

Kilometraje:  CHasis:

Color:  Anio:

**Articulos que tiene el vehiculo**

BOTQUIN

ENCENDEDOR

EXTINGUIDOR

GATA

**Ingresar Datos Orden**

\* Responsable:  \* Kilometraje Actual:

\* Fecha Recepcion:  \* Fecha Entrega:

\* Nivel de Gasolina:

\* Orden:

\* Detalle Orden:

#	Servicio	Detalle	
0	CAMBIO FILTRO DE AIRE	EFWRWEGFDSGDFGD	✘
1	CAMBIO ACEITE	DSFADSGFSDGSDFG	✘
2	CAMBIO DE BUJIAS	CCCCCCCCCCCCCCCCCCCC	✘

Copyright © 2012 Camilo Gonzalez. Todos los derechos reservados.

Gráfico 26: Orden Detalle

En el formulario **Parametrización** se realiza el listado, ingreso, actualización, búsqueda y eliminación de los registros.



Gráfico 27: Parametrización

En el formulario **Campañas** se realiza el listado, ingreso, actualización, búsqueda y eliminación de los registros.



Gráfico 28: Campañas

El formulario **Facturación** emite la factura.

**Lista de Facturas**

Ingrese Nº de Factura:

NºFactura	RUC	Subtotal	IVA	Total	Fecha	Anulada?
0000001	222222222	80	10	90	2012-09-09	<input checked="" type="checkbox"/>
0000002	222222222	362	43	405	2012-09-10	<input type="checkbox"/>
0000003	222222222	483	58	541	2012-09-10	<input type="checkbox"/>
0000004	111111111	282	34	316	2012-09-10	<input type="checkbox"/>
0000005	111111111	181	22	202	2012-09-10	<input type="checkbox"/>
0000006	111111111	383	46	429	2012-09-10	<input type="checkbox"/>
0000007	222222222	81	10	91	2012-09-10	<input checked="" type="checkbox"/>

Gráfico 29: Facturación Maestro

**Crear Factura**

Nº de Orden: 0

Descripción Producto:  Detalles Orden:

Código:  Cantidad: 1.0

Precio:  Cantidad Existente:

Cliente: SHIRLEY JAZMINE ORTEGA PRADO  
 RUC/C: 3333333333 Telefono: 022345678  
 Dirección: safdsghsssfdfgsfgsgsf  
 Fecha: 14/09/2012

Factura 001 - 001  
 Nº 0000014  
 AUT.S.R.I: 1110901201

Cant	Servicio	Descripción	V.Unitario	V.Total
1.00	CAMBIO ACEITE	REPIESTO 1	81.00	81.00
3.00	CAMBIO DE BUJIAS	REPUESTO 2	80.00	240.00
<b>Subtotal</b>				<b>\$321.00</b>
<b>I.V.A%</b>				<b>\$38.52</b>
<b>Total</b>				<b>\$359.52</b>

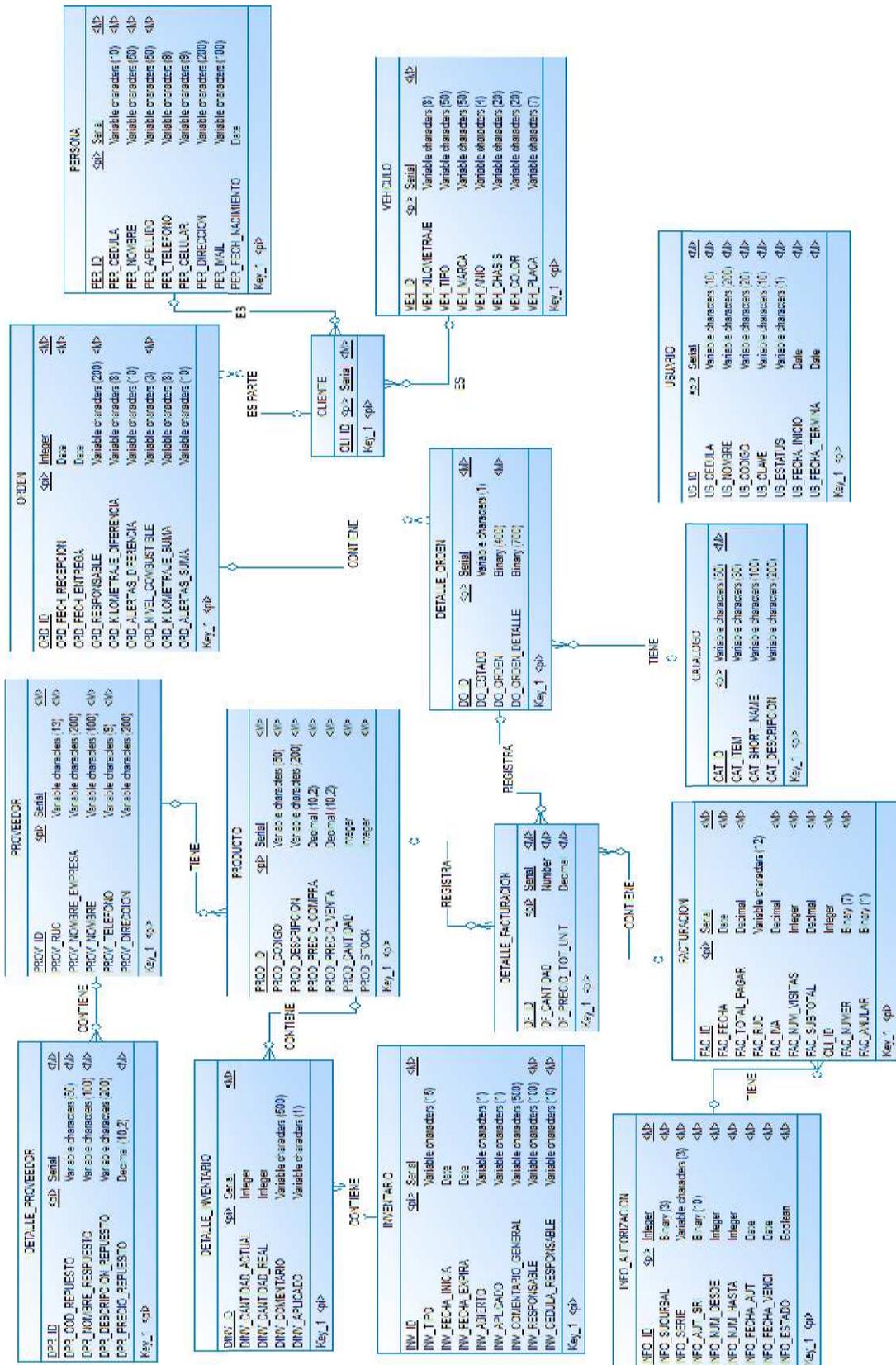
Gráfico 30: Facturación Detalle

### **3.1.2.4Capa de Datos**

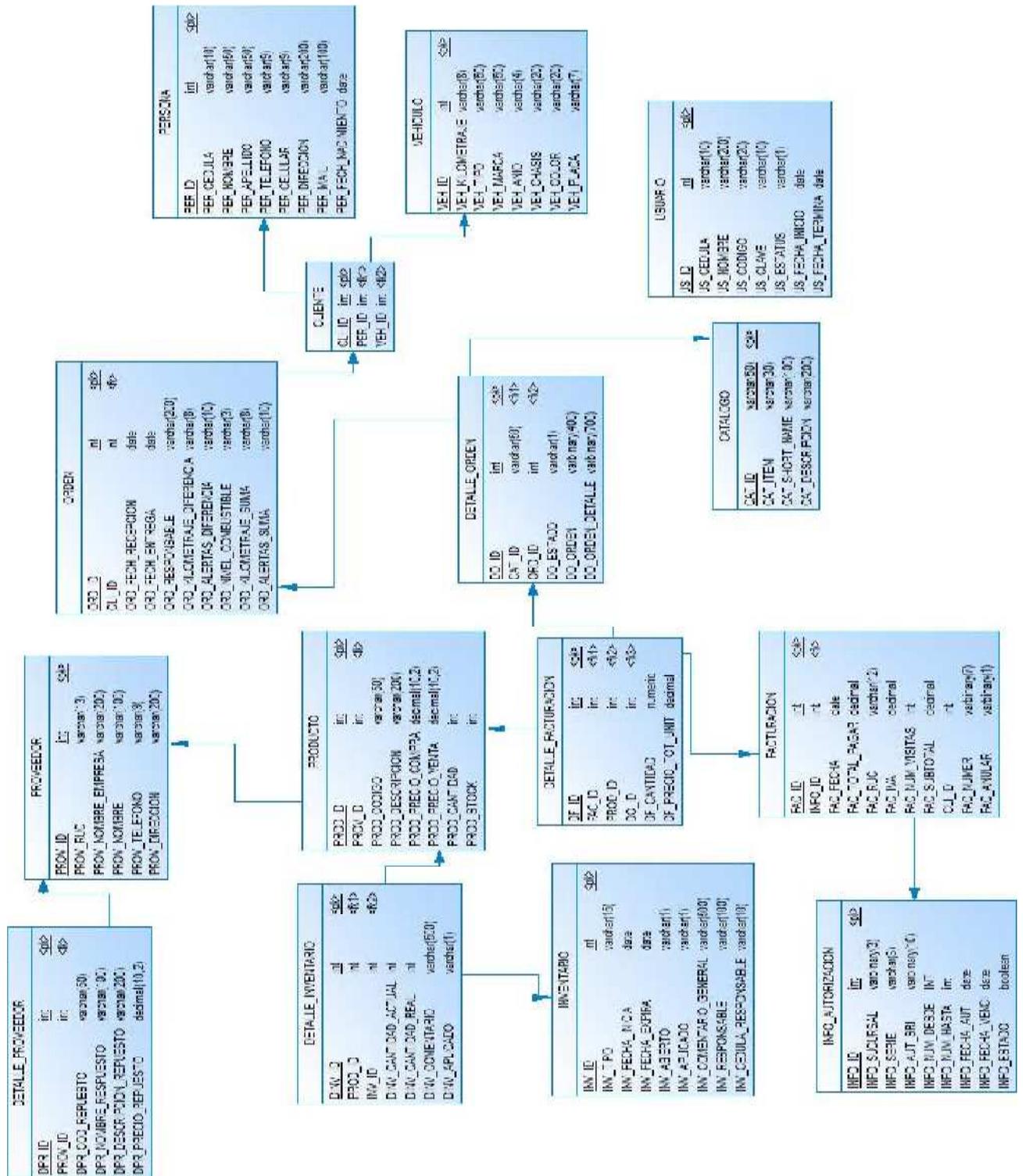
Para poder entender la forma en que se debe almacenar los datos, permitirá entender mejor al sistema, para esto se realizó el modelado de la Base de Datos del proyecto.



• **Modelo Conceptual**



- Modelado Físico



### **3.1.2.5 Creación de escenarios y de Tarjetas CRC**

Para poder realizar las Tarjetas CRC primero se tiene que realizar una lluvia de ideas para poder identificar cada uno de los elementos que intervienen en las tarjetas. A continuación se presenta los identificados:

- PERSONA
- VEHÍCULO
- CLIENTE
- ÓRDEN
- DETALLE ORDEN
- PRODUCTO
- PROVEEDOR
- DETALLE PROVEEDOR
- INVENTARIO
- DETALLE INVENTARIO
- FACTURACIÓN
- DETALLE FACTURACIÓN
- PERFIL DE USUARIO
- CATÁLOGOS
- INFORMACIÓN PARA LA AUTORIZACIÓN DE LA FACTURA

#### **a) Escenarios**

Antes de la realización de las tarjetas se tiene que describir los posibles escenarios en los que el sistema y los actores interactúan.

#### **Escenario 1: Ingreso al sistema**

- El usuario ingresa su código "1717197915" y su clave "\*\*\*\*\*"

- El sistema validó la existencia de su código y la clave ingresada por El usuario. Si esta cuenta existe el sistema continúa con el ingreso cargando las variables con los permisos respectivos y por último presentando el Menú principal.

### **Escenarios secundarios**

- Si no existe la cuenta de usuario o la clave es incorrecta se presentará una mensaje indicando que “la cuenta no existe o que la clave es incorrecta”
- Si la cuenta del usuario caduco, mandará automáticamente a una ventana en donde se puede actualizar el código y la clave del usuario dando esta un nuevo tiempo de vigencia por tres meses adicionales.

### **Escenario 2: Creación, actualización y eliminación de un usuario**

- El usuario ingresa un nuevo usuario, ingresa al sistema y selecciona en el menú principal la opción de usuario en donde desplegará la lista de todos los usuarios que pueden ingresar al sistema. El usuario ingresa los siguientes datos:

<i>CÉDULA:</i>	<i>17171979154</i>
<i>NOMBRES:</i>	<i>Camilo González</i>
<i>CÓDIGO:</i>	<i>1717197914</i>
<i>CLAVE:</i>	<i>*****</i>
<i>CONFIRMAR CLAVE:</i>	<i>*****</i>

*Tabla 20: Datos Usuario*

- El usuario actualizará los datos de un usuario, consulta la lista de todos los usuarios, selecciona el usuario “Roberto” y Cambia la clave del mismo.
- El usuario eliminará una cuenta de usuario, consulta la lista de todos los usuarios, selecciona el usuario “Roberto” y lo elimina.

### Escenario 3: Creación, actualización y eliminación de una persona

- El usuario ingresa una nueva persona, ingresa al sistema y selecciona en el menú principal la opción de clientes y escoge la opción de personas, en donde desplegará la lista de todas las personas que constan en el sistema. El usuario ingresa los siguientes datos:

<i>CÉDULA:</i>	<i>17171979154</i>
<i>NOMBRES:</i>	<i>JUAN CARLOS</i>
<i>APELLIDOS:</i>	<i>BAEZ PONCE</i>
<i>TELF. DOMICILIO:</i>	<i>022657465</i>
<i>TELF. CELULAR:</i>	<i>098456347</i>
<i>MAIL:</i>	<i>JC@hotmail.com</i>
<i>FECHA NACIMIENTO:</i>	<i>15/08/1988</i>
<i>DIRECCIÓN:</i>	<i>ANA DE ALFARO #371 Y DOLORES DE GALINDO</i>

*Tabla 21: Datos Persona*

- El usuario actualizará los datos de una persona, consulta la lista de las personas, selecciona a “María José Espinosa” o si prefiere buscar

por el número de cédula, cambia la dirección” *ANA DE ALFARO #389 Y DOLORES DE GALINDO*”.

- El usuario quiere eliminar una persona, consulta la lista de personas, selecciona a “María José Espinosa” y la elimina.

### **Escenario 5: Creación, actualización y eliminación de un cliente**

- El usuario ingresa un nuevo cliente, ingresa al sistema y selecciona en el menú principal la opción de clientes, en donde se ingresa la cédula de una persona ya registrada, esta permitirá agregarle los vehículos que se desea para esta persona y se convertirá en un cliente. El usuario ingresa los siguientes datos del vehículo:

<i>PLACA:</i>	<i>IOI0989</i>
<i>TIPO:</i>	<i>AUTO</i>
<i>MARCA:</i>	<i>TOYOTA YARIS</i>
<i>COLOR:</i>	<i>NEGRO</i>
<i>AÑO:</i>	<i>2011</i>
<i>KILOMETRAJE:</i>	<i>9876</i>
<i>CHASIS:</i>	<i>657HYDT4673GYS</i>

*Tabla 22: Datos Cliente*

- El usuario actualizará los datos de un cliente, consulta la lista los clientes, ingresa el número de cédula del cliente que va a modificar “1719171596” y escogerá el vehículo que desea modificar en este caso “*IOI0989*” y modifica el kilometraje “10000”.
- El usuario eliminará los datos de un cliente, consulta la lista los clientes, ingresa el número de cédula del cliente que va a eliminar “1719171596” y escogerá el vehículo que desea eliminar en este caso “*IOI0989*”.

## Escenario 6: Creación, actualización y eliminación de órdenes

- El usuario ingresa una nueva orden, ingresa al sistema y selecciona en el menú principal la opción de órdenes, en donde desplegará la lista de todos las ordenes que constan en el sistema. Al crear una nueva orden el sistema pide que se ingrese el número de placa del vehículo en este caso será "1010989" a quien se va a realizar la orden, el mismo que desplegará la información del cliente y vehículo El usuario ingresa los siguientes datos para la orden:

### **ARTÍCULOS CON LOS QUE INGRESA EL VEHÍCULO.**

GATA
EXTINGUIDOR

Tabla 23: Datos Ingresar Vehículo

### **TOMA DE LA ORDEN**

RESPONSABLE:	JUAN MUÑOZ
KILOMETRAJE ACTUAL:	12000
FECHA DE RECEPCIÓN:	09/20/2012
FECHA ENTREGA:	09/25/2012
NIVEL DE GASOLINA:	90%
ORDEN:	CAMBIO ACEITE
DETALLE ORDEN:	CAMBIO DE ACEITE POR MANTENIMIENTO

Tabla 24: Datos Toma de Orden

- El usuario actualizará los datos de una orden, consulta la lista de las ordenes, selecciona la orden "# 6" o si prefiere buscar por el número de orden, cambia la fecha de entrega "09/27/2012".

- El usuario quiere eliminar una orden, consulta la lista de personas, selecciona la orden “# 6” y la elimina.

### Escenario 7: Creación, actualización y eliminación de proveedores

- El usuario ingresa un nuevo proveedor, ingresa al sistema y selecciona en el menú principal la opción de proveedores, en donde desplegará la lista de todos los proveedores que constan en el sistema. El usuario ingresa los siguientes datos:

<i>RUC:</i>	<i>1635475894001</i>
<i>PROPIETARIO:</i>	<i>JUAN TORAL</i>
<i>TELÉFONO:</i>	<i>022675884</i>
<i>NOMBRE EMPRESA:</i>	<i>TORAL Y ASOCIADOS Ltd.</i>
<i>DIRECCIÓN:</i>	<i>MANUELA CARBAJAL #543 Y AMAZONAS</i>

*Tabla 25: Datos Proveedor*

- El usuario ingresará los datos de los productos que proporciona este proveedor y serán los siguientes:

<i>CÓDIGO:</i>	<i>FIL_001</i>
<i>NOMBRE REPUESTO:</i>	<i>FILTRO DE AIRE</i>
<i>PRECIO:</i>	<i>\$ 85.50 DÓLARES</i>
<i>DESCRIPCIÓN:</i>	<i>FILTRO DE AIRE PARA MAZDA 3</i>

*Tabla 26: Datos Repuesto Proveedor*

- El usuario actualizará los datos de un proveedor, consulta la lista de los proveedores, selecciona un proveedor “TORAL Y ASOCIADOS Ltd.” o si prefiere buscar por el nombre de la empresa, se cambia la dirección de la empresa “MANUELA CARBAJAL #400 Y AMAZONAS”.

- El usuario quiere eliminar un proveedor, consulta la lista de proveedores, selecciona un proveedor “TORAL Y ASOCIADOS Ltd.” y la elimina.

### Escenario 8: Creación, actualización y eliminación de inventarios

- El usuario ingresa un nuevo inventario, ingresa al sistema y selecciona en el menú principal la opción de inventarios, en donde desplegará la lista de todos los inventarios que constan en el sistema. El usuario ingresa los siguientes datos:

<i>CÉDULA:</i>	<i>6475893647</i>
<i>NOMBRE RESPONSABLE:</i>	<i>PABLO QUINTANA</i>
<i>FECHA INICIO:</i>	<i>29/08/2012</i>
<i>FECHA EXPIRA:</i>	<i>31/08/2012</i>
<i>COMENTARIO GENERAL:</i>	<i>INVENTARIO SIN NINGUNA NOVEDAD</i>

*Tabla 27: Datos Inventario*

- El usuario ingresará los datos de las cantidades nuevas del inventario por cada repuesto y los datos a ingresar son:

<i>NUEVA CANTIDAD:</i>	<i>56</i>
<i>COMENTARIOS:</i>	<i>EXISTÍAN ANTES DEL INVENTARIO 48 UNIDADES AL MOMENTO DEL INVENTARIO SE INGRESO 56 UNIDADES</i>
<i>APLICADO</i>	<i>SI</i>

*Tabla 28: Datos Repuesto Inventario*

- El usuario actualizará los datos de un inventario, consulta la lista de los inventarios, tomando en cuenta que el inventario no esté aplicado aun, selecciona un inventario en este caso el inventario perteneciente a la fecha de inicio “29/08/2012” o si prefiere buscar por la fecha de inicio de inventario, se cambia la fecha de expiración “01/09/2012”.
- El usuario quiere eliminar un inventario, consulta la lista de inventarios, tomando en cuenta que el inventario no conste como aplicado, selecciona un inventario “29/08/2012.” y lo elimina.

### **Escenario 9: Creación, anulación de facturación**

- El usuario ingresa una nueva factura, ingresa al sistema y selecciona en el menú principal la opción de facturación, en donde desplegará la lista de todas las facturas realizadas y que constan en el sistema, se deberá ingresar el número de orden para poder facturar dicha orden, El usuario ingresa los siguientes datos:

<i>NÚMERO DE ORDEN:</i>	<i>12</i>
<i>DETALLE ORDEN:</i>	<i>CAMBIO DE ACEITE</i>
<i>DESCRIPCIÓN DEL PRODUCTO:</i>	<i>GALÓN DE ACEITE 20W50CANTIDAD: 2</i>
<i>FECHA:</i>	<i>09/20/2012</i>

*Tabla 29: Datos Facturación*

- El usuario anulará una factura, consulta la lista de facturación, selecciona una factura de la lista en este caso la factura perteneciente al número “0000003” o si prefiere buscar por el número de factura, se marca como factura anulada.

### **Escenario 10: Creación, actualización y eliminación de repuestos**

- El usuario ingresa un nuevo repuestos, ingresa al sistema y selecciona en el menú principal la opción de repuestos en donde desplegará la lista de todos los repuestos que tiene el sistema. El usuario ingresa los siguientes datos:

<i>CÓDIGO REPUESTO:</i>	<i>FIL_001</i>
<i>NOMBRE REPUESTO:</i>	<i>FILTRO DE AIRE TOYOTA</i>
<i>YARISPROVEEDOR:</i>	<i>FRANQUICIA S.A.</i>
<i>PRECIO COMPRA:</i>	<i>\$ 76 DÓLARES</i>
<i>PRECIO VENTA:</i>	<i>\$ 81 DÓLARES</i>
<i>CANTIDAD ACTUAL:</i>	<i>48 UNIDADES</i>
<i>STOCK MÍNIMO:</i>	<i>5 UNIDADES</i>

*Tabla 30: Datos Repuesto*

- El usuario actualizará los datos de un repuesto, consulta la lista de todos los repuestos, selecciona el repuesto o busca el repuesto por el nombre en este caso será “*FILTRO DE AIRE TOYOTA YARIS*” y Cambia el precio venta “*\$ 91 DÓLARES*” .
- El usuario eliminará una repuesto, consulta la lista de todos los repuestos, selecciona repuesto “*FILTRO DE AIRE TOYOTA YARIS*” y lo elimina.

### Escenario 11: Creación, actualización y eliminación de parametrización

- El usuario ingresa un nuevo servicio, ingresa al sistema y selecciona en el menú principal la opción de parametrización en donde desplegará la lista de todos los servicios que tiene el sistema. El usuario ingresa los siguientes datos:

CÓDIGO SERVICIO:	CAM_BUJIA
NOMBRE SERVICIO:	CAMBIO DE BUJÍAS
DESCRIPCIÓN DEL SERVICIO:	CAMBIO DE BUJÍAS PARA TOYOTA YARIS

Tabla 31: Datos Servicio Parametrización

- El usuario actualizará los datos de un servicio, consulta la lista de todos los servicios, selecciona el servicio o busca el por el nombre de servicio en este caso será “CAMBIO DE BUJÍAS” y Cambia la descripción del servicio en este caso será “CAMBIO DE BUJÍAS TOYOTA COROLA “.
- El usuario eliminará un servicio, consulta la lista de todos los servicios, selecciona servicio “CAMBIO DE BUJÍAS” y lo elimina.

### Escenario 12: Listado de campañas

- El usuario busca lista de clientes que tienen acceso a promociones según el número de visitas, ingresa al sistema y selecciona en el menú principal la opción de campañas en donde desplegará una ventana. El usuario ingresa los siguientes datos:

INGRESE NÚMERO DE VISITAS:	4
----------------------------	---

Tabla 32: Datos Número de Visitas en Campañas

- El usuario dará por aplicada la promoción al cliente, consulta la lista de todos clientes que tienen acceso a la campana parametrizada, selecciona al cliente en este caso “JUAN CARLOS BAEZ PONCE” y lo suprime de la lista dando a entender que este cliente ya tuvo acceso a esta promoción.

## b) Elaboración de las tarjetas CRC

Las tarjetas CRC (Cargo o Clase, Responsabilidad y Colaboración) Facilitan el diseño del sistema y es una de las partes importantes en la metodología XP.

(Tarjeta CRC Usuario)

<b>Nombre de la clase: UsuarioBean</b>	
<b>Responsabilidad</b>	<b>Colaboración</b>
Ingreso a una cuenta de usuario	UsuarioServiceImpl, newUser
Actualizar la cuenta del usuario	UsuarioServiceImpl, editUser
Eliminar cuenta de usuario	UsuarioServiceImpl,remove
Validar el ingreso al sistema	UsuarioServiceImpl,validarUsuario
Actualizar cuenta de usuario	UsuarioServiceImpl,guardarClaveNew

*Tabla 33: Tarjeta CRC Usuario*

(Tarjeta CRC Persona)

<b>Nombre de la clase: PersonaBean</b>	
<b>Responsabilidad</b>	<b>Colaboración</b>
Ingreso datos de la persona	PersonaServiceImpl, newPer
Actualizar los datos de la persona	PersonaServiceImpl, editPer
Eliminar los datos de la persona	PersonaServiceImpl,remove

*Tabla 34: Tarjeta CRC Persona*

(Tarjeta CRC Vehículo)

<b>Nombre de la clase: VehiculoBean</b>	
<b>Responsabilidad</b>	<b>Colaboración</b>
Ingreso datos del vehículo	VehiculoServiceImpl, newVeh
Actualizar los datos del vehículo	VehiculoServiceImpl, editVeh
Eliminar los datos del vehículo	VehiculoServiceImpl,remove

*Tabla 35: Tarjeta CRC Vehículo*

(Tarjeta CRC Cliente)

<b>Nombre de la clase: ClienteBean</b>	
<b>Responsabilidad</b>	<b>Colaboración</b>
Ingreso datos del cliente	ClienteServiceImpl, newVehCli

*Tabla 36: Tarjeta CRC Cliente*

(Tarjeta CRC Detalle Orden)

<b>Nombre de la clase: OrdenBean</b>	
<b>Responsabilidad</b>	<b>Colaboración</b>
Ingreso datos de detalle orden	DetalleOrdenServiceImpl, CatalogoServiceImpl,saveOrden,agregarServicio
Actualizar los datos de detalle orden	DetalleOrdenServiceImpl, CatalogoServiceImpl, saveOrden(),agregarServicio
Eliminar los datos de detalle orden	DetalleOrdenServiceImpl,removeDetOrden

*Tabla 37: Tarjeta CRC Detalle Orden*

(Tarjeta CRC Orden)

<b>Nombre de la clase: OrdenBean</b>	
<b>Responsabilidad</b>	<b>Colaboración</b>
Ingreso datos de orden	OrdenServiceImpl, CatalogoServiceImpl, VehiculoServiceImpl,saveOrden

Actualizar los datos de orden	OrdenServiceImpl, CatalogoServiceImpl, VehiculoServiceImpl, saveOrden
Eliminar los datos de orden	OrdenServiceImpl,remove
Validar kilometraje en la orden	OrdenServiceImpl, CatalogoServiceImpl, VehiculoServiceImpl,actualizacionKm

Tabla 38: Tarjeta CRC Orden

(Tarjeta CRC Detalle Proveedor)

<b>Nombre de la clase:</b> <b>ProveedorBean</b>	
<b>Responsabilidad</b>	<b>Colaboración</b>
Ingreso datos del detalle del proveedor	DetalleProveedorServiceImpl, guardarProveedor, newToolPopUp
Actualizar los datos del detalle del proveedor	DetalleProveedorServiceImpl, guardarProveedor,newToolPopUp
Eliminar los datos del detalle del proveedor	DetalleProveedorServiceImpl,removeDetProv

Tabla 39: Tarjeta CRC Detalle Proveedor

(Tarjeta CRC Proveedor)

<b>Nombre de la clase:</b> <b>ProveedorBean</b>	
<b>Responsabilidad</b>	<b>Colaboración</b>
Ingreso datos del proveedor	ProveedorServiceImpl, DetalleProveedorServiceImpl, guardarProveedor
Actualizar los datos del proveedor	ProveedorServiceImpl, DetalleProveedorServiceImpl, guardarProveedor

Eliminar los datos del proveedor	ProveedorServiceImpl,,remove
----------------------------------	------------------------------

Tabla 40: Tarjeta CRC Proveedor

(Tarjeta CRC Detalle Inventario)

<b>Nombre de la clase:</b> InventarioBean	
<b>Responsabilidad</b>	<b>Colaboración</b>
Ingreso datos del detalle inventario	DetalleInventarioServiceImpl, ProductoServiceImpl, guardarInvGeneral(),aceptarDetalleInv,
Actualizar los datos del detalle inventario	DetalleInventarioServiceImpl, ProductoServiceImpl, guardarInvGeneral,editDetInv
Eliminar los datos del detalle inventario	DetalleInventarioServiceImpl,removeDetInv
Lista los productos pendientes para realizar el inventario	listaProdInvPend
Carga datos de los repuestos	Inventariar

Tabla 41: Tarjeta CRC Detalle Inventario

(Tarjeta CRC Inventario)

<b>Nombre de la clase:</b> InventarioBean	
<b>Responsabilidad</b>	<b>Colaboración</b>
Ingreso datos de inventario	InventarioServiceImpl, DetalleInventarioServiceImpl, guardarInvGeneral,
Actualizar los datos de inventario	InventarioServiceImpl, DetalleInventarioServiceImpl, guardarInvGeneral
Eliminar los datos de inventario	InventarioServiceImpl,

	DetalleInventarioServiceImpl,remove
--	-------------------------------------

Tabla 42: Tarjeta CRC Inventario

(Tarjeta CRC Detalle Facturación)

<b>Nombre de la clase:</b> <b>DatosFacturacionBean</b>	
<b>Responsabilidad</b>	<b>Colaboración</b>
Ingreso datos de Detalle Factura	DetallefacturacioServiceImpl,insertarDetalleFac
Actualizar los datos de Detalle Factura	DetallefacturacioServiceImpl,actualizarDetalleFac
Eliminar los datos de Detalle Factura	DetallefacturacioServiceImpl,eliminarDetalleFac

Tabla 43: Tarjeta CRC Detalle Facturación

(Tarjeta CRC Facturación)

<b>Nombre de la clase:</b> <b>DatosFacturacionBean</b>	
<b>Responsabilidad</b>	<b>Colaboración</b>
Ingreso datos de Factura	FacturacionServiceImpl,DetallefacturacioServiceImpl,generarFactura
Actualizar los datos de Factura	FacturacionServiceImpl,DetallefacturacioServiceImpl,modificarFactura

Tabla 44: Tarjeta CRC Facturación

(Tarjeta CRC Producto)

<b>Nombre de la clase:</b> <b>ProductoBean</b>	
<b>Responsabilidad</b>	<b>Colaboración</b>
Ingreso datos del producto	ProductoServiceImpl,

	ProveedorServiceImpl,gurdarProducto
Actualizar los datos del producto	ProductoServiceImpl, ProveedorServiceImpl, gurdarProducto,editProd
Eliminar los datos del producto	ProductoServiceImpl,remove

*Tabla 45: Tarjeta CRC Producto*

(Tarjeta CRC Parametrización)

<b>Nombre de la clase:</b> <b>ParametrizacionBean</b>	
<b>Responsabilidad</b>	<b>Colaboración</b>
Ingreso datos de parametrización	CatalogoServiceImpl,guardarServicio
Actualizar los datos de parametrización	CatalogoServiceImpl, guardarServicio,editServicio
Eliminar los datos de parametrización	CatalogoServiceImpl, remove

*Tabla 46: Tarjeta CRC Parametrización*

(Tarjeta CRC Campañas)

<b>Nombre de la clase:</b> <b>CampaniasBean</b>	
<b>Responsabilidad</b>	<b>Colaboración</b>
Actualizar los datos de parametrización	FacturacionServiceImpl, PersonaServiceImpl,descartarPromocionPersona

*Tabla 47: Tarjeta CRC Campañas*

### 3.1.3 Desarrollo e Implementación

Las estrategias utilizadas en este proyecto se las describirá a continuación:

La integración es una estrategia importante en la fase de desarrollo determinando cambio y recodificación, se intenta con esto tener fracciones de código sencillo con el fin de integrar lo más rápido posible con el resto sin tener conflicto alguno.

Para lograr esto se tiene que contar con pruebas de unidad incluso antes de desarrollar el código, esto permitiendo tener una visión de lo que el código debe realizar.

Las estrategias como la propiedad colectiva y la programación en parejas no se utilizaron en el proyecto ya que es desarrollado por una sola persona, pero sus principios como (realimentación rápida, asumir la sencillez, cambio incremental, aceptar el cambio, trabajo con calidad) se las tomo ya que son muy necesarias.

Se tomó en cuenta las recomendaciones de XP en la fase de desarrollo como:

El cliente está siempre disponible, se debe escribir código de acuerdo a los estándares, desarrollar las pruebas de unidad primero, integración frecuente, dejar las optimizaciones para el final, no trabajar más de 40 horas a la semana.

Adoptando la cultura XP, con sus estrategias de desarrollo, principios, recomendaciones se desarrollará pensando en obtener un software de calidad.

### **3.1.3.1 Codificación**

Contando con las herramientas y con la ayuda en la fase de diseño del sistema se podrá desarrollar pensando en la calidad.

**Java:** Java es un lenguaje de programación con el que se puede realizar cualquier tipo de programa. En la actualidad es un lenguaje muy extendido y cada vez cobra más importancia tanto en el ámbito de Internet como en la informática en general. Está desarrollado por la compañía Sun Microsystems con gran dedicación y siempre enfocado a cubrir las necesidades tecnológicas más punteras.

**JSF:** JavaServer Faces es un estándar oficial JEE 5, se trata de un framework que define un modelo de componentes de interfaz de usuario y de eventos. Java Server Faces permite manejar el estado de los componentes de interfaz de usuario, manejar sus eventos, la validación y

conversión del lado del servidor y centralizar la navegabilidad de las páginas de la aplicación.

**MySQL 5.0:** Es un motor de base de datos open source, se lo puede administrar por medio de consola.

### 3.1.3.2 Creación de la red del sistema

Para entender más el funcionamiento de la instalación de los equipos se muestra la siguiente red para tener como referencia en la instalación de la misma.

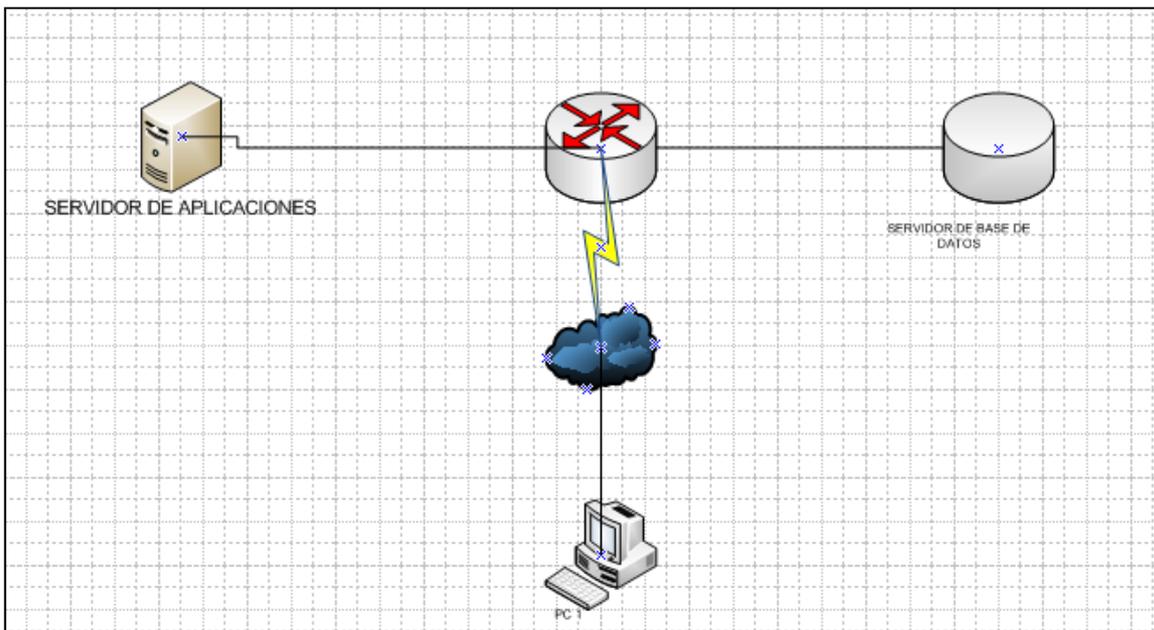


Gráfico 31: Red del Sistema

### 3.1.3.3 Pruebas XP

Se desarrollará pruebas unitarias para cada fracción de código, logrando que las pruebas se completen en un 100%.

Las pruebas de aceptación se las realizará con el cliente, ya que es la persona que da el visto bueno al sistema.

### 3.1.3.3.1 Pruebas

Unas de las grandes ventajas de la metodología XP es poder contar con una herramienta que permite las pruebas automatizadas.

Las pruebas en XP constituyen un gran valor, para mejorar la calidad del software. La utilización de una gran cantidad de tiempo en la realización de las mismas, provoca un miedo en los desarrolladores, que utilizan metodologías tradicionales, es por eso que en esta metodología, se decidió utilizar el siguiente proceso de pruebas de desarrollo según se muestra en el gráfico:

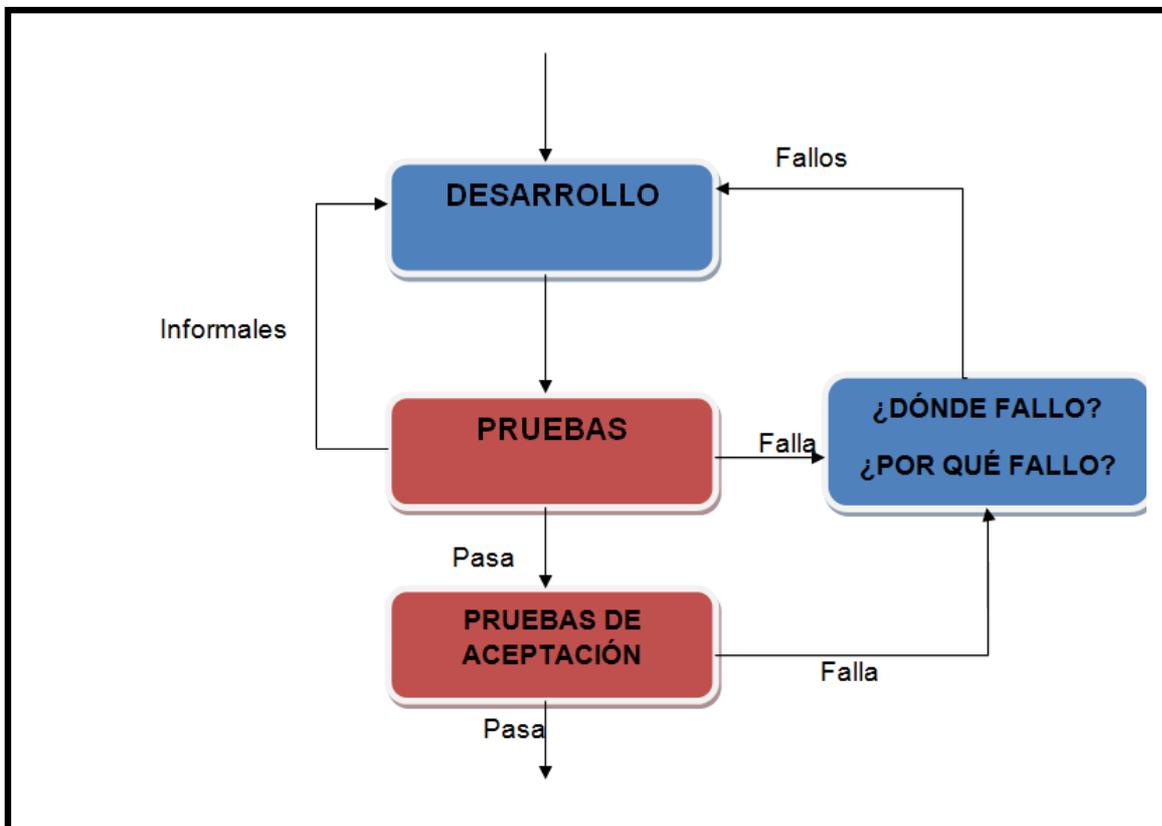


Gráfico 32: Proceso de Pruebas

## **Base de Datos**

Al terminar el diseño y la implementación de la base de datos, se insertaron datos directamente en ésta y se realizaron consultas por medio del asistente de consultas de MySQL Tool, para verificar que el modelo entidad relación estaba bien diseñado.

### **3.1.3.3.2 Módulos**

Al terminar cada módulo el equipo de desarrollo realizó las pruebas correspondientes para verificar que cada una cumpliera con sus tareas de forma correcta, fue así como por ejemplo en la parte de inserción de datos se verificó que los datos ingresados estén correctamente guardados en la base de datos.

Las pruebas de aceptación se hicieron cada vez que se sacó una nueva versión, consistía en entregar al cliente dicha versión para que estos ingresen datos, consultas y/o ediciones según fuera el caso, para verificar y dejar constancia que la aplicación respondía bien.

Al tratarse de una aplicación Web una de las pruebas más importantes fue la navegación, y los tiempos de respuesta, para estas pruebas se certificó con el navegador FileZilla V16.0. Y dio como resultado un funcionamiento óptimo en este navegador.

### **3.1.3.3.3 Pruebas de unidad y aceptación**

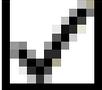
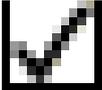
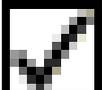
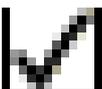
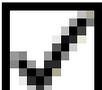
Las pruebas de unidad realizadas por el desarrollador permiten verificar el funcionamiento del código.

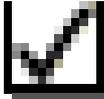
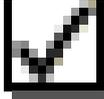
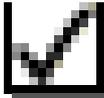
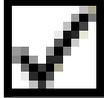
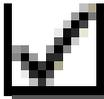
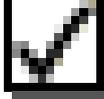
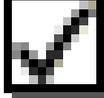
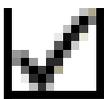
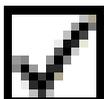
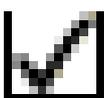
Las pruebas de aceptación son verificadas por el usuario para determinar el correcto funcionamiento del sistema

Mientras se tenga una batería de pruebas será más fácil verificar y realizar cambios en el código sin correr el riesgo de errar o que otras clases tengan conflictos con el mismo.

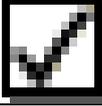
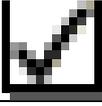
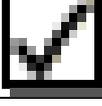
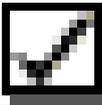
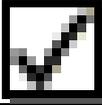
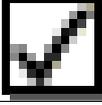
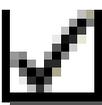
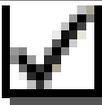
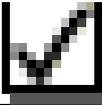
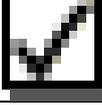
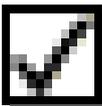
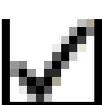
**a) Pruebas automatizadas de caja negra**

Se centran en los requisitos funcionales del software y es un complemento a las pruebas de caja blanca.

Nombre de la clase	Nombre prueba caja negra	Descripción	Resultado
Login	Buscar	Verifica que los datos ingresados del usuario sean correctos para poder ingresar al sistema	
usuarios	Buscar	Verifica que la búsqueda de toda la información de la tabla Usuario sea correcta	
	Eliminar	Verifica que la eliminación de toda la información de la tabla Usuario sea correcta	
	Modificar	Verifica que la fecha de inicio sea mayor a la fecha de expiración, para el permiso de ingreso al sistema	
	Insertar	Verifica que todos los datos sean ingresados correctamente y toma en cuenta que los datos obligatorios no pueden estar vacíos. Verifica que la clave sea igual a la confirmación de clave.	
personas	Buscar	Verifica que la búsqueda de toda la información de la tabla Persona sea correcta	
	Modificar	Verifica la modificación de la persona sea correcta, viendo que no se dejen campos vacíos y que sean obligatorios.	

	Eliminar	Verifica que todos los datos de la eliminación de la persona sea correcta	
	Insertar	Verifica que no se ingrese cédulas duplicadas y los campos obligatorios no estén vacíos	
clientes	Buscar	Verifica que la búsqueda de toda la información de la persona sea correcta. En el caso de la persona ya tener asignado su o sus vehículos verifica que la información sea correcta.	
	Modificar	Verifica que la los datos cambiados estén guardados correctamente y que no estén vacíos los campo obligatorios.	
	Eliminar	Verifica que al eliminar el vehículo corresponda al ítem seleccionado	
	Insertar	Verifica la asignación del vehículo a la persona buscada.	
Ordenes	Buscar	Verifica que la búsqueda de toda la información de la tabla Ordenes sea correcta	
	Insertar	Verifica datos obligatorios ingresados.	
	Modificar	Verifica que los datos cambiados estén correctos y que los campos obligatorios no estén vacíos	
	Eliminar	Verifica que el ítem corresponda a al número de orden seleccionado para ser eliminado	
	Buscar Vehículo y Persona	Verifica que la búsqueda de toda la información de la tabla vehículos y personas sea correcta	

Detalle Orden	Insertar	Verifica que se ingrese el servicios correctamente	<input checked="" type="checkbox"/>
	Eliminar	Verifica que el servicio seleccionado que se va a eliminar sea el correcto	<input checked="" type="checkbox"/>
Facturación	Buscar	Verifica que el número de factura exista en la lista de facturas	<input checked="" type="checkbox"/>
	Insertar	Verifica el ingreso de los datos de la factura sean correctos	<input checked="" type="checkbox"/>
	Anular	Verifica que cargue el visto de factura anulada	<input checked="" type="checkbox"/>
	Buscar Orden	Verifica que el número de orden exista y carga datos del cliente	<input checked="" type="checkbox"/>
	Agregar Servicio	Verifica que el servicio, con su repuesto correspondiente sea agregado a la lista a facturar	<input checked="" type="checkbox"/>
Inventario	Buscar	Verifica que al buscar la fecha de inicio de inventario exista en la tabla de inventarios	<input checked="" type="checkbox"/>
	Insertar	Verifica que al crear un nuevo inventario todos los inventarios anteriores consten como aplicados	<input checked="" type="checkbox"/>
	Modificar	Verifica que para poder modificar un inventario no conste como aplicado	<input checked="" type="checkbox"/>
	Eliminar	Verifica que para poder eliminar un inventario el mismo no conste como aplicado	<input checked="" type="checkbox"/>
Detalle Inventario	Inventariar producto	Muestra lista de productos a inventariar pero verifica solo los faltantes	<input checked="" type="checkbox"/>
	Modificar	Verifica que para poder modificar no debe constar como aplicado	<input checked="" type="checkbox"/>
	Eliminar	Verifica que para poder eliminarlo, el mismo no conste como aplicado	<input checked="" type="checkbox"/>

	Insertar	Verifica que los datos estén ingresados correctamente tales como nueva cantidad mayor a cantidad existente y mayor a stock mínimo	
Proveedor	Buscar	Verifica que el nombre de la empresa conste en la tabla de Proveedores	
	Insertar	Verifica que los campos obligatorios estén ingresados correctamente	
	Modificar	Verifica que los campos obligatorios no estén vacíos	
	Eliminar	Verifica que el ítem seleccionado corresponda al proveedor que se desea eliminar	
Detalle Proveedor	Insertar	Verifica que todos los campos obligatorios fueron ingresados	
	Modificar	Verifica que los campos modificados no se estén vacíos	
	Eliminar	Verifica que el ítem seleccionado a eliminar sea el correcto	
Productos	Buscar	Verifica que el nombre del repuesto conste en la tabla de Productos	
	Insertar	Verifica que todos los campos obligatorios fueron ingresados	
	Modificar	Verifica que los campos modificados no se estén vacíos	
	Eliminar	Verifica que el ítem seleccionado corresponda al producto que se desea eliminar	
Campañas	Buscar	Verifica que se liste todos los usuarios de acuerdo a la parametrización ingresada	

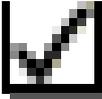
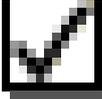
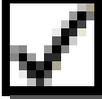
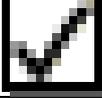
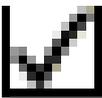
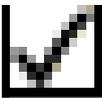
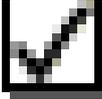
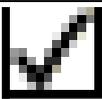
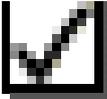
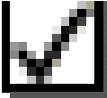
	Descartar	Verifica que el cliente se descarte de la lista de clientes que pueden ingresar a determinada promoción	
Parametrización /servicios	Buscar	Verifica que el nombre del servicio conste en la tabla de Catálogos	
	Insertar	Verifica que todos los campos obligatorios fueron ingresados	
	Modificar	Verifica que los campos modificados no se estén vacíos	
	Eliminar	Verifica que el ítem seleccionado corresponda al servicio que se desea eliminar	

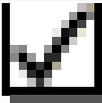
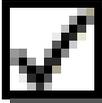
Tabla 48: Pruebas automatizadas de caja negra

#### b) Pruebas automatizadas de Caja Blanca

Aseguran que la operación interna del programa se ajusta a las especificaciones y que todos los componentes internos se han probado adecuadamente.

Login	Ingreso por Teclado	Verifica que los atributos de la tabla Usuario tengan concordancia con el tipo de dato ingresados	
Usuario	Valida	Valida que la cédula no tenga más de diez dígitos. Valida que la clave sea mayor a cinco caracteres	
Personas	Valida	Valida el formato de correo electrónico. Valida el número de dígitos para teléfonos. Valida el número de cédula con diez caracteres	
Clientes	Valida	Valida el ingreso de la placa con el formato establecido	

Ordenes	Valida	Valida el ingreso a buscar de solo números. Valida el kilometraje para el mensaje de alertas. Valida el formato de fechas	
Facturación	Valida	Valida que el número de orden a buscar sea numérico. Valida la anulación de una factura deshabilitando los campos de agregar servicios. Verifica que los atributos de la tabla Factura tengan concordancia con el tipo de dato. Validó formato de fecha. Verifica que al tener la lista en la factura sume el precio correctamente y agrega a la suma los impuestos pertinentes	
Inventario	Valida	Valida que la búsqueda por fecha solo sea ingresada una fecha. Valida que el ruc del responsable del inventario sea de trece caracteres. Verifica que los atributos de la tabla Inventario tengan concordancia con el tipo de dato. Valida que la nueva cantidad sea numérica.	
Proveedores	Valida	Valida el ruc del propietario tenga entre 10 y 13 caracteres. Valida el número de caracteres correspondiente a los teléfonos. Verifica que los atributos de la tabla Proveedor tengan concordancia con el tipo de dato. Valida que el precio del repuesto que ofrece el proveedor sea de tipo numérico.	
Productos	Valida	Valida que en los campos de tipo numérico se ingresen solo números. Verifica que los atributos de la tabla Producto tengan	

		concordancia con el tipo de dato.	
Campañas	Valida	Valida que según la parametrización se liste los clientes.	
Parametrización / Servicios	Valida	Verifica que los atributos de la tabla Catálogos tengan concordancia con el tipo de dato	

*Tabla 49: Pruebas automatizadas de Caja Blanca*



## CAPÍTULO 4: Conclusiones y Recomendaciones

### 4.1 Contenido

#### 4.1.1 Conclusiones

La del desarrollo del proyecto resulto satisfactoria. La elección y aplicación de dicha metodología dadas las características del problema arrojó resultados positivos en términos de satisfacción del cliente, cumplimiento de los plazos y una magnífica colaboración con los dueños del taller. Se encontró que la metodología se ajustó muy bien, no solo al tipo de cliente y a las características del problema, también resulto adecuada para el entorno del trabajo y las características de los desarrolladores.

- Se concluyó que la metodología **extreme programming** asegura la calidad del software, en todo su ciclo de vida desde la planificación hasta llegar a las pruebas, ya que en cada fase del ciclo de vida se presentan productos que permiten evaluar el desarrollo del software.
- Después de tomar en cuenta algunas metodologías ágiles se decidió que la metodología XP es una metodología madura para implantar proyectos rápidamente y dentro del proyecto se evaluó la documentación y capacitación de la metodología.
- Al utilizar el motor de base de datos MYSQL se obtuvo un rendimiento adecuado para el tipo de negocio.
- Al dividir en capas el desarrollo, permite dividir la capa lógica de la capa del negocio y la capa de presentación, según el número de capas con la que se desarrolla, esto facilita la mantenibilidad y escalabilidad de las aplicaciones
- Entregas frecuentes como elemento motivados del proyecto.

- La comunicación entre el cliente y el desarrollador es fundamental para obtener resultados óptimos.
- Las metodologías ágiles da mayor ventaja para proyectos pequeños.
- Al utilizar XP se realizó un sistema simple sencillo y fácil de usar, concluyendo que la recodificación es el fuerte más grande de la metodología, permitiendo optimizar aún más el código.
- Se comprobó que XP está preparada para cuando se presentan cambios rápidos y se los puede aplicar de una manera flexible, debido a sus ciclos de vida cortos y su codificación sencilla.
- No se puede olvidar que XP necesita de sus estándares para su aplicación.
- Los estándares en XP aseguran la calidad de la aplicación, facilitan la resolución de todas las fases disminuyendo los tiempos de desarrollo y aumentando la calidad del producto.

#### **4.1.2 Recomendación**

- Se debe fijar una serie de reglas generales en la comunicación con el cliente ya que por el grado de informalidad que la metodología presenta, puede surgir diferencias que pongan en peligro la culminación exitosa del proyecto.
- Debe hacerse una capacitación al cliente sobre XP antes de iniciar el proyecto debido que este hace parte del equipo de desarrollo.
- Tener un buen conocimiento de las herramientas para la implantación antes de iniciar dicha etapa.

- Es recomendable para aplicaciones Web desarrolladas desde tres capas en adelante
- Plantear como unidad de tiempo horas en lugar de días para la asignación de tareas.
- Considerar el internet y herramientas basadas en él como mecanismo de comunicación, válido dentro de XP, y discutir las necesidades de un único sitio geográfico de trabajo.
- Sería factible la utilización de MySQL ya que es un motor de base de datos robusto, con un buen soporte técnico y gratuito.
- Convendría utilizar XP en proyectos cortos y medianos, para disminuir el tiempo de desarrollo.
- En proyectos grandes debido al formalismo no es recomendable la aplicación de XP, no obstante también se lo podría utilizar en proyectos grandes siempre y cuando estos sean separados por módulos pequeños.
- Las pruebas unitarias se realizarán antes de la codificación ya que estas ayudarán a entender mejor el funcionamiento del sistema.
- El código debe ser lo más sencillo, con el fin de que esta pueda someterse a cambios, en el caso de ser necesario y estos no sean tan complejos realizarlos.

## 4.2 BIBLIOGRAFÍA

- XP, Recuperado 14 de septiembre de 2012, de [http://es.wikipedia.org/wiki/Programaci%C3%B3n\\_extrema](http://es.wikipedia.org/wiki/Programaci%C3%B3n_extrema)
- XP, Recuperado 14 de septiembre de 2012, de <http://procesosdesoftware.wikispaces.com/METODOLOGIA+XP>
- XP, Recuperado 14 de septiembre de 2012, de <http://es.scribd.com/doc/57257203/Metodologia-XP>
- XP, Recuperado 14 de septiembre de 2012, de <http://www.willydev.net/descargas/prev/ExplicaXP.pdf>
- XP, Recuperado 14 de septiembre de 2012, de [http://users.dsic.upv.es/asignaturas/facultad/Isi/ejemploxp/Gestion\\_Proyecto.html](http://users.dsic.upv.es/asignaturas/facultad/Isi/ejemploxp/Gestion_Proyecto.html)
- XP, Recuperado 14 de septiembre de 2012, de <http://xpgestionestadisticas.wordpress.com/2008/06/23/hello-world/>
- JEE, Recuperado 27 de noviembre, 2011, de <http://www.oracle.com>
- JBOSS, Recuperado 6 de diciembre de 2011, <http://www.osmosislatina.com/jboss/basico.htm>
- Eclipse, Recuperado 19 de marzo, 2012, de <http://www.eclipse.org/>
- MySQL, Recuperado 20 de marzo, 2012, de <http://www.mysql.com>
- JSF, Recuperado 20 de marzo, 2012, de <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=IntroduccionJSFJava>
- Richfaces, Recuperado 20 de marzo, 2012, de <http://es.scribd.com/doc/26352343/JBoss-RichFaces-Capitulo-1-%C2%BFQue-es-RichFaces>
- Power Designer, Recuperado 27 de noviembre, 2011, de [http://www.sybven.com/portales/images/stories/documentos/PowerDesigner15\\_Brochure\\_Espanol.pdf](http://www.sybven.com/portales/images/stories/documentos/PowerDesigner15_Brochure_Espanol.pdf)
- Hibernate, Recuperado 21 de septiembre de 2011, de <http://www.hibernate.org>