

ESCUELA POLITÉCNICA DEL EJÉRCITO

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

CARRERA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

ANÁLISIS, DESARROLLO E IMPLEMENTACIÓN DEL
SISTEMA SEGURO TOTAL PARA LA EMPRESA
SOLMOVSA.

Previa a la obtención del Título de:

INGENIERO EN SISTEMAS E INFORMÁTICA

POR:

FERNANDO MANUEL MORALES MORA

LUIS HORACIO SALAZAR ESTÉVEZ

SANGOLQUÍ, Junio del 2012

CERTIFICACIÓN

Certifico que el presente trabajo fue realizado en su totalidad por los Sr(s).
FERNANDO MANUEL MORALES MORA y LUIS HORACIO SALAZAR
ESTÉVEZ como requerimiento parcial a la obtención del título de INGENIEROS
EN SISTEMAS E INFORMÁTICA.

Sangolquí, Junio del 2012

ING. DIEGO MARCILLO

DEDICATORIA

A mis padres Carlos y Nidia quienes con su esfuerzo, apoyo y trabajo me permitieron estudiar esta carrera e ingresar en una universidad de alto prestigio como es la Escuela Politécnica del Ejército. A mis hermanos Dudley, Mónica, Carlos y Nidya que también fueron de mucho apoyo durante toda mi vida estudiantil. A mi esposa Lorena y mi hijo Derek para quienes mi esfuerzo y sacrificio está entregado con todo el amor que les tengo.

Fernando Morales

DEDICATORIA

Mi tesis la dedico con mucho amor y cariño a Dios quien esta junto a mí día a día enseñándome en cada experiencia el valor de la vida. A mis padres Ruperto y Teresa quienes creyeron en mi apoyándome a cumplir mis sueños y metas planteadas. A mis hermanos Gilma, Ximena y David que con su experiencia profesional fueron un apoyo e incentivo para continuar.

A mi esposa Dianita y mi hijo Matías quienes han estado junto a mí apoyando y en parte compartiendo el sacrificio que he realizado para culminar y alcanzar este sueño tan anhelado.

Luis Salazar

AGRADECIMIENTOS

A mis padres Carlos y Nidia, a mis hermanos Dudley, Mónica, Carlos y Nidya pues sin su apoyo no hubiese alcanzado este objetivo en mi vida. A mi abuelita Filo que con sus bendiciones me supo impulsar a salir adelante.

A mi esposa Lorena quien ha visto sacrificado tiempo de compartir juntos por este objetivo y quien su amor y motivación me ha ayudado para alcanzar algo que lo veía inalcanzable.

Un gran agradecimiento a mi director Ing. Diego Marcillo y mi codirector Ing. Mauricio Campaña quienes con su conocimiento e impulso contribuyeron a que se pueda concluir este proyecto de tesis.

A la Universidad quien con sus puertas abiertas ha permitido que miles de personas puedan crecer profesionalmente. A todos mis profesores universitarios pues la transferencia de su conocimiento han hecho que sea una mejor persona tanto moral como profesionalmente.

A mi amigo y compañero de tesis Luis quien con su insistencia, conocimiento y trabajo permitió que este proyecto de tesis vea su finalización.

Fernando Morales

AGRADECIMIENTOS

Mis agradecimientos están dirigidos a toda mi familia quienes han estado directa e indirectamente apoyando, incentivándome a culminar mi tesis.

A mis padres Ruperto y Teresa por su amor y apoyo incondicional, quienes siempre serán mis héroes y ejemplo a seguir. A mi esposa Dianita quien siempre ha compartido mis momentos más difíciles y alegrías dándome la fortaleza de seguir adelante.

Mi agradecimiento muy especial al Ing. Diego Marcillo y al Ing. Mauricio Campaña por los conocimientos compartidos para mi desarrollo profesional.

A mi amigo y compañero de tesis Fernando quien aportó con su experiencia y conocimiento a culminar este proyecto.

Luis Salazar

ÍNDICE DE CONTENIDOS

CERTIFICACIÓN.....	ii
DEDICATORIA.....	iii
AGRADECIMIENTOS.....	v
ÍNDICE DE CONTENIDOS.....	vii
LISTADO DE TABLAS.....	x
LISTADO DE FIGURAS.....	xi
LISTADO DE ANEXOS.....	xii
RESUMEN.....	1
CAPÍTULO 1.....	2
GENERALIDADES.....	2
1.1 Introducción.....	2
1.2 Objetivos.....	3
1.2.1 Objetivo general.....	3
1.2.2 Objetivos específicos.....	3
1.3 Alcance.....	4
CAPÍTULO 2.....	5
MARCO TEÓRICO.....	5
2.1 Seguros Vehiculares.....	5
2.1.1 Conceptos básicos de seguros vehiculares.....	5
2.1.1.1 Seguro Vehicular.....	5
2.1.1.2 Póliza.....	6
2.1.1.3 Asegurador.....	7
2.1.1.4 Asegurado.....	7
2.1.1.5 Riesgo.....	7
2.1.1.6 Siniestro.....	7
2.1.1.7 Prima.....	7
2.1.1.8 Suma Asegurada.....	8
2.1.1.9 Deducible.....	8
2.1.1.10 Cobertura.....	8
2.1.2 Impuestos de un seguro vehicular.....	10
2.2 Programación Extrema.....	10
Introducción.....	10
2.2.1 Definición.....	11
2.2.1.1 Objetivos de la programación extrema.....	12
2.2.2 Variables.....	12
2.2.3 Fases de la metodología de programación extrema.....	13
2.2.3.1 Planificación.....	14

2.2.3.2	Diseño	19
2.2.3.3	Desarrollo	21
2.2.3.4	Pruebas	23
2.2.4	UML.....	24
2.2.4.1	Diagrama de casos de uso	25
2.2.4.2	Diagrama de clases	26
2.2.4.3	Diagrama de Objetos.....	27
2.2.4.4	Diagrama de comportamiento.....	28
2.2.4.5	Diagrama de estados.....	28
2.2.4.6	Diagrama de Actividades	29
2.2.4.7	Diagramas de interacción	30
2.2.4.8	Diagramas de implementación	30
2.2.5	Plataforma .NET	33
2.2.5.1	Introducción.....	33
2.2.5.2	Características principales de la plataforma .NET	34
2.2.5.3	Elementos de la plataforma .NET	34
2.2.5.4	Arquitectura .NET	35
2.2.5.5	Características principales de la arquitectura .NET	36
2.2.5.6	.Net Framework	37
2.2.5.7	.Net Compact Framework	42
2.2.6	Capa de Interfaz	44
2.2.6.1	Visual Studio .NET	44
2.2.6.2	Lenguaje de desarrollo	45
2.2.7	Capa Lógica	45
2.2.7.1	Web Services	46
2.2.7.2	Protocolo SOAP	47
2.2.7.3	Multifuncionalidad entre Sistemas Operativos	48
2.2.8	Capa de Datos.....	48
2.2.8.1	Microsoft SQL Server	48
2.2.8.2	Microsoft SQL CE (Compact Edition).....	49
2.2.9	Dispositivos Móviles	49
2.2.9.1	Pocket PC	50
2.2.9.2	Windows Mobile 6.5.....	52
2.2.9.3	Windows CE	52
2.2.10	Nunit Testeador de pruebas unitarias	53
2.2.10.1	Prueba Unitaria.....	53
CAPÍTULO 3.....		54
IMPLEMENTACIÓN DEL SISTEMA.....		54
3.1	Planificación	54

3.1.1	Historias de Usuario	56
3.1.2	Plan de entregas	62
3.1.3	Velocidad del Proyecto	63
3.1.4	Iteraciones.....	63
3.1.5	Rotaciones	64
3.1.6	Reuniones	64
3.2	Diseño	64
3.2.1	Metáfora del Sistema.....	64
3.2.2	Tarjetas CRC.....	66
3.2.3	Soluciones puntuales.....	70
3.2.4	Funcionalidad Mínima.....	71
3.2.5	Reciclaje.....	74
3.2.6	Diagramas de secuencias.....	74
3.2.6.1	Diagrama de componentes.....	79
3.2.6.2	Diagrama de estados.....	79
3.3	Desarrollo	80
3.3.1	Disponibilidad del cliente	81
3.3.2	Unidad de pruebas	81
3.3.3	Programación por parejas.....	82
3.3.4	Integración.....	82
CAPÍTULO 4.....		84
IMPLANTACIÓN DEL SISTEMA.....		84
4.1	Pruebas de Aceptación.....	84
4.2	Implantación	88
4.3	Aplicación de Servicios Web.....	88
4.4	Aplicación de Sitio Web	90
4.5	Aplicación Móvil.....	91
CAPÍTULO 5.....		93
CONCLUSIONES Y RECOMENDACIONES.....		93
5.1	Conclusiones	93
5.2	Recomendaciones.....	94
BIBLIOGRAFÍA.....		95
ANEXOS.....		¡Error! Marcador no definido.

LISTADO DE TABLAS

Tabla 3.1	Tabla de planificación del proyecto	54
Tabla 3.1	Historia de usuario, Acceso al sistema.....	56
Tabla 3.2	Historia de usuario, Registro de datos del bien.....	57
Tabla 3.3	Historia de usuario, Administración del aplicativo web	58
Tabla 3.4	Historia de usuario, Registro de datos del cliente.....	58
Tabla 3.5	Historia de usuario, Registro de forma de pago	59
Tabla 3.6	Historia de usuario, Registro de datos específicos del bien	59
Tabla 3.7	Historia de usuario, Verificación, análisis y aprobación de solicitudes	60
Tabla 3.8	Historia de usuario, Carga de catálogos.....	60
Tabla 3.9	Historia de usuario, Envío de información	61
Tabla 3.10	Historia de usuario, Listado impresión.....	61
Tabla 3.11	Historia de usuario, Reportes gerenciales.....	62
Tabla 3.12	Tabla de identificación de entregables del proyecto.....	63
Tabla 3.12	Tarjeta CRC, CRC Usuario	67
Tabla 3.13	Tarjeta CRC, CRC Perfil usuario	67
Tabla 3.14	Tarjeta CRC, CRC Bien.....	68
Tabla 3.15	Tarjeta CRC, CRC Catálogo	68
Tabla 3.16	Tarjeta CRC, CRC Cliente.....	69
Tabla 3.17	Tarjeta CRC, CRC Solicitud	69
Tabla 3.18	Tarjeta CRC, CRC Análisis	70
Tabla 3.19	Tarjeta CRC, CRC Reporte	70
Tabla 4.1	Prueba de Aceptación, Creación de usuario	84
Tabla 4.2	Prueba de Aceptación, Registro de pólizas de aseguramiento	86
Tabla 4.3	Prueba de Aceptación, Verificación, análisis y aprobación	87

LISTADO DE FIGURAS

Figura 2.1	Tareas de la programación extrema.....	11
Figura 2.2	Características básicas de la programación extrema.....	11
Figura 2.3	Variables de la programación extrema.....	12
Figura 2.4	Fases de la programación extrema	13
Figura 2.5	Tarjeta de una historia de usuario	15
Figura 2.6	Esquema de un plan de entregas	16
Figura 2.7	Plan de iteraciones	18
Figura 2.8	Tarjeta CRC	20
Figura 2.9	Modelos UML	25
Figura 2.10	Caso de uso	26
Figura 2.11	Diagrama de clases.....	27
Figura 2.12	Diagrama de objetos	27
Figura 2.13	Diagrama de estados de solicitudes.....	28
Figura 2.14	Diagrama de actividades.....	29
Figura 2.15	Diagrama de componentes	31
Figura 2.16	Diagrama de plataforma o despliegue	32
Figura 2.17	Plataforma Microsoft .NET	33
Figura 2.18	Elementos de la plataforma .NET.....	35
Figura 2.19	Capas de la plataforma .NET	36
Figura 2.20	Arquitectura del .Net Framework	38
Figura 2.21	Biblioteca de clases de .Net Framework	40
Figura 2.22	Arquitectura de compact Framework .Net	43
Figura 2.23	Arquitectura general del sistema	46
Figura 2.24	Esquema de Web Services	47
Figura 2.25	Pocket Pc / Smartphone	50
Figura 3.1	Cronograma del proyecto.....	55
Figura 3.2	Diagrama de infraestructura de la solución.....	65
Figura 3.3	Diagrama de arquitectura de la solución	66
Figura 3.4	Flujo del negocio	72
Figura 3.5	Flujo del sistema	73
Figura 3.6	Diagrama de secuencia de registro de clientes.....	75
Figura 3.7	Diagrama de secuencia de registro de solicitudes	76
Figura 3.8	Diagrama de secuencia de actualización de clientes y solicitudes	77
Figura 3.9	Diagrama de secuencia de generación de reportes	78
Figura 3.10	Diagrama de componentes	79
Figura 3.11	Diagrama de estados de una solicitud	80
Figura 3.12	Prueba unitaria	81
Figura 4.1	Publicación paso 1	89
Figura 4.2	Publicación paso 2	89
Figura 4.3	Creación de un directorio virtual.....	90
Figura 4.4	Propiedades del directorio virtual	91
Figura 4.5	Generación de instalador de aplicativo Pocket	92

LISTADO DE ANEXOS

ANEXO A: Manual de instalación del sistema Seguro Total ¡Error! Marcador no definido.

ANEXO B: Manual de usuario Seguro Total Móvil ¡Error! Marcador no definido.

ANEXO C: Manual de usuario Seguro Total Web ¡Error! Marcador no definido.

ANEXO D: Modelo lógico de la base de datos ¡Error! Marcador no definido.

ANEXO E: Modelo físico de la base de datos ¡Error! Marcador no definido.

RESUMEN

La empresa “SOLMOVSA” ubicada en Quito, Provincia de Pichincha, enfocada a la industria del software en el Ecuador brinda soluciones tecnológicas a los requerimientos que tienen las empresas en el giro de sus negocios, por esta razón se ha visto la necesidad de disponer de un sistema de cotización y contratación de seguros vehiculares, el mismo que servirá como base para un mejor funcionamiento de las actividades que realiza una empresa aseguradora, y a la vez permitir establecer objetivos y estrategias para mejorar las operaciones de la misma.

En la presente tesis el lector encontrará una breve descripción de la teoría de seguros vehiculares y de una de las principales metodologías en el desarrollo ágil, la metodología que se describe es programación extrema, la principal característica de esta metodología es el enfoque que da al usuario, así como la manera de capturar los requerimientos, la realización del diseño del software y el manejo de las pruebas unitarias. Así como también una descripción del Framework de .Net, sus diferentes componentes, una introducción al uso de base de datos empotradas, la descripción de Servicios Web y una introducción a uno de los lenguajes de programación orientado a objetos creado por Microsoft, Visual C#.

El lector podrá ver como basado en el proceso metodológico que establece la programación extrema, se ha creado una solución que funciona en dispositivos móviles usando Visual C#, y una solución web las cuales se integran utilizando servicios para su comunicación.

CAPÍTULO 1

GENERALIDADES

1.1 Introducción

En la actualidad las soluciones informáticas han causado una gran ola de nuevos requerimientos empresariales; es decir, nuevas facilidades para satisfacer sus necesidades. El uso de nuevos dispositivos tecnológicos, como es el caso de dispositivos móviles y el uso de aplicativos Web, traen consigo la ejecución de nuevos planes en cuanto a la organización empresarial o específicamente en su forma de trabajo actual.

Utilizar las facilidades que presentan los dispositivos electrónicos y adaptarlos a satisfacer los requerimientos es el objetivo de todas las empresas de desarrollo y en este ámbito la empresa SOLMOVSA está dispuesta a emplear recursos en la investigación de las nuevas herramientas tecnológicas para brindar así a sus clientes las soluciones de vanguardia óptimas que ellos requieren.

SOLMOVSA se ha encargado de desarrollar soluciones de software de última tecnología ofreciendo productos a medida de los requerimientos de sus clientes. Entendiendo que el nicho de mercado de SOLMOVSA es el desarrollo de aplicaciones móviles es necesario entonces ofrecer nuevos aplicativos que garanticen su competitividad.

Entrando en el campo de las necesidades empresariales actuales es importante reconocer que las empresas requieren tener organizada su información y sobre todo poder acceder a ésta cuando sea necesario como es

en el caso de los seguros. Es por tanto que las aplicaciones móviles están tomando una gran acogida en la actualidad y por eso se hace imprescindible el conocimiento de herramientas de desarrollo en éste ámbito.

Se experimenta así la necesidad de emprender un estudio de los dispositivos móviles actuales, en especial sobre Pocket Pc y Smartphone, su uso y sobre todo como adaptarlos a satisfacer los requerimientos mediante el desarrollo de aplicaciones y su integración a los sistemas distribuidos que las empresas posean.

1.2 Objetivos

1.2.1 Objetivo general

Analizar, desarrollar e implementar el sistema “SEGURO TOTAL” para la empresa SOLMOVSA.

1.2.2 Objetivos específicos

1. Realizar el análisis, diseño del sistema “Seguro Total” para la empresa SOLMOVSA que integrará el aplicativo cotizador de seguros vehicular sobre Pocket PC y el aplicativo Web para manejo de datos gestionados por la aplicación móvil.
2. Implementar y probar el funcionamiento, integración del aplicativo sobre Pocket PC y el manejo de datos sobre la aplicación web
3. Implantar el sistema “Seguro Total” para la empresa SOLMOVSA en producción.

1.3 Alcance

Se realizará el desarrollo de una aplicación Web y una aplicación para Pocket PC. Para esto se utilizará una base de datos central en SQL Server 2008 y una base de datos distribuida en los dispositivos móviles mediante SQL Server CE. La integración entre las dos aplicaciones se la realizará a través de servicios Web.

El aplicativo para la Pocket PC se encargará de realizar las cotizaciones de seguros vehiculares, ingresando información al dispositivo así como imágenes del vehículo cotizado.

La aplicación Web se encargará de gestionar toda la información ingresada por las aplicaciones móviles. Se manejarán reportes así como indicadores que permitan visualizar las ventas de pólizas de seguro a través de los dispositivos móviles, y además realizará la administración de clientes.

La información utilizada para el cotizador vehicular será proporcionada por la propia empresa auspiciante, para demostrar la autenticidad de los resultados.

Se realizará una transferencia del conocimiento adquirido en el desarrollo del presente estudio a miembros de la empresa auspiciante.

Se implantará el sistema en la empresa auspiciante.

CAPÍTULO 2

MARCO TEÓRICO

2.1 Seguros Vehiculares

Introducción

En la actualidad asegurar un bien, la salud o la vida ante todo riesgo ya no es un lujo, es una necesidad. En el país muchas son las empresas que dan este servicio privado reservando derechos de conformidad con las condiciones generales y particulares que se haya celebrado en el contrato.

“El objeto del Seguro es reducir su exposición al riesgo de experimentar grandes pérdidas y garantizar la protección contra siniestros importantes y problemáticos, a cambio de pagos fijos“ ¹

2.1.1 Conceptos básicos de seguros vehiculares

2.1.1.1 Seguro Vehicular

Constituye una solución a la necesidad de proteger un vehículo ante la ocurrencia de hechos imprevistos, cuyas consecuencias superen nuestra capacidad individual para repararlas, garantizando el resarcimiento de un capital para reparar o cubrir la pérdida o daño que aparezca en cualquier momento, recibiendo como contraprestación un precio por adelantado por el servicio de protección que ofrece.

Las aseguradoras no únicamente pagan con dinero el daño que el asegurado o alguna de sus pertenencias hayan sufrido, sino que, según el tipo de aseguradora y de póliza le brindan comodidad al asegurado, como puede ser

¹ Objeto del Seguro, http://www.elprisma.com/apuntes/administracion_de_empresas/seguroconcepto/

arreglándole su vehículo, reponiendo piezas nuevas, o prestarle un vehículo hasta que el suyo haya quedado totalmente arreglado.

2.1.1.2 Póliza

Es el contrato de seguros, mediante el cual una de las partes, el asegurador, se compromete a que si la persona que compró el seguro sufre algún daño en su persona (enfermedades o accidentes e incluso la muerte) , al usar el bien (automóvil) por cualquier motivo (robo, siniestro), dicha persona (o quien ella haya designado como beneficiario) recibirá la cantidad de dinero acordada en la póliza, garantizándole, a cambio de recibir una prima, el pago de una suma predeterminada o el valor de la pérdida al producirse el siniestro amparado por el riesgo. Siendo un contrato tiene el carácter de bilateral, comercial, oneroso, solemne y real entre otros.

a.- **Condiciones Generales**, son disposiciones impresas sobre deberes y derechos de las partes, formas de atención de siniestros, riesgos cubiertos y excluidos, materias de orden jurídico general.

b.- **Condiciones Particulares**, generalmente son las particularidades del propio asegurado, como son el objeto específico del seguro, ubicación del riesgo, suma asegurada, vigencia del seguro y otras referidas a la materia concreta del riesgo cubierto, inclusive limitaciones de cobertura sobre lo señalado ampliamente en las condiciones generales. Hay un principio contractual, que declara que las condiciones particulares prevalecen sobre las condiciones generales en caso de discrepancia entre ambas.

2.1.1.3 Asegurador

Es la persona jurídica llamada Compañía de Seguros que asume de forma profesional el riesgo mediante la percepción de un precio llamado prima total celebrado en una póliza de seguros.

2.1.1.4 Asegurado

Es la persona natural o jurídica que se encuentra expuesta al riesgo, en su persona, sus bienes o en su patrimonio y recibe el servicio de protección contra el riesgo cubierto por el asegurador.

2.1.1.5 Riesgo

Es la posibilidad de pérdida o daño, es decir la constante amenaza que pesa sobre el vehículo.

2.1.1.6 Siniestro

Cuando un vehículo sufre un daño material en su estructura el seguro materializa su acción de protección e indemnización.

2.1.1.7 Prima

Es el precio del seguro que paga el asegurado en el momento de la emisión de la póliza. La prima es por lo general para una vigencia anual del seguro y ya incluye todos los impuestos, siendo el pago de la misma de carácter obligatorio según las condiciones establecidas en la póliza de seguros.

- **Prima Neta:** Es el valor de la prima sin incluir los impuestos y cargos.
- **Prima Total:** Es el valor de la prima neta más los impuestos y cargos que se agregan por ley.

2.1.1.8 Suma Asegurada

Es la cantidad fijada en las condiciones particulares de la póliza y representa la valorización del riesgo cubierto o suma hasta cuyo límite está obligado el asegurador a indemnizar en caso de pérdida total del bien u objeto asegurado.

2.1.1.9 Deducible

Es la cantidad o porcentaje establecido en una póliza cuyo importe ha de superarse para que se pague una reclamación.

2.1.1.10 Cobertura

Compromiso aceptado por un asegurador en virtud del cual se hace cargo, hasta el límite estipulado, de las consecuencias económicas derivadas de un siniestro. Cubriendo las siguientes coberturas:

- a) Responsabilidad Civil: Se ampara la pérdida del patrimonio del asegurado ante la reclamación legal de un tercero, hasta el límite fijado en las condiciones particulares de la póliza por:
 - Lesiones corporales a terceras personas
 - Daños a la propiedad de terceras personas.
 - Gastos y costos judiciales del proceso.
- b) Accidentes Personales: Ampara los gastos por el daño corporal que sufre tanto el conductor como sus pasajeros, en lo que respecta a muerte, invalidez, gastos de curación y/o gastos de sepelio, siempre y cuando las causas que ocasionó cualquiera de los daños descritos anteriormente estén debidamente amparados en la póliza sin que los gastos sobrepasen los límites fijados en las condiciones particulares de la misma.

- c) Gastos Médicos: Ampara los gastos por el daño corporal que sufre un tercero que salga afectado en su siniestro que tuviese el asegurado.
- d) Daño Total: Ampara el 100% del valor de la suma asegurada del vehículo
- e) Daño Parcial: Ampara un porcentaje del valor de la suma asegurada del vehículo, según se haya estipulado en las condiciones particulares de la póliza.
- f) Robo Total: Esta cobertura garantiza la devolución del 100% del valor asegurado.
- g) Robo Parcial: Ampara un porcentaje del valor asegurado según se haya estipulado en las condiciones particulares de la póliza.
- h) Servicios Extras: esta cobertura es la que hace la diferencia entre compañías o empresas de seguros, ya que le brindan comodidad al asegurado de tener por cualquier eventualidad uno o todos los siguientes servicios:
- Servicio de wincha o arrastre
 - Servicio mecánico
 - Servicio de ambulancia
 - Servicio de Auto por Auto.

2.1.2 Impuestos de un seguro vehicular

Los siguientes impuestos son cargados al asegurado ya que son obligatorios en el país:

- a) Seguro Campesino: Es una contribución obligatoria que se debe sumar a la Prima Neta, siendo el 0.5% de la prima neta.
- b) Derechos de Emisión: Es el valor que se reserva la aseguradora por tramitar la póliza.
- c) Superintendencia de Bancos: Es el valor que según las leyes ecuatorianas se debe sumar a la Prima Neta, siendo este el 3.5% de la prima neta.
- d) IVA: Es el impuesto que se paga por la transferencia de bienes y por la prestación de servicios, siendo el 12% de la Prima Neta.

2.2 Programación Extrema

Introducción

Programación Extrema o también conocida como XP (Extreme Programming) es una metodología ágil que forma parte del movimiento de desarrollo ágil de software y que nace basada en la simplicidad, la comunicación y el reciclado continuo de código. Pensada entre los años 1996 y 1999 durante un proyecto de pago de roles para la empresa automotriz Chrysler cuando Kent Beck, Ward Cunningham, Ron Jeffereis y otros trabajaban juntos extendiendo sus ideas de un desarrollo de software adaptable y orientado a la gente.

En la figura 2.1 Kent Beck define cuatro grandes tareas a realizar en el desarrollo de todo proyecto:



Figura 2.1 Tareas de la programación extrema

En la figura 2.2 se detalla las cinco características que debe reunir un programador XP.



Figura 2.2 Características básicas de la programación extrema

2.2.1 Definición

La metodología de Programación Extrema es una disciplina para el desarrollo de software, se trata de una metodología ligera que está en contraposición a las metodologías tradicionales, basada en la simplicidad, comunicación y retroalimentación o reutilización del código desarrollado.

La metodología es importante por dos razones:

- Constituye un método de control para las actividades de desarrollo de software que se han convertido en métodos operativos Standard
- Es una de las pocas y nuevas metodologías ligeras desarrolladas para reducir el costo del software

De todo lo anteriormente mencionado se puede decir que XP es:

“Un proceso ligero, de bajo riesgo, flexible, predecible, científico y divertido de desarrollar software”²

2.2.1.1 Objetivos de la programación extrema

- Dar satisfacción del cliente. Dándole el software que él necesita y cuando lo necesita.
- Potenciar al máximo el trabajo en grupo. Tanto los jefes de proyecto, los clientes y desarrolladores, son parte del equipo y están involucrados en el desarrollo del software.

2.2.2 Variables

Programación Extrema define cuatro variables para cualquier proyecto software. En la figura 2.3 se detalla las variables de la programación extrema.



Figura 2.3 Variables de la programación extrema

De las cuatro variables sólo tres de ellas podrán ser fijadas por las fuerzas externas al proyecto (clientes y jefes de proyecto), mientras que el valor de la variable libre será establecido por el equipo de desarrollo en función de los valores de las otras tres. En muchos casos normalmente los clientes y jefes de proyecto se creen capaces de fijar de antemano el valor de todas las variables, cuando esto ocurre la calidad es lo primero que se esfuma de la ecuación.

² Extreme Programming Explained: Embrace Change, by Kent Beck, Addison Wesley, 2004.

Un parámetro que se debe tomar en cuenta es la calidad, aumentar la calidad conduce a que el proyecto pueda realizarse en menos tiempo, no tener esto en cuenta conduce a la desmoralización del equipo y, con ello, a la larga, a la ralentización del proyecto mucho más allá del tiempo que hubiera podido ganarse al principio con esta reducción de calidad.

2.2.3 Fases de la metodología de programación extrema

Hay diversas prácticas inherentes a la Programación Extrema, en cada uno de los ciclos de desarrollo del proyecto. En la figura 2.4 se visualiza las fases que tiene la programación extrema.

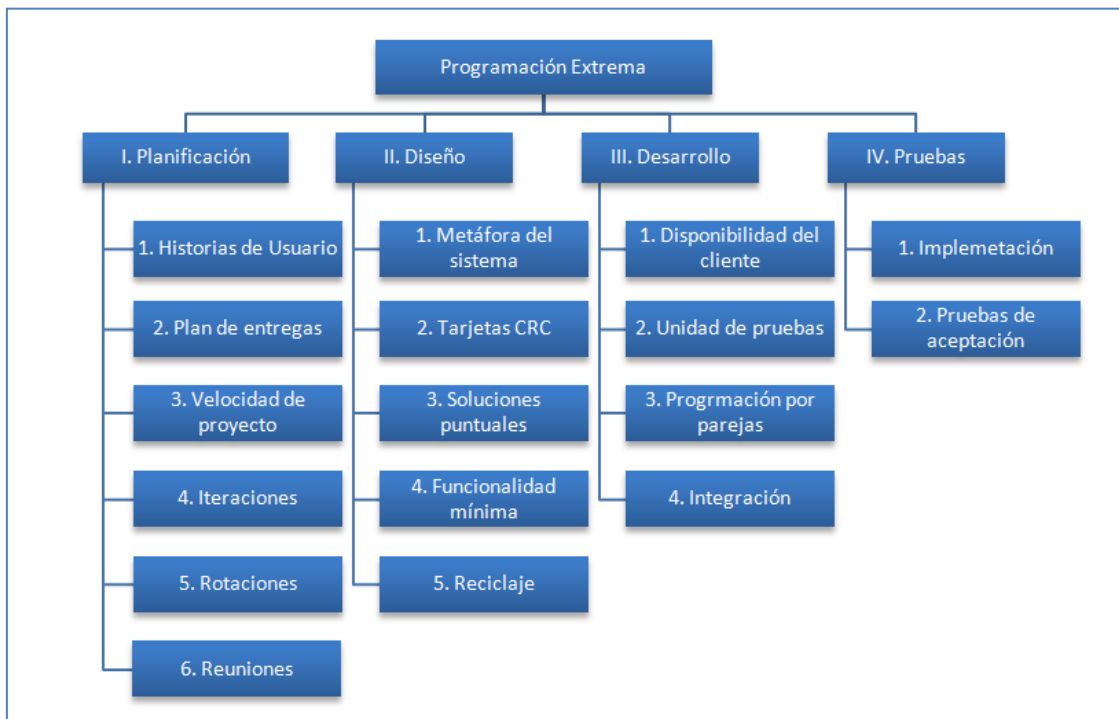


Figura 2.4 Fases de la programación extrema ³

³Diseño por autores de la tesis basado en: <http://www.info-ab.uclm.es/asignaturas/42551/trabajosAnteriores/Presentacion-XP.pdf>

2.2.3.1 Planificación

Es un permanente diálogo entre la parte empresarial y técnica del proyecto en la que se define:

- Alcance: ¿Qué es lo que el software debe resolver para que este genere valor?
- Prioridad: ¿Qué debe ser hecho en primer lugar?
- Composición de las versiones: ¿Cuánto es necesario hacer para saber si el negocio va mejor con el software que sin el?
- Fechas de entrega de versiones: ¿Cuáles son las fechas en las cuales se presentan cada una de las iteraciones?
- Estimaciones: ¿Cuánto lleva implementar una característica?
- Consecuencias: Informar sobre las consecuencias de la toma de decisiones por parte del negocio.
- Procesos: ¿Cómo se organiza el trabajo y el equipo?
- Programación Detallada: Dentro de una versión ¿Qué problemas se resolverán primero?

A continuación se detalla las fases de la planificación:

- Historias de usuario

Las historias de usuario tienen el mismo propósito que los casos de uso pero no son lo mismo, siendo una técnica para especificar los requisitos del software, escritas por los propios clientes en donde expresan las necesidades del sistema con definiciones cortas y en su propio lenguaje en una tarjeta de papel.

Estas historias de usuario solamente proporcionan los detalles sobre la estimación del riesgo y cuánto tiempo conlleva la implementación de dicha historia de usuario. El nivel de detalle de las historias debe ser el mínimo posible que permita hacerse una ligera idea de cuánto cuesta implementar el sistema.

Los desarrolladores deben hacer una estimación de cuánto tiempo, idealmente, les lleva implementar cada historia de usuario. Las condiciones ideales son aquellas en las que se codifica la historia de usuario sin otras distracciones y sabiendo exactamente qué es lo que hay que implementar. Como resultado se debe obtener un periodo ideal de 1, 2 ó 3 semanas. Más de 3 semanas implica que se debe dividir la historia de usuario en partes. Menos de 1 semana implica que la historia de usuario es demasiado sencilla y se debe unir dos o más de ellas.

En la figura 2.5 Kent Beck en su libro presenta un ejemplo de la ficha de una historia de usuario en la cual se puede reconocerse varios contenidos.

Customer Story and Task Card BIN Development / CCLA

DATE: 3/19/99 TYPE OF ACTIVITY: NEW FIX ENHANCE POC TEST

STORY NUMBER: 1275 PRIORITY: USER: TECH:

PRIOR REFERENCE: RISK: TECH ESTIMATE:

TASK DESCRIPTION:
 SPLIT COLA: When the COLA rate changes in the middle of the BiW Pay Period, we will want to pay the 1st week of the pay period at the Old COLA rate and the 2nd week of the pay period at the NEW COLA rate. Should occur automatically based on system design.

NOTES:
 For the OI, we will write a m/frame program that will pay or calculate the COLA in the 2nd week of OT. The plant currently retransmits hours data for the 2nd week exclusively when we run the EOL. This will come into the Model as a "2144" CCLA.

TASK TRACKING: Gross Pay Adjustment Create RM Boundary and Place in BEG+Gross COLA

Date	Status	To Do	Comments	INI

Figura 2.5 Tarjeta de una historia de usuario ⁴

⁴ Extreme Programming Explained: Embrace Change, by Kent Beck, Addison Wesley, 2004.

- Plan de entregas

Con las historias de usuario ya definidas se crea el plan estimado de entrega o también llamado (*Release plan: Planificación de publicaciones), se convoca a reuniones para crear dicho plan, este plan se usa para crear los planes de iteración para cada iteración, tomando en cuenta las historias de los usuarios cuya implementación se considera primordial y se añaden aquellas que no han pasado las pruebas de aceptación de anteriores iteraciones. En esta reunión están presentes tanto los desarrolladores como los usuarios. La planificación de una iteración también hace uso de tarjetas en las que se escriben tareas, que duran entre unos tres días (la duración la deben decidir los propios desarrolladores). El plan de entregas se da en función de dos parámetros: tiempo de desarrollo ideal y grado de importancia para el cliente.

Las iteraciones individuales son planificadas en detalle justo antes de que comience cada iteración. A modo de esquema de detalla en la figura 2.6 la planificación de las historias de usuario antes de entregar el plan.

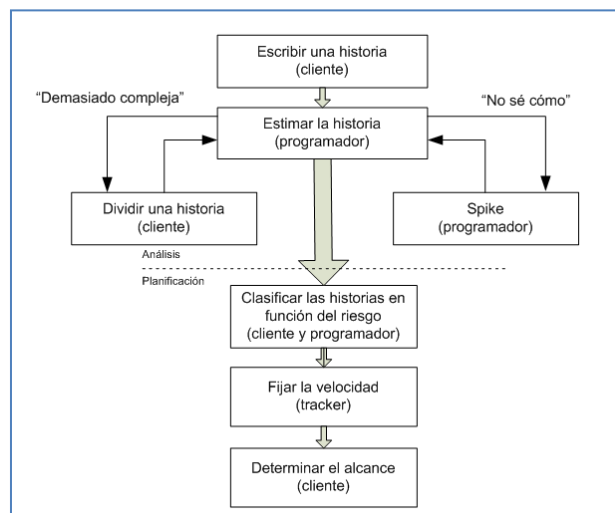


Figura 2.6 Esquema de un plan de entregas ⁵

⁵ Planificación Historias de Usuario, Fernández Escribano, Gerardo. Ingeniería de Software II. Introducción a Extreme Programming 9-12-2002.

- Velocidad del proyecto

Es utilizada para determinar cuántas historias de usuario pueden ser implementadas antes de una fecha dada o cuánto tiempo es necesario para llevar a cabo un conjunto de historias en una iteración. Es conveniente reevaluar esta medida cada 3 ó 4 iteraciones y si se aprecia que no es adecuada hay que negociar con el cliente un nuevo "Release Plan"⁶.

- Iteraciones

Cada iteración corresponde a un periodo de tiempo de desarrollo del proyecto de entre una y tres semanas de tiempo de programación. Al principio de cada iteración se convoca a una reunión con el fin de realizar el plan de iteración correspondiente.

La suma de los días que toma desarrollar todas las tareas de la iteración no debe sobrepasar la velocidad del proyecto de la iteración anterior. Si la iteración está sobrecargada, el cliente debe decidir que historias de usuario retrasar a una iteración posterior. Si, por el contrario, la iteración tiene huecos se debe rellenar con otras historias de usuario.

Un plan de iteración puede verse en la figura 2.7 el cual consiste en seleccionar las historias de usuario que según el plan de entregas corresponden a la iteración y se visualiza que cada historia se transforma en tareas de desarrollo.

⁶ Release Plan: nuevo plan liberado.

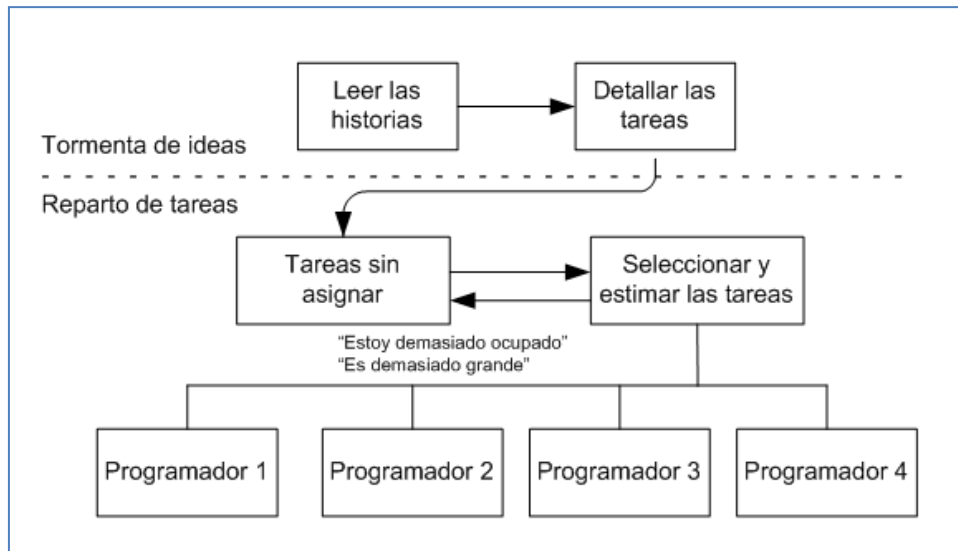


Figura 2.7 Plan de iteraciones ⁷

- Rotaciones

Una gran característica de esta metodología es el código con propiedad compartida, evitando los famosos cuellos de botella que pueden ser ocasionados por los desarrolladores. La ventaja de las rotaciones es que cada persona conoce cómo funciona el sistema en general y se puede realizar un reparto más equitativo del trabajo. El hecho de que se asignen por parejas permite entrenar a un nuevo miembro, simplemente dividiendo el grupo original en dos.

- Reuniones de Seguimiento

Es fundamental mantener la comunicación entre las diferentes partes que intervienen, es por ello que las reuniones se debe tratar de que estas sean frecuentes, de poca duración y diarias; en estas se puede analizar los

⁷ <http://www.info-ab.uclm.es/asignaturas/42551/trabajosAnteriores/Presentacion-XP.pdf>

problemas y las soluciones señalando los puntos a los que se debe dar más importancia por su dificultad o por su punto crítico.

2.2.3.2 Diseño

Las cosas que se realice deben ser lo más sencillo posible, fácilmente entendible e implementable, si alguna parte de la implementación resulta compleja, se debe replantear, de esta manera cualquier cambio y modificación cuesta menos tiempo y esfuerzo desarrollar.

A continuación se detalla las fases del diseño:

- Metáfora

“Una metáfora para el sistema es una historia que todo el mundo puede contar acerca de cómo el sistema funciona”⁸

El elegir una metáfora permite mantener la coherencia de nombres de todo aquello que se va implementar, se debe elegir un sistema de nombres que permita que cualquiera que vea el diseño lo pueda interpretar la relación entre el objeto y aquello que representa.

- Tarjetas CRC (clases, responsabilidad y colaboración)

Hay ocasiones que los desarrolladores se reúnen para diseñar partes muy complejas en las cuales actúan los que tienen mayor experiencia e incluso el cliente, en estas reuniones se emplea un tipo de tarjetas denominadas CRC (Class, Responsibilities and Collaboration) cuyo objetivo es facilitar la comunicación y documentar los resultados. Para cada clase identificada se

⁸ <http://www.willydev.net/descargas/prev/ExplicaXP.pdf> (Kent Beck).

llena una tarjeta de este tipo y se especifica su finalidad así como otras clases con las que se interaccione.

Usar estas tarjetas permite desprenderse del método de trabajo basado en procedimientos y trabajar con una metodología basada en objetos.

En la figura 2.8 el nombre de la clase se coloca a modo de título en la tarjeta, las responsabilidades se colocan a la izquierda, y las clases que se implican en cada responsabilidad a la derecha, en la misma línea que su requerimiento correspondiente.

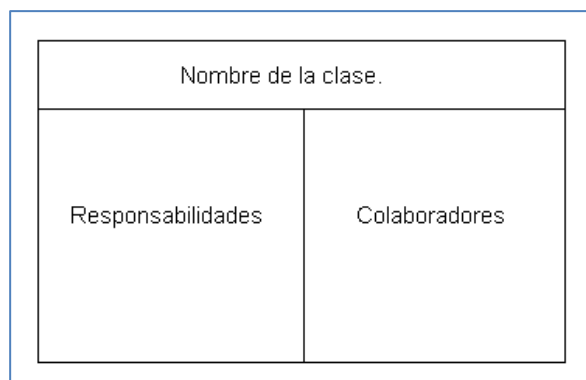


Figura 2.8 Tarjeta CRC ⁹

- Soluciones puntuales para reducir riesgos

En este tema se hace referencia a los programas spike, los cuales son programas que exploran una posible solución al problema, a partir de estos pequeños programas se va contribuyendo a la solución del problema.

Producir rápidamente versiones del sistema que sean operativas, aunque no cuenten con toda la funcionalidad del sistema. Esta versión ya constituye un resultado de valor para el negocio. Una entrega no debe tardar más de 3 meses.

⁹ <http://robiol.wordpress.com/tag/ejemplo-tarjeta-crc/>

- Funcionalidad mínima

Permite a los equipos de programadores XP mejorar el diseño del sistema a través de todo el proceso de desarrollo. Los programadores evalúan constantemente el diseño y recodifican lo necesario. La finalidad es mantener un sistema enfocado a proveer el valor del negocio mediante la minimización de código duplicado y/o ineficiente.

No se debe caer en el caso de ir añadiendo funcionalidades según se vaya ocurriendo así se tenga conocimiento como implementarlas. Por lo que se debe centrar en las tareas que se fija para hoy, y hacerla lo mejor posible, sólo el 10% de la misma es utilizada, lo que implica que el desarrollo de funcionalidad extra es un desperdicio de tiempo y recursos.

- Reciclaje

Se debe eliminar la redundancia o sea la funcionalidad inútil, mejorando y modificando la estructura y codificación de códigos ya creados sin alterar su funcionalidad, el reciclaje permite ahorrar tiempo e incrementa la calidad ya que procura optimizar su funcionamiento.

2.2.3.3 Desarrollo

Esta etapa debe reunir las siguientes características y cualidades:

- Disponibilidad del Cliente

Una de las condiciones que impone la metodología es tener al usuario siempre disponible, formando parte del equipo de desarrollo, todas las fases que abarca la metodología requieren de comunicación con el usuario, si es posible cara a cara, en persona, sin intermediarios, con esto se consigue disminuir el tiempo

de comunicación y la cantidad de documentación junto con los altos costos de su creación y mantenimiento.

- Unidad de Pruebas

El tiempo que toma realizar determinados test es aproximadamente el mismo tiempo que se emplea en crear exclusivamente dicho código, la creación de dichas unidades ayuda al programador a tener una visión acerca del cómo?, en otras palabras, del comportamiento del programa. Las pruebas unitarias son establecidas antes de escribir el código y son ejecutadas constantemente ante cada modificación del sistema. Los clientes escriben las pruebas funcionales para cada historia de usuario que deba validarse. En este contexto de desarrollo evolutivo y de énfasis en pruebas constantes, la automatización para apoyar esta actividad es crucial.

- Programación por parejas

Uno de los aspectos a tomar en cuenta en este punto es que todo el código que forma parte del plan, es desarrollado por dos personas, el propósito es incrementar la calidad de software sin afectar de manera alguna el tiempo de entrega. Se debe tomar en cuenta que este grupo de dos personas parten de principios o conocimientos similares, esto significa que deben encontrarse en un mismo nivel.

Para alcanzar la calidad del software cuando se trabaja entre varias personas se debe considerar los siguientes puntos:

- Estándares de Codificación

Se debe seguir los estándares en el código a ser desarrollado, esto permite facilitar su lectura y comprensión por ende en caso de ser necesario realizar modificaciones por cualquier miembro del equipo.

- No trabajar más de 40 horas semanales

Aquí se utilizan las conocidas reuniones de plan de entregas para cambiar los objetivos del proyecto. No es recomendable que se incorpore gente al proyecto, una vez que ya se ha empezado, se debe trabajar un máximo de 40 horas semanales, no es eficiente trabajar 60 horas a la semana ya que lo único que producirá es un producto de software con menor calidad.

- Todo el código es común para todos

Cualquier programador puede cambiar una línea de código para añadir funcionalidad o eliminar algún error, pero esto siempre será probado antes de ser publicado para mantener la consistencia del programa.

- Integración continua

Es importante en este punto señalar que los programadores deben actualizar sus módulos con las versiones más recientes de trabajo, logrando así que todo el mundo trabaje con la última versión. La actualización de cada módulo es responsabilidad propia de cada pareja de programadores así como las pruebas que se haga a la versión a ser actualizada.

2.2.3.4 Pruebas

Un principio muy esencial de la programación extrema es que todo el código se prueba, debiendo estar los test lo más automatizados posible y empaquetados

junto al código. Siempre que se detecte un fallo se debe crear una unidad de pruebas para protegernos de este, de esta manera se asegura que la localización del mismo sea más fácil por parte de los programadores.

Las pruebas unitarias ayudan también a la refactorización, ya que aseguran que los cambios que se hayan introducido en la iteración actual no afecten a la funcionalidad. Pero no se debe olvidar de las pruebas de aceptación que son creadas a partir de las pruebas de usuario, durante una iteración la historia de usuario seleccionada en la planificación de iteraciones se convierte en una prueba de aceptación.

2.2.4 UML

Introducción

UML o Lenguaje unificado de modelado es un lenguaje que permite representar de forma gráfica los elementos que conforman un sistema software orientado a objetos, UML es considerado como un estándar para el desarrollo de sistemas permitiendo generar diseños que capturan las ideas y poder transmitirlos a otras personas. UML está compuesto por distintos elementos gráficos que se combinan para formar diagramas cuya finalidad es representar diversas perspectivas a las que se conoce como modelos.

Un modelo es una representación de un sistema software desde una perspectiva específica, en un dibujo técnico como el rebatimiento de un cuerpo el mismo que muestra un mismo grafico visto desde diferentes ángulos, los modelos permiten centrar la atención en los distintos aspectos de un sistema.

UML cuenta con los siguientes tipos de modelos:

En la figura 2.9 se muestra los diferentes diagramas que se deben construir en el análisis, diseño y desarrollo de un proyecto de software.

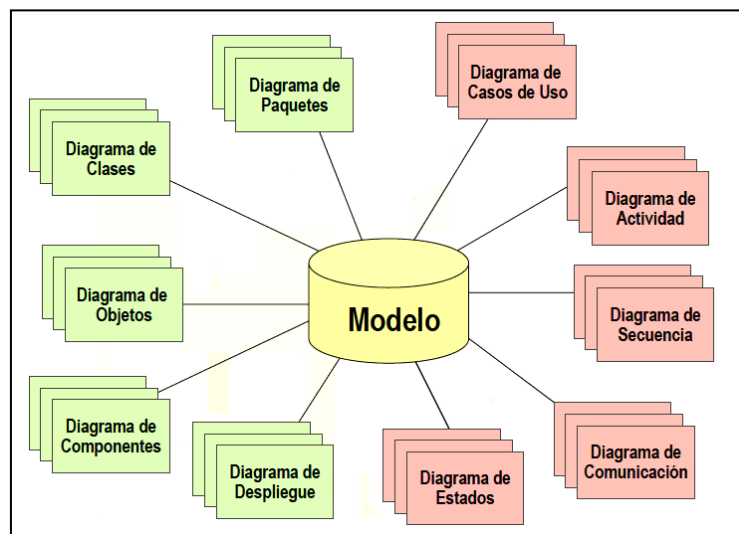


Figura 2.9 Modelos UML ¹⁰

2.2.4.1 Diagrama de casos de uso

Un caso de uso describe las acciones que se producen entre el actor y el sistema cuando el actor usa el sistema para realizar una tarea específica; para los desarrolladores del sistema esta es una herramienta de gran utilidad, ya que permite determinar requerimientos desde el punto de vista del usuario, además de plantear un enfoque bajo el cual el sistema pueda ser usado por cualquier tipo de usuario y no solo por aquellos con conocimientos de computación.

A continuación en la figura 2.10 se muestra como el actor administrador está encargado de realizar algunas actividades dentro de un SI, el actor puede ingresar la información, registrar información, especificar perfiles y generar claves.

¹⁰ JOSEP VILALTA MARZO, Introducción a UML, Vico Open Modeling, S.I,
http://www.vico.org/aRecursosPrivats/UML_TRAD/talleres/mapas/UMLTRAD_101F/LinkedDocuments/TRAD_introUML.pdf

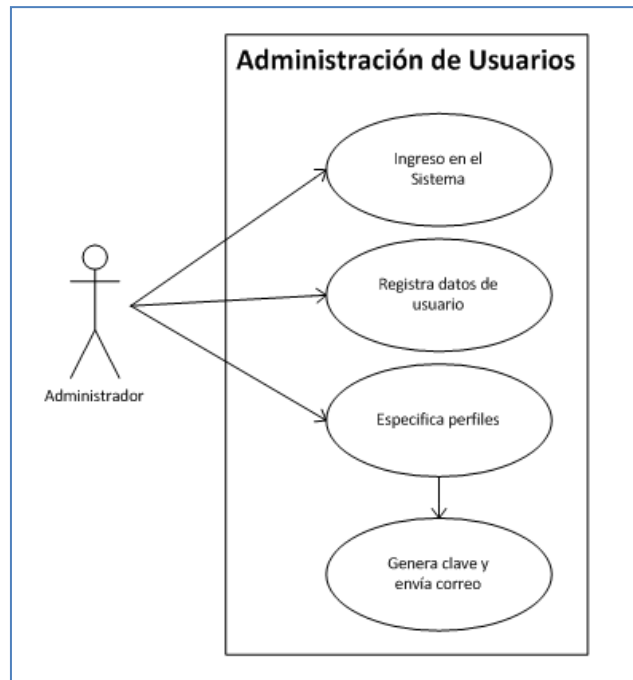


Figura 2.10 Caso de uso

2.2.4.2 Diagrama de clases

Una clase es la agrupación de atributos y acciones que tienen similar comportamiento, por lo general son representaciones del mundo real.

Un diagrama de clase es la representación de las relaciones que existen entre las diferentes clases que conforman el sistema, este diagrama se va refinando a lo largo del análisis y diseño hasta lograr un modelo ideal, con el cual los desarrolladores pueden trabajar, además permite al analista hablarle al cliente con terminología que pueda entender.

En la figura 2.11 se diagrama 4 clases que permiten obtener el usuario, la agencia y el cliente de la solicitud. El diagrama muestra los atributos y métodos que son utilizados en cada clase.

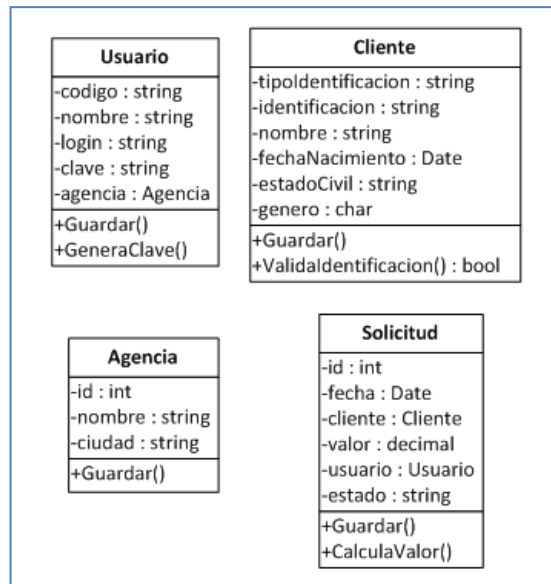


Figura 2.11 Diagrama de clases

2.2.4.3 Diagrama de Objetos

Un objeto es una instancia de una clase, la misma que permite acceder a los atributos y funciones de una clase; un diagrama de objetos muestra una serie de objetos y sus relaciones.

En la figura 2.12 se muestra el diagrama de objetos que tiene relación con un catálogo y sus elementos principales.

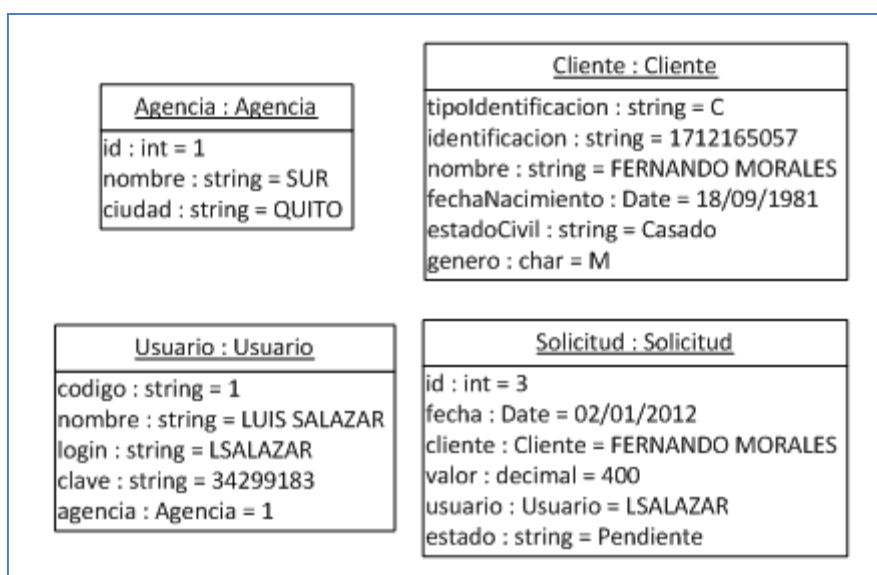


Figura 2.12 Diagrama de objetos

2.2.4.4 Diagrama de comportamiento

Un diagrama de comportamiento define la secuencias de estado por la que pasa un objeto a largo de su vida en respuesta a eventos.

2.2.4.5 Diagrama de estados

Un diagrama de estados muestra los posibles estados por los que pasa un objeto en ciertos puntos del tiempo; los eventos representan a los incidentes que hacen que un objeto pase de un estado a otro, cabe recalcar que todo objeto tiene un estado inicial y un estado final.

En la figura 2.13 se detalla en el diagrama los posibles estados que tiene una solicitud.

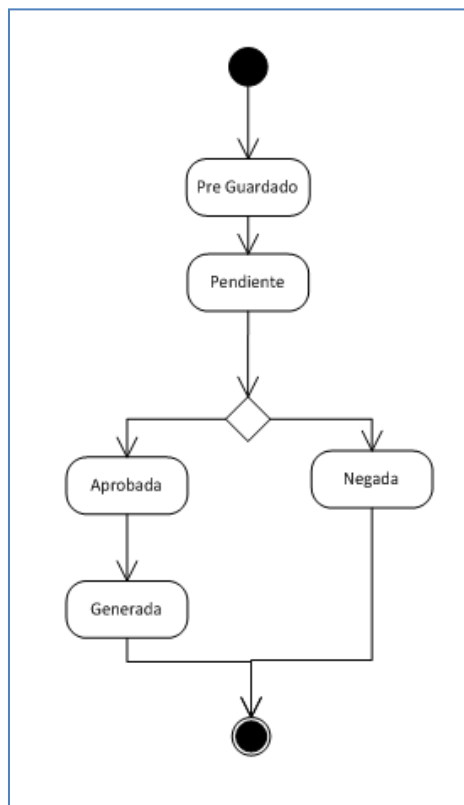


Figura 2.13 Diagrama de estados de solicitudes

2.2.4.6 Diagrama de Actividades

Un diagrama de actividades muestra el flujo de actividades que se presentan a lo largo del tiempo considerando las peticiones que se realizan entre clases.

La idea de este tipo de diagramas es generar una especie de Pert donde se puede ver el flujo de actividades; este se representa con una serie de arcos y nodos los mismos que definen a las actividades de una clase y las relaciones entre las actividades de varias clases.

El diagrama de actividades en la figura 2.14 muestra la administración de usuarios donde se puede observar cada uno de los pasos.

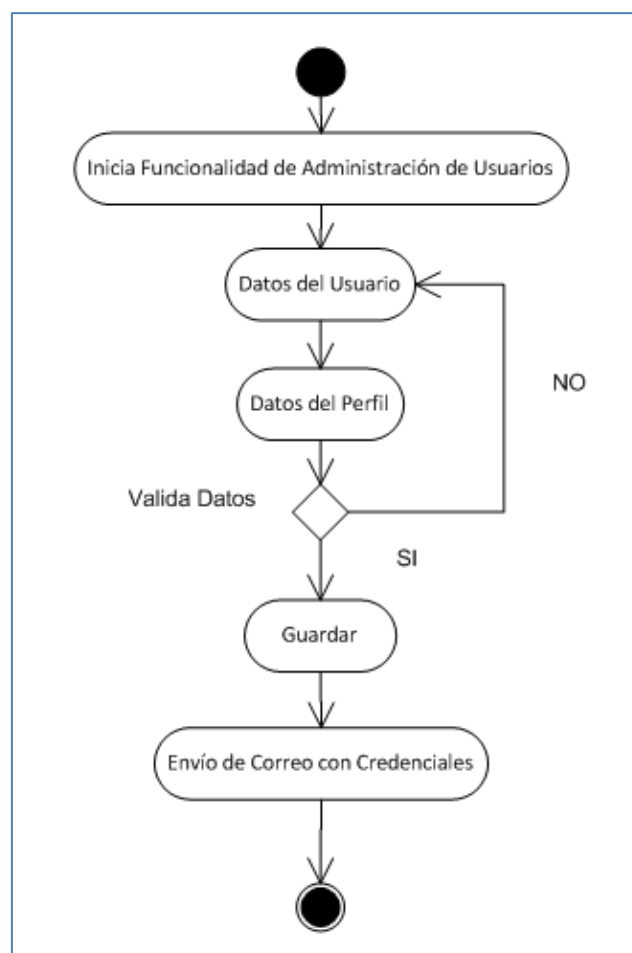


Figura 2.14 Diagrama de actividades

2.2.4.7 Diagramas de interacción

Este tipo de diagrama es utilizado para enfatizar la interacción que existe entre objetos; por lo cual este tipo de diagramas se dividen en dos tipos:

- Diagrama de secuencia

Permite mostrar la mecánica de cómo interactúan los objetos entre sí de forma ordenada en una secuencia de tiempo; donde el eje vertical representa el tiempo y el eje horizontal los mensajes entre objetos, se pueden utilizar flechas para indicar la dirección de los mensajes, cabe recalcar que el tiempo se recorre de arriba hacia abajo e igualmente y la interacciones van apareciendo una detrás de otra.

- Diagrama de colaboración

Un diagrama de colaboración permite representar a nivel de detalle un diagrama de secuencia centrándose en las interacciones de cada objeto y el orden en que estas se presentan, los objetos se conectan por medio de enlaces donde cada enlace representa una instancia de asociación entre los objetos implicados.

2.2.4.8 Diagramas de implementación

Son aquellos utilizados para representar aspectos relacionados con la implementación como la estructura del código fuente o también la distribución del hardware del sistema; existen dos tipos de diagramas de implementación:

- Diagrama de componentes

En UML todo elemento físico se modela como componente, siendo este un grupo de objetos que interactúan entre sí o la agrupación de funcionalidad.

Los servicios de una clase o un componente se definen a través de un interfaz el mismo que funciona como una unión entre componentes, compartiendo así la funcionalidad de los mismos

UML define cinco estereotipos estándar que se aplican a los componentes:

- Executable.- Componente que se puede ejecutar en un nodo.
- Library.- Biblioteca de objetos estática o dinámica.
- Table.- Componentes que representa una tabla de una base de datos.
- File.- Componente que representa un documento que contiene código fuente o datos.
- Document.- Componente que representa un documento.

UML no especifica iconos predefinidos para estos estereotipos.

La figura 2.15 muestra el diagrama de un componente que contiene una clase en código java y una imagen, los dos componentes se relacionan entre sí de forma que el usuario puede ver los componentes a través de una interfaz.

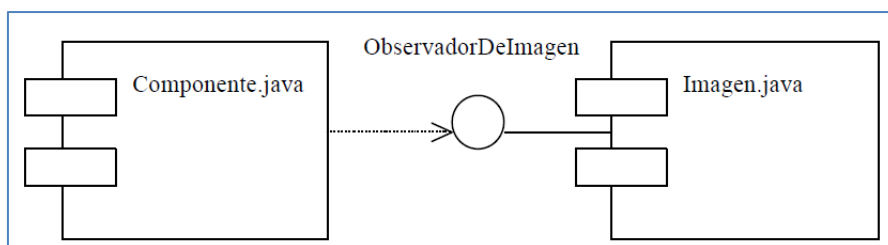


Figura 2.15 Diagrama de componentes ¹¹

¹¹ UML Resource Center, Rational Software. <http://www.rational.com/uml/>

- Diagrama de plataforma o despliegue

Muestra la configuración de los componentes hardware, los procesos, los elementos de procesamiento en tiempo de ejecución y los objetos que existen en tiempo de ejecución. En este tipo de diagramas intervienen:

- Nodos
- Asociaciones de comunicación
- Componentes dentro de los nodos y objetos que se encuentran a su vez dentro de los componentes.

La figura 2.16 muestra el Diagrama de los elementos de procesamiento que intervienen dentro de un computador y funcionan sobre un sistema operativo para permitir operar un sistema de nómina a un usuario.

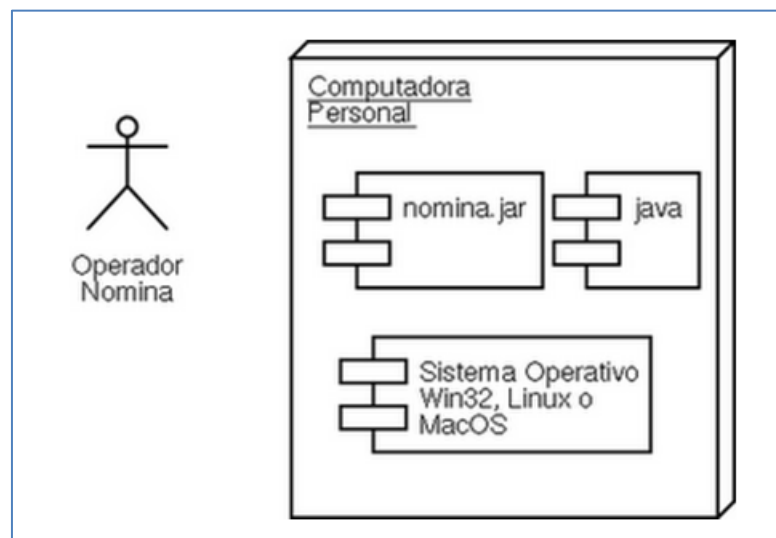


Figura 2.16 Diagrama de plataforma o despliegue ¹²

¹² Diagrama de Plataforma o despliegue, <http://3.bp.blogspot.com>

2.2.5 Plataforma .NET

2.2.5.1 Introducción

Microsoft .NET es una plataforma que engloba distintas aplicaciones, servicios Web XML, conceptos y que en conjunto permiten el desarrollo y la ejecución de aplicaciones, también provee de mecanismos robustos, seguros y eficientes para asegurar que la ejecución de las mismas sea óptima

Plataforma orientada a internet y entornos distribuidos, siendo los servicios web (Web Services) un medio que permitirá a distintas tecnologías interoperar entre sí, así como conectar diversos sistemas operativos, dispositivos, información y usuarios dando a los desarrolladores las herramientas y tecnologías necesarias para desarrollar soluciones de negocios de manera rápida sin importar que involucren diversos medios y tecnologías.

En la figura 2.17 se muestra las partes que componen la plataforma .NET



Figura 2.17 Plataforma Microsoft .NET ¹³

¹³ <http://www.slideshare.net/PauloGuerraT/1-plataforma-net>

2.2.5.2 Características principales de la plataforma .NET

- **Portabilidad:** Una aplicación desarrollada en .NET puede ser ejecutada en cualquier sistema operativo de cualquier maquina que disponga de una versión de la plataforma.
- **Multilenguaje:** Cualquier lenguaje de programación puede adaptarse a la plataforma .NET y ejecutarse en ella.
- **Modelo de programación:** Único para todo tipo de aplicaciones y dispositivo de hardware (PC's, Pocket PC's, Teléfonos Celulares Inteligentes, también llamados "SmartPhones", Tablet PC's, etc.).
- **Integración:** Se integra fácilmente con otras aplicaciones desarrolladas en plataformas Microsoft de una forma sencilla y rápida.

2.2.5.3 Elementos de la plataforma .NET

- Un modelo de programación pasado en XML.
- Un conjunto de servicios Web para facilitar a los desarrolladores la integración de los servicios.
- Un conjunto de servidores que permiten ejecutar estos servicios.
- Software en el cliente para poder utilizar estos servicios (como Windows XP, agendas electrónicas, etc.)
- Herramientas para el desarrollo como Visual Studio .NET.

En la figura 2.18 se muestra los elementos que pueden componer la plataforma .NET

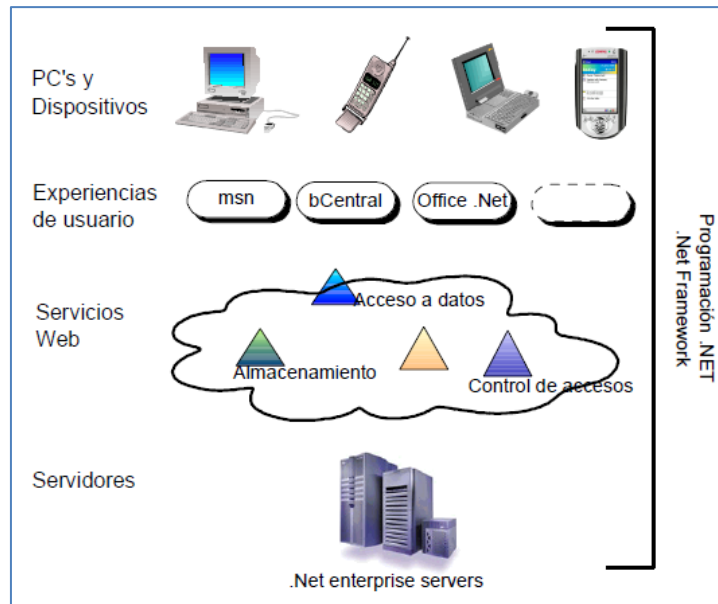


Figura 2.18 Elementos de la plataforma .NET ¹⁴

2.2.5.4 Arquitectura .NET

La arquitectura .NET (.NET Framework) es el modelo de programación de la plataforma .NET para construir y ejecutar los servicios .NET, reduciendo la complejidad en el desarrollo, permitiendo a los programadores centrarse en escribir la lógica específica del servicio a desarrollar y consiste en tres partes fundamentales.

- Common Language Runtime (CLR) entorno de ejecución
- Framework Clases (Clases de la plataforma)
- ASP.NET

¹⁴ Elementos de la plataforma .NET, <http://www.disca.upv.es/enheror/pdf/ActaNET.PDF>

En la figura 2.19 se detalla las capas de la plataforma .NET

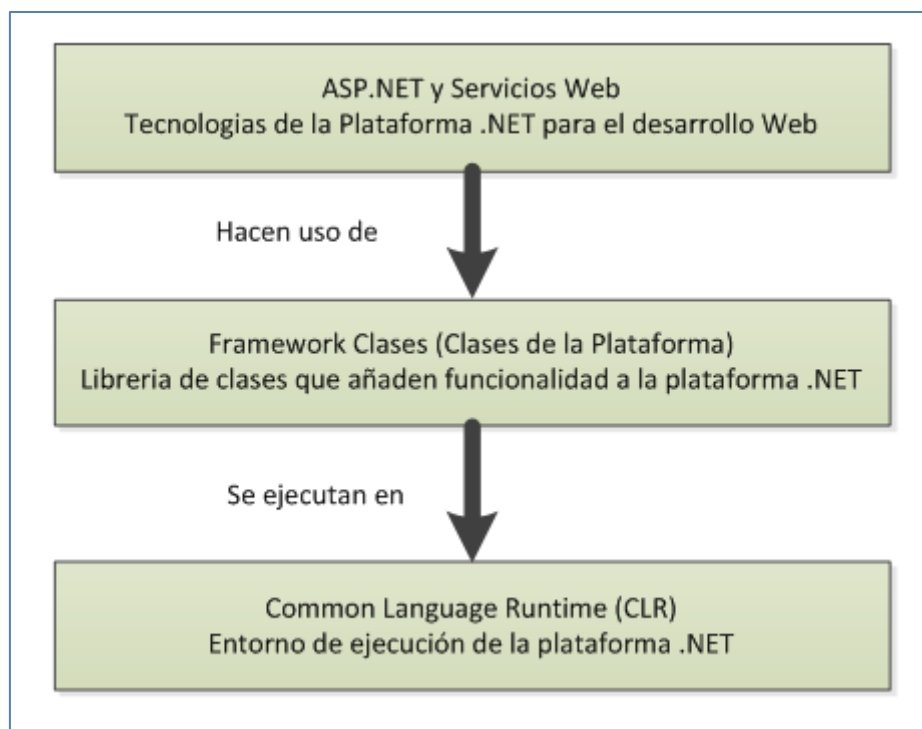


Figura 2.19 Capas de la plataforma .NET ¹⁵

2.2.5.5 Características principales de la arquitectura .NET

- Un entorno de ejecución de aplicaciones, también llamado “Runtime”, que es un componente de software cuya función es la de ejecutar las aplicaciones .NET e interactuar con el sistema operativo ofreciendo sus servicios y recursos.
- Un conjunto de bibliotecas de funcionalidades y controles reutilizables, con una enorme cantidad de componentes ya programados listos para ser consumidos por otras aplicaciones.

¹⁵ Capas de la plataforma .NET, Unai Extremo Baigorri, Borja Sotomayor Basilio.

- Un conjunto de lenguajes de programación de alto nivel, junto con sus compiladores y linkers, que permitirán el desarrollo de aplicaciones sobre la plataforma .NET.
- Un conjunto de utilitarios y herramientas de desarrollo para simplificar las tareas más comunes del proceso de desarrollo de aplicaciones
- Documentación y guías de arquitectura, que describen las mejores prácticas de diseño, organización, desarrollo, prueba e instalación de aplicaciones .NET

2.2.5.6 .Net Framework

Es el componente fundamental de la plataforma Microsoft .NET, necesario tanto para poder desarrollar aplicaciones como para poder ejecutarlas. Siendo el marco de trabajo sobre la que se reúne todo un conjunto de lenguajes y servicios que simplifican enormemente el desarrollo de aplicaciones. Mediante esta herramienta se ofrece un entorno de ejecución altamente distribuido, que permite crear aplicaciones robustas y escalables. El avance mas significativo del .NET Framework 4.0 es la mejora de la compatibilidad con el desarrollo de sitios web habilitados para AJAX. ASP.NET mediante un conjunto de nuevos controles de servidor y nuevos API.

- **.NET Framework Redistributable Package** Mínimo componente de la plataforma .NET que se necesita para poder ejecutar aplicaciones. Se instala en el ambiente de producción, una vez que el desarrollo y las pruebas de la aplicación han finalizado.
- **NET Framework SDK** Versión dirigida al ambiente de desarrollo de aplicaciones, contiene herramientas de desarrollo de línea de comandos

(compiladores, depuradores, etc.), documentación de referencia, ejemplos y manuales.

Está compuesto por:

- El entorno de ejecución de la plataforma .NET
- Las bibliotecas de funcionalidad reutilizable

Los principales componentes de este entorno son:

- Lenguajes de compilación
- Biblioteca de clases de .Net
- CLR (Common Language Runtime)

En la figura 2.20 se detalla los componentes de la arquitectura .NET framework.

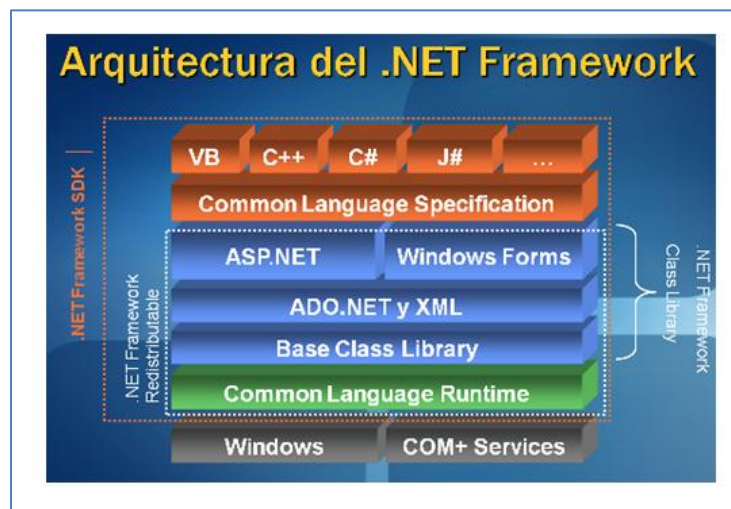


Figura 2.20 Arquitectura del .Net Framework ¹⁶

- Common Language Runtime (CLR)

Es el núcleo del Framework de .Net, ya que es el entorno de ejecución en el que se cargan las aplicaciones desarrolladas en los distintos lenguajes.

¹⁶ <http://www.slideshare.net/PauloGuerraT/1-plataforma-net>

La herramienta de desarrollo compila el código fuente de cualquiera de los lenguajes soportados por .Net en un mismo código, denominado código intermedio (MSIL, Microsoft Intermediate Language). Para generar dicho código el compilador se basa en el Common Language Specification (CLS) que determina las reglas necesarias para crear código MSIL compatible con el CLR.

De esta forma, indistintamente de la herramienta de desarrollo utilizada y del lenguaje elegido, el código generado es siempre el mismo, ya que el MSIL es el único lenguaje que entiende directamente el CLR. Este código es transparente al desarrollo de la aplicación ya que lo genera automáticamente el compilador.

Sin embargo, el código generado en MSIL no es código máquina y por tanto no puede ejecutarse directamente. Se necesita un segundo paso en el que una herramienta denominada compilador JIT (Just-In-Time) genera el código máquina real que se ejecuta en la plataforma que tenga la computadora.

De esta forma se consigue con .Net cierta independencia de la plataforma, ya que cada plataforma puede tener su compilador JIT y crear su propio código máquina a partir del código MSIL.

La compilación JIT la realiza el CLR a medida que se invocan los métodos en el programa y, el código ejecutable obtenido, se almacena en la memoria caché de la computadora, siendo recompilado sólo cuando se produce algún cambio en el código fuente.

- Biblioteca de clases de .Net

Cuando se está programando una aplicación muchas veces se necesitan realizar acciones como manipulación de archivos, acceso a datos, conocer el estado del sistema, implementar seguridad, etc. El Framework organiza toda la funcionalidad del sistema operativo en un espacio de nombres jerárquico de forma que a la hora de programar resulta bastante sencillo encontrar lo que se necesita. Para ello, el Framework posee un sistema de tipos universal, denominado Common Type System (CTS). Este sistema permite que el programador pueda interactuar los tipos que se incluyen en el propio Framework (biblioteca de clases de .Net) con los creados por él mismo (clases). De esta forma se aprovechan las ventajas propias de la programación orientada a objetos, como la herencia de clases predefinidas para crear nuevas clases, o el polimorfismo de clases para modificar o ampliar funcionalidades de clases ya existentes.

En la figura 2.21 se detalla las clases de la biblioteca de .NET Framework,

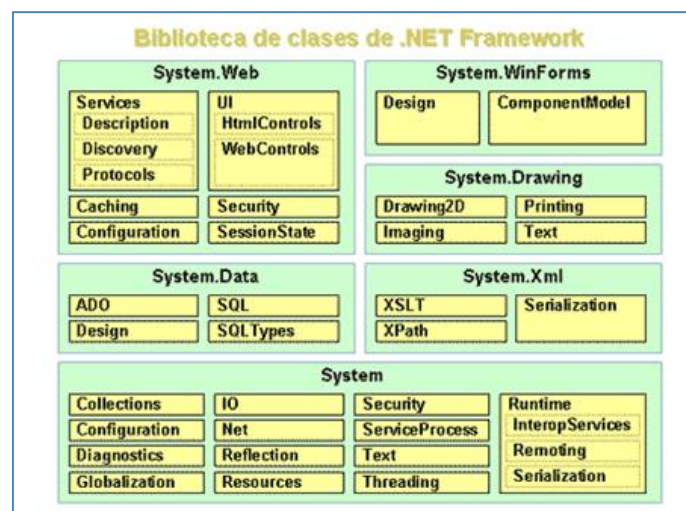


Figura 2.21 Biblioteca de clases de .Net Framework ¹⁷

¹⁷ Biblioteca de clases, [http://msdn.microsoft.com/es-es/library/9s7k7ce5\(v=vs.90\).aspx](http://msdn.microsoft.com/es-es/library/9s7k7ce5(v=vs.90).aspx)

La biblioteca de clases de .Net Framework incluye, entre otros, tres componentes clave:

- **ASP.NET:** para construir aplicaciones y servicios Web.
- **WINDOWS FORMS:** para desarrollar interfaces de usuario.
- **ADO.NET** para conectar las aplicaciones a bases de datos.

La forma de organizar la biblioteca de clases de .Net dentro del código es a través de los espacios de nombres (namespaces), donde cada clase está organizada en espacios de nombres según su funcionalidad. Por ejemplo, para manejar ficheros se utiliza el espacio de nombres System.IO y si lo que se quiere es obtener información de una fuente de datos se utilizará el espacio de nombres System.Data.

La principal ventaja de los espacios de nombres de .Net es que de esta forma se tiene toda la biblioteca de clases de .Net centralizada bajo el mismo espacio de nombres (System). Además, desde cualquier lenguaje se usa la misma sintaxis de invocación, ya que a todos los lenguajes se aplica la misma biblioteca de clases.

- **Ensamblados**

Uno de los mayores problemas de las aplicaciones actuales es que en muchos casos tienen que tratar con diferentes archivos binarios (DLL's), elementos de registro, conectividad abierta a bases de datos (ODBC), etc.

Para solucionarlo el Framework de .Net maneja un nuevo concepto denominado ensamblado. Los ensamblados son ficheros con forma de EXE o

DLL que contienen toda la funcionalidad de la aplicación de forma encapsulada. Por tanto la solución al problema puede ser tan fácil como copiar todos los ensamblados en el directorio de la aplicación.

Con los ensamblados ya no es necesario registrar los componentes de la aplicación. Esto se debe a que los ensamblados almacenan dentro de sí mismos toda la información necesaria en lo que se denomina el manifiesto del ensamblado. El manifiesto recoge todos los métodos y propiedades en forma de meta-datos junto con otra información descriptiva, como permisos, dependencias, etc.

Para gestionar el uso que hacen las aplicaciones de los ensamblados .Net utiliza la llamada caché global de ensamblados (GAC, Global Assembly Cache). Así, .Net Framework puede albergar en el GAC los ensamblados que puedan ser usados por varias aplicaciones e incluso distintas versiones de un mismo ensamblado, algo que no era posible con el anterior modelo COM.

2.2.5.7 .Net Compact Framework

Es una versión reducida del .NET Framework Redistributable, especialmente pensada para ser instalada en dispositivos móviles como Pocket PC's y SmartPhones.

La versión 2.0 incluye importantes mejoras sobre su predecesora, orientadas en la mejora de la productividad, una mayor compatibilidad con desarrollos .NET y ayudas para implementar los recursos de los dispositivos.

Para crear aplicaciones .NET Compact Framework, es necesario Microsoft Visual Studio .NET 2005. Para las aplicaciones móviles, que se ejecutan sobre

Windows Mobile en algún dispositivo tipo Pocket PC o SmartPhone, es necesario tener instalado el .NET Compact Framework en el dispositivo. En la figura 2.22 se detalla los componentes de la arquitectura Compact Framework que se aloja en los dispositivos móviles.

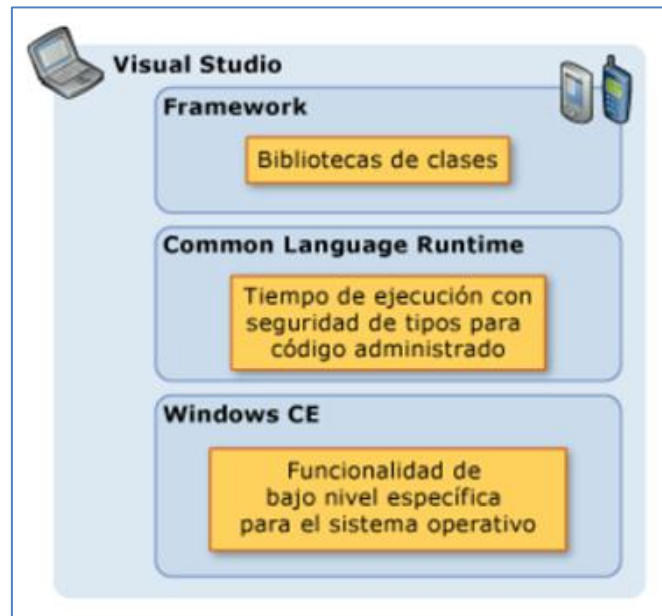


Figura 2.22 Arquitectura de compact Framework .Net ¹⁸

La nueva versión de Compact Framework 2.0 aporta numerosas novedades, se debe recordar que Compact Framework .NET proviene de Framework .NET por lo que muchas de estas novedades y/o mejoras suponen la incorporación de nuevas clases y métodos que provienen de Framework .NET o bien optimizaciones de las clases y métodos existentes.

.NET Compact Framework tiene dos componentes principales:

- El tiempo de ejecución en lenguaje común.

¹⁸ Arquitectura de Compact Framework .Net, [http://msdn.microsoft.com/es-es/library/9s7k7ce5\(v=vs.90\).aspx](http://msdn.microsoft.com/es-es/library/9s7k7ce5(v=vs.90).aspx),

Se encarga de administrar el código en el momento de la ejecución, proporcionando servicios esenciales como la administración de la memoria y de los subprocesos, al mismo tiempo que garantiza la seguridad y la precisión.

- La biblioteca de clases de .NET Compact Framework

Es una colección de clases reutilizables que se puede utilizar para desarrollar aplicaciones de manera fácil y rápida, este marco se ha diseñado pensando en la portabilidad tanto para plataformas Microsoft como de otros fabricantes.

2.2.6 Capa de Interfaz

2.2.6.1 Visual Studio .NET

Visual Studio .NET 2010 es la versión más reciente de esta herramienta que Microsoft distribuye junto a la plataforma que permite construir y desarrollar aplicaciones permitiendo la elección de programación más adecuado.

Características

- **Ejecutivo común:** Todos los lenguajes en la arquitectura .NET utilizan un modulo de ejecución común con librerías comunes.
- **Clases unificadas:** En la plataforma .NET las librerías o clases son comunes para todos los lenguajes.
- **Integración Multilenguaje:** .NET incluye la posibilidad de llamada a métodos de otros objetos desarrollados en otros lenguajes e incluso su herencia. Esto permite desarrollar objetos en el lenguaje mas apropiado para el problema a solucionar.
- **ASP.NET:** Permite crear gráficamente paginas Web utilizando una serie de controles.

- **ADO.NET:** Esta librería proporciona un acceso común a los datos, ya sea en base de datos o XML.
- **Plataforma Abierta:** A este entorno de desarrollo se le puede añadir herramientas o nuevos lenguajes de programación, de tal forma que estén perfectamente integrados en Visual Studio.

2.2.6.2 Lenguaje de desarrollo

Visual C# .NET

Visual C# .NET (C Sharp) esta diseñado para crear una amplia gama de aplicaciones orientadas a objetos que se ejecutan en .NET y el Framework 4.0. Este lenguaje se acopla perfectamente con el desarrollo Web y cuyo objetivo primordial es reducir el coste y los tiempos de desarrollo de los servicios .NET, facilitando la detección de errores y siendo capaz de crear una gran variedad de aplicaciones ya sea de tipo consola, Windows o para la Web.

2.2.7 Capa Lógica

En la actualidad trabajar con n capas es una gran ventaja ya que el código es mas ordenado y puede ser acceder desde cualquier parte de la solución como lo muestra la figura 2.23.

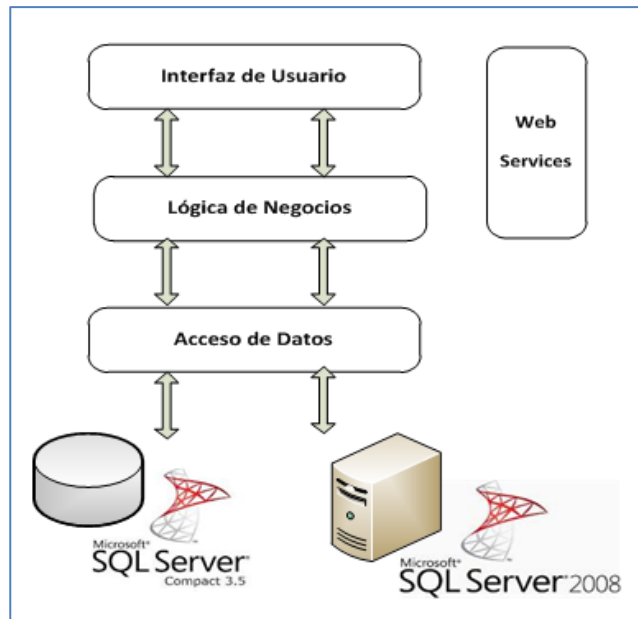


Figura 2.23 Arquitectura general del sistema

2.2.7.1 Web Services

También llamados Servicios Web son un conjunto de protocolos y estándares que sirven para intercambiar datos entre distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma.

Capaz de recibir una petición, activar unos procesos y devolver los resultados. Todo esto, en Internet y a través de protocolos de red (HTTP Hypertext Transfer Protocol, FTP File Transfer Protocol, SMTP Simple Mail Transfer Protocol). La comunicación entre los diferentes entornos del Web Services se realiza mediante XML(Extensible Markup Language).

Para establecer un diálogo coherente entre el WSC (Web Services Cliente), que envía la petición y recibe la respuesta y el WSS (Web Services Servidor), el que ejecuta el proceso y envía la respuesta, se utiliza SOAP (Simple Objecte Access Protocol), que es una codificación basada en XML.

En la figura 2.24 podemos visualizar un Web Services, en vez de obtener peticiones desde un navegador y devolver páginas web como respuesta, recibe peticiones, mediante un mensaje formateado con SOAP, desde otras aplicaciones realiza la labor que le han pedido y devuelve un mensaje de respuesta también con formato SOAP.

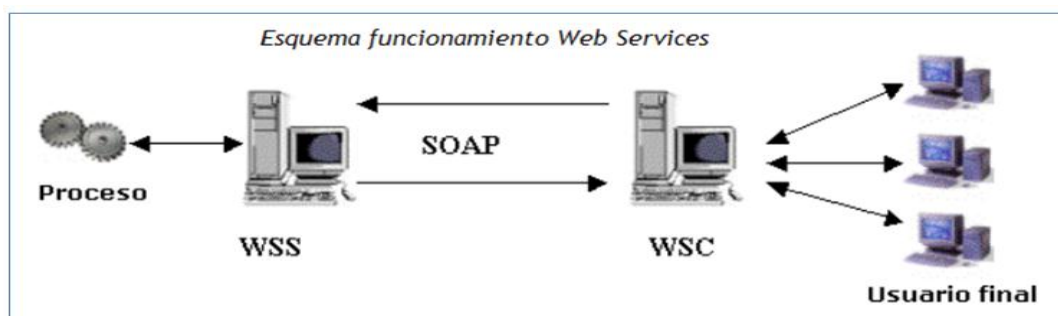


Figura 2.24 Esquema de Web Services ¹⁹

2.2.7.2 Protocolo SOAP

SOAP (Simple Object Access Protocol) Se trata de un protocolo "liviano" que permite la comunicación entre aplicaciones a través de mensajes por medio de Internet, basado en XML y es la base de los Web Services, manteniendo una independencia de la plataforma y del lenguaje.

Características principales del protocolo SOAP

- Mantiene un estándar de invocación de servicios remotos, basado en protocolos estándares de Internet: HTTP (Protocolo de transporte de Hipertexto) para la transmisión y XML (lenguaje de marcado extensible) para la codificación de datos.
- Independencia de plataforma, lenguaje de desarrollo e implementación (modelo de objetos).

¹⁹ Esquema de Web Services, <http://www.geoportal-idec.cat/geoportal/cas/geoserveis/ws-i/>

- No se encuentra fuertemente asociado a ningún protocolo de transporte
- No está atado a ninguna infraestructura de objeto distribuido
- Aprovecha los estándares existentes en la industria.
- Permite la interoperabilidad entre múltiples entornos.

2.2.7.3 Multifuncionalidad entre Sistemas Operativos

Gracias a los protocolos que permiten las transacciones con el Web Services, pueden conectarse a los datos y aplicaciones que pueden estar corriendo en diferentes sistemas operativos, conectados a la web pueden realizar las conexiones respectivas con la base de datos deseada por medio del servicio web el cual es manejado por el lenguaje XML.

2.2.8 Capa de Datos

2.2.8.1 Microsoft SQL Server

SQL Server 2008 R2 es la versión más reciente que ofrece una plataforma de gestión de datos muy óptima, con los más altos niveles de seguridad, fiabilidad y escalabilidad, a la cual se puede acceder desde cualquier lugar y en cualquier momento; además se puede almacenar datos estructurados, semi-estructurados, no estructurados y documentos de forma directamente en el base de datos.

Principales Características

- Gran capacidad de escalabilidad y rendimiento
- Proporciona las capacidades necesarias de programación de base de datos basadas en estándares web.

- Permite la realización de respaldos automáticamente, lo cual asegura la información de la empresa.
- Seguridades de cifrado para mantener la integridad de los datos almacenados.
- Apoyado en el lenguaje estructurado del consulta SQL.

2.2.8.2 Microsoft SQL CE (Compact Edition)

El uso de bases locales en dispositivos móviles se lo puede realizar mediante SQL CE, siendo un motor de base relacional de libre descarga y distribución ofrece unas características especialmente útiles para clientes ligeros. La versión más reciente es SQL Server Compact 3.5 SP2 está diseñado para integrarse con el Microsoft .NET Compact Framework por medio del Visual Studio simplificando el desarrollo de aplicaciones con bases de datos en dispositivos inteligentes. Las bases creadas por el motor se tienen la extensión *.sdf, la cual puede ser abierta en el administrador de bases de SQL 2008.

2.2.9 Dispositivos Móviles

Los dispositivos móviles también conocidos como computadora de mano, son aparatos de pequeño tamaño, en la actualidad con muy buenas capacidades de procesamiento, con conexión permanente o intermitente a una red, con memoria limitada o escalable.

2.2.9.1 Pocket PC

En la figura 2.25 se visualizan dispositivos móviles con Windows Mobile que permiten instalar aplicaciones de terceros.



Figura 2.25 Pocket Pc / Smartphone ²⁰

Un Pocket PC no es más que un asistente personal digital o PDA con el sistema operativo de Microsoft. Dicha empresa producto de la necesidad de un sistema operativo para ciertos dispositivos que no eran computadores personales, se dispuso a desarrollar uno con las características más relevantes de Windows desde su versión 3.1, dando surgimiento por medio de esa iniciativa al sistema operativo Windows CE (Compact Edition o edición compacta).

La plataforma Pocket PC es la denominación empleada en dispositivos de pequeño tamaño, con capacidades de computador personal. La base de su funcionamiento es un potente y compacto sistema operativo, denominado oficialmente Microsoft Windows CE. Este dio lugar a un sistema operativo denominado comúnmente como Pocket PC 2000 y, posteriormente, a una edición más evolucionada llamada Pocket PC 2002 hasta contar ahora con una

²⁰ <http://www.xataka.com/moviles/htc-touch2-un-poco-mas-de-lo-que-esperabamos>

versión denominada Windows Mobile. Es así que se adoptó el nombre de Pocket PC a la implantación del sistema operativo Windows CE sobre un dispositivo de mano PDA, esto sí en conjunto con varias aplicaciones de uso cotidiano en una PC.

Por otra parte el término PDA fue establecido por John Sculley, un alto ejecutivo de Apple en 1992, durante una exposición en Las Vegas donde anunciaba la intención de Apple por desarrollar un computador de pequeñas dimensiones en el que el usuario utilizaba un lápiz y una pantalla táctil para introducir la información dejando de lado el uso del teclado.

Smartphone es un teléfono celular inteligente con características similares a las de una computadora personal, que soporta tener correo electrónico, la funcionalidad completa de un organizador personal, la instalación de aplicaciones propias del fabricante desarrolladas por un tercero para incrementar sus funcionalidad, acceso a diferentes servicios vía web.

Lo interesante de estos dispositivos que pueden realizar varias tareas al mismo tiempo como recibir llamadas, revisar la agenda mientras se ve unos videos en Media Player, o mientras se sincroniza el dispositivo con otros, y todo esto sin necesidad de interrumpir alguna de las tareas.

Características importantes de un smartphone

- Pantalla táctil y el teclado completo
- Permite de instalación de programas de terceros
- Permite edición y almacenamiento de videos y fotos.

- Sistemas Operativos que se puede instalar según la casa comercial: iOS 5, Android 2.3 Gingerbread, Windows Phone 7.5 Mango , Blackberry 7 OS
- Lectura de archivos en diversos formatos de acuerdo a las aplicaciones previamente instaladas
- Batería con capacidad de ofrecer tiempos de uso bastante prolongados

2.2.9.2 Windows Mobile 6.5

Windows Mobile es un sistema operativo móvil compacto desarrollado por Microsoft, y diseñado para su uso en teléfonos inteligentes (Pocket Pc, Smartphones) y otros dispositivos móviles. Windows Mobile hace parte de los sistemas operativos con interfaz natural de usuario y se basa en el núcleo del sistema operativo Windows CE y cuenta con un conjunto de aplicaciones básicas utilizando las API de Microsoft Windows, además es compatible con dos plataformas de programación más populares y modernas.

2.2.9.3 Windows CE

Los ordenadores de bolsillo basados en este sistema operativo suelen tener algo más de velocidad de procesador, algo más de memoria y mayor resolución de pantalla, que es en color, pero al tratarse de una versión reducida de Windows carga más el sistema, suelen ser de dimensiones mayores y las baterías duran menos. Por otra parte, si se es usuario de PC, los programas que se usan en el Smart phone son gemelos de los que operan en el PC. El correo electrónico, la agenda y el calendario funcionan mediante Outlook CE.

Los archivos Excel y Word son perfectamente intercambiables con los del PC. El navegador es Internet Explorer. También, al ser más potentes, pueden ejecutar aplicaciones multimedia como la reproducción de MP3 o grabar sonido. El sistema también se puede actualizar mediante conexión al ordenador usando el puerto serie, USB o infrarrojos.

2.2.10 Nunit Testeador de pruebas unitarias.

NUnit es un framework de pruebas unitarias, la versión de producción actual, la versión 2.6, es la séptima versión mayor de este xUnit herramienta basada en la unidad de pruebas para Microsoft. NET, y esta escrito enteramente en C #. que ofrece las funcionalidades necesarias para implementar pruebas en un proyecto. Además provee una interfaz grafica para ejecutar y administrar las mismas.

Consiste en un conjunto de metaatributos y aserciones que permiten probar los métodos de una clase especificada.

Para poder utilizar Nunit 2.2 debe ser referenciada la dll, nunit.framework.dll dentro del proyecto que contiene el código a probar.

2.2.10.1 Prueba Unitaria

Es un procedimiento usado para validar que un módulo de código fuente funciona apropiadamente, siendo el principal beneficio el que permite aislar segmentos del programa que pueden ser probados, y validados: esto hace que el software que esta siendo desarrollado tenga mayor calidad si se corrigen los bugs que esta herramienta puede encontrar.

CAPÍTULO 3

IMPLEMENTACIÓN DEL SISTEMA

3.1 Planificación

Se establecieron reuniones de trabajo con la empresa auspiciante con la finalidad de conocer el detalle de la solución requerida, con lo cual se pudo determinar el ámbito de la misma estableciendo a su vez prioridades dentro del desarrollo y los entregables durante el proceso de producción. Se pudo establecer un cronograma de actividades con las tareas y los hitos a cumplir.

Tabla 3.1 Tabla de planificación del proyecto

Nombre de tarea	Duración
Seguro Total	134 días
Planificación	8 días
Levantamiento de Requerimientos	3 días
Historias de Usuarios	5 días
Diseño	38 días
Modelamiento de Arquitectura	5 días
Diagrama de infraestructura	3 días
Diagrama de procesos	5 días
Flujo de negocio	2 días
Flujo de sistema	3 días
Diagramas de secuencia	2 días
Diagrama de componentes	3 días
Diagrama de navegación	3 días
Diagrama de estados	3 días
Diagrama de base de datos	3 días
Diseño de objetos	3 días
Diagrama de clases	3 días
Construcción	78 días
Preparación de entorno	2 días
Diseño de soluciones	3 días
Desarrollo de aplicación móvil	68 días
Iteración 1	10 días
Acceso	1 día
Bien a Asegurar	3 días
Coberturas	3 días
Accesorios	3 días
Iteración 2	8 días
Cliente	3 días
Forma de Pago	3 días
Datos generales del bien	2 días
Iteración 3	9 días
Carga de imagen	3 días
Carga de catálogos	4 días
Envío de solicitudes	2 días
Desarrollo de aplicación Web	63 días

Iteración 1	11 días
Acceso	1 día
Administración	10 días
Iteración 2	10 días
Verificación de Información de Solicitudes	5 días
Clientes	5 días
Iteración 3	5 días
Reportes	5 días
Implantación	62 días
Iteración 1	10 días
Pruebas	2 días
Certificación	1 día
Ajustes y retroalimentación	3 días
Creación de manuales	2 días
Capacitación	1 día
Puesta en producción	1 día
Iteración 2	10 días
Pruebas	2 días
Certificación	1 día
Ajustes y retroalimentación	3 días
Creación de manuales	2 días
Capacitación	1 día
Puesta en producción	1 día
Iteración 3	10 días
Pruebas	2 días
Certificación	1 día
Ajustes y retroalimentación	3 días
Creación de manuales	2 días
Capacitación	1 día
Puesta en producción	1 día

En la figura 3.1 se puede visualizar la planificación del proyecto basada en la metodología.

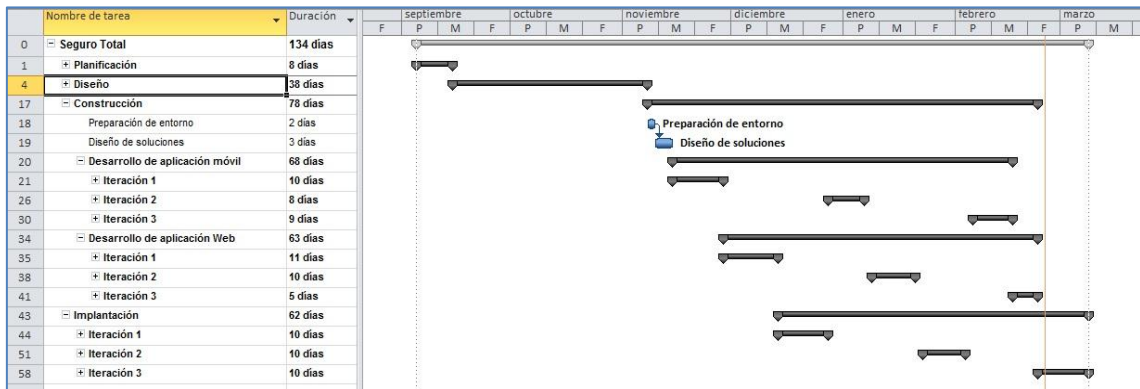


Figura 3.1 Cronograma del proyecto

3.1.1 Historias de Usuario

En conjunto con los auspiciantes del sistema se generó las siguientes historias de usuarios que reflejan los requerimientos y necesidades que debe cumplir el aplicativo.

Tabla 3.1 Historia de usuario, Acceso al sistema

Historia de Usuario	
Número: 1	Usuario: Vendedor, Analista, Supervisor, Emisor, Administrador
Nombre historia: Acceso al Sistema	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 2	Iteración asignada: 1
Programador responsable: Fernando Morales – Luis Salazar	
Descripción: Los usuarios dispondrán de una pantalla de acceso al sistema por medio de la cual se solicitará un código de usuario y contraseña tanto en el aplicativo web como en el aplicativo móvil con la finalidad de que sus operaciones y registros se los asignen a quien corresponda. En el aplicativo web restringirá el acceso si la contraseña es incorrecta y en el aplicativo móvil se impedirá el envío de información de solicitudes registradas si el usuario no envía las credenciales correctas.	

Tabla 3.2 Historia de usuario, Registro de datos del bien.

Historia de Usuario	
Número: 2	Usuario: Vendedor
Nombre historia: Registro de datos del bien (vehículo)	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 1	Iteración asignada: 1, 3
Programador responsable: Fernando Morales – Luis Salazar	
<p>Descripción: El vendedor registra de forma manual escribiendo la información correspondiente al bien (vehículo) en los campos definidos para la captura de datos, además de las fotografías correspondientes que garanticen el estado actual del bien; dicha información deberá registrarse en la base de datos del dispositivo móvil.</p> <p>La información puede ser objeto de modificación siempre y cuando la solicitud a la que hace referencia no se haya enviado para su posterior aprobación. Adicionalmente se registrará a que coberturas se suscribiría el cliente y los accesorios que posea el bien identificando el estado material de cada accesorio.</p> <p>Luego del registro en la base del desmotivo móvil se borrara la información de la solicitud realizada.</p>	
<p>Observaciones: Este registro se lo hará desde el dispositivo móvil. La carga de imágenes se asignará a la iteración 3.</p>	

Tabla 3.3 Historia de usuario, Administración del aplicativo web

Historia de Usuario	
Número: 3	Usuario: Administrador
Nombre historia: Administración aplicativo web	
Prioridad en negocio: Baja	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Fernando Morales – Luis Salazar	
Descripción: El usuario con perfil administrador deberá contar con las opciones que le permita hacer el mantenimiento de los catálogos usados así también contar con administración de usuarios en el sistema.	

Tabla 3.4 Historia de usuario, Registro de datos del cliente

Historia de Usuario	
Número: 4	Usuario: Vendedor
Nombre historia: Registro de datos del cliente	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 2	Iteración asignada: 2
Programador responsable: Fernando Morales – Luis Salazar	
Descripción: El vendedor registra de forma manual escribiendo la información del usuario en los campos definidos para la captura de datos del cliente. Dicha información deberá registrarse en la base de datos del dispositivo móvil, además esta puede ser objeto de modificación siempre y cuando la solicitud a la que hace referencia no se haya enviado para su posterior aprobación.	
Observaciones: Ésta información es producto de verificación la web.	

Tabla 3.5 Historia de usuario, Registro de forma de pago

Historia de Usuario	
Número: 5	Usuario: Vendedor
Nombre historia: Registro de forma de pago	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 2	Iteración asignada: 2
Programador responsable: Fernando Morales – Luis Salazar	
Descripción: El vendedor registra la forma de pago una vez que el cliente ha aceptado los valores de aseguramiento. Éste registro contempla información si el pago es de contado o a crédito especificando información adicional de ser el caso.	
Observaciones: Ésta información es producto de verificación en la web.	

Tabla 3.6 Historia de usuario, Registro de datos específicos del bien

Historia de Usuario	
Número: 6	Usuario: Vendedor
Nombre historia: Registro de datos específicos del bien	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 2	Iteración asignada: 2
Programador responsable: Fernando Morales – Luis Salazar	
Descripción: El vendedor registra información específica del bien asegurado como el número de motor, número de chasis, color y placa del vehículo o número de factura en caso de ser vehículo nuevo.	
Observaciones: Ésta información es producto de verificación en la web.	

Tabla 3.7 Historia de usuario, Verificación, análisis y aprobación de solicitudes

Historia de Usuario	
Número: 7	Usuario: Analista
Nombre historia: Verificación, Análisis y Aprobación de solicitudes	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Puntos estimados: 2	Iteración asignada: 2
Programador responsable: Fernando Morales – Luis Salazar	
<p>Descripción: El analista deberá contar con un listado de solicitudes las mismas que serán aquellas que fueron asignadas a su perfil, de este listado este deberá seleccionar solo una a la vez para su posterior análisis que consistirá en la verificación de la información teniendo la posibilidad de modificar y actualizar todos los datos de cliente y solicitud. Además dependiendo del criterio del analista este podrá decidir si la solicitud continúa en el proceso o si la operación se da por terminada. En caso de que si continúe es decir sea aprobada se remitirá al perfil emisor para imprimir el certificado.</p>	

Tabla 3.8 Historia de usuario, Carga de catálogos

Historia de Usuario	
Número: 8	Usuario: Vendedor
Nombre historia: Carga de catálogos	
Prioridad en negocio: Baja	Riesgo en desarrollo: Bajo
Puntos estimados: 2	Iteración asignada: 3
Programador responsable: Fernando Morales – Luis Salazar	
<p>Descripción: Se deberá disponer de la opción de sincronización de los catálogos en el dispositivo móvil.</p>	
<p>Observaciones: Esta actualización se lo hará desde el dispositivo móvil</p>	

Tabla 3.9 Historia de usuario, Envío de información

Historia de Usuario	
Número: 9	Usuario: Vendedor
Nombre historia: Envío de la información	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 2	Iteración asignada: 3
Programador responsable: Fernando Morales – Luis Salazar	
<p>Descripción: El vendedor una vez que la información acerca de la solicitud este completa en su totalidad procederá al envío de la misma desde el dispositivo móvil hacia la aplicación de administración. Como el vendedor puede estar ubicado en cualquier parte del mundo se deberá diseñar un mecanismo que permita enviar los datos de aplicación a aplicación recibiendo una señal de confirmación que garantice que dicha operación se haya efectuado con éxito. Además cuando la información de la solicitud haya llegado a su destino entrara en un proceso de asignación por carga en la cual se asignará al usuario de perfil analista para su posterior verificación</p>	

Tabla 3.10 Historia de usuario, Listado impresión

Historia de Usuario	
Número: 10	Usuario: Emisor
Nombre historia: Listado impresión	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 2	Iteración asignada: 3
Programador responsable:	
<p>Descripción: El emisor debe contar con un listado de solicitudes las mismas que serán aquellas que anteriormente fueron aprobadas de las cuales este podrá seleccionar solo una a la vez. El sistema cuenta con la posibilidad de generar un reporte con el formato correspondiente al certificado de aprobación.</p>	

Tabla 3.11 Historia de usuario, Reportes gerenciales

Historia de Usuario	
Número: 11	Usuario: Supervisor
Nombre historia: Reportes Gerenciales	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 1	Iteración asignada: 3
Programador responsable:	
<p>Descripción: El usuario con perfil supervisor deberá contar con una opción que le permita obtener los siguientes reportes de nivel gerencial:</p> <p>Reporte comparativo de número de solicitudes aprobadas por vendedor.</p> <p>Reporte comparativo de número de solicitudes generadas por matriz, sucursal, provincia, ciudad</p>	

3.1.2 Plan de entregas

Para la elaboración del Plan de Entregas del presente proyecto y aplicando los parámetros de desarrollo bajo la metodología XP se considera dentro de la planificación se estableció 3 entregables del software después de agrupar las historias de usuarios y estableciendo el cronograma de actividades. Dichos entregables contemplan pruebas de usuarios, certificaciones, documentación y capacitación.

Tabla 3.12 Tabla de identificación de entregables del proyecto

Nombre de tarea
Implantación
Iteración 1
Pruebas
Certificación
Ajustes y retroalimentación
Creación de manuales
Capacitación
Puesta en producción
Iteración 2
Pruebas
Certificación
Ajustes y retroalimentación
Creación de manuales
Capacitación
Puesta en producción
Iteración 3
Pruebas
Certificación
Ajustes y retroalimentación
Creación de manuales
Capacitación
Puesta en producción

3.1.3 Velocidad del Proyecto

En base a estimados se establecieron tiempos de desarrollo para las diferentes historias de usuarios los mismos que se detallan dentro del cronograma diseñado en la planificación.

3.1.4 Iteraciones

Una vez analizadas las historias de usuarios y en común acuerdo con los auspiciantes de la tesis se estableció el agrupamiento de las mismas para la generación de 3 iteraciones dentro del proyecto que tendrán sus entregables relacionados. En el cronograma planificado se contempla las actividades de cada iteración, así también en cada historia de usuario se identifica en que iteración será desarrollado luego del análisis correspondiente.

3.1.5 Rotaciones

El código de las aplicaciones del proyecto será de mutuo conocimiento de los desarrolladores de la tesis pues contempla además la programación en pareja, lo que conlleva un desprendimiento de propiedad exclusiva del código.

3.1.6 Reuniones

En la planificación se ha contemplado las reuniones con los usuarios del sistema tanto para la ejecución de pruebas como para la certificación del software. Otras reuniones planificadas serán las de capacitación donde se pondrá en conocimiento a los usuarios la funcionalidad de las aplicaciones.

3.2 Diseño

Para el diseño de las aplicaciones del proyecto a más de basarse en la metodología extrema, también se detalla otros tipos de diagramas no contemplados dentro de la misma que los detallaremos a continuación.

3.2.1 Metáfora del Sistema

El sistema contempla el desarrollo de dos aplicaciones una web y otra móvil que mediante el uso de servicios web interactuarán entre sí.

La aplicación móvil establece el uso de una base de datos local con SQL Server CE en el dispositivo que permita el registro de solicitudes de seguros vehiculares con la información del bien a asegurar más datos adicionales del cliente, dichos datos serán enviados a través de un servicio web que almacenará dicha información en la base de datos centralizada en SQL Server.

El aplicativo web se encargará de hacer la lectura de los datos registrados en la base de datos a través de servicios web y de ejecutar acciones como la verificación de los datos, la aprobación de la solicitud, impresión de solicitudes aprobadas y la emisión de reportes.

Para apoyar la metáfora se hizo uso del siguiente diagrama de infraestructura que se detalla en la figura 3.2 donde se contempla los servidores involucrados los posibles canales de comunicación y los clientes asociados a la aplicación:

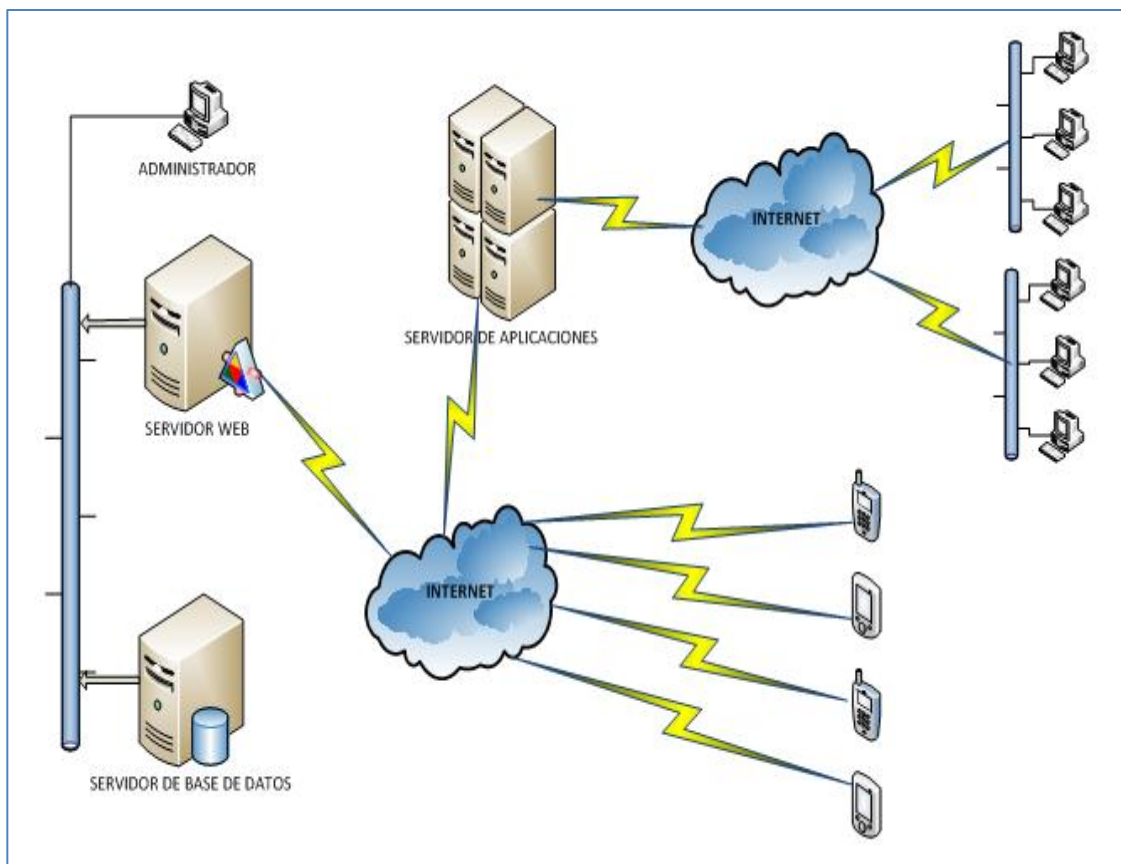


Figura 3.2 Diagrama de infraestructura de la solución

Adicionalmente el diagrama que muestra la arquitectura del sistema en la figura 3.3 contemplando las 2 aplicaciones y su interacción con los servicios web que se desarrollarán en el sistema:

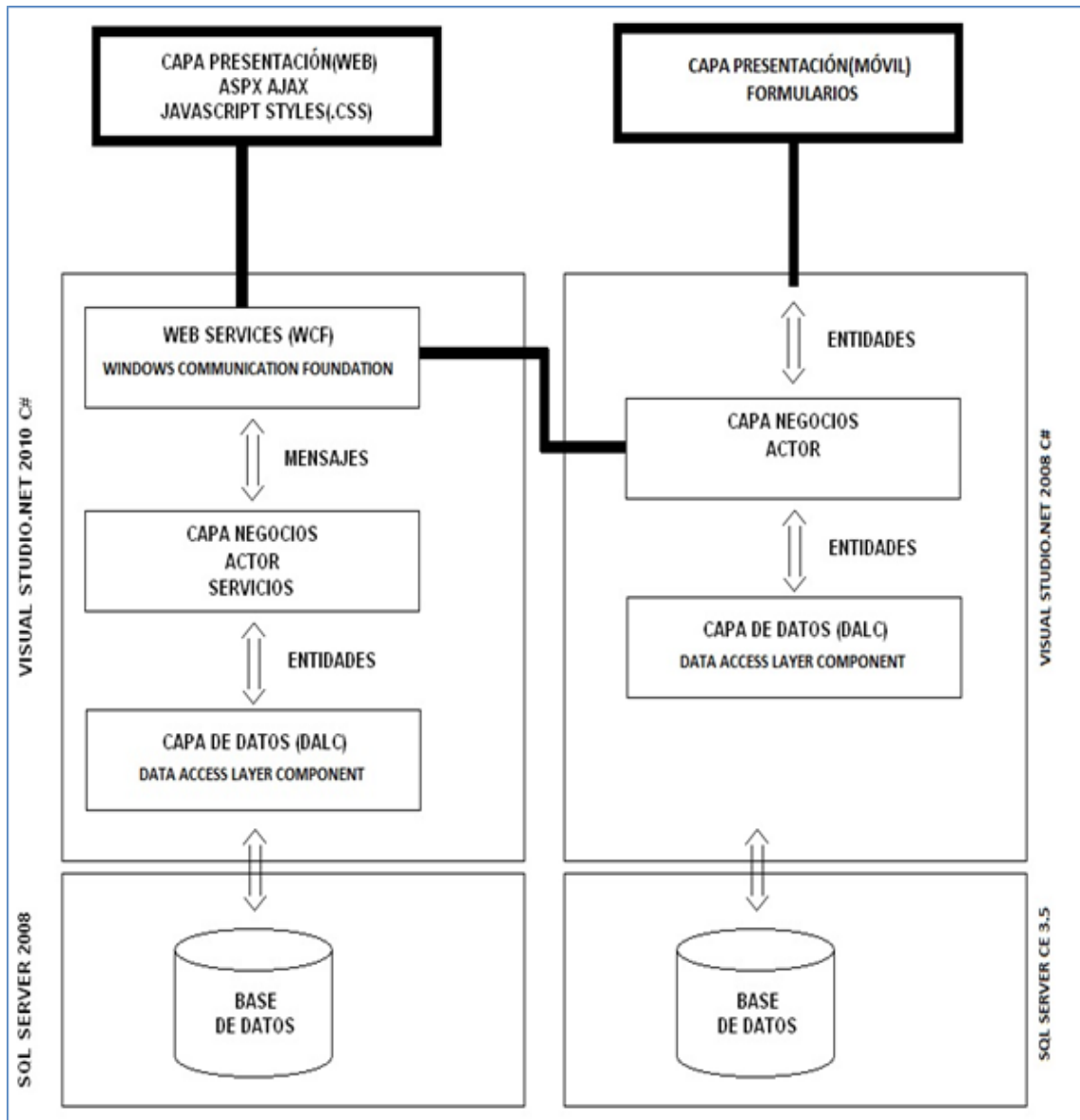


Figura 3.3 Diagrama de arquitectura de la solución

3.2.2 Tarjetas CRC

Tomando en cuenta cada una de las historias de usuario se ha contemplado la implementación de las siguientes tarjetas de tipo Clase, Responsabilidad y Colaboración:

Tabla 3.12 Tarjeta CRC, CRC Usuario

La tarjeta CRC Usuario permite la creación, actualización, eliminación y recuperación de usuarios

Usuario	
Responsabilidad	Colaboración
1.- Registrar (login, password, nombre, correo, departamento, agencia, perfil)	
2.- Validar Credenciales (login, password)	Perfil
3.- Cambiar credenciales (login, password)	
4.- Actualizar usuario ((login, password, nombre, correo, departamento, agencia, perfil)	Perfil
5.- Recuperar usuario (login)	Perfil, Agencia

Tabla 3.13 Tarjeta CRC, CRC Perfil usuario

La tarjeta CRC Perfil usuario permite la parametrización de los actores que interactúan con el sistema

Perfil Usuario	
Responsabilidad	Colaboración
1.- Registrar perfiles	Perfil
2.- Registrar perfiles por usuario	Usuario, Perfil
3.- Actualizar perfiles por usuario	Usuario, Perfil
4.- Asignar Agencia	Usuario, Agencia, Ciudad

Tabla 3.14 Tarjeta CRC, CRC Bien

La tarjeta CRC Bien permite la creación, modificación, eliminación y recuperación de un bien

Bien	
Responsabilidad	Colaboración
1.- Registra bien	Marca, Modelo, Año, estado, Tasa
2.- Modificación del Bien	
3.- Eliminación Bien	
4.- Recuperar Bien	
5.- Registra Coberturas	Cobertura
6.- Registra Accesorio	Accesorio
7.- Registrar Fotografías Bien	Fotografía

Tabla 3.15 Tarjeta CRC, CRC Catálogo

La tarjeta CRC Catálogo permite la creación, modificación, eliminación y recuperación de catálogos que alimentara la información al sistema

Catalogo	
Responsabilidad	Colaboración
1.- Registro ítem	Catálogo
2.- Modificación ítem	
3.- Eliminación ítem	
4.- Recuperar ítem	

Tabla 3.16 Tarjeta CRC, CRC Cliente

La tarjeta CRC Cliente permite la creación, modificación, eliminación y recuperación de clientes

Cliente	
Responsabilidad	Colaboración
1.- Registra	Tipo Identificación, Nacionalidad, Estado Civil, Provincia, Ciudad, Genero
2.- Modificación	
3.- Eliminación	
4.- Recuperar Cliente	Tipo Identificación, Nacionalidad, Estado Civil, Provincia, Ciudad, Genero

Tabla 3.17 Tarjeta CRC, CRC Solicitud

La tarjeta CRC Solicitud permite la creación, modificación, eliminación de la solicitud

Solicitud	
Responsabilidad	Colaboración
1.- Registra	Usuario, Bien, Accesorio, Cobertura, Forma Pago, Medio Pago, Fotografía
2.- Modificación	
3.- Eliminación	
4.- Recuperar	

Tabla 3.18 Tarjeta CRC, CRC Análisis

La tarjeta CRC Análisis permite la creación, modificación, eliminación de la solicitud

Análisis	
Responsabilidad	Colaboración
1.- Recuperar	Solicitud, Cliente, Bien, Accesorio, Cobertura, Fotografía
2.- Modificación	
3.- Actualización	

Tabla 3.19 Tarjeta CRC, CRC Reporte

La tarjeta CRC Reporte permite la consulta y generación de un reporte en base a la información de la solicitud

Reporte	
Responsabilidad	Colaboración
1.- Recuperar	Solicitud, Cliente, Bien, Accesorio, Cobertura, Fotografía
2.- Visualizar	

3.2.3 Soluciones puntuales

Se desarrollarán 3 soluciones dentro del sistema la de los servicios web, la de la página web y la del aplicativo móvil.

La de los servicios web se encargará de hacer la conexión con la base de datos centralizada permitiendo:

- Almacenar solicitudes generadas desde el dispositivo móvil.
- Registro de clientes que fueron levantados con el dispositivo móvil.

- Extraer solicitudes para su verificación.
- Extraer información para uso en reportes.
- Almacenar datos de catálogos y registro de usuarios.

La solución de la página web permitirá establecer una presentación de los datos que son devueltos por el servicio web interactuando con el usuario para ejecutar tareas como:

- Verificación de solicitudes y clientes.
- Impresión de solicitudes aprobadas.
- Despliegue de reportes para los usuarios con el perfil correspondiente.
- Generar interfaces que permitan el mantenimiento de catálogos.
- Generar la interface para el registro de usuarios en sistema.

El aplicativo móvil es el que concentra la mayor funcionalidad o la más importante al permitir capturar información de las solicitudes y permitiendo además:

- Sincronizar catálogos de la base centralizada a través de los servicios web con la base local.
- Registrar solicitudes contemplando información del bien a asegurar, el cliente asociado, las coberturas y formas de pago.
- Hacer la toma de fotografías del bien a asegurar.
- Envío de la información recabada por medio de los servicios web para su uso en el aplicativo web.

3.2.4 Funcionalidad Mínima

Como parte de la metodología de desarrollo el sistema está diseñado para cubrir las historias del usuario sin contemplar funcionalidades adicionales no propuestas en el levantamiento de las mismas.

En la figura 3.4 y 3.5 se detallan el flujo que debe realizar una solicitud en el sistema Seguro Total

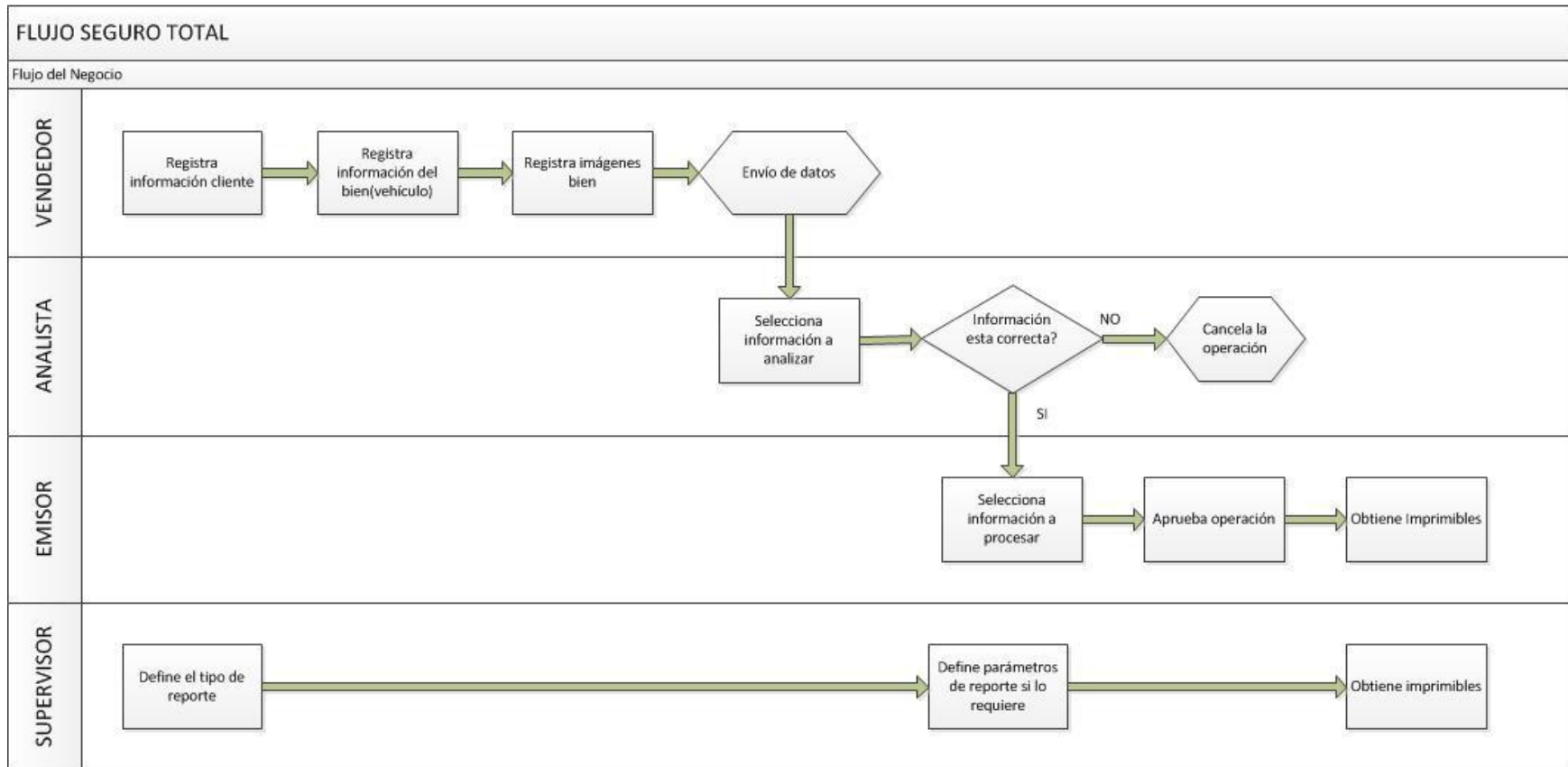


Figura 3.4 Flujo del negocio

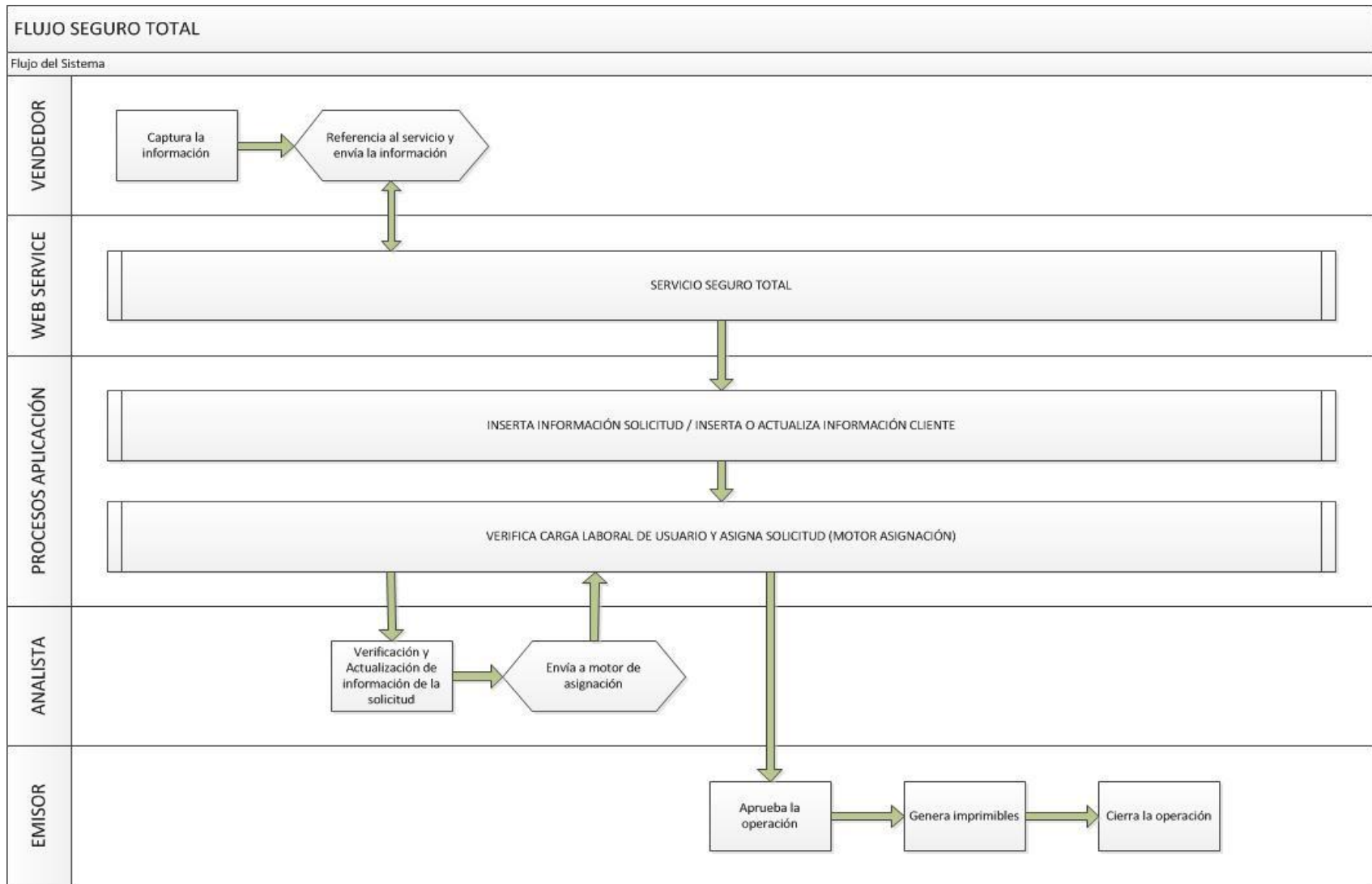


Figura 3.5 Flujo del sistema

3.2.5 Reciclaje

El sistema está diseñado para reutilizar al máximo el código generado concentrando funcionalidad en la capa de servicios y haciendo de uso de secciones de código en su núcleo que permiten que no se repita código ya generado. A nivel general la solución tiene una arquitectura que obliga a que se use la capa de actores para el acceso a la base de datos devolviendo entidades de negocio con las cuales se puede completar cualquier proceso.

3.2.6 Diagramas de secuencias

Como diagramas adicionales a los propuestos por la metodología de programación extrema se ha decidido implementar diagramas de secuencias de las principales operaciones que cubre el aplicativo brindándonos una idea clara de cuales son los diferentes pasos por los que un proceso de negocio debe atravesar para cubrir su objetivo y dar así cumplimiento al alcance propuesto en una historia de usuario.

En la figura 3.6 se detalla el diagrama de secuencia de registros de cliente desde la entrevista con el vendedor hasta guardar la información en la base de datos. Además en la figura 3.7 y 3.8 se detallan la secuencia que se realiza en el proceso de registro de una solicitud.

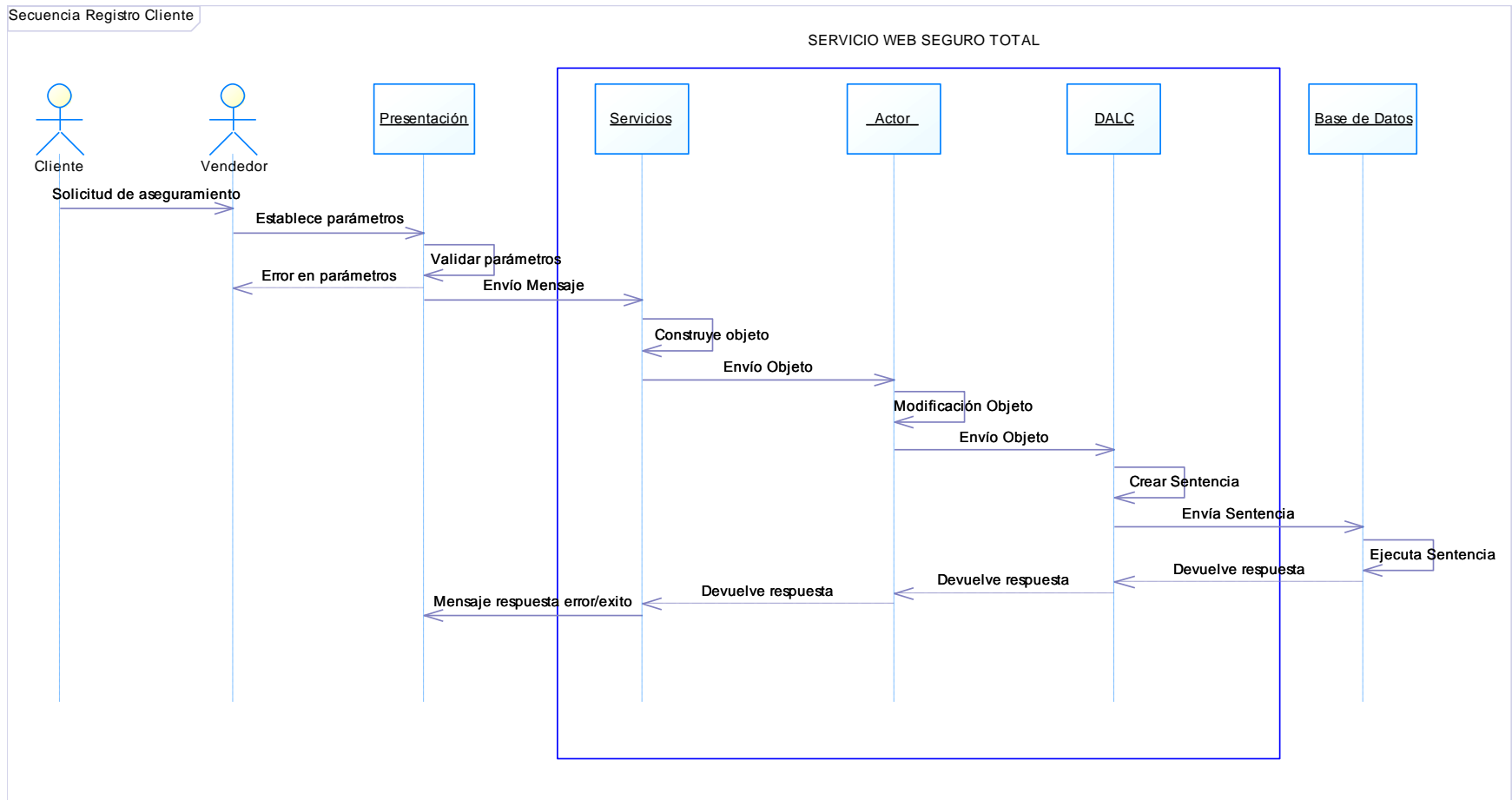


Figura 3.6 Diagrama de secuencia de registro de clientes

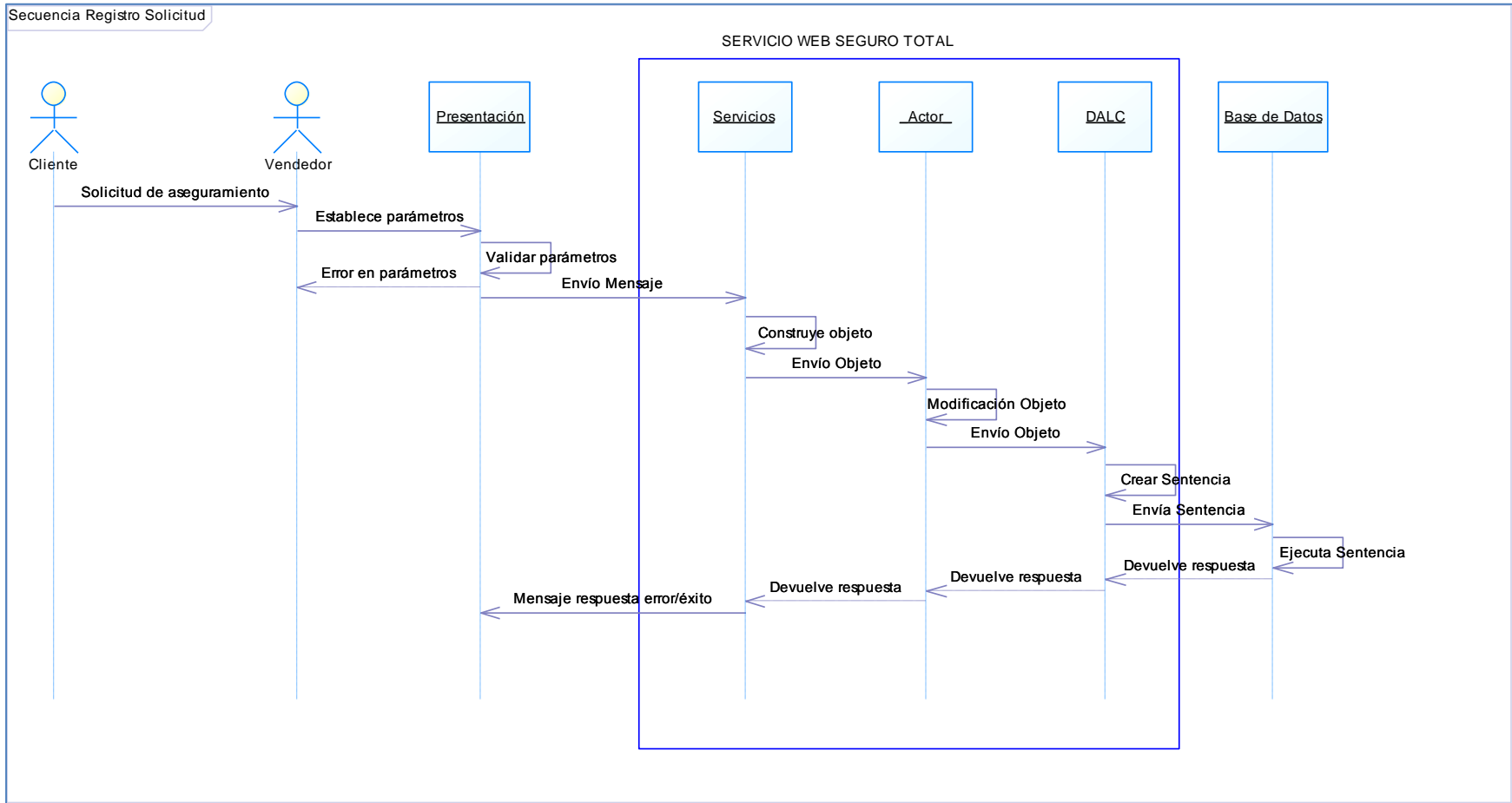


Figura 3.7 Diagrama de secuencia de registro de solicitudes

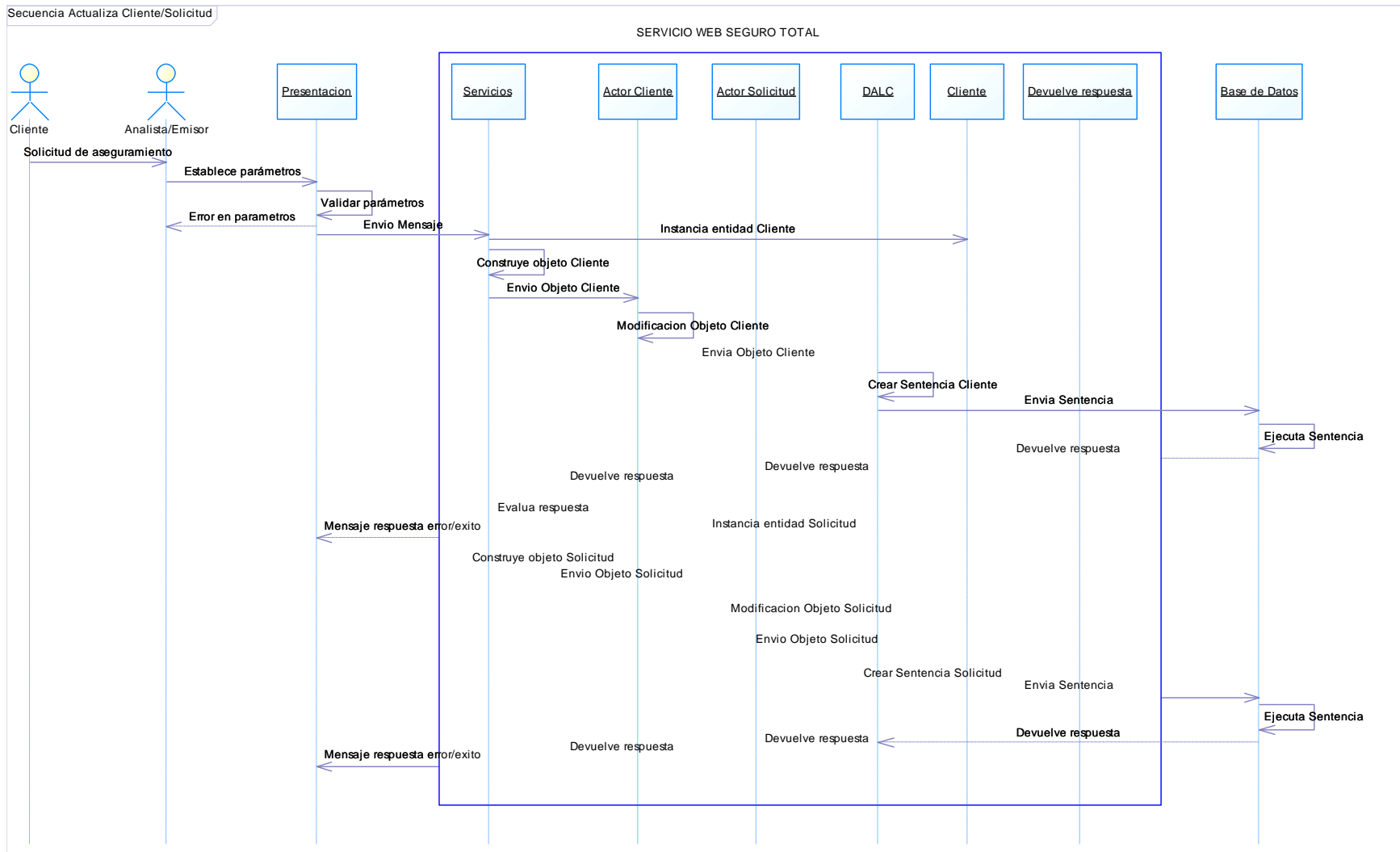


Figura 3.8 Diagrama de secuencia de actualización de clientes y solicitudes

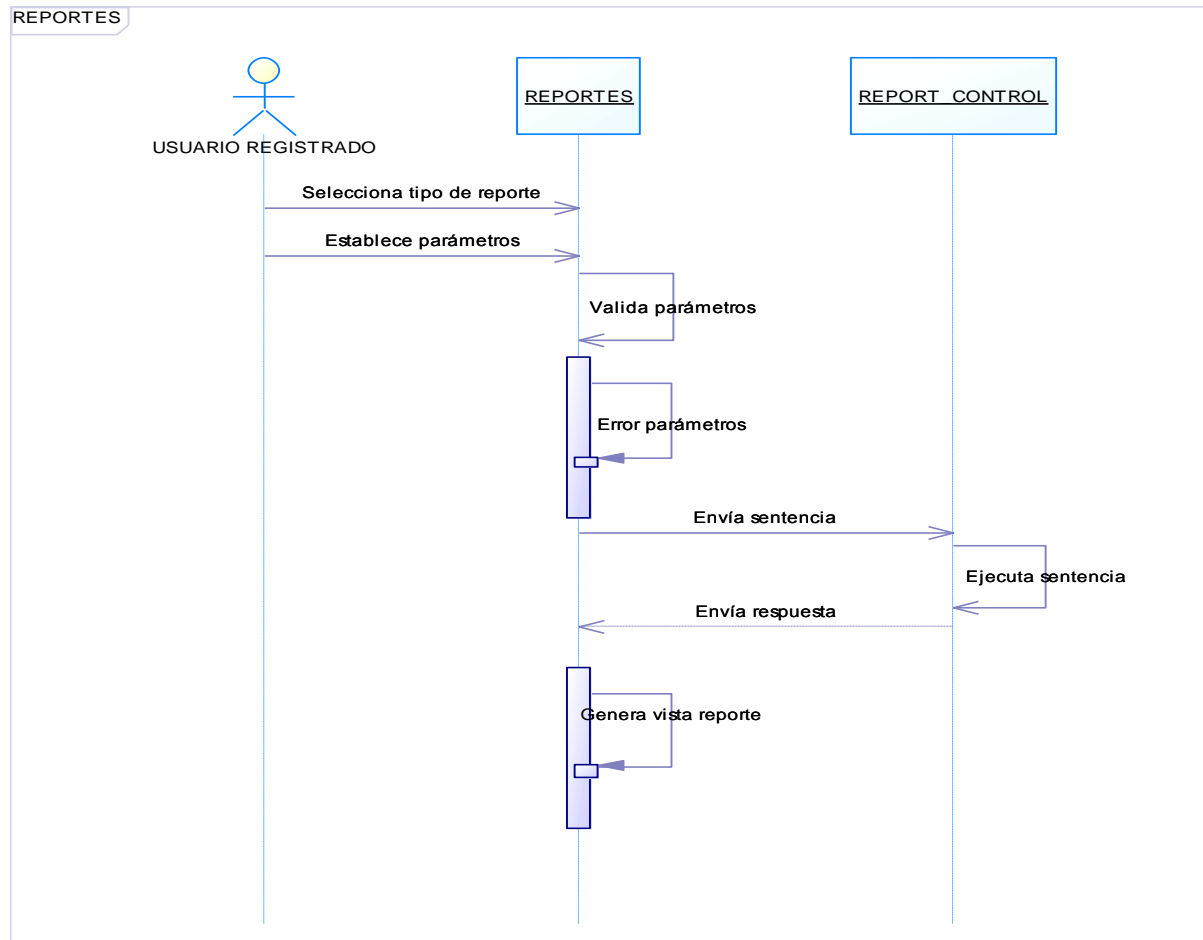


Figura 3.9 Diagrama de secuencia de generación de reportes

3.2.6.1 Diagrama de componentes

Permite visualizar los aspectos físicos del sistema.

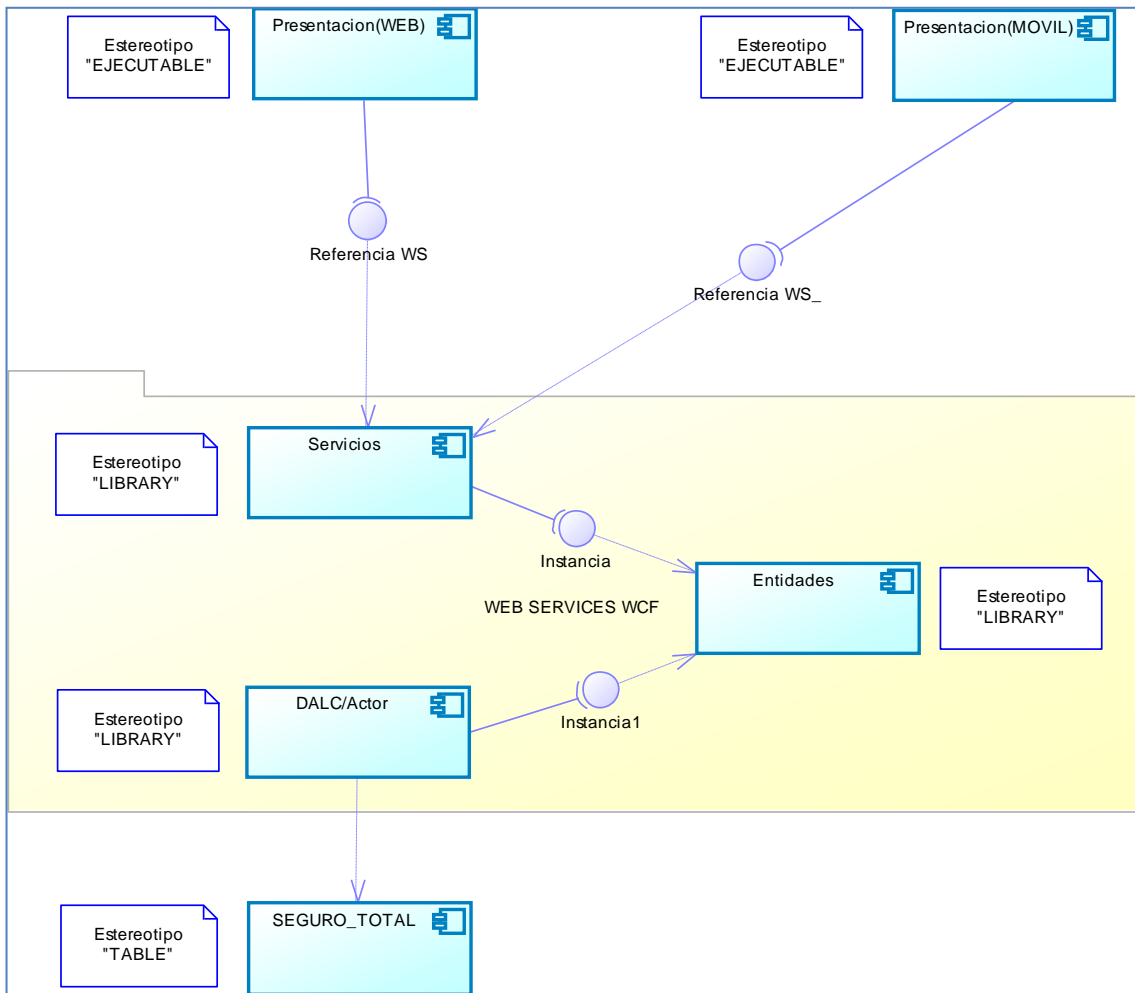


Figura 3.10 Diagrama de componentes

3.2.6.2 Diagrama de estados

A través del diagrama de la figura 3.11 se identifica los diferentes estados por los que atravesaría una solicitud en el sistema.

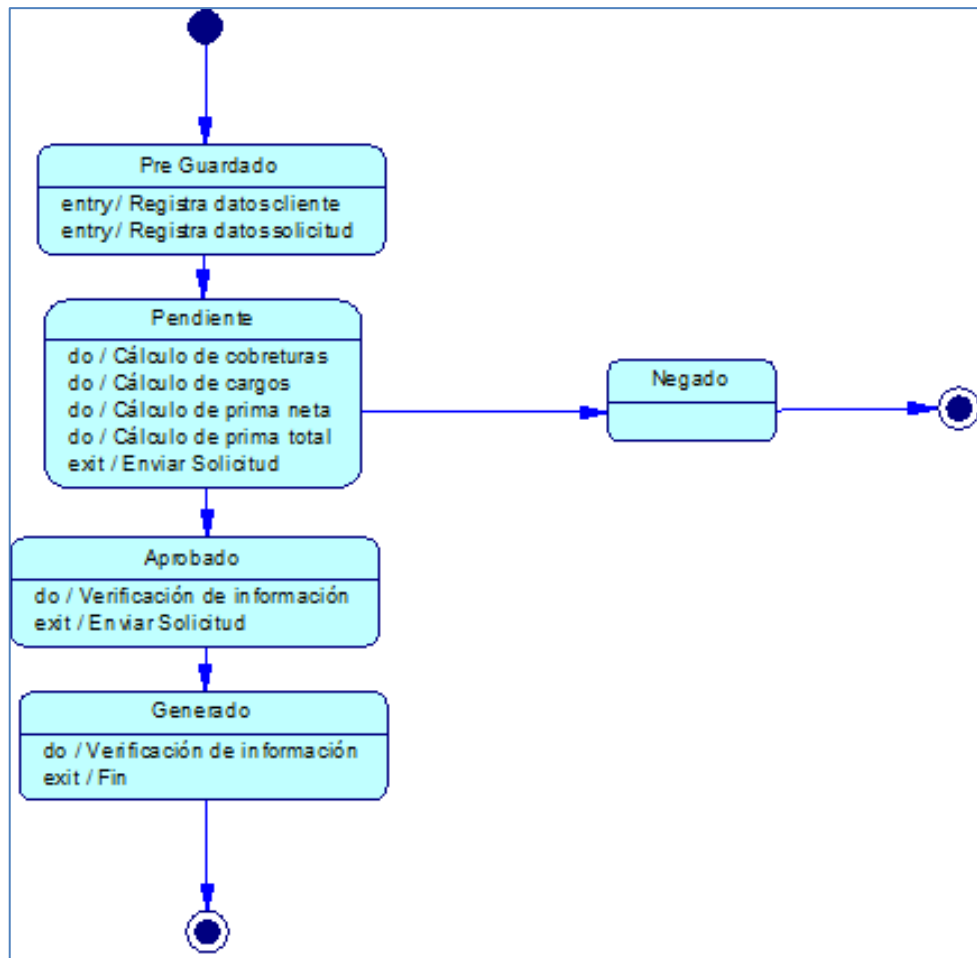


Figura 3.11 Diagrama de estados de una solicitud

3.3 Desarrollo

En el desarrollo de las aplicaciones del sistema se hizo uso de las herramientas de Microsoft Visual Studio.Net 2010 para la solución de los servicios web y el aplicativo web, Visual Studio.Net 2008 para el aplicativo móvil y SQL Server Management Studio para el diseño de base de datos centralizada como para la base distribuida en el dispositivo móvil.

3.3.1 Disponibilidad del cliente

Durante el desarrollo de la solución seguro total ha sido fundamental el trabajo en conjunto con los auspiciantes de la tesis por la retroalimentación en ciertos procesos o historias de usuarios permitiendo brindar mayor flujo en la construcción a través de la explicación de incertidumbres o dudas por parte de los programadores.

Otro punto fundamental de que el cliente esté presente en esta etapa es la aplicación de pruebas inmediatas que permitan no desviarse del objetivo analizado en cada historia de usuario.

3.3.2 Unidad de pruebas

Para la implementación de pruebas unitarias se usó la herramienta de Visual Studio .Net 2010 Team Tester estableciendo pruebas sobre las porciones de código más sensibles y que representan el negocio de la solución como muestra la figura 3.12

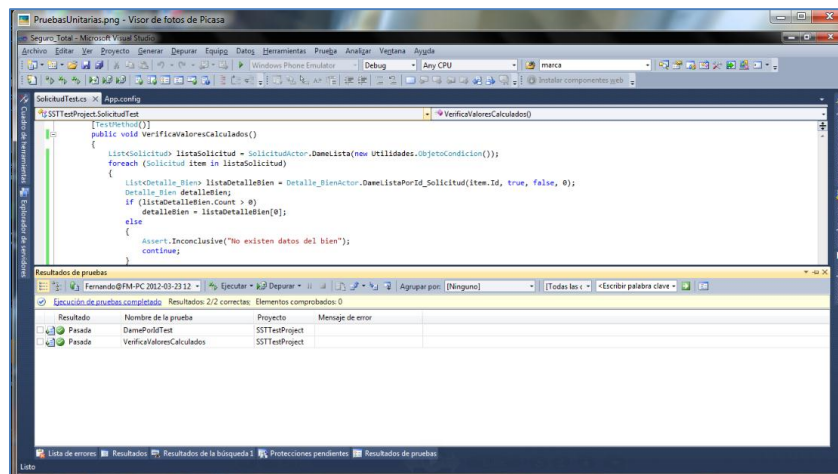


Figura 3.12 Prueba unitaria

3.3.3 Programación por parejas

Todo la construcción del sistema fue desarrollado por los dos autores de la tesis permitiendo así cumplir con este principio de la metodología de programación extrema y el conocimiento del sistema por parte de los dos involucrados.

3.3.4 Integración

En el sistema en su totalidad no hubo necesidad de usar una herramienta de administración de código fuente ya que el desarrollo se implementó en pareja y los dos autores de la tesis trabajaban sobre la misma solución.

La integración global del sistema se la podrá contemplar a través de las pruebas de aceptación que los usuarios finales diseñen para identificar que se cumplan todas las historias de usuario.

Estándares

Para que el código que se genere sea de fácil lectura y entendimiento se ha decidido adoptar ciertos estándares de codificación que permitirán tener uniformidad en la especificación de nombres.

- Se omitirán las tildes en todos los casos.
- Las letras “ñ” se deben remplazar por la silaba ni.
- Se procurará no utilizar abreviaturas tratando siempre de especificar nombres completos que mantengan la unión de varias palabras.

- Uso de convenciones de nombres tipo CAMEL. Ejemplos: idCliente, idBien, nombre, tipoidentificación.
- Los atributos de clases comenzarán con guion bajo “_”. Ejemplos: _nombre, _descripción.
- Las variables y parámetros no usarán el guion bajo. Ejemplos: auxiliar, temporal, primaNeta.
- Las propiedades de clases usarán convenciones de nombre tipo PASCAL. Ejemplo: Id, Activo, PaisOrigen.
- Los métodos y funciones usarán convenciones de nombres tipo PASCAL. Ejemplo: AutenticaUsuario, DamePorId, etc.
- Los nombres de Clases utilizarán nombres en singular, con convención de nombre tipo PASCAL. Ejemplo: Usuario, Solicitud, Accesorio, etc.

CAPÍTULO 4

IMPLANTACIÓN DEL SISTEMA

4.1 Pruebas de Aceptación

Las pruebas de aceptación se concentraron en las funcionalidades más relevantes de la aplicación para lo cual se muestran las siguientes ejecuciones:

Tabla 4.1 Prueba de Aceptación, Creación de usuario

La prueba de aceptación Creación de usuario detalla los pasos y condiciones con las cuales se realizó la prueba.

Caso de Prueba de Aceptación	
Número: 1	Historia de Usuario No. 1. Acceso al Sistema. Historia de Usuario No. 3. Administración aplicativo web.
Nombre: Creación de Usuarios	
Descripción: Se propone la verificación de la creación de un nuevo usuario, identificando el uso de un código de acceso no usado y constatando el envío del correo electrónico con su contraseña al usuario creado. Identificar el acceso a la aplicación validando contraseña.	
Condiciones de Ejecución: <ul style="list-style-type: none">• Creación de un usuario con código de acceso ya existente.• Creación de un usuario no existente en base de datos.• Acceso de un usuario no creado.• Acceso de un usuario con contraseña incorrecta.	

Entrada / Pasos de Ejecución:

- Acceso como administrador.
- Ejecución de opción de administración de usuarios.
- Registro de nuevo usuario.
- Constatar correo con clave de acceso.
- Ingreso a la aplicación con el nuevo usuario.

Resultado esperado:

- No creación de usuario con datos obligatorios incompletos.
- Validación de código de acceso ya registrado.
- Envío de correo electrónico con clave de acceso.
- Ingreso a la aplicación con el nuevo usuario con contraseña correcta
- Denegación de acceso a la aplicación con credenciales inválidas.

Evaluación de la Prueba: EXITOSA

En la tabla 4.2, La prueba de aceptación Registro de pólizas de aseguramiento detalla los pasos y condiciones con las cuales se realizó el registro de la solicitud, bajo las condiciones de la creación de una nueva solicitud hay que tomar en cuenta que se origina desde el dispositivo móvil.

Tabla 4.2 Prueba de Aceptación, Registro de pólizas de aseguramiento

Caso de Prueba de Aceptación	
Número: 2	<p>Historia de Usuario No. 2. Registro de datos del bien.</p> <p>Historia de Usuario No. 4. Registro de datos cliente.</p> <p>Historia de Usuario No. 5. Registro de forma de pago.</p> <p>Historia de Usuario No. 9. Envío de la información.</p>
Nombre: Registro de póliza de aseguramiento desde aplicativo móvil.	
Descripción: Se propone la verificación de alta de una nueva solicitud de póliza de aseguramiento de vehículo, contemplando el acceso de todos los datos desde el aplicativo móvil hasta el envío o registro en la base centralizada.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Registro con clientes ya creados. • Catálogos previamente sincronizados en la base del dispositivo móvil. 	
Entrada / Pasos de Ejecución:	
<ul style="list-style-type: none"> • Acceso al aplicativo móvil. • Registro de datos base del bien a asegurar. • Registro de coberturas. • Registro de accesorios. • Registro o captura de datos del cliente. • Datos específicos del bien. • Toma de fotografías. • Envío de información a base centralizada. 	
Resultado esperado:	
<ul style="list-style-type: none"> • Registro de datos del bien asegurado en el dispositivo móvil. • Captura de información de clientes ya registrados. • Toma correcta de fotografías del bien. • Envío de información a base central consistente. 	
Evaluación de la Prueba: EXITOSA	

Tabla 4.3 Prueba de Aceptación, Verificación, análisis y aprobación

En la tabla 4.3 se detalla las condiciones con las que se realizó la prueba como también los resultados obtenidos.

Caso de Prueba de Aceptación	
Número: 3	Historia de Usuario No. 7. Verificación, Análisis y Aprobación de Solicitudes.
Nombre: Verificación, análisis y aprobación adecuada de solicitudes.	
Descripción: Identificar el correcto registro de información enviada desde el dispositivo móvil y proceso de verificación de las solicitudes registradas terminando con su aprobación o anulación.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Registro de las solicitudes previamente ejecutadas desde el dispositivo móvil. • Usuario verificador con acceso a las solicitudes. 	
Entrada / Pasos de Ejecución:	
<ul style="list-style-type: none"> • Acceso al aplicativo web. • Selección de la solicitud a verificar. • Verificación y actualización de información de datos del cliente. • Validación de datos del bien a asegurar con sus fotografías. • Aprobación o negación de la solicitud. 	
Resultado esperado:	
<ul style="list-style-type: none"> • Correcta actualización de información de solicitud y cliente. • Solicitud aprobada o anulada según recomendación del verificador. 	
Evaluación de la Prueba: EXITOSA	

4.2 Implantación

Para la puesta en producción de los entregables del sistema en cada iteración es necesario generar una publicación de las aplicaciones desarrolladas utilizando diferentes mecanismos para cada aplicación.

4.3 Aplicación de Servicios Web

El primer paso a ejecutar es compilar la aplicación en modo Release lo que hará que las dll que se generan tengan un peso menor que una compilación en modo Debug o de depuración.

El segundo paso es colocarse sobre el proyecto de exposición de los servicios web y con el uso de click derecho ejecutar la opción que dice Publicar, esto producirá que en una carpeta, en el lugar que especifiquemos, copie archivos .svc, las dlls necesarias y el archivo de configuración que permite especificar la cadena de conexión a la base de datos. La publicación no expone código fuente en el sitio web donde se publique.

En las figura 4.1 y 4.2 se muestra el paso 1 y paso 2 para publicar la solución con la herramienta de desarrollo.

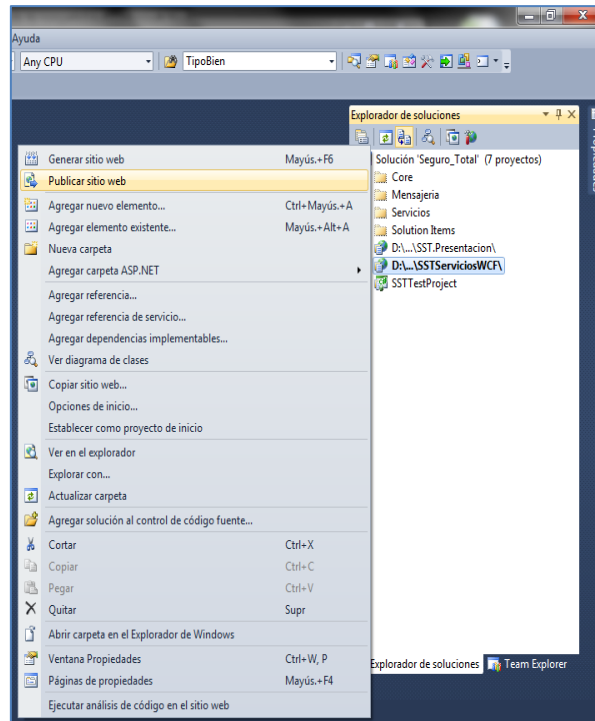


Figura 4.1 Publicación paso 1

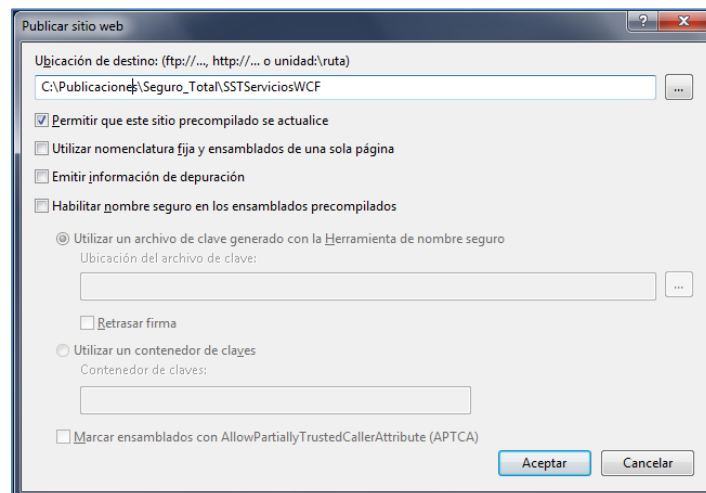


Figura 4.2 Publicación paso 2

En la figura 4.3 se muestra el último paso que corresponde a la agregación de la aplicación dentro del Internet Information Services dentro del Sitio Web por

Defecto y especificando el lugar físico en disco de donde se haya copiado la publicación del sitio.

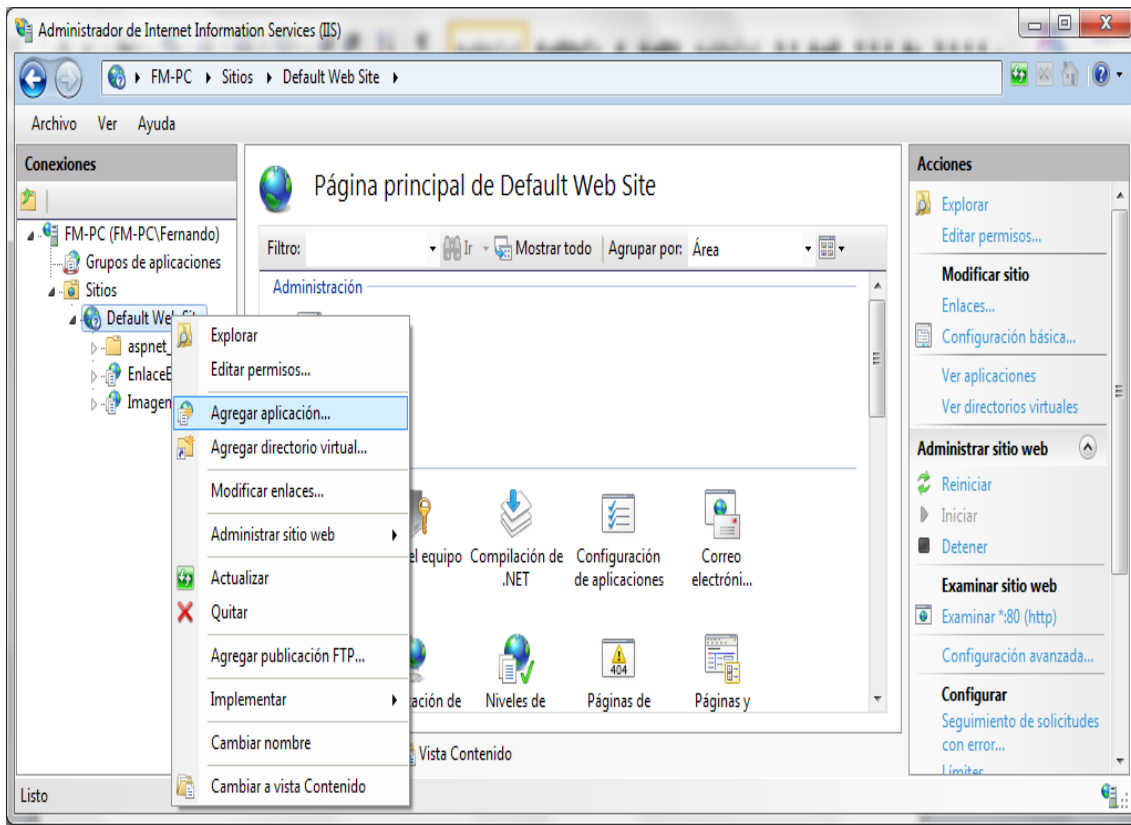


Figura 4.3 Creación de un directorio virtual

4.4 Aplicación de Sitio Web

Los pasos a seguir son los mismos que la publicación de los servicios web y como adicional se incluirán archivos con extensión aspx de los formularios web que se posea.

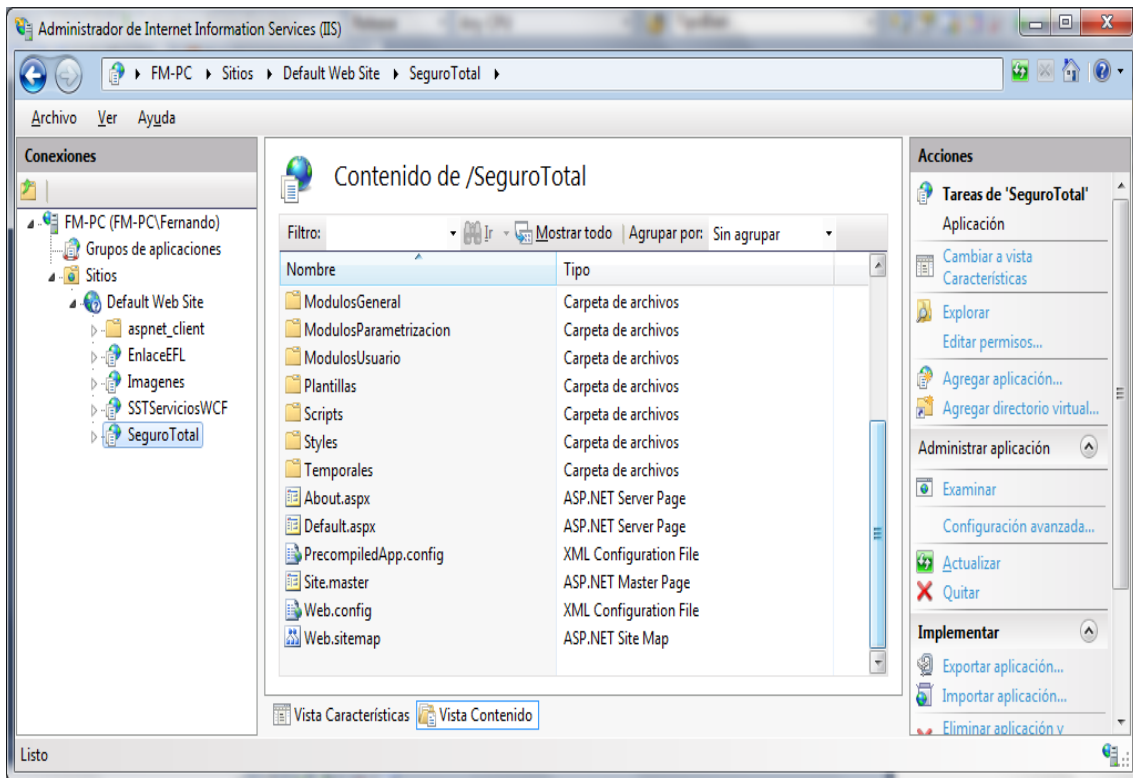


Figura 4.4 Propiedades del directorio virtual

4.5 Aplicación Móvil

Se genera un instalador el cual distribuirá el SQL Server CE 3.5, la aplicación móvil y la base de datos de tal manera que al ejecutar dicho instalador produce una instalación de lo antes señalado.

En la figura 4.5 se muestra el explorador de soluciones del aplicativo móvil en el cual se encuentra el proyecto generador del instalador SeguroTotal.cab el cual se copia al dispositivo y se ejecuta para que se instale en el mismo.

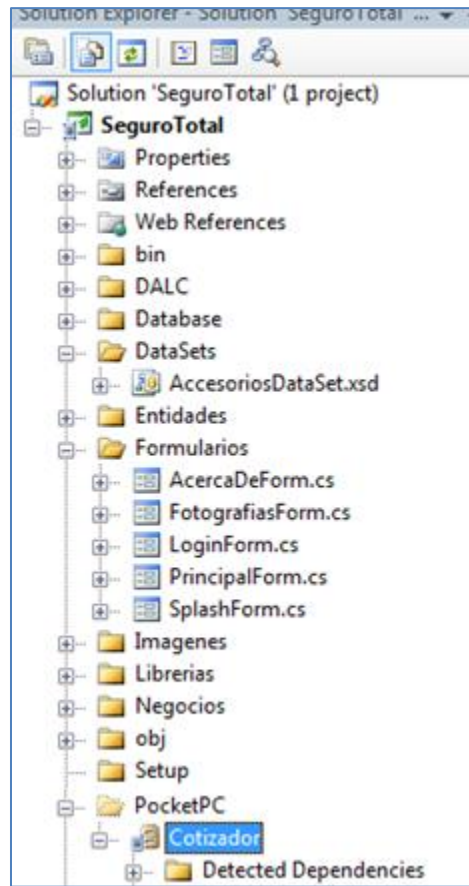


Figura 4.5 Generación de instalador de aplicativo Pocket

CAPÍTULO 5

CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

- Al analizar los procesos manuales que se venían realizando al registrar una solicitud de seguro vehicular, se observó que daba lugar a posibles alteraciones en los resultados de una cotización, lo cual originó la necesidad de implementar el sistema Seguro Total para transparentar y garantizar la fidelidad de la información.
- Para el desarrollo del proyecto fue una excelente selección la metodología de desarrollo de programación extrema ya que tiene lineamientos bien definidos como la recolección de requerimientos del sistema a través de historias de usuario en términos naturales dejando a un lado los tecnicismos, además contempla el desarrollo en parejas evitando que el código sea de propiedad de un solo programador, lo que fue ideal para el desarrollo de la tesis, pero ésta característica no es muy aplicable en el mundo laboral por la percepción de desperdicio de un recurso.
- Al trabajar en proyectos donde no se cuente con la disponibilidad del cliente o usuario de la aplicación que interactúe permanentemente con los desarrolladores no resultaría adecuado el uso de la metodología desarrollo programación extrema.

- El uso de Visual Studio .NET facilitó el desarrollo, pruebas e implementación del sistema Seguro Total permitiendo construir los aplicativos móvil y web con un lenguaje de programación en común y en un mismo entorno de desarrollo integrado.
- Las aplicaciones desarrolladas permitieron cumplir el propósito para el cual fueron creadas, sistematizando el proceso de las cotizaciones vehiculares, entregando resultados de forma eficaz, eficiente y manteniendo integridad en el manejo de la información, confiabilidad y disponibilidad de los datos.

5.2 Recomendaciones

- Para mantener la fidelidad y confianza en los datos es necesario aplicar mecanismos de seguridad a nivel administrativo y de base de datos en los aplicativos web y móvil.
- Dentro del proceso de investigación en el desarrollo del aplicativo móvil se intentó hacer uso de tecnología de última generación como es el caso del sistema operativo Windows Phone 7 para teléfonos inteligentes, pero la implementación de la solución sobre el dispositivo no fue posible ya que tuvo restricciones de liberación para Ecuador.
- En el mercado existen varios tipos de plataformas tecnológicas de desarrollo para aplicativos móviles como para los sistemas operativos Android, IOS y Windows Phone 7, se debe considerar que se encuentren los permisos liberados de implementación y publicación para el país.

BIBLIOGRAFÍA

TEXTOS

- LETELIER, Patricio PENANÉS, M. Carmen, Metodologías ágiles para el desarrollo de software, Extreme Programming (XP). Universidad Politécnica de Valencia
- Ing. José Joskowicz, Reglas y Prácticas en eXtreme Programming, Junio 2005 Technical report TR 05/03 Department of Computer Science, University of Manitoba, Winnipeg, Manitoba, Canada R3T 2N2

DIRECCIONES ELECTRONICAS

- **PROGRAMACIONEXTREMA.ORG**
<http://www.programacionextrema.org/>
29 Octubre 2011
- **FUNDAMENTOS Y EJEMPLIFICACIÓN DE LA METODOLOGÍA XP**
<http://users.dsic.upv.es/asignaturas/facultad/lsi/ejemploxp/>
14 Diciembre 2011
- **FUNDAMENTOS UML**
<http://www.ingenierosoftware.com/analisisydiseno/uml.php>
FEBRERO 2012

HOJA DE LEGALIZACIÓN DE FIRMAS

ELABORADO POR

Luis Horacio Salazar Estévez

Fernando Manuel Morales Mora

DIRECTOR DE LA CARRERA

Ing. Mauricio Campaña

Sangolqui 29 de Junio del 2012

ESCUELA POLITÉCNICA DEL EJÉRCITO
CARRERA DE INGENIERÍA E INFORMÁTICA

SALAZAR ESTÉVEZ LUIS HORACIO

MORALES MORA FERNANDO MANUEL

Autorizamos a la Escuela Politécnica del Ejército la publicación de la Tesis de Grado titulada “ANÁLISIS, DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA SEGURO TOTAL PARA LA EMPRESA SOLMOVSA.” en la biblioteca virtual de la institución; cuyo contenido, ideas y criterios es de nuestra exclusiva responsabilidad y autoría.

Sangolquí 29 de Junio del 2012

Luis Horacio Salazar Estévez

Fernando Manuel Morales Mora