

# ESCUELA POLITÉCNICA NACIONAL

## FACULTAD DE INGENIERÍA DE SISTEMAS

DESARROLLO DEL SISTEMA DE ADMINISTRACIÓN DE  
HISTORIAS CLÍNICAS DEL DEPARTAMENTO MÉDICO DE LA  
EMPRESA PRODUCTORA Y EXPORTADORA DE FLORES  
ROSADEX CIA. LTDA.

PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN  
SISTEMAS INFORMÁTICOS Y COMPUTACIÓN

### AUTORES:

PAMELA PAOLA ESTRELLA RUBIO  
[estrellita8\\_8@hotmail.com](mailto:estrellita8_8@hotmail.com)

HÉCTOR OMAR VÁSQUEZ BARRIONUEVO  
[omar.vasquez@hotmail.com](mailto:omar.vasquez@hotmail.com)

DIRECTORA: ING. MAYRITA VALLE  
[mayra\\_valle@hotmail.com](mailto:mayra_valle@hotmail.com)

QUITO, AGOSTO 2008

## DECLARACIÓN

Nosotros, Pamela Paola Estrella Rubio y Héctor Omar Vásquez Barrionuevo, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos nuestros derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

---

**Pamela Estrella**

---

**Omar Vásquez**

## **CERTIFICACIÓN**

Certifico que el presente trabajo fue desarrollado por Pamela Paola Estrella Rubio y Héctor Omar Vásquez Barrionuevo, bajo mi supervisión.

---

**Ing. Mayrita Valle**

**DIRECTOR DE PROYECTO**

## **DEDICATORIA**

Este Proyecto de Titulación dedico de una manera muy especial a Dios, quien ha bendecido y guiado mi camino, a mis padres y a mi hermana, gracias a su apoyo, comprensión y sabios consejos me permitieron culminar mi carrera y la terminación del presente trabajo.

**Pamela**

## **DEDICATORIA**

Dedico este Proyecto de Titulación especialmente a mi madre y a mis hermanos, por su incondicional apoyo y estar en los momentos buenos y malos.

También dedico este Proyecto de Titulación a mi esposa Elizabeth y a mi hija Sarah, por ser mi inspiración para seguir adelante.

**Omar**

## AGRADECIMIENTOS

Primeramente quiero agradecer a Dios quien ha llenado de bendiciones mi vida y la de toda mi familia, ha guiado mi camino y ha permitido la culminación de mi proyecto de titulación.

Agradezco a mis padres Héctor y Gladys por ser los mejores padres del mundo que Dios me pudo dar, por brindarme su apoyo y estar conmigo en los momentos más difíciles de mi vida, a mi hermana Patricia quien me ha dado ánimo y su comprensión a lo largo de mi carrera, ustedes se han convertido en el pilar fundamental de mi vida para haber logrado conseguir este objetivo.

Agradezco de una manera especial a Paul, por estar siempre a mi lado, darme sus consejos y siempre estar pendiente de mí.

También quiero agradecer a mi gran amigo y compañero Omar, una persona inteligente, capaz pero sobre todo de una gran calidad humana quien siempre me ha brindado su ayuda incondicional, gracias por tu esfuerzo y gran dedicación puesto en este proyecto.

A mis amigos más cercanos de facultad: Andrea, Anita, Carmen, Paúl C, Wilson, Jorge N, Stefy, Jorge P. por haberme brindado su amistad con los cuales he contado e incondicionalmente me han ayudado.

A la Ing. Mayrita Valle por su dirección y apoyo al realizar este proyecto.

**Pamela**

## **AGRADECIMIENTOS**

A mi madre por su apoyo en todas las etapas de mi vida, ejemplo para ser una mejor persona cada día, y por su amor incondicional.

A mi hermano Polo por sus consejos y apoyo en los momentos difíciles de mi vida.

A mi hermano Vinicio por su guía en la universidad de la vida, y por su ayuda siempre que lo necesité.

A mi esposa Elizabeth quien me apoyó en todo momento para la finalización de este Proyecto de Titulación, y en todos los aspectos de mi vida; también a mi hija por ser mi fuente de inspiración para lograr lo que me proponga.

A mis amigos Andrea, Anita, Carmen, Pamela, Paul E, Paúl C, Wilson, Jorge, Mary, Rubén, Denys, Bengy, Byron por su amistad e incondicional apoyo en todo momento.

A mi amiga Pamela por su apoyo en el transcurso de la realización de esta etapa final para lograr el título de Ingeniero en Sistemas, y por la amistad que nos une.

A la Ing. Mayrita Valle por su amistad, consejos, dirección, apoyo, y ejemplo que ha demostrado en todo momento.

**Omar**

# CONTENIDO

INTRODUCCIÓN .....	1
CAPITULO I .....	4
DEFINICIÓN DEL PROBLEMA.....	4
1.1. ANTECEDENTES Y DESCRIPCIÓN DEL PROBLEMA .....	4
1.2. SELECCIÓN DE LA METODOLOGÍA Y HERRAMIENTAS A UTILIZAR..	5
1.2.1. METODOLOGÍA.....	5
1.2.2. HERRAMIENTA CASE PARA EL MODELO RELACIONAL.....	13
1.2.3. SISTEMA DE ADMINISTRACIÓN DE BASE DE DATOS .....	17
1.2.4. TECNOLOGÍAS DE DESARROLLO. ....	21
CAPITULO II .....	25
ANÁLISIS Y DISEÑO .....	25
2.1 ESPECIFICACIÓN DE REQUERIMIENTOS .....	25
2.1.1. VISIÓN DEL SISTEMA.....	25
2.1.2 ESCENARIOS .....	27
2.2 ANÁLISIS.....	31
2.2.1 CASOS DE USO .....	31
2.2.2 REQUISITOS DE CALIDAD .....	35
2.2.3 ROLES .....	36
2.2.4 PLAN DE ITERACIÓN.....	37
2.3 DISEÑO .....	38
2.3.1 DIAGRAMA DE APLICACIÓN.....	38
2.3.2 COMPONENTES DE LA APLICACIÓN.....	38
2.3.3 ARQUITECTURA DEL SISTEMA.....	39
2.3.4 DIAGRAMA DE CLASES .....	42
2.3.5 MODELO RELACIONAL .....	44



CAPITULO III .....	46
IMPLEMENTACIÓN, PRUEBAS Y EVALUACIÓN.....	46
3.1 IMPLEMENTACIÓN.....	46
3.1.1 ESTÁNDAR DE CODIFICACIÓN PARA EL DESARROLLO DE CLASES EN C# .....	46
3.1.2 CAPAS .....	55
3.2 PRUEBAS.....	66
3.2.1 ESCENARIO: GESTIONAR LAS HISTORIAS CLÍNICAS: REGISTRAR PACIENTES.....	66
3.2.2 ESCENARIO: GESTIONAR LAS CONSULTAS: REGISTRAR CONSULTA.....	74
3.3 EVALUACIÓN.....	81
3.3.1 CARACTERÍSTICA: FUNCIONALIDAD .....	81
3.3.2 CARACTERÍSTICA: USABILIDAD .....	83
3.3.3 CARACTERÍSTICA: EFICIENCIA .....	85
3.3.4 VALORAR DE RESULTADOS .....	87
3.3.5 CONCLUSIÓN DE LA EVALUACIÓN .....	87
3.3.6 ANÁLISIS DE RESULTADOS .....	88
CAPITULO IV.....	91
CONCLUSIONES Y RECOMENDACIONES .....	91
4.1 CONCLUSIONES.....	91
4.2 RECOMENDACIONES .....	93
BIBLIOGRAFÍA .....	95
ANEXOS .....	98

## ÍNDICE DE FIGURAS

Figura 1.2.1.1 Fases y workflows de RUP basado en [BMP98].....	6
Figura 1.2.1.2 Ciclo de vida de XP, adaptado de [Beck99a] .....	8

Figura 1.2.1.3 Modelo de Proceso para el Desarrollo de Aplicaciones .....	10
Figura 1.2.2.1 Arquitectura de Power Designer .....	13
Figura 1.2.2.2 Microsoft Visio .....	15
Figura 1.2.3.1 Componentes Básicos de SQL Server 2005.....	17
Figura 1.2.3.2 Arquitectura de Oracle .....	18
Figura 1.2.3.3 BenchMark TPC entre Microsoft SQL Server 2005 y Oracle .....	21
Figura 1.2.4.1 Arquitectura de Java .....	22
Figura 2.3.1.1 Caso de Uso Principal.....	32
Figura 2.3.1.2 Caso de Uso Gestionar Exámenes .....	33
Figura 2.3.1.3 Caso de Uso Medicamentos .....	34
Figura 2.3.1.4 Caso de Uso Pacientes.....	34
Figura 2.3.1.5 Caso de Uso Reportes .....	35
Figura 2.3.4.1 Plan de Iteración .....	37
Figura 2.4.1.1 Diagrama de Aplicación .....	38
Figura 2.4.3.1 Arquitectura del Sistema .....	1
Figura 2.4.4.1 Diagrama de Clases.....	42
Figura 2.4.5.1 Modelo Relacional.....	45
Figura 3.1.2.1 Estructura de la Capa Entidades Sección 1 .....	55
Figura 3.1.2.2 Estructura de la Capa Entidades Sección 2.....	56
Figura 3.1.2.3 Estructura de la Capa Entidades Sección 3.....	56
Figura 3.1.2.4 Estructura de la Capa Entidades Sección 4.....	57
Figura 3.1.2.5 Estructura de la Capa Entidades Sección 5.....	58
Figura 3.1.2.6 Estructura de la Capa Entidades Sección 6.....	58
Figura 3.1.2.7 Estructura de la Capa Entidades Sección 7.....	59
Figura 3.1.2.8 Estructura de la Capa Entidades Sección 8.....	60
Figura 3.1.2.9 Estructura de la Capa Entidades Sección 9.....	61
Figura 3.1.2.10 Estructura de la Capa Entidades Sección 10 .....	61
Figura 3.1.2.11 Estructura de la Capa Entidades Sección 11 .....	62
Figura 3.1.2.12 Estructura de la Capa Entidades Sección 12 .....	62
Figura 3.1.2.13 Estructura de la Capa Entidades Sección 13 .....	63
Figura 3.1.2.14 Estructura de la Capa de Lógica .....	64
Figura 3.1.2.15 Estructura de la Capa de Servicios .....	65
Figura 3.1.2.16 Estructura de la Capa de GUI .....	66

## ÍNDICE DE TABLAS

Tabla 2.1.2.1 Roles Alternativos del Escenario Gestionar las Historias Clínicas: Registrar Pacientes .....	27
Tabla 2.1.2.2 Roles Alternativos del Escenario Gestionar las Consultas: Registrar Consulta .....	29
Tabla 2.3.2.1 Requisitos de Calidad.....	36
Tabla 2.4.2.1 Descripción de los Componentes de la Aplicación.....	39
Tabla 2.4.3.1 Descripción de los Componentes del Servidor .....	40
Tabla 2.4.3.2 Descripción de los Componentes del Cliente .....	41
Tabla 3.1.1.1 Nomenclatura de Clases .....	47
Tabla 3.1.1.2 Nomenclatura de Enumeraciones, Valores de Enumeraciones, Eventos e Interfaz .....	48
Tabla 3.1.1.3 Nomenclatura de Eventos .....	48
Tabla 3.1.1.4 Nomenclatura de Interfaz .....	49
Tabla 3.1.1.5 Nomenclatura de Métodos .....	49
Tabla 3.1.1.6 Nomenclatura de Nombres (Namespaces) .....	50
Tabla 3.1.1.7 Nomenclatura de Parámetros.....	50
Tabla 3.1.1.8 Tipos de Excepciones Estándar .....	54
Tabla 3.2.1.1 Pruebas del Escenario Gestionar la Historia Clínica: Registrar Pacientes.....	73
Tabla 3.2.2.1 Pruebas del Escenario Gestionar las Consultas: Registrar Consulta .....	80
Tabla 3.3.1.1 Métrica Interna de Seguridad de Acceso.....	81
Tabla 3.3.1.2 Métrica Interna de Cumplimiento de la Funcionalidad.....	82
Tabla 3.3.2.1 Métrica Interna de Capacidad para ser entendido.....	83
Tabla 3.3.2.2 Métrica Interna de Operabilidad .....	84
Tabla 3.3.3.1 Métrica Externa de Comportamiento del Tiempo .....	85
Tabla 3.3.3.2 Métrica Externa de comportamiento del Tiempo .....	86
Tabla 3.3.4.1 Sumarización de Valores Obtenidos .....	87
Tabla 3.3.6.1 Análisis de Resultados, característica Funcionalidad.....	88
Tabla 3.3.6.2 Análisis de Resultados, característica Usabilidad .....	88

Tabla 3.3.6.3 Análisis de Resultados, característica Eficiencia.....	89
Tabla 3.3.6.4 Análisis Final de Resultados .....	90

## **RESUMEN**

El presente trabajo corresponde al desarrollo de un sistema que automatiza todos los registros necesarios para crear la historia clínica de cada empleado según las normas laborales establecidas por el seguro social; permitiendo al médico obtener la información de sus pacientes de manera oportuna, registrando las citas de los empleados según cronogramas específicos y necesidades de atención del empleado; una vez que el médico haya realizado el diagnóstico pertinente, el sistema le permitirá seleccionar los medicamentos a prescribir junto con su posología y medicamentos genéricos; así el médico puede realizar las recetas médicas de manera rápida, eficiente y legible. Al tener todos los datos en el sistema, el médico puede generar reportes a la Gerencia con información resumida sobre Enfermedades, Consultas, Medicamentos y Exámenes como apoyo a la planificación de los recursos humanos.

## INTRODUCCIÓN

La empresa Rosadex Cia. Ltda. se dedica a la producción y exportación de flores a Europa, Norteamérica, y Asia. La mayoría de empleados trabaja en contacto constante con fungicidas y fertilizantes que afectan a su salud; por lo que se exige un control médico conducido, basado en las sugerencias indicadas por el seguro social, quien establece los exámenes y controles necesarios para este tipo de empresa.

El departamento médico requiere una gestión eficiente y confiable de las historias clínicas de cada empleado, el médico necesita registrar los medicamentos recetados y entregados a sus pacientes. Además, con la automatización de la información el médico agilizará su trabajo y tendrá reportes resumidos sobre Enfermedades, Consultas, Medicamentos y Exámenes realizados a los pacientes, como apoyo a la toma de decisiones de la Gerencia en relación al manejo de rotación y salida del personal.

El presente proyecto consta de cuatro capítulos que se describen a continuación:

En el primer capítulo se describe brevemente los antecedentes y la descripción del problema que se pretende resolver con el desarrollo de este proyecto. Además se describen las Metodologías y herramientas más difundidas y utilizadas en diversos proyectos de desarrollo de software, de las metodologías descritas en el capítulo se seleccionó Microsoft Solution Framework for Agile y las herramientas: Case Studio para realizar el modelado de Base de Datos, Microsoft SQL Server 2005 como Sistema de Base de Datos y como Tecnología de Desarrollo a Microsoft .Net Framework acompañado del IDE Microsoft Visual Studio 2008 y el lenguaje de programación C# 3.5.

El segundo capítulo está dividido en Especificación de Requerimientos, Análisis y Diseño, en la especificación de requerimientos se establece la visión del sistema en donde se describe el producto que se va a crear, el valor que tendrá y los participantes que originarán ese valor. Además, se determinan los objetivos del sistema para identificar los escenarios, cada escenario registra los pasos específicos que se pueden realizar para alcanzar ese objetivo. Una vez que se

tiene un listado de escenarios se establece la prioridad de cada uno de ellos, este listado recoge el trabajo pendiente acumulado de los escenarios del proyecto.

Se establecen los roles apropiados del sistema y se realiza su respectiva descripción, en donde se identifica las destrezas típicas, las capacidades, las necesidades, los deseos, los hábitos de trabajo, las tareas y la experiencia de un determinado grupo de usuarios.

En el Análisis, se determinan los casos de uso, en donde se encuentran como actores los roles definidos para el sistema, los casos de uso nos muestran gráficamente la descripción de los escenarios identificados.

Además se identifica los objetivos de calidad de servicio, los factores de calidad o los requisitos que no están expresados en los escenarios, los requisitos de calidad documentan características del sistema, como rendimiento, carga, disponibilidad, esfuerzo, accesibilidad, operabilidad y mantenimiento. Estos requisitos normalmente adquieren la forma de restricciones bajo las cuales debe funcionar el sistema. Una vez identificados se determina la importancia de cada requisito y la prioridad general de requisitos formando un listado de requisitos de calidad.

En el Diseño, se muestra el diagrama de aplicación en donde se puede observar los elementos que se pueden implementar como servicios Web, aplicaciones Web, aplicaciones Windows y aplicaciones de referencia como bases de datos externas y servicios Web externos.

Se diseña la arquitectura del sistema en donde se pueden definir elementos estructurales que hacen posible que la aplicación pueda controlar mejor aquellos requisitos que cambien y permitir que las diferentes áreas de la aplicación se desarrollen de forma independiente.

Además se diseña el diagrama de clases, y se describe el diccionario de datos en el Anexo 3. También se puede observar el modelo relacional del sistema.

El tercer capítulo está dividido en Implementación, Pruebas y Evaluación, en la Implementación se determina el estándar de codificación para el desarrollo de clases en C#.

Además, se identifican las 4 capas que componen el sistema, las mismas que son: Entidades, Lógica, Servicios y GUI.

En la capa de Entidades, se puede observar que las entidades están basadas en DataSet's tipados, los mismos que viajarán por todas las capas del sistema y que sirven de persistencia en la base de datos.

En la capa de Lógica se pueden observar los métodos creados y agrupados según la necesidad del negocio.

En la capa de Servicios se muestra los WebService's que maneja el sistema para la utilización de los métodos definidos en la capa de Lógica.

En la capa de GUI se muestra los formularios creados para la interacción con el usuario.

En las Pruebas, se valida la funcionalidad del sistema por cada escenario definido en el Análisis, para lo cual se definen casos de prueba según el escenario evaluado, en cada caso de prueba se especifica las precondiciones, entrada y el resultado esperado.

En la Evaluación se realiza la evaluación del sistema utilizando métricas internas y externas de calidad del producto de software, una vez que se aplican las métricas al caso de estudio, se procede a la sumarización de los datos obtenidos y se describe la conclusión de la evaluación con su respectiva representación gráfica.

Finalmente en el cuarto capítulo se presentan las conclusiones del presente trabajo y recomendaciones para trabajos similares que pueden efectuarse a futuro.



## **CAPITULO I**

### **DEFINICIÓN DEL PROBLEMA**

#### **1.1. ANTECEDENTES Y DESCRIPCIÓN DEL PROBLEMA**

La empresa Rosadex Cia. Ltda. se dedica a la producción y exportación de flores a Europa, Norteamérica, y Asia. El crecimiento de la demanda de sus productos requiere un número considerable de trabajadores para las áreas de producción, postCosecha y exportación de su producto. La mayoría de empleados de Rosadex trabaja en contacto constante con fungicidas y fertilizantes que afectan a su salud; por lo que se exige un control médico conducido, basado en las sugerencias indicadas por el seguro social, quien establece los exámenes y controles necesarios para este tipo de empresa.

La importancia de la información que debe llevar el departamento médico requiere una gestión eficiente y confiable de las historias clínicas de cada empleado. Además, la mayoría de los diagnósticos realizados por el médico, requieren su medicamento y tratamiento pertinente, razón por la cual el médico necesita registrar los medicamentos recetados y entregados a sus pacientes.

La automatización de la información le permitirá al médico agilizar su trabajo y tener reportes resumidos sobre Enfermedades, Consultas, Medicamentos y Exámenes realizados a los pacientes, como apoyo a la toma de decisiones de la Gerencia en relación al manejo de rotación y salida del personal.

La gran cantidad de datos clínicos que maneja el médico de la empresa dificulta el desempeño de su trabajo, y en ocasiones llega a descuidar su principal labor de atender a los empleados, por la necesidad de presentar la documentación pertinente y los reportes resumidos.

## **1.2. SELECCIÓN DE LA METODOLOGÍA Y HERRAMIENTAS A UTILIZAR**

Para realizar la mejor selección se presenta a continuación un breve análisis de las metodologías y herramientas más difundidas y utilizadas en diversos proyectos de desarrollo de software

### **1.2.1. METODOLOGÍA**

#### **1.2.1.1. Rational Unified Process (RUP)**

RUP es un proceso de desarrollo de software, una forma disciplinada de asignar tareas y responsabilidades en una empresa de desarrollo (quién hace qué, cuándo y cómo).

Aumenta la productividad de los desarrolladores mediante acceso a:

- Base de conocimiento, plantillas y herramientas.
- Se centra en la producción y mantenimiento de modelos del sistema más que en producir documentos.

Ciclos y Fases:

El proceso de ciclo de vida de RUP se divide en cuatro fases bien conocidas llamadas Incepción, Elaboración, Construcción y Transición (Ver Figura 1.2.1.1). Esas fases se dividen en iteraciones, cada una de las cuales produce un fragmento de software demostrable. La duración de cada iteración puede extenderse desde dos semanas hasta seis meses.

Cada fase concluye con un hito bien definido donde deben tomarse ciertas decisiones.

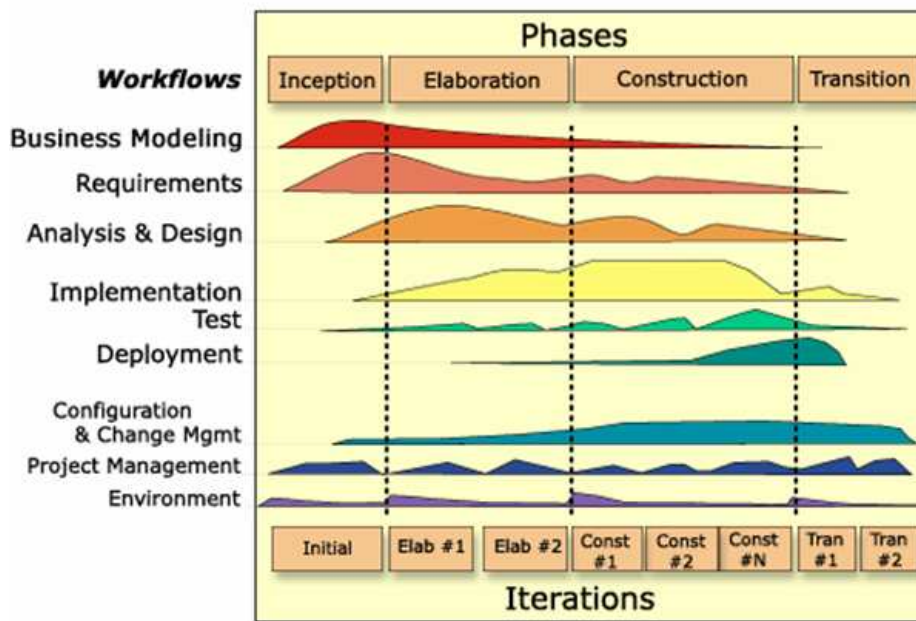


Figura 1.2.1.1 Fases y workflows de RUP basado en [BMP98] [7]

A través de las fases se desarrollan en paralelo nueve workflows o disciplinas: Modelado de Negocios, Requerimientos, Análisis & Diseño, Implementación, Prueba, Gestión de Configuración & Cambio, Gestión del Proyecto y Entorno.

Proceso Iterativo e Incremental:

Las actividades se encadenan en una mini-cascada con un alcance limitado por los objetivos de la iteración. RUP sigue un modelo iterativo que aborda las tareas más riesgosas primero con lo cual se logra reducir los riesgos del proyecto y tener un subsistema ejecutable tempranamente.

Iteraciones:

- Cada fase del RUP se descompone en iteraciones.
- Una iteración es un ciclo completo de desarrollo que produce una versión de un producto ejecutable.
- Cada iteración pasa a través de varios flujos de trabajo del proceso.

[7] <https://pid.dsic.upv.es/C1/Material/Documentos%20Disponibles/Introducci%C3%B3n%20a%20RUP.ppt>

#### Artefactos:

- Documento, informe o ejecutable que se produce, se manipula o se consume.
- Cada actividad del RUP lleva algunos artefactos asociados, bien sean requeridos como entradas o bien sean generados como salidas.

#### Roles:

Un rol define el comportamiento y las responsabilidades de un individuo, una persona puede tener varios roles.

RUP define un total de 30 roles: Modelado de Negocios tiene 3, Requerimientos tiene 5, Análisis & Diseño tiene 6, Implementación tiene 3, Pruebas tiene 2, Despliegue tiene 4, Gestión de Configuración & Cambios tiene 2, Gestión del Proyecto tiene 2 y Entorno tiene 3.

#### Responsabilidades:

Significa hacer una serie de actividades o ser el responsable de una serie de artefactos.

#### Actividad:

Describe los pasos de concepción, realización y revisión que llevan a cabo los trabajadores para crear o modificar los artefactos.

#### Ventajas:

- RUP es uno de esos enfoques de ciclo de vida que se adapta especialmente bien a UML
- Permite producir software de calidad para satisfacer las necesidades de los usuarios finales.
- RUP captura algunas de las mejores prácticas de desarrollo de software, haciéndolo adaptable a un amplio rango de proyectos y organizaciones.

Desventajas:

- Si el conjunto de documentos y artefactos no son concebidos tal y como se plantea en RUP, dicha documentación solo servirá para ser archivada lo cual no genera valor respecto a la calidad del desarrollo, y evoluciona en problemas más complejos.
- Es necesario incluir a más personas en el equipo de desarrollo.
- En RUP se gasta demasiado tiempo en la documentación antes de pasar a la fase de desarrollo.
- RUP sugiere que trabajar tiempo extra no es una condición crónica, pero puede perjudicar la motivación del grupo o el rendimiento de forma considerable.
- Para utilizar RUP, es recomendable que los desarrolladores tengan conocimiento previo en UML.

### 1.2.1.2. eXtreme Programming (XP)

eXtreme Programming es un conjunto de técnicas y prácticas para el desarrollo de software. Se establece dentro de la familia de metodologías ligeras, tratando de obtener métodos sencillos de obtener software de calidad.

Sus principios básicos son dos: la mejora de la comunicación con los usuarios, para retroalimentar el proceso de desarrollo; y obtener cuanto antes un programa que haga algo, para ir añadiendo incrementalmente nuevas características.

El ciclo de vida es iterativo. El siguiente diagrama describe su cuerpo principal:

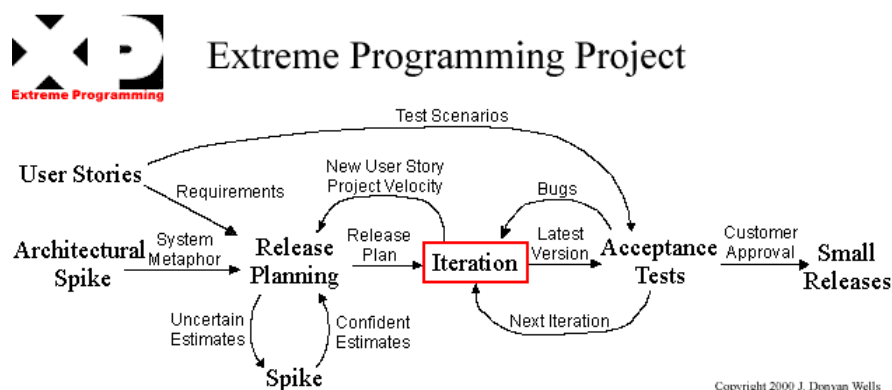


Figura 1.2.1.2 Ciclo de vida de XP, adaptado de [Beck99a] [8]

[8] <http://oness.sourceforge.net/proyecto/html/ch05s02.html>

Entre los artefactos que se utilizan en XP se puede mencionar a las tarjetas de historias (story cards) o historias de usuario; son tarjetas comunes de papel en que se escriben breves requerimientos de un rasgo; pueden adoptar el esquema CRC. Las tarjetas tienen una granularidad de diez o veinte días y se usan para estimar prioridades, alcance y tiempo de realización; en caso de discrepancia, gana la estimación más optimista.

Las historias que se implementan en cada iteración se deciden al principio de estas. Se establecen plazos para realizar cada iteración, y se realizan de forma secuencial. No es recomendable estar pendiente de historias a implementar en futuras iteraciones. Preocuparse de cumplir el plan de versiones es la prioridad.

Los roles de XP son pocos. Un cliente que escribe las historias y las pruebas de aceptación; programadores en pares; verificadores que ayudan al cliente a desarrollar las pruebas; consultores técnicos; y, como parte del management, un coach o consejero que es la conciencia del grupo, interviene y enseña, y un seguidor de rastros (tracker) que colecta las métricas y avisa cuando hay una estimación alarmante, además del Jefe.

Ventajas:

- XP se ha clasificado en las metodologías ágiles las cuales plantean aumentar la velocidad del proyecto.
- Reducen la documentación asociada a cada proyecto.
- Incide en mejorar la comunicación con los usuarios, la retroalimentación en el proceso.

Desventajas:

- Puede ser riesgoso con proyectos de complejidad alta.
- Se recomienda programar en pareja. Dos cabezas piensan mejor que una, sin embargo, pueden existir opiniones diferentes provocando guerras entre desarrolladores.

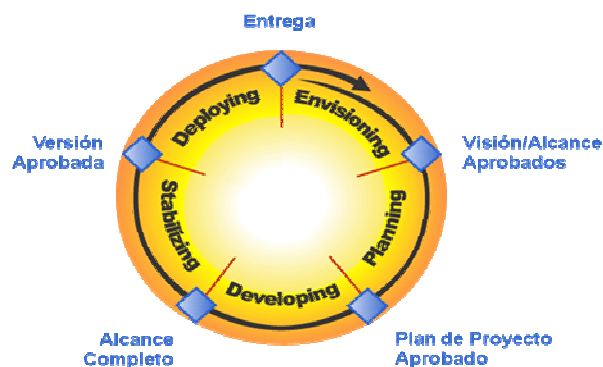
- Al ser una metodología ligera, la hace especialmente idónea para un entorno heterogéneo con un grupo grande de desarrolladores, además de que algunas de las medidas que propone no tienen sentido para proyectos pequeños.

### 1.2.1.3. Microsoft Solution Framework for Agile (MSF)

Microsoft Solution Framework versión 4.0 es el framework basado en las mejores prácticas de Microsoft y sus clientes. MSF toma las mejores prácticas y las integra en distintos modelos, principios y guías.

Presenta un metamodelo descriptivo y dos templates de procesos prescriptivos que implementan MSF 4.0 en Team System.

Team System proporciona un conjunto de herramientas integradas y extensibles que se pueden utilizar para activar el proceso de desarrollo (Ver Figura 1.2.1.3).



**Figura 1.2.1.3 Modelo de Proceso para el Desarrollo de Aplicaciones [9]**

Iteraciones:

La definición, el desarrollo y la comprobación del producto tienen lugar en iteraciones que se solapan y dan como resultado la finalización incremental del proyecto. Las distintas iteraciones tienen un foco diferente conforme se acerca el lanzamiento del proyecto. Las iteraciones pequeñas permiten reducir el margen de error de las estimaciones y proporcionan comentarios rápidos sobre la precisión de sus planes de proyecto. Cada iteración debe dar como resultado una parte estable del sistema global.

[9] <http://download.microsoft.com/download/4/4/E/44E1B331-E509-4D10-A9E3-B60640A3A403/20051206-ARC-BA.ppt>

Modelo de equipo:

El modelo de equipo de Microsoft Solutions Framework (MSF) describe el enfoque de Microsoft en lo que respecta a la estructuración de personas y sus actividades con el fin de asegurarse el éxito del proyecto.

Grupos de Apoyo:

Existen siete grupos de apoyo: El grupo de administración del programa apoya la entrega de la solución, el grupo de arquitectura apoya el sistema en su totalidad, el grupo de desarrollo apoya la solución técnica, el equipo de pruebas apoya la calidad de la solución desde la perspectiva del cliente, el equipo de lanzamiento y operaciones apoya la distribución e implementación de forma progresiva de la solución en la infraestructura adecuada, el grupo que se ocupa de la experiencia del usuario apoya la mejor solución para los usuarios finales y el grupo de administración de productos apoya las actividades empresariales de los clientes.

En MSF for Agile Software Development, se forma un equipo de pares para representar a todas las áreas que participan en la creación, utilización y mantenimiento del producto. Cada miembro del equipo o función es responsable de representar las necesidades específicas de sus áreas y ninguno es más importante que el resto. Juntas, estas vistas, proporcionan las comprobaciones y ajustes necesarios para asegurarse de que el equipo obtiene la solución correcta.

Existen seis funciones en MSF for Agile:

- Administración de programas: Jefe de Proyecto.
- Arquitectura: Arquitecto.
- Desarrollo: Desarrollador.
- Prueba: Ingeniero de Pruebas.
- Operaciones de Lanzamiento: Jefe de Lanzamiento.
- Experiencia del Usuario: Analista de Negocios.
- Administración de Productos: Analista de Negocios.



#### Ventajas:

- Ayuda al desarrollo rápido de aplicaciones, y separa bien la documentación de las fases de desarrollo.
- Es un proceso ágil de desarrollo de software, dirigido por escenarios y basado en contenido para crear aplicaciones .NET y otras aplicaciones orientadas a objetos.
- Incorpora directamente prácticas para controlar requisitos de calidad de servicio como, por ejemplo, rendimiento y seguridad.

#### Desventajas:

- A los miembros del equipo se les asignan una o más funciones.
- Las actividades pueden producir determinados productos de trabajo y puede que necesiten que los productos de trabajo tengan un determinado estado antes de que se puedan llevar a cabo.
- Todos los miembros del equipo tienen que estar enterados de todo el contenido y actividades del proyecto.

Después de haber realizado un análisis acerca de las características descritas anteriormente, se ha concluido lo siguiente:

- RUP para algunos proyectos es demasiado complejo.
- XP para algunos proyectos es demasiado ligero
- Los productos de trabajo de MSF for Agile son documentos, hojas de cálculo, planes de proyecto, código fuente y otros resultados que son tangibles que se derivan de las actividades.
- MSF for Agile está orientado a crear aplicaciones .Net con orientación a objetos.

Por estas razones se ha determinado utilizar la metodología MSF for Agile que se acopla a nuestro proyecto.

## 1.2.2. HERRAMIENTA CASE PARA EL MODELO RELACIONAL

### 1.2.2.1. Power Designer

Power Designer es un conjunto de herramientas que combina distintas técnicas de modelamiento a través de UML, Procesos Empresariales y Base de datos.

El diagrama es sobre un esquema bastante exacto y completo de cómo los componentes de PowerDesigner trabajan juntos (Ver Figura 1.2.2.1). Pero se muestran todos los archivos y bases de datos.

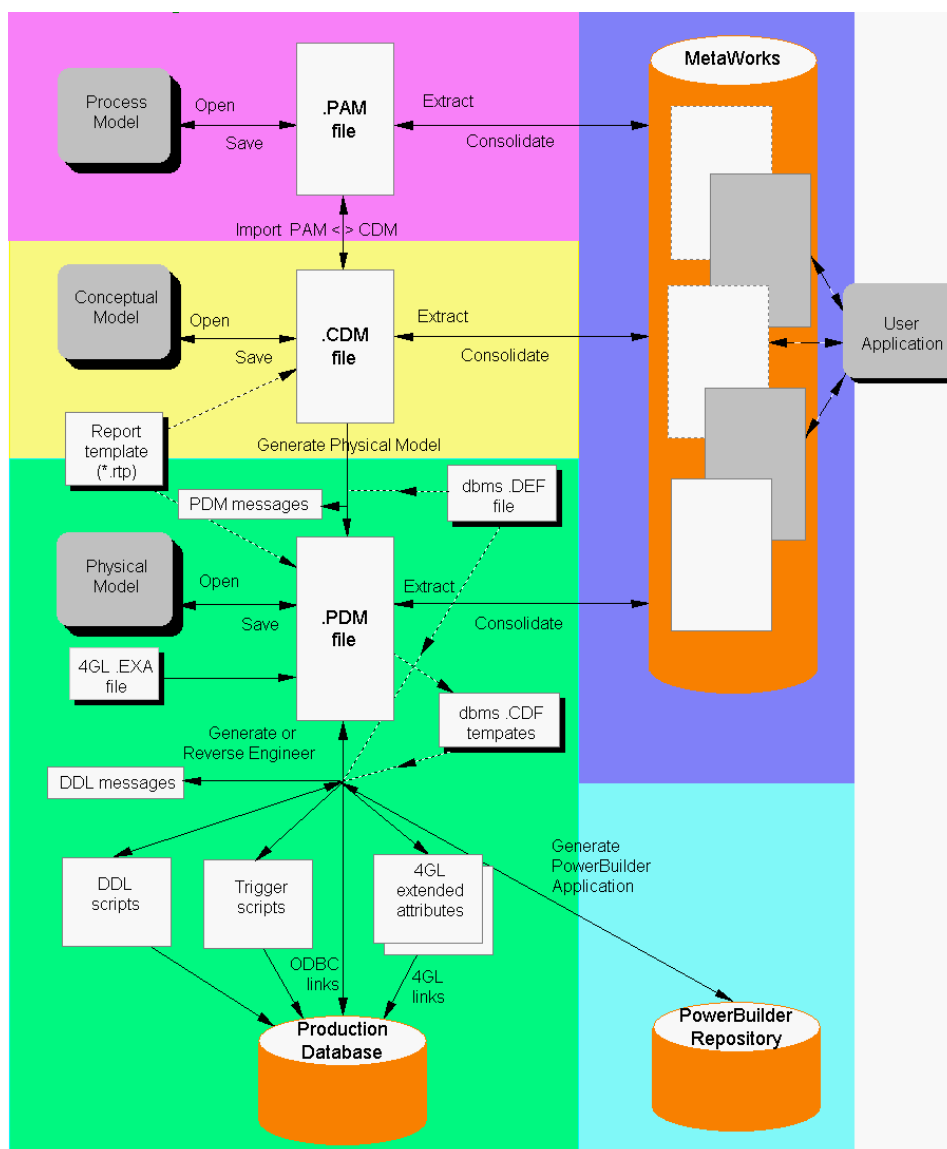


Figura 1.2.2.1 Arquitectura de Power Designer [10]

[10] [http://www.aisintl.com/case/user\\_groups/PowerDesigner/library/sd\\_architecture.html](http://www.aisintl.com/case/user_groups/PowerDesigner/library/sd_architecture.html)

Ventajas:

- Es una herramienta común para el modelado relacional de la base de datos.
- Permite manejar varios modelos al mismo tiempo
- Se puede generar el modelo físico y el script de la base de datos para los motores de bases de datos más comunes.
- Permite hacer reingeniería.

Desventajas:

- Separa el modelo lógico del modelo físico.
- Es complejo manejar varios modelos de la misma base de datos.
- Es complejo manejar características específicas, como nombres de claves foráneas, dominios, etc.

#### **1.2.2.2. Case Studio**

CASE Studio es una herramienta profesional con la que se puede diseñar bases de datos, además facilita herramientas para la creación de diagramas de relación, modelado de datos y gestión de estructuras.

Tiene soporte para trabajar con una amplia variedad de formatos de base de datos (Oracle, SQL, MySQL, PostgreSQL, Access, etc.) y permite además generar scripts SQL, aplicar procesos de retroingeniería (*reverse engineering*) a las bases de datos, usar plantillas de diseño personalizables y crear detallados informes en HTML y RTF. Además, su manejo es intuitivo y fácil de organizar, permitiendo dividir en submodelos y tener una visión a nivel macro de todo el modelo relacional.

Ventajas:

- Es netamente un software para el modelado de la base de datos
- Permite tener distintas vistas del modelo total, agrupando según características comunes.
- Mantiene íntimamente ligados el modelo lógico con el modelo físico.
- Permite hacer reingeniería de base de datos y conversión de modelos a otro tipo de base de datos.

## Desventajas

- No es una herramienta conocida.
- No es una herramienta de libre difusión.
- No se puede trabajar con varios modelos lógicos a la vez.

### 1.2.2.3. Microsoft Visio

Microsoft Visio es un conjunto de software de dibujo vectorial para Microsoft Windows que forma parte de las aplicaciones de la suite ofimática Microsoft Office.

Las herramientas que lo componen permiten realizar diagramas de oficinas, diagramas de bases de datos, diagramas de flujo de programas, UML, y más, que permiten iniciar al usuario en los lenguajes de programación (Ver Figura 1.2.2.2).

Visio comenzó a formar parte de los productos de Microsoft cuando fue adquirida la compañía Visio en el año 2000. Aunque originalmente apuntaba a ser una aplicación para dibujo técnico para el campo de Ingeniería y Arquitectura; con añadidos para desarrollar diagramas de negocios, su adquisición por Microsoft implicó drásticos cambios de directrices de tal forma que a partir de la versión de Visio para Microsoft Office 2003 el desarrollo de diagramas para negocios pasó de añadido a ser el núcleo central de negocio, minimizando las funciones para desarrollo de planos de Ingeniería y Arquitectura que se habían mantenido como principales hasta antes de la compra.

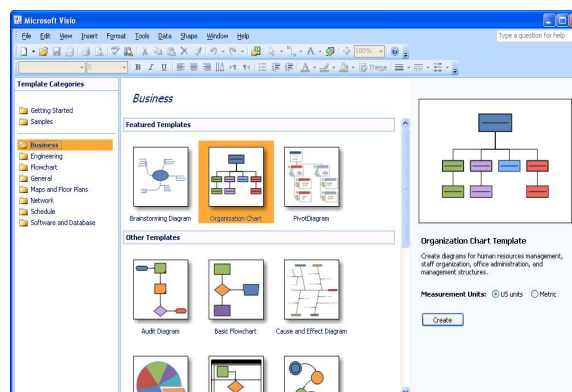


Figura 1.2.2.2 Microsoft Visio [11]

[11] [http://download.microsoft.com/download/8/2/7/82717b95-b244-4f2e-8d31-5580adc84471/Refresh\\_Visio2007\\_Datasheet\\_readyforweb.doc](http://download.microsoft.com/download/8/2/7/82717b95-b244-4f2e-8d31-5580adc84471/Refresh_Visio2007_Datasheet_readyforweb.doc)

#### Ventajas:

- Permite actualizar automáticamente todos los datos de los diagramas y evitar tener que hacerlo manualmente.
- Se puede elegir entre barras, indicadores e íconos de datos para crear una vista de escritorio digital de fácil comprensión que permitirá ver el progreso, las tendencias clave y las excepciones de los proyectos en curso, con lo que se puede detectar más rápidamente los problemas.
- Permite realizar ingeniería reversa.

#### Desventajas:

- La herramienta es muy genérica para diagramas y no se enfoca solamente al modelado de datos, por lo tanto carece de ciertas opciones.
- Permite realizar el modelo de la base de datos pero únicamente la versión Architect permite generar un script para Microsoft SQL Server 2005.
- Dependiendo de la versión Microsoft Visio permite realizar modelos de bases de datos

Después de haber realizado un análisis acerca de las características descritas anteriormente, se ha concluido lo siguiente:

- Power Designer es complejo de utilizar
- Power Designer separa el modelo lógico del modelo físico, dificultando cualquier cambio en el modelo.
- Case Studio es fácil de utilizar
- Case Studio permite realizar conversiones de modelos y mantiene ligados los modelos lógicos y físicos, permitiendo una coordinación simétrica.
- Case Studio permite trabajar con los principales motores de bases de datos conocidos actualmente.

Por estas razones se ha seleccionado la herramienta Case de modelado de Base de Datos "Case Studio".

### 1.2.3. SISTEMA DE ADMINISTRACIÓN DE BASE DE DATOS

#### 1.2.3.1. Microsoft SQL Server 2005

Microsoft SQL Server 2005 es un sistema de administración de base de datos relacionales acompañado de herramientas inteligencia de negocios (Business Intelligence). El motor de la base de datos SQL Server 2005 ofrece acceso controlado y procesamiento rápido de transacciones para mantener una alta disponibilidad de acceso y procesamiento de los datos.

El diagrama mostrado en la Figura 1.2.3.1 ilustra los componentes de Microsoft SQL Server 2005; expone la integración de varias herramientas de negocios que nos permiten almacenar bases de datos relacionales, diseño e implementación de reportes, Análisis de información (Cubos OLAP y Data Mining), y Replicación.



Figura 1.2.3.1 Componentes Básicos de SQL Server 2005 [12]

Ventajas:

- Contiene la versión Express que no necesita licencias y es de libre difusión e incluye el servicio de reportes.
- Tiene las mismas funcionalidades y más de Microsoft SQL Server 2000.
- Trabaja con el Microsoft .Net Framework 2.0
- No solo es un motor de base de datos, sino todo un conjunto de herramientas de negocio.
- Se integra con los usuarios de Active Directory para manejar la seguridad

[12] <http://www.microsoft.com/spain/sql/productinfo/overview/what-is-sql-server.msp>.

Desventajas:

- Trabaja únicamente en plataformas Windows.
- Necesita las últimas actualizaciones de los sistemas operativos para poder instalar.
- Si no se instala en un sistema operativo con dominio, no se puede utilizar los usuarios de Active Directory; solamente la autenticación propia de Microsoft SQL Server.
- Se requiere de licencias para su utilización.

### 1.2.3.2. Oracle

Oracle es un herramienta cliente/servidor para la gestión de base de datos que ofrece desempeño y escalabilidad en servidores Unix y Windows, automatiza tareas administrativas y ofrece Real Application Clusters. Es un producto vendido a nivel mundial, aunque la gran potencia que tiene y su elevado precio hace que solo se vea en empresas muy grandes y multinacionales, por norma general. Se puede observar su estructura en la Figura 1.2.3.2:

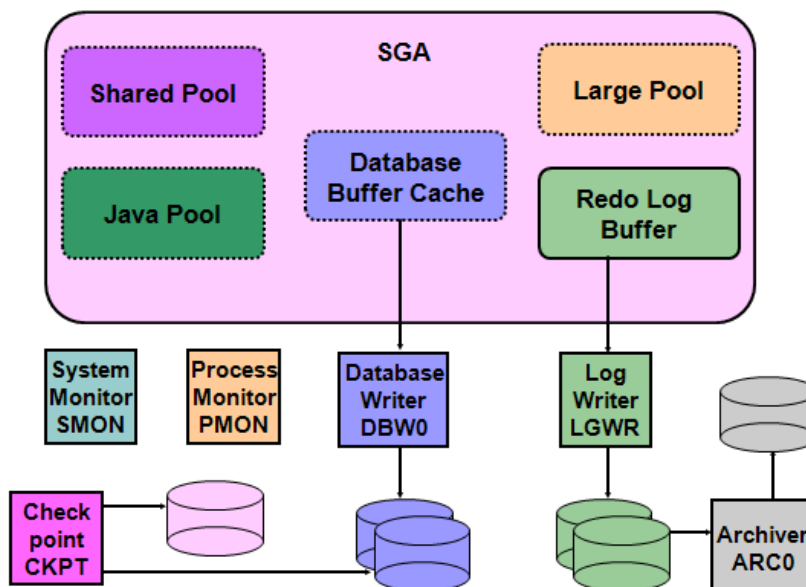


Figura 1.2.3.2 Arquitectura de Oracle [13]

[13] [http://www.bd.cesma.usb.ve/ci5313/docs/taller1\\_Edna.pdf](http://www.bd.cesma.usb.ve/ci5313/docs/taller1_Edna.pdf)

Ventajas:

- Soporta varias transacciones en línea.
- Trabaja en varias plataformas de Sistemas Operativos
- Es uno de los Sistemas Manejadores de Base de Datos Relacional (RDBMS) más usados.

Desventajas:

- Necesita gran capacidad de hardware para su buen desempeño
- El Front End para utilizar el motor de base de datos es muy complejo de utilizar.
- Se requiere instalar varios componentes en el cliente para su utilización.

### **1.2.3.3. Adaptive Server Enterprise (ASE)**

ASE es directo descendiente de Sybase SQL Server (lanzada al mercado en 1988 como la primera base de datos con arquitectura cliente/servidor) y su cambio de nombre se produjo a partir de la versión 11.5, en 1996, para evitar confusiones con Microsoft SQL Server, con el que comparte un origen común (Sybase licenció el código a Microsoft para el sistema operativo Windows). En 1998, se lanzó ASE 11.9.2, con soporte al bloqueo a nivel de registro y rendimiento mejorado en ambientes SMP. ASE 12.0 fue liberado en 1999, brindando soporte para Java en la base de datos, alta disponibilidad y gestión de transacciones distribuidas. En 2001, ASE 12.5 fue lanzada, con características tales como asignación dinámica de memoria, soporte para XML en la base de datos y conexiones seguras con SSL, entre otros. En septiembre de 2005, Sybase lanzó al mercado ASE 15.

Ventajas:

- Posee soporte a múltiples herramientas de desarrollo y lenguajes de programación, como PowerBuilder, Visual Basic, Java, C, PHP, etc.
- Posee soporte a múltiples protocolos de conectividad, como Open Client (propio de Sybase), ODBC, OLE DB, ADO.NET y JDBC.



- Tiene un administrador lógico de recursos y tareas y permite realizar copias de respaldo en línea y de alto rendimiento.

#### Desventajas:

- Es compatible solo con aplicaciones realizadas en Java
- Para implantaciones en ambientes Windows se necesita de licencias, solo hay una versión libre limitada para Linux.
- Cuando ocurre un error en el servidor de base de datos, es muy complejo volverlo a instalar o crear una nueva instancia.

Después de haber realizado un análisis acerca de las características descritas anteriormente, se ha concluido lo siguiente:

- La versión Express de Microsoft SQL Server 2005 no necesita de licencias para su instalación y no necesita muchas características de hardware.
- SQL Server 2005 al ser compatible con SQL Server 2000, facilita la curva de aprendizaje de los usuarios.
- Oracle es utilizado para sistemas complejos.
- Sybase ASE u Oracle garantizan la integridad, consistencia y congruencia de la información que gestionan, pero necesitan de un gran soporte de hardware.
- SQL Server 2005 se acopla perfectamente con aplicaciones .Net
- En la Figura 1.2.3.3 podemos ver el benchmarking TPC (Transaction Processing Council) que indica que Microsoft SQL Server 2005 supera a Oracle en transacciones por minuto y ahorra en costos.



### Figura 1.2.3.3 BenchMark TPC entre Microsoft SQL Server 2005 y Oracle [14]

Por las razones anteriormente descritas se escoge como Sistema de Base de Datos a Microsoft SQL Server 2005.

## 1.2.4. TECNOLOGÍAS DE DESARROLLO.

### 1.2.4.1. Microsoft .Net Framework

.Net Framework es la plataforma de desarrollo de software de Microsoft para los sistemas operativos Windows; permite un desarrollo rápido de aplicaciones por las funcionalidades que trae consigo. Utiliza lenguajes de programación orientados a objetos y su fuerte es su arquitectura para ambientes distribuidos.

#### Ventajas

- Permite utilizar componentes escritos en varios lenguajes de programación.
- Permite crear componentes reutilizables
- Según un estudio de “Evans Data” Microsoft Visual Studio es el más usado y dominante entorno de desarrollo por encima de Borland JBuilder, Eclipse, NetBeans, IBM Rational Application Developer. [15]
- Puede conectarse a las bases de datos más comunes, especialmente con Microsoft SQL Server 2005.
- Es netamente orientado a objetos.
- Existen versiones Express de libre difusión del IDE que permiten crear aplicaciones robustas y no necesitan licencias.
- La versión del Microsoft .Net Framework 3.5 trae mejoras en el lenguaje como la inclusión de programación funcional y nuevas arquitecturas que permiten desarrollar de manera más rápida las aplicaciones.

#### Desventajas

- Trabaja únicamente solo en plataformas Windows
- Requiere las últimas versiones de Sistemas Operativos Windows.

---

[14] <http://www.microsoft.com/sql/prodinfo/compare/oracle>

[15] <http://www.internetnews.com/dev-news/article.php/3616626>

- El IDE Visual Studio 2008 en las versiones Estándar, Profesional, Team Suite necesitan de licencias para su uso, ya que contienen herramientas que agilitan el desarrollo de software.

#### 1.2.4.2. Java

Java es un lenguaje de programación orientado a objetos que pertenece a Sun Microsystems, que permite crear aplicaciones de escritorio, aplicaciones web, aplicaciones móviles en distintos sistemas operativos El estándar que más se utiliza para desarrollar aplicaciones en Java es J2EE. En la Figura 1.2.4.1 se muestra la arquitectura de Java:

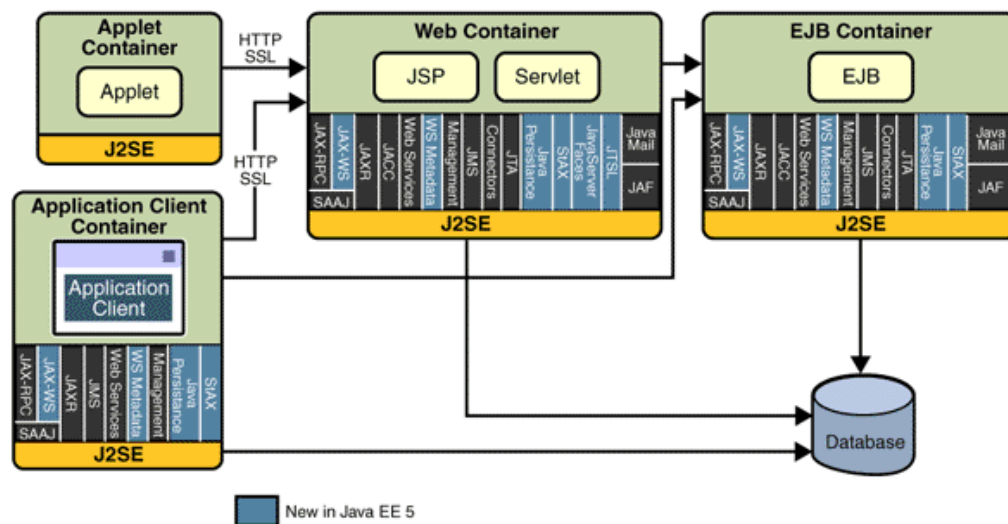


Figura 1.2.4.1 Arquitectura de Java [16]

Ventajas:

- Es netamente orientado a objetos
- Trabaja en varias plataformas
- Es portable
- Es de difusión libre

Desventajas:

- La mayoría de entornos de desarrollo que soportan Java no son muy amigables.
- Es complejo realizar una interfaz de usuario amigable.

[16] <http://www.jtech.ua.es/j2ee/2006-2007/jee.html>

- Requiere gran capacidad de memoria.
- Utiliza un solo lenguaje

Después de haber realizado un análisis acerca de las características descritas anteriormente, se ha concluido lo siguiente:

- Microsoft .Net Framework trabaja con varios lenguajes
- Microsoft .Net Framework se acopla perfectamente a sistemas operativos Windows y a Microsoft SQL Server 2005.
- El Entorno de Desarrollo de Microsoft .Net Framework permite realizar un desarrollo rápido de aplicaciones.
- Java requiere muchos recursos para correr sus programas
- El Entorno de Desarrollo de Java no es muy amigable y requiere mucha más programación para el desarrollo de los sistemas.

Por las razones antes descritas y por la experiencia de haber desarrollado con estas herramientas se escoge como Tecnología de Desarrollo a Microsoft .Net Framework acompañado del IDE Microsoft Visual Studio 2008 y el lenguaje de programación C# 3.5.

## **CAPITULO II**

### **ANÁLISIS Y DISEÑO**

En este capítulo se identifican los escenarios, para cada escenario se elije el rol apropiado y se describe la descripción del mismo. Una vez que se tiene un listado de escenarios se establece la prioridad de cada uno de ellos. Se determinan los casos de uso, se describe brevemente cada uno de ellos y se identifican los roles del sistema.

Además se identifica los factores de calidad o los requisitos que no están expresados en los escenarios, una vez identificados se determina la importancia de cada requisito y la prioridad general de requisitos. Se diseña el diagrama de aplicación, la arquitectura del sistema, el diagrama de clases y el modelo relacional.

#### **2.1 ESPECIFICACIÓN DE REQUERIMIENTOS**

##### **2.1.1. VISIÓN DEL SISTEMA**

###### **2.1.1.1. Orígenes**

La empresa Rosadex Cia. Ltda. se dedica a la producción y exportación de flores a Europa, Norteamérica, y Asia. El crecimiento de la demanda de sus productos requiere un número considerable de trabajadores para las áreas de producción, postCosecha y exportación de su producto. La mayoría de empleados de Rosadex trabaja en contacto constante con fungicidas y fertilizantes que afectan a su salud; por lo que se exige un control médico conducido, basado en las sugerencias indicadas por el seguro social, quien establece los exámenes y controles necesarios para este tipo de empresa. La importancia de la información que debe llevar el departamento médico requiere una gestión eficiente y confiable de las historias clínicas de cada empleado. Además, la mayoría de los diagnósticos realizados por el médico, requieren su medicamento y tratamiento

pertinente, razón por la cual el médico necesita registrar los medicamentos recetados y entregados a sus pacientes.

El médico de la empresa requiere generar reportes resumidos sobre Enfermedades, Consultas, Medicamentos y Exámenes realizados a los pacientes, como apoyo a la toma de decisiones de la Gerencia en relación al manejo de rotación y salida del personal. La gran cantidad de datos clínicos y los medicamentos recetados que maneja el médico de la empresa dificulta el desempeño de su trabajo, y en ocasiones llega a descuidar su principal labor de atender a los empleados, por la necesidad de presentar la documentación pertinente y los reportes resumidos.

#### **2.1.1.2. Factores determinantes**

La empresa está logrando una gran cobertura en el mercado por lo cual cada vez demanda más empleados, esto implica que se requiere generar reportes, más seguidos y precisos para conseguir una adecuada y efectiva gestión de rotación y salida de personal, como apoyo a la toma de decisiones en el ámbito de recursos humanos.

La falta de control de medicamentos sugiere que el sistema además de registrar los medicamentos existentes en el departamento médico también registre aquellos medicamentos recetados y entregados por el médico a sus pacientes, también se deben registrar las consultas y exámenes realizados a cada empleado, especialmente los exámenes exigidos por el IESS.

El sistema no necesita generar facturas de las consultas, porque el servicio médico corre por cuenta de la empresa; pero si necesita registrar los medicamentos entregados a cada paciente para su posterior cobro por parte de recursos humanos.

El sistema debe generar reportes de a la Gerencia con información resumida sobre Enfermedades, Consultas, Medicamentos y Exámenes como apoyo a la planificación de los recursos humanos.

### 2.1.1.3. Definición de la visión

Para facilitar el trabajo del médico de Rosadex, quien atiende a los pacientes y registra la información de las fichas médicas, se desarrollará un sistema informático, que le permitirá registrar las historias clínicas, exámenes, citas, consultas, y medicamentos prescritos a los empleados, así como también generar reportes resumidos de la información registrada, como apoyo a la toma de decisiones de la gerencia; a diferencia del proceso actual que dificulta consolidar la información de manera rápida y fácil.

## 2.1.2 ESCENARIOS

### 2.1.2.1 Gestionar las Historias Clínicas: Registrar Pacientes

#### 2.1.2.1.1 Roles alternativos

El médico es el usuario encargado de ingresar los datos de los pacientes en el sistema, una vez que haya registrado toda la información puede generar la historia clínica respectiva. El médico puede usar la información recopilada para realizar el diagnóstico adecuado, definir el tratamiento y medicamentos que serán prescritos a su paciente. También puede actualizar la información con los exámenes realizados a los pacientes y evolución del tratamiento administrado.

Rol	Desviaciones (cuando corresponda)
Médico	Actualiza la información referente a los empleados de la empresa.  Consulta de los datos ingresados.
Secretaria	Consulta de los datos ingresados.

**Tabla 2.1.2.1 Roles Alternativos del Escenario Gestionar las Historias Clínicas: Registrar Pacientes**



#### *2.1.2.1.2 Descripción de escenario*

Para registrar los datos del paciente, el médico ingresa a “Historia Clínica”, en donde existe la sección “Paciente” y escoge la opción “Datos Personales” y se presenta una nueva pantalla que permite la gestión de datos personales del paciente.

El médico presiona el botón <Nuevo> para ingresar la información requerida, al presionar el botón <Modificar> edita los datos ingresados, una vez que termina la edición y/o inserción de los datos, presiona el botón <Guardar> para guardar la información, como desea visualizar los datos que ingresó presiona el botón <Imprimir> y aparece un reporte con los datos existentes de la historia clínica que acaba de ingresar, pero desea borrar los datos ya ingresados así que presiona el botón <Eliminar> y aparece un mensaje que solicita la confirmación para eliminar los datos, entonces el médico acepta el mensaje y los datos registrados se borran del sistema, también se puede cancelar el ingreso o modificación de los datos al presionar el botón <Cancelar>.

Los datos del paciente incluyen los nombres, apellidos, dirección, teléfono, el nombre de una persona de referencia, el teléfono de la persona de referencia, fecha de nacimiento, ciudad, nacionalidad, ciudad de residencia actual, ciudad de residencia anterior, sexo, estado civil, número de hijos, profesión, grupo sanguíneo, plan médico, tipo de seguro, seguro, cargo anterior y cargo actual.

#### **2.1.2.2 Gestionar las Consultas: Registrar Consulta**

##### *2.1.2.2.1 Roles alternativos*

El médico es el encargado de registrar las consultas a las cuales acuden los empleados de la empresa. El médico requiere esta información para llevar un control del estado de salud del paciente y la frecuencia con la que asiste a una consulta. El médico puede utilizar esta información para generar los informes resumidos para la gerencia.

Rol	Desviaciones (cuando corresponda)
Médico	Actualizar los datos del informe Consulta. Consulta de los datos ingresados.
Secretaria	Consulta de los datos ingresados.

**Tabla 2.1.2.2 Roles Alternativos del Escenario Gestionar las Consultas: Registrar Consulta**

#### 2.1.2.2.2 Descripción de escenario

Para registrar los datos de la Consulta, el médico ingresa a “Historia Clínica”, en donde existe la sección “Servicios” y escoge la opción “Consulta” y aparece una nueva pantalla que permite la gestión de las consultas.

Esta pantalla tiene los botones: Nuevo, Modificar, Guardar, Eliminar, Imprimir, Cancelar y Salir.

El médico presiona el botón <Nuevo> para ingresar la información requerida en la consulta, al presionar el botón <Modificar> edita los datos ingresados, una vez que termina la edición y/o inserción de los datos, presiona el botón <Guardar> para guardar la información, como desea visualizar los datos que ingresó presiona el botón <Imprimir> y aparece un reporte con los datos existentes de la consulta, pero desea borrar los datos ya ingresados así que presiona el botón <Eliminar> y aparece un mensaje que solicita la confirmación para eliminar los datos, entonces el médico acepta el mensaje y los datos registrados se borran del sistema, también se puede cancelar el ingreso o modificación de los datos al presionar el botón <Cancelar>.

Además desde esta pantalla puede recetar medicamentos y registrar exámenes a través de los botones <Recetar Medicamentos> y <Registrar Exámenes> respectivamente.

Los datos de la consulta incluyen el código de la historia clínica, departamento, médico, paciente, cita, fecha de la consulta, motivo de la consulta, si toleró la

medicación, la evolución de la enfermedad, si cumplió las indicaciones, tipo de impedimento y el diagnóstico presuntivo.

### **2.1.2.3 Gestionar Toma de Decisiones: Generar Reportes Resumidos**

#### *2.1.2.3.1 Roles alternativos*

El médico es el encargado de generar los reportes resumidos como apoyo a la Gerencia en la toma de decisiones con respecto a la gestión de personal. La información que se obtiene de los empleados de la empresa, permite al médico generar los reportes estadísticos de Enfermedades, Consultas, Medicamentos y Exámenes como apoyo a la planificación de los recursos humanos.

#### *2.1.2.3.2 Descripción de escenario*

Para generar los reportes resumidos, el médico ingresa a “Reportes” en donde se muestra un menú con las opciones de “Enfermedades”, “Consultas”, “Medicamentos”, y “Exámenes”.

El médico hace clic sobre la opción que desea para generar el reporte, y aparece una nueva pantalla con las opciones para generar según cada reporte:

Enfermedades: Se muestra en una grilla las patologías existentes y hace clic en cada patología que desea mostrar en el reporte, escoge la fecha inicial y la fecha final haciendo clic en un calendario para cada fecha.

Consultas: escoge la fecha inicial y la fecha final haciendo clic en un calendario para cada fecha.

Medicamentos: se muestra en una grilla los medicamentos existentes y hace clic en cada medicamento que desea mostrar en el reporte, escoge la fecha inicial y la fecha final haciendo clic en un calendario para cada fecha.

Exámenes: se muestra en una grilla los diferentes tipos de exámenes y hace clic en cada tipo de examen que desea mostrar en el reporte, escoge la fecha inicial y la fecha final haciendo clic en un calendario para cada fecha.

Para todos los reportes antes mencionados presiona el botón <Generar Reporte> entonces se visualiza en un gráfico de barras y una tabla con los datos estadísticos, presiona el botón <imprimir> y se envía a la impresora que escoja; o puede presionar el botón <exportar> y escoge el formato en el que quiere exportar (Excel o PDF).

En el reporte de Enfermedades se obtiene el número de pacientes por cada enfermedad registrada en cada consulta.

En el reporte de Consultas se obtiene el número de consultas por fecha y los motivos de consulta más comunes.

En el reporte de Medicamentos se obtiene el número de pacientes por cada medicamento que se ha recetado en cada consulta.

En el reporte de Exámenes se obtiene el número de exámenes por cada tipo de examen.

*La descripción completa de los escenarios se encuentra disponible en el Anexo 1*

## **2.2 ANÁLISIS**

### **2.2.1 CASOS DE USO**

#### **2.2.1.1 Principal**

El Médico y la secretaria son los encargados de gestionar los medicamentos para su prescripción y entrega, y las citas de los pacientes; así como también el médico es el encargado de registrar los antecedentes y las consultas del paciente, y generar los reportes estadísticos. El diagrama del caso de uso lo podemos ver en la siguiente figura:

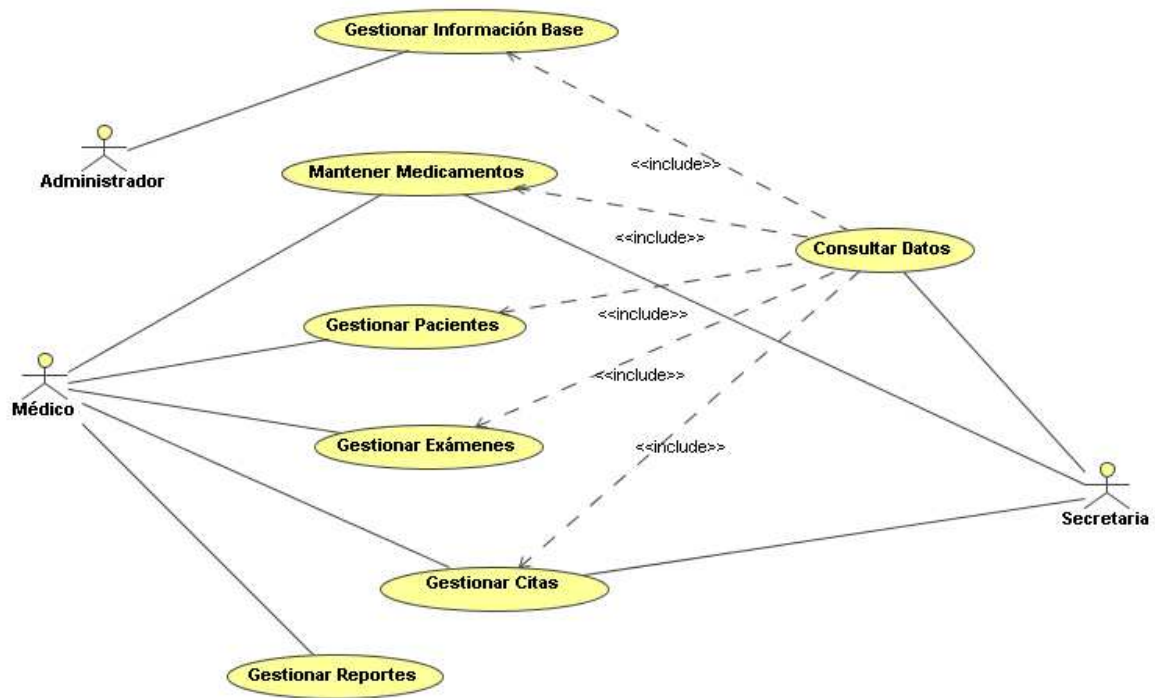
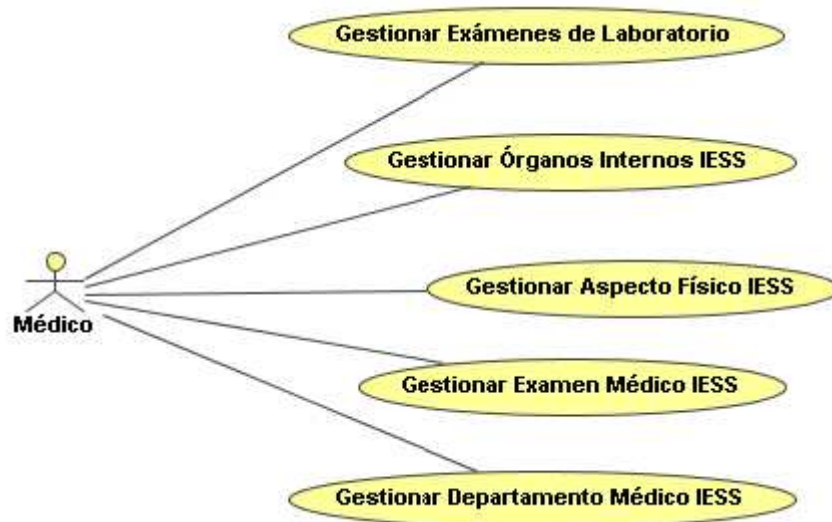


Figura 2.2.1.1 Caso de Uso Principal

### 2.2.1.2 Caso de Uso Gestionar Exámenes

En este caso de uso el médico es el que se encarga de registrar los diferentes tipos de exámenes, tanto exámenes del IESS como exámenes de laboratorio recetados. Cada examen del IESS contiene campos específicos que tienen que ser llenados dependiendo del tipo de examen, en cambio en los exámenes de laboratorio se registra el tipo de examen y los resultados. El diagrama del caso de uso lo podemos ver en la siguiente figura:



**Figura 2.2.1.2 Caso de Uso Gestionar Exámenes**

### **2.2.1.3 Caso de Uso Medicamentos**

En este caso de uso se maneja toda la gestión de los medicamentos, tanto para la entrega como para la medicación. El médico y la secretaria se encargan de registrar la creación, ingreso, y entrega de medicamentos; así como también el doctor se encarga de recetar los medicamentos en la consulta que realiza al paciente. El diagrama del caso de uso lo podemos ver en la siguiente figura:

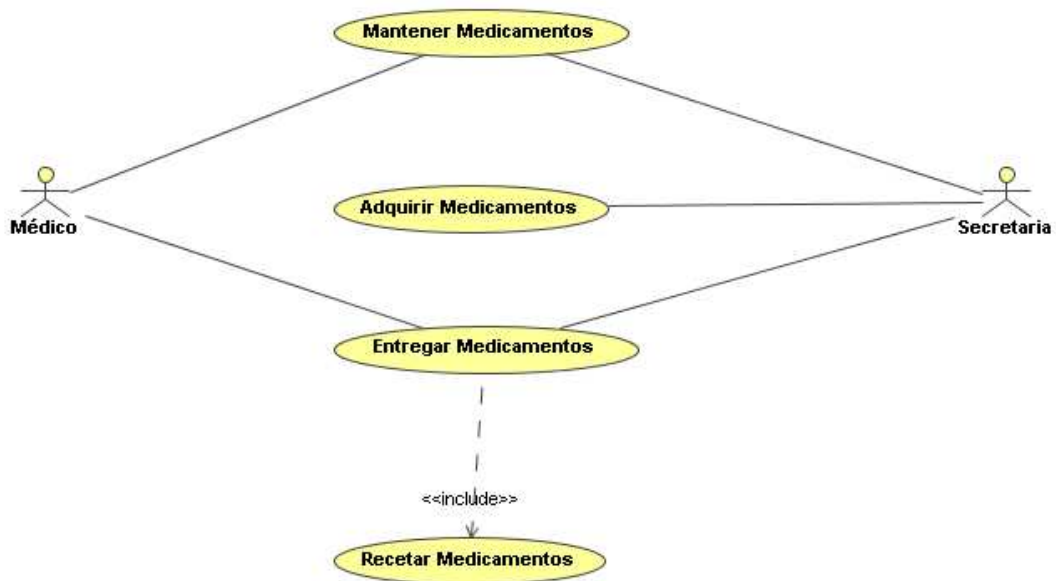


Figura 2.2.1.3 Caso de Uso Medicamentos

#### 2.2.1.4 Caso de Uso Pacientes

En este caso de uso el médico gestionar todos los datos necesarios para formar la historia clínica del paciente, desde los antecedentes del mismo hasta las consultas y medicamentos recetados. El diagrama del caso de uso lo podemos ver en la siguiente figura:

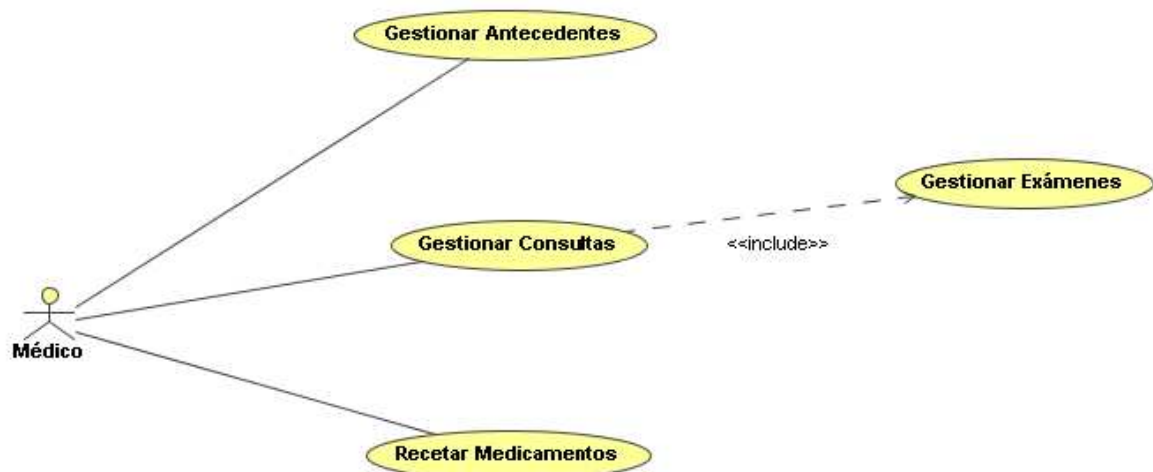


Figura 2.2.1.4 Caso de Uso Pacientes

### 2.2.1.5 Caso de Uso Reportes

En este caso de uso el médico genera reportes estadísticos de exámenes, enfermedades, consultas, y medicamentos; de esta forma el médico puede ver de forma global el estado de salud de los empleados y los medicamentos utilizados y recetados. El diagrama del caso de uso lo podemos ver en la siguiente figura:

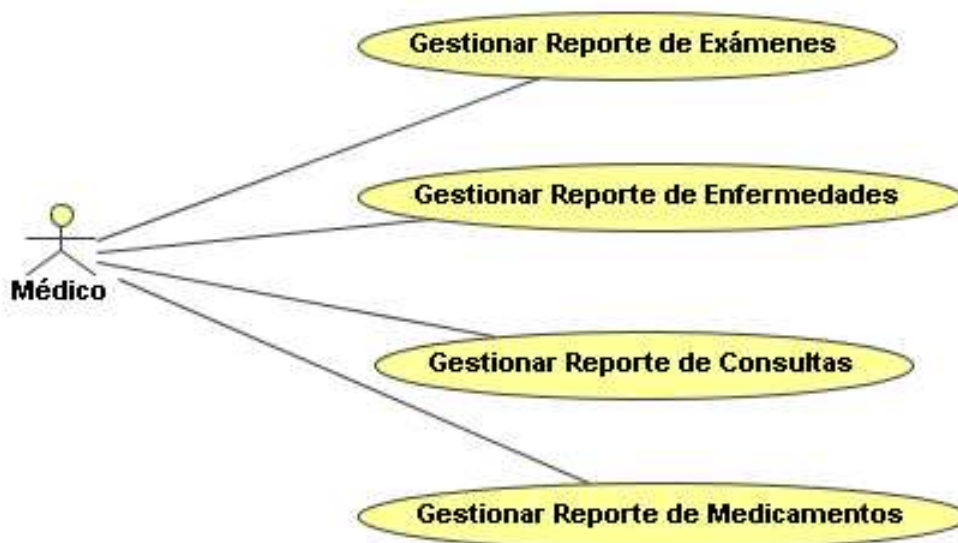


Figura 2.2.1.5 Caso de Uso Reportes

### 2.2.2 REQUISITOS DE CALIDAD

ID	Tipo de elemento de Trabajo	Título	Tipo de calidad de servicio	Complejidad	Descripción
1	Requisito de calidad de servicio	Usuarios concurrentes	Cargar	Media	Existe un número máximo de usuarios que pueden acceder concurrentemente al sistema.
2	Requisito de calidad de servicio	Inicio de Sesión	Seguridad	Media	Cada usuario debe tener un login y un password para acceder al sistema.
3	Requisito de calidad de servicio	Sistema Operativo	Plataforma	Baja	La aplicación se ejecutará sobre plataformas Windows, estándar de la organización.



4	Requisito de calidad de servicio	Tolerancia a Errores	Rendimiento	Alta	Validar el ingreso de la información y garantizar la ejecución correcta de los procesos.
5	Requisito de calidad de servicio	Gestión de Perfiles de Usuarios	Seguridad	Media	La disponibilidad de cada una de las funcionalidades del sistema operativo dependerán del perfil de usuario seleccionado.
6	Requisito de calidad de servicio	Interfaz familiar	Usabilidad	Baja	El sistema será fácil de usar y la curva de aprendizaje será mediante una interfaz conocida

**Tabla 2.2.2.1 Requisitos de Calidad**

## 2.2.3 ROLES

### 2.2.3.1 Administrador

Su función es registrar los respectivos usuarios del sistema así como asignar los diferentes perfiles que le corresponda a cada uno.

### 2.2.3.2 Médico

Su función es registrar los diagnósticos, exámenes y medicamentos prescritos a sus pacientes, en sus respectivas historias clínicas. Ingresar al Sistema las consultas y tratamientos asignados a los empleados.

### 2.2.3.3 Secretaria

Su función es registrar y verificar las citas a las cuales los empleados fueron convocados por el Médico para su respectiva evaluación y diagnóstico.

*La descripción completa de los roles se encuentra disponible en el Anexo 2*

## 2.2.4 PLAN DE ITERACIÓN

A continuación se muestra el Plan de Iteración:

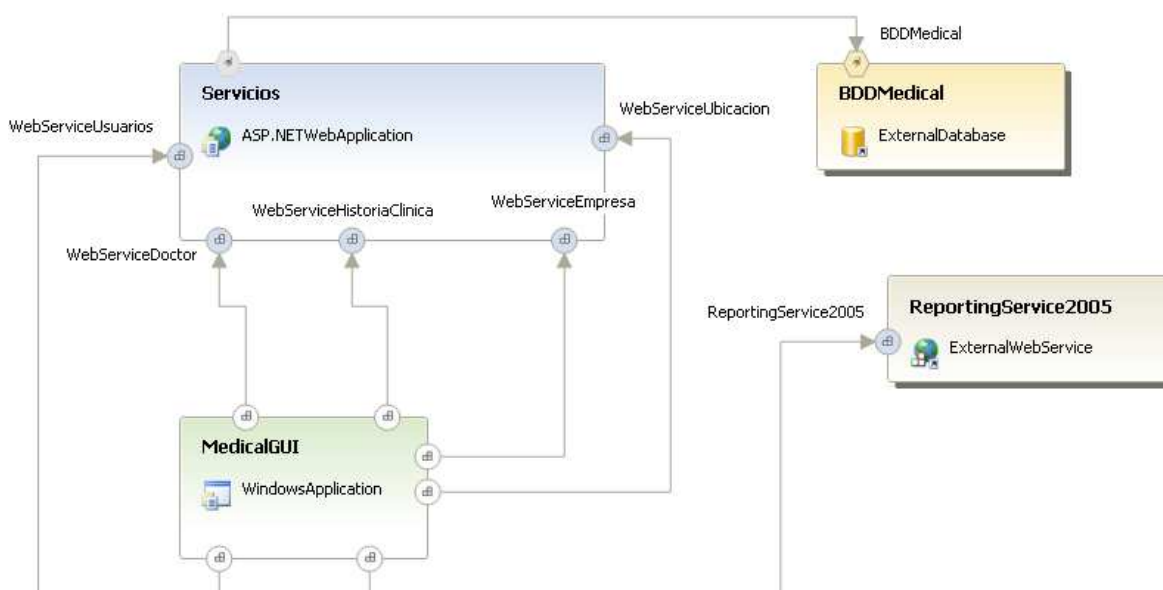
	Nombre de tarea	Duración	Comienzo	Fin
	<b>Medical</b>	<b>234 días</b>	<b>lun 16/07/07</b>	<b>jue 05/06/08</b>
	<b>Iteración 0</b>	<b>29 días</b>	<b>lun 16/07/07</b>	<b>jue 23/08/07</b>
	Visión del Sistema	5 días	lun 16/07/07	vie 20/07/07
	Diagrama de Casos de Uso	3 días	lun 23/07/07	mié 25/07/07
	Especificación de los Roles	1 día	jue 26/07/07	jue 26/07/07
	<b>Descripción de Escenarios</b>	<b>18 días</b>	<b>vie 27/07/07</b>	<b>mar 21/08/07</b>
	Escenario: Gestionar Personal Médico	3 días	vie 27/07/07	mar 31/07/07
	Escenario: Gestionar Historia Clínica	3 días	mié 01/08/07	vie 03/08/07
	Escenario: Administración de Medicamentos	3 días	lun 06/08/07	mié 08/08/07
	Escenario: Gestionar Consultas	3 días	jue 09/08/07	lun 13/08/07
	Escenario: Controlar Exámenes del IESS	3 días	mar 14/08/07	jue 16/08/07
	Escenario: Gestionar Toma de decisiones	3 días	vie 17/08/07	mar 21/08/07
	Requisitos de Calidad de Servicio	2 días	mié 22/08/07	jue 23/08/07
	<b>Iteración 1</b>	<b>25 días</b>	<b>vie 24/08/07</b>	<b>jue 27/09/07</b>
	Diagrama de la Aplicación	5 días	vie 24/08/07	jue 30/08/07
	Arquitectura del Sistema	5 días	vie 31/08/07	jue 06/09/07
	Diagrama de Clases	10 días	vie 07/09/07	jue 20/09/07
	Diagrama del Modelo Relacional	5 días	vie 21/09/07	jue 27/09/07
	<b>Iteración 2</b>	<b>165 días</b>	<b>vie 28/09/07</b>	<b>jue 15/05/08</b>
	Desarrollo de la Capa de Entidades	20 días	vie 28/09/07	jue 25/10/07
	Desarrollo de la capa de Negocio	20 días	vie 26/10/07	jue 22/11/07
	Desarrollo de la Capa de Servicios	20 días	vie 23/11/07	jue 20/12/07
	Investigación Smart Client Software Factory	20 días	vie 21/12/07	jue 17/01/08
	<b>Desarrollo de la Interfaz de Usuario</b>	<b>65 días</b>	<b>vie 18/01/08</b>	<b>jue 17/04/08</b>
	Desarrollo de la Interfaz de Usuario de Personal Médico	5 días	vie 18/01/08	jue 24/01/08
	Desarrollo de la Interfaz de Usuario de Historia Clínica	15 días	vie 25/01/08	jue 14/02/08
	Desarrollo de la Interfaz de Usuario de Medicamentos	10 días	vie 15/02/08	jue 28/02/08
	Desarrollo de la Interfaz de Usuario de Consultas	15 días	vie 29/02/08	jue 20/03/08
	Desarrollo de la Interfaz de Usuario de Exámenes del IESS	20 días	vie 21/03/08	jue 17/04/08
	Desarrollo de Reportes	10 días	vie 18/04/08	jue 01/05/08
	Generación del Producto	10 días	vie 02/05/08	jue 15/05/08
	<b>Iteración 3</b>	<b>15 días</b>	<b>vie 16/05/08</b>	<b>jue 05/06/08</b>
	Pruebas del Sistema	10 días	vie 16/05/08	jue 29/05/08
	Evaluación del Sistema	5 días	vie 30/05/08	jue 05/06/08

Figura 2.2.4.1 Plan de Iteración

## 2.3 DISEÑO

### 2.3.1 DIAGRAMA DE APLICACIÓN

A continuación se puede observar el diagrama de la Aplicación (Ver Figura 2.3.1.1):



**Figura 2.3.1.1 Diagrama de Aplicación**

### 2.3.2 COMPONENTES DE LA APLICACIÓN

A continuación se describen cada uno de los componentes que integran la Aplicación.

<b>BDDMedical:</b>	La base de datos se la implementará en un servidor Sql Server 2005
<b>Servicios:</b>	Se utilizarán WebServices para trabajar de manera distribuida con los clientes, y para evitar el acceso directo a la base de datos por parte de la aplicación cliente; los webservices funcionarán sobre un servidor de aplicaciones con Internet Information Server
<b>MedicalGUI:</b>	La aplicación cliente será implementada en una Aplicación Windows, porque sistema correrá dentro de la intranet
<b>Reporting Services 2005:</b>	El sistema contará con el servidor de reportes de Sql Server 2005, el cual expone los reportes para que sean consumidos desde la

	aplicación cliente
--	--------------------

Tabla 2.3.2.1 Descripción de los Componentes de la Aplicación

### 2.3.3 ARQUITECTURA DEL SISTEMA

A continuación se puede observar la Arquitectura del Sistema:

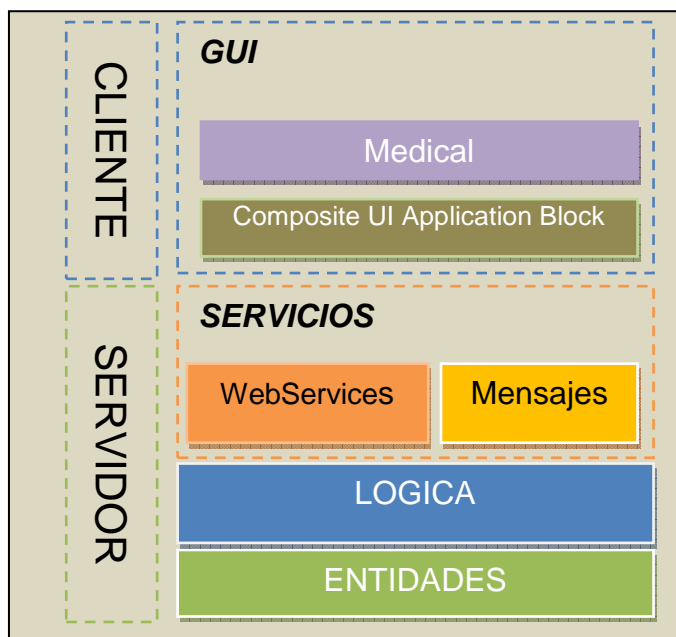


Figura 2.3.3.1 Arquitectura del Sistema

#### 2.3.3.1 Servidor:

A continuación se describen los componentes del Servidor:

<b>Entidades:</b>	La base de datos se la implementará en un servidor Sql Server 2005
<b>Logica:</b>	Se utilizarán WebServices para trabajar de manera distribuida con los clientes, y para evitar el acceso directo a la base de datos por parte de la aplicación cliente; los webservices funcionarán sobre un servidor de aplicaciones con Internet Information Server
<b>WebServices/Mensajes:</b>	Se utilizarán WebServices para trabajar de manera distribuida con los clientes, y para evitar el acceso directo a la base de datos por parte de la aplicación cliente; los webservices funcionarán sobre un servidor de aplicaciones con Internet Information Server

**Tabla 2.3.3.1 Descripción de los Componentes del Servidor**

### 2.3.3.2 Cliente:

A continuación se describen los componentes del Cliente:

<b>Composite UI Application Block:</b>	Es un application block <sup>1</sup> de Microsoft Patterns && Practices que contiene las mejores prácticas para el desarrollo de “Smart Clients”. <sup>2</sup>
<b>Medical:</b>	La aplicación cliente será implementada en una Aplicación Windows, porque sistema correrá dentro de la intranet

**Tabla 2.3.3.2 Descripción de los Componentes del Cliente**

<sup>1</sup> Application Block: Es un framework de desarrollo de software que permite construir aplicaciones basadas en patrones de diseño; sin tener que comenzar desde cero.

<sup>2</sup> Smart Clients: Aplicaciones clientes fáciles de distribuir y gestionar que proporcionan una experiencia adaptativa, responsiva y rica a través del aprovechamiento de recursos locales e inteligentemente conectando a fuentes distribuidas de datos.

### 2.3.4 DIAGRAMA DE CLASES

**Figura 2.3.4.1 Diagrama de Clases**

*El diccionario de datos de las clases se encuentra disponible en el Anexo 3*



### **2.3.5 MODELO RELACIONAL**

**Figura 2.3.5.1 Modelo Relacional**

## CAPITULO III

### IMPLEMENTACIÓN, PRUEBAS Y EVALUACIÓN

En este capítulo se determina el estándar de codificación para el desarrollo de clases en C#, y se identifican las capas que componen el sistema.

Para realizar las pruebas se valida la funcionalidad del sistema por cada escenario definido, se realiza la evaluación utilizando métricas internas y externas de calidad del producto de software.

#### 3.1 IMPLEMENTACIÓN

##### 3.1.1 ESTÁNDAR DE CODIFICACIÓN PARA EL DESARROLLO DE CLASES EN C#

**Nota:** *No se utilizará notación húngara. No se utilizará el carácter de subrayado.*

###### 3.1.1.1 Definiciones

###### 2.3.5.1.1 *Mayúsculas y minúsculas Pascal*

La primera letra del identificador y la primera letra de las siguientes palabras concatenadas están en mayúsculas. El estilo de mayúsculas y minúsculas Pascal se puede utilizar en identificadores de tres o más caracteres. Por ejemplo:

**BackColor**

###### 2.3.5.1.2 *Mayúsculas y minúsculas Camel*

La primera letra del identificador está en minúscula y la primera letra de las siguientes palabras concatenadas en mayúscula. Por ejemplo:

**backColor**

###### 3.1.1.2 Nomenclatura

###### 2.3.5.1.3 *Nomenclatura de una Clase*

A continuación se describe la Nomenclatura de una Clase:

<b>Estructura:</b>	<b>Clase</b>
<b>Estilo:</b>	Mayúsculas y minúsculas Pascal
<b>Observaciones:</b>	Para nombrar a las clases heredadas se debe poner al último el nombre de la clase base solo cuando sea necesario de especificar, en clases muy generales no es necesario poner la palabra de la clase base.
<b>Ejemplo:</b>	<b>MiPrimeraClase</b>
<b>Estructura:</b>	<b><i>Propiedades, Enumeraciones, Valores de Enumeraciones, Eventos, Interfaz:</i></b>
<b>Estilo:</b>	Mayúsculas y minúsculas Pascal
<b>Observaciones:</b>	Los atributos privados que hagan referencia a las propiedades se debe usar el “_” guión bajo seguido del nombre del atributo con estilo camel.
<b>Ejemplo:</b>	<b>BackColor</b>

**Tabla 3.1.1.1 Nomenclatura de Clases**

#### 2.3.5.1.4 Nomenclatura de Enumeraciones, Valores de Enumeraciones, Eventos e Interfaz

A continuación se describe la Nomenclatura de Enumeraciones, Valores de Enumeraciones, Eventos e Interfaz:

<b>Estructura:</b>	<b><i>Enumeraciones, Valores de Enumeraciones, Eventos, Interfaz:</i></b>
<b>Estilo:</b>	Mayúsculas y minúsculas Pascal
<b>Observaciones:</b>	Se recomienda utilizar Flags Attributes que son usados para determinar el tipo de enumeración

<b>Ejemplo:</b>	<pre>// This enum is used for demonstration of flags editor functionality [Flags] enum MyFlagsEnum {     // Hidden fields (will not be displayed in editor)     [Browsable(false)] None=0,     [Browsable(false)] BothFlags=3,      [Description("First flag")] FirstFlag=1,     [Description("Second flag")] SecondFlag=2,     [Description("Third flag")] ThirdFlag=4 }  // This enum is used for demonstration of options editor functionality enum MyOptionsEnum {     [Description("First option")] FirstOpt=1,     [Description("Second option")] SecondOpt=2,     [Description("Third option")] ThirdOpt=3 }</pre>
-----------------	---

**Tabla 3.1.1.2 Nomenclatura de Enumeraciones, Valores de Enumeraciones, Eventos e Interfaz**

#### 2.3.5.1.5 Nomenclatura de Eventos

A continuación se describe la Nomenclatura de Eventos:

<b>Estructura:</b>	<b>Eventos:</b>
<b>Estilo:</b>	Mayúsculas y minúsculas Pascal
<b>Observaciones:</b>	Especifique dos parámetros denominados sender y e.
<b>Ejemplo:</b>	

**Tabla 3.1.1.3 Nomenclatura de Eventos**

#### 2.3.5.1.6 Nomenclatura de una Interfaz

A continuación se describe la Nomenclatura de una Interfaz:

<b>Estructura:</b>	<b>Interfaz:</b>
<b>Estilo:</b>	Mayúsculas y minúsculas Pascal

<b>Observaciones:</b>	Todos los nombres de interfaces comienzan con la letra "I" Mayúscula, seguido del nombre de la interfaz.  Se utiliza un sustantivo descriptivo para el nombre o un adjetivo.
<b>Ejemplo:</b>	<b>IDisposable</b>

**Tabla 3.1.1.4 Nomenclatura de Interfaz**

#### 2.3.5.1.7 Nomenclatura de Métodos

A continuación se describe la Nomenclatura de Métodos:

<b>Estructura:</b>	<b>Métodos:</b>
<b>Estilo:</b>	Mayúsculas y minúsculas Camel
<b>Observaciones:</b>	Se recomienda utilizar un verbo conjugado seguido del sustantivo al que se aplica.  <b>Sobrecarga de Funciones:</b>  Cuando se haga sobrecarga de funciones, utilizar el mismo nombre para los parámetros que se repitan, el nombre debe describir para que va a ser utilizado no el tipo de datos que se va a utilizar.  En WebServices no se debe hacer sobrecarga de funciones para los métodos del servicio.
<b>Ejemplo:</b>	<b>getAll()</b>

**Tabla 3.1.1.5 Nomenclatura de Métodos**

#### 2.3.5.1.8 Nomenclatura de Espacios de Nombres (Namespaces)

A continuación se describe la Nomenclatura de Nombres (Namespaces):

<b>Estructura:</b>	<b>Espacios de Nombres(Namespaces):</b>
<b>Estilo:</b>	Mayúsculas y minúsculas Pascal

<b>Observaciones:</b>	<p>Cuando son clases genéricas para varios sistemas se utiliza el nombre de la empresa más la utilidad más la especialidad más subdivisiones si existiese.</p> <p>Para namespaces de sistemas en especial se pone el nombre del sistema más el nombre de la capa más la división a la que pertenece.</p> <p>No utilizar el mismo nombre para un espacio de nombres y para una clase.</p>
<b>Ejemplo:</b>	Pamince.Tools.Windows, AsFlor.Servicios, AsFlor.GUI.Packing

**Tabla 3.1.1.6 Nomenclatura de Nombres (Namespaces)**

#### 2.3.5.1.9 Nomenclatura de Parámetros

A continuación se describe la Nomenclatura de Parámetros:

<b>Estructura:</b>	<b>Parámetros:</b>
<b>Estilo:</b>	Mayúsculas y minúsculas Camel
<b>Observaciones:</b>	El parámetro debe ser descriptivo a su función más no a su tipo.
<b>Ejemplo:</b>	getAll(

**Tabla 3.1.1.7 Nomenclatura de Parámetros**

### 3.1.1.3 Observaciones en la Codificación

#### 2.3.5.1.10 Utilizar Métodos o Propiedades

Hay que considerar que las propiedades siempre deben ser independientes, así que no importa el orden en que se asigne. Las funciones se utilizan cuando se necesitan los valores en conjunto para realizar determinada acción; en cambio las propiedades significan un cambio de estado en la clase.

#### 2.3.5.1.11 *Propiedades de Solo Lectura y Solo Escritura*

Se utiliza propiedades de solo lectura cuando el usuario no puede modificar la propiedad, solo consultarla; no se debe utilizar propiedades de solo escritura.

No hay que sobreentender que una propiedad nunca se va a escribir y dejar abierta la opción; habilitar las opciones de escritura y lectura solamente si se utilizarán las dos, caso contrario solo de lectura.

#### 2.3.5.1.12 *Eventos*

Para los parámetros de los eventos (e) siempre derivar de la clase System.EventArgs. Cuando se va a derivar una clase que va a controlar los eventos se debe utilizar en la clase padre el modificador protected.

#### 2.3.5.1.13 *Sobrecarga de métodos*

- Se utiliza para proporcionar métodos diferentes que hagan lo mismo
- Se utiliza para proporcionar valores por defecto en ciertos parámetros
- El método más complejo debería ser llamado por los métodos más simples
- Cuando se va a llamar a más de un parámetro se utiliza la palabra clave **params** seguido de la declaración del array; esto se ve claramente en la función printf de C.

#### 2.3.5.1.14 *Constructores*

Reducir al mínimo la carga en los constructores, usar solamente para inicializar variables

#### 2.3.5.1.15 *Parámetros*

Cuando existen parámetros incorrectos votar la excepción System.ArgumentException

#### 2.3.5.1.16 *Estructuras*

Utilizar estructuras cuando se necesitan solo almacenamiento de valores, más no operaciones.



### 2.3.5.1.17 Enumeraciones

- No utilizar la palabra enum en la declaración de la enumeración.
- Cuando se va a utilizar enumeraciones que provoquen combinaciones utilizar la clase System.FlagsAttribute y valores de los atributos con potencias de dos.

```
[Flags()]
public enum WatcherChangeTypes
{
    Created = 1,
    Deleted = 2,
    Changed = 4,
    Renamed = 8,
    All = Created | Deleted | Changed | Renamed
};
```

### 2.3.5.1.18 Tipos Anidados

Los tipos anidados son clases que son declaradas dentro de otra clase

Los tipos anidados no se deben utilizar si cumplen las siguientes condiciones:

- Las instancias del tipo deben crearse mediante el código de cliente. Si un tipo tiene un constructor público, probablemente no debería anidarse. La razón tras esta directriz es que si se pueden crear instancias de un tipo anidado, esto indicaría que el tipo ocupa un lugar en su propia biblioteca. Se puede crear, usar y destruir sin usar el tipo externo; por lo tanto, no debería estar anidado. Un tipo interno no se debería reutilizar ampliamente fuera del tipo externo sin una relación con el tipo externo.
- Las referencias al tipo se declaran normalmente en el código de cliente.

### 2.3.5.1.19 Provocar y Controlar Excepciones

Para cada excepción debe haber un método que permita verificar si se va a provocar la excepción. Ejemplo: Para evitar una excepción FileNotFoundException, se puede invocar File.Exists.

Los nombres deben termina con el sufijo Exception.

Las Excepciones deben heredar de la clase `ApplicationException` y no de `SystemException`.

Utilizar propiedades para indicar más detalladamente la excepción.

### 3.1.1.4 Tipos de Excepciones Estándar

En la tabla siguiente se incluyen las excepciones estándar que proporciona el tiempo de ejecución y las condiciones para las que deberá crear una clase derivada.

Tipo de excepción	Tipo base	Descripción	Ejemplo
<a href="#">Exception</a>	<b>Object</b>	Clase base de todas las excepciones.	Ninguno (utilice una clase derivada de esta excepción).
<a href="#">SystemException</a>	<b>Exception</b>	Clase base de todos los errores que genera el motor de tiempo de ejecución.	Ninguno (utilice una clase derivada de esta excepción).
<a href="#">IndexOutOfRangeException</a>	<b>SystemException</b>	La inicia el motor de tiempo de ejecución sólo cuando no se indiza correctamente una matriz.	Indizar una matriz fuera del intervalo válido: <code>arr[arr.Length+1]</code>
<a href="#">NullReferenceException</a>	<b>SystemException</b>	La inicia el motor de tiempo de ejecución sólo cuando se hace referencia a un objeto nulo.	<code>object o = null;</code> <code>o.ToString();</code>
<a href="#">InvalidOperationException</a>	<b>SystemException</b>	La inician los métodos que se encuentran en un estado no válido.	Llamar a <code>Enumerator.GetNext()</code> después de eliminar un <code>Item</code> de la colección subyacente.
<a href="#">ArgumentException</a>	<b>SystemException</b>	Clase base de todas las excepciones de argumento.	Ninguno (utilice una clase derivada de esta excepción).
<a href="#">ArgumentNullException</a>	<b>ArgumentException</b>	La inician los métodos que no permiten que un argumento sea nulo.	<code>String s = null;</code> <code>"Calculate".IndexOf(s);</code>
<a href="#">ArgumentOutOfRangeException</a>	<b>ArgumentException</b>	La inician métodos que comprueban que los argumentos están en un intervalo dado.	<code>String s = "string";</code> <code>s.Chars[9];</code>

<a href="#">ExternalException</a>	<b>SystemException</b>	Clase base para excepciones que se generan o que tienen como destino entornos fuera del motor de ejecución.	Ninguno (utilice una clase derivada de esta excepción).
<a href="#">COMException</a>	<b>ExternalException</b>	Excepción que encapsula la información Hresult de COM.	Se usa en la interoperabilidad COM.
<a href="#">SEHException</a>	<b>ExternalException</b>	Excepción que encapsula la información de control estructurado de excepciones Win32.	Se utiliza en la interoperabilidad de código no administrado.

**Tabla 3.1.1.8 Tipos de Excepciones Estándar**

#### 2.3.5.1.20 Sobrecarga de Operadores

- Proporcione métodos de sobrecarga de operadores sólo en la clase en la que se definen los métodos. El compilador de C# cumple esta directriz.
- Utilice la sobrecarga de operadores en los casos en los que el resultado de la operación es obvio. Por ejemplo, tiene sentido poder restar un valor Time de otro valor Time y obtener un TimeSpan. No obstante, no es adecuado utilizar el operador **or** para crear la unión de dos consultas a la base de datos, o utilizar **shift** para escribir una secuencia.
- Sobrecargue los operadores de forma simétrica. Por ejemplo, si sobrecarga el operador de igualdad (**==**), también debe sobrecargar el operador distinto de (**!=**).

```
public struct DateTime
{
    public static TimeSpan operator -(DateTime t1, DateTime t2) { }
    public static TimeSpan Subtract(DateTime t1, DateTime t2) { }
}
```

#### 2.3.5.1.21 Referencias Culturales

Para evitar problemas de conversión siempre se utilizará una misma cultura para mantener un estándar en la conversión.

### 3.1.2 CAPAS

#### 3.1.2.1 Entidades

Las Entidades se van a basar en DataSet's tipados que viajarán por todas las capas del sistema y sirven de persistencia en la base de datos.

La Figura 3.1.2.1 muestra el DataSet Departamento Médico IESS que contiene los DataTable's de Antecedentes Ambientales, Perfil Ocupacional e Inventario Exposición Profesional; con el cual podemos manipular los datos del Informe del Departamento Médico del IESS.

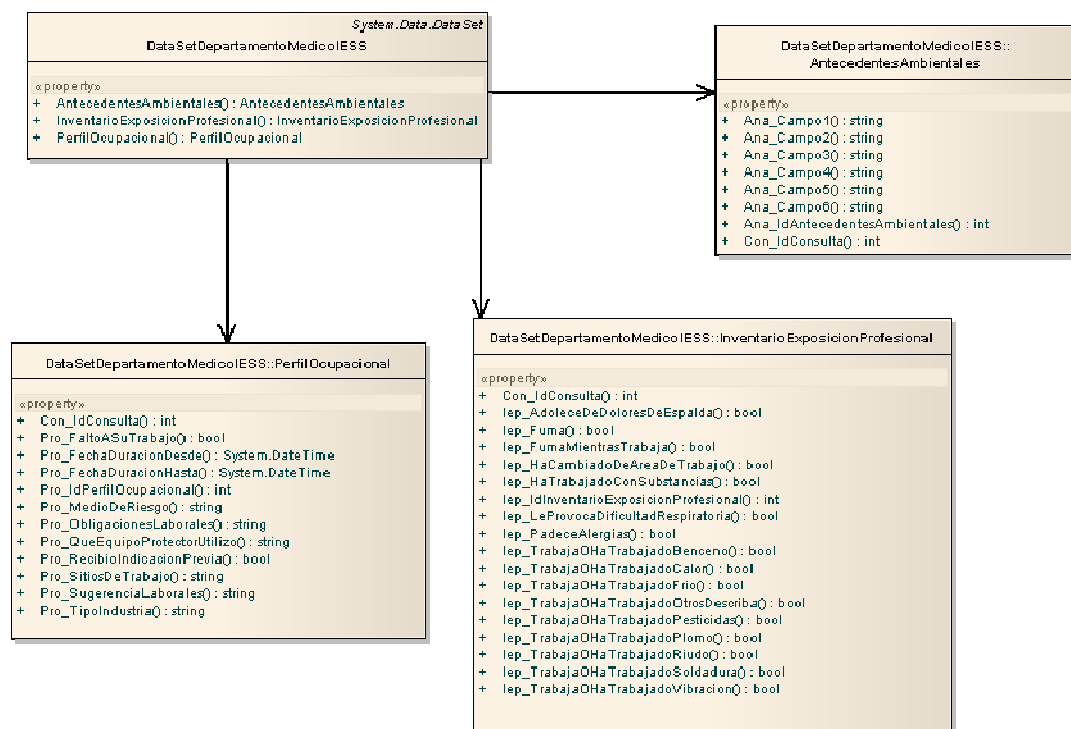


Figura 3.1.2.1 Estructura de la Capa Entidades Sección 1

La Figura 3.1.2.2 muestra el DataSet Doctor que contiene los DataTable's del doctor, cargo, especialidad y departamento; con el cual podemos manipular los datos del Médico.

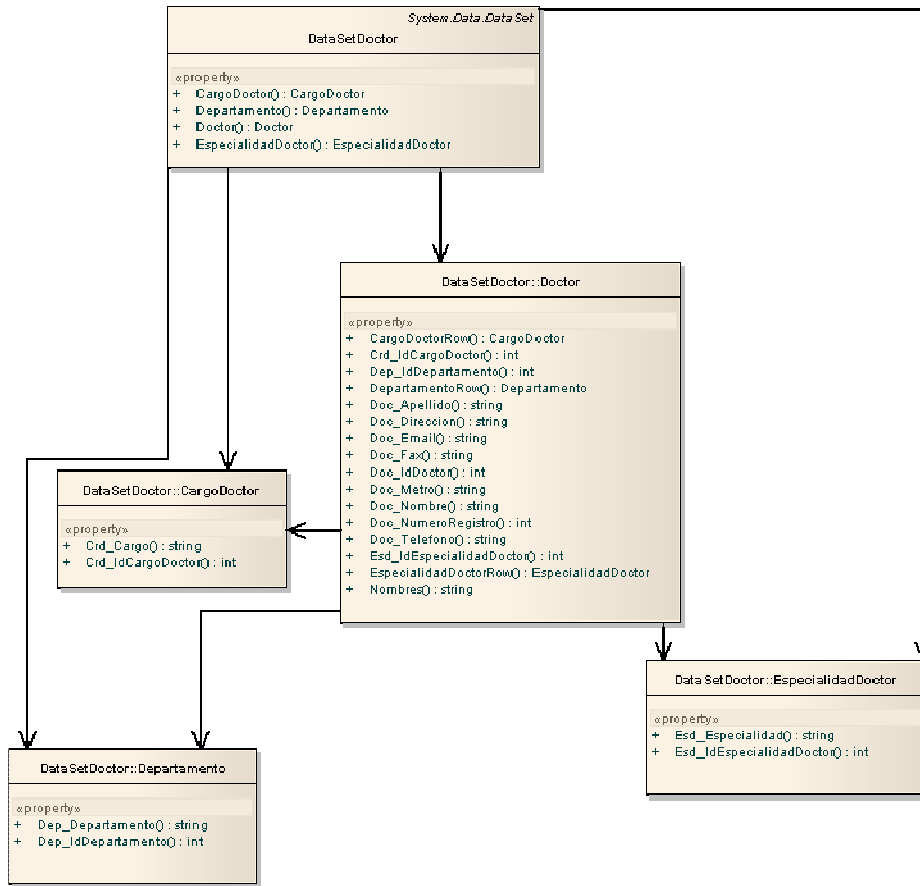


Figura 3.1.2.2 Estructura de la Capa Entidades Sección 2

La Figura 3.1.2.3 muestra el DataSet Empresa que contiene el DataTable de Empresa; con el cual podemos manipular los datos de la Empresa.

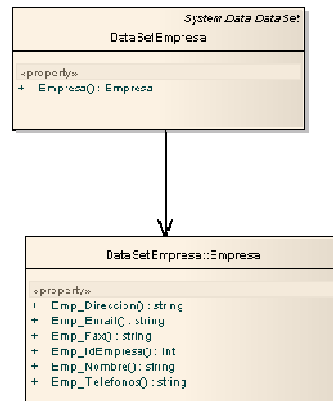
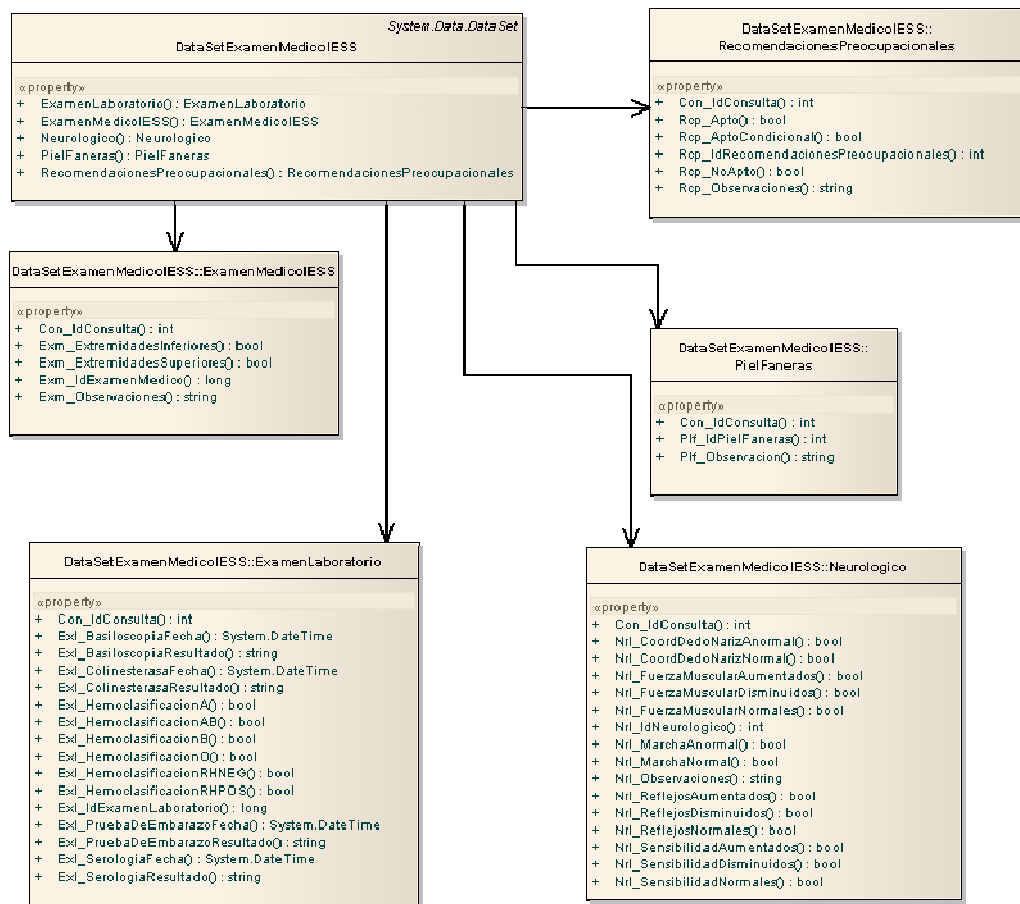


Figura 3.1.2.3 Estructura de la Capa Entidades Sección 3

La Figura 3.1.2.4 muestra el DataSet Examen Médico IESS que contiene los DataTable's de examen médico, examen de laboratorio, neurológico, piel faneras y recomendaciones preocupacionales; con el cual podemos manipular los datos del Informe del Examen Médico del IESS.



**Figura 3.1.2.4 Estructura de la Capa Entidades Sección 4**

La Figura 3.1.2.5 muestra el DataSet Exámenes que contiene los DataTable's de exámenes, tipo de examen y resultados del examen; con el cual podemos manipular los datos de los exámenes.

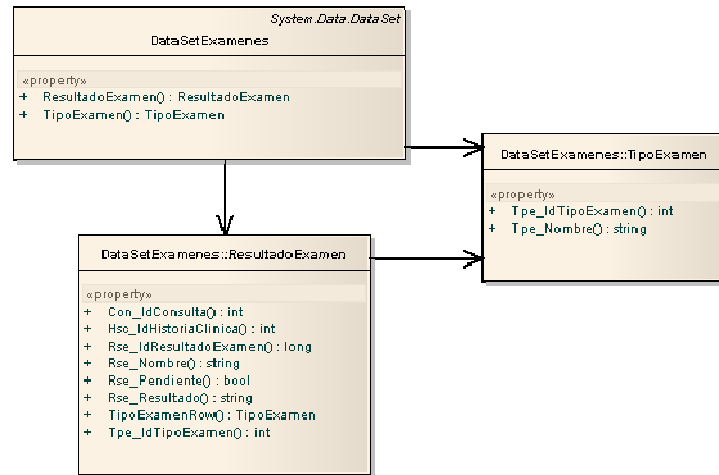


Figura 3.1.2.5 Estructura de la Capa Entidades Sección 5

La Figura 3.1.2.6 muestra el DataSet Maestro Paciente que contiene los DataTable's de paciente, profesión, parentesco, cargo, categoría, patología, tipo de patología, seguro, tipo de seguro, grupo sanguíneo, estado civil y sexo; con el cual podemos manipular la información adicional para los datos del Paciente.

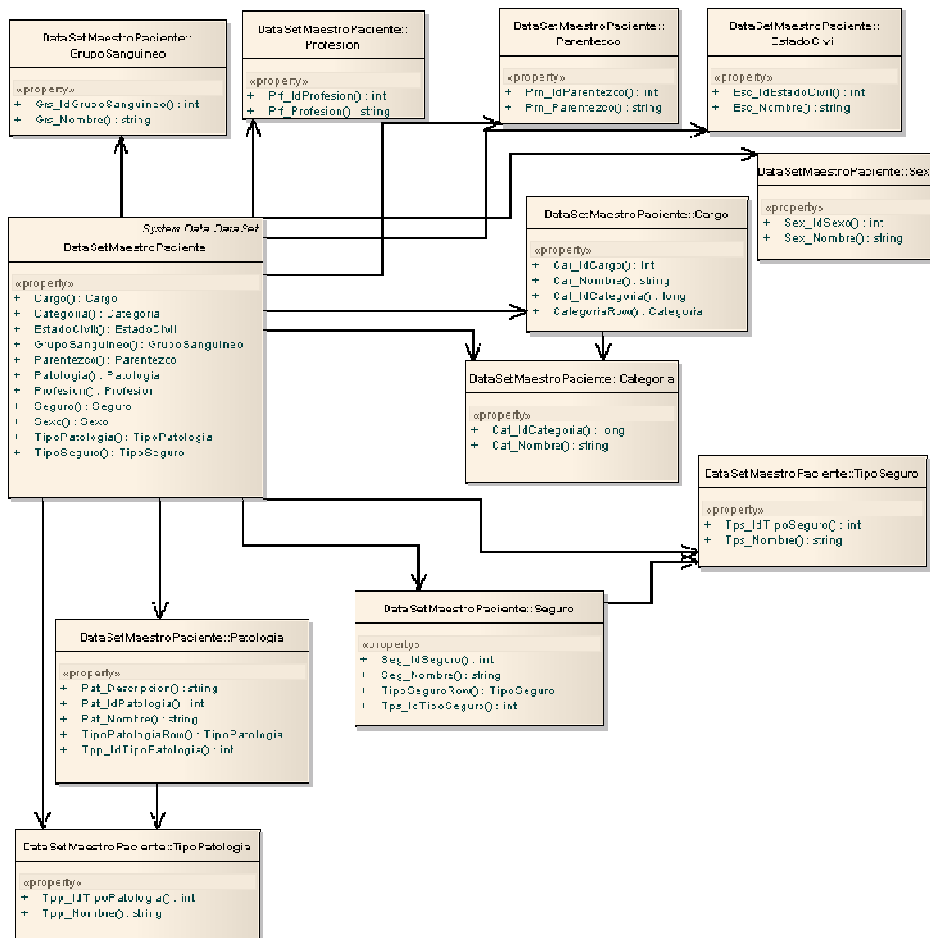


Figura 3.1.2.6 Estructura de la Capa Entidades Sección 6

La Figura 3.1.2.7 muestra el DataSet Medicamentos que contiene los DataTable's de medicamento, casa comercial, especialidad y genérico; con el cual podemos manipular los datos de los Medicamentos.

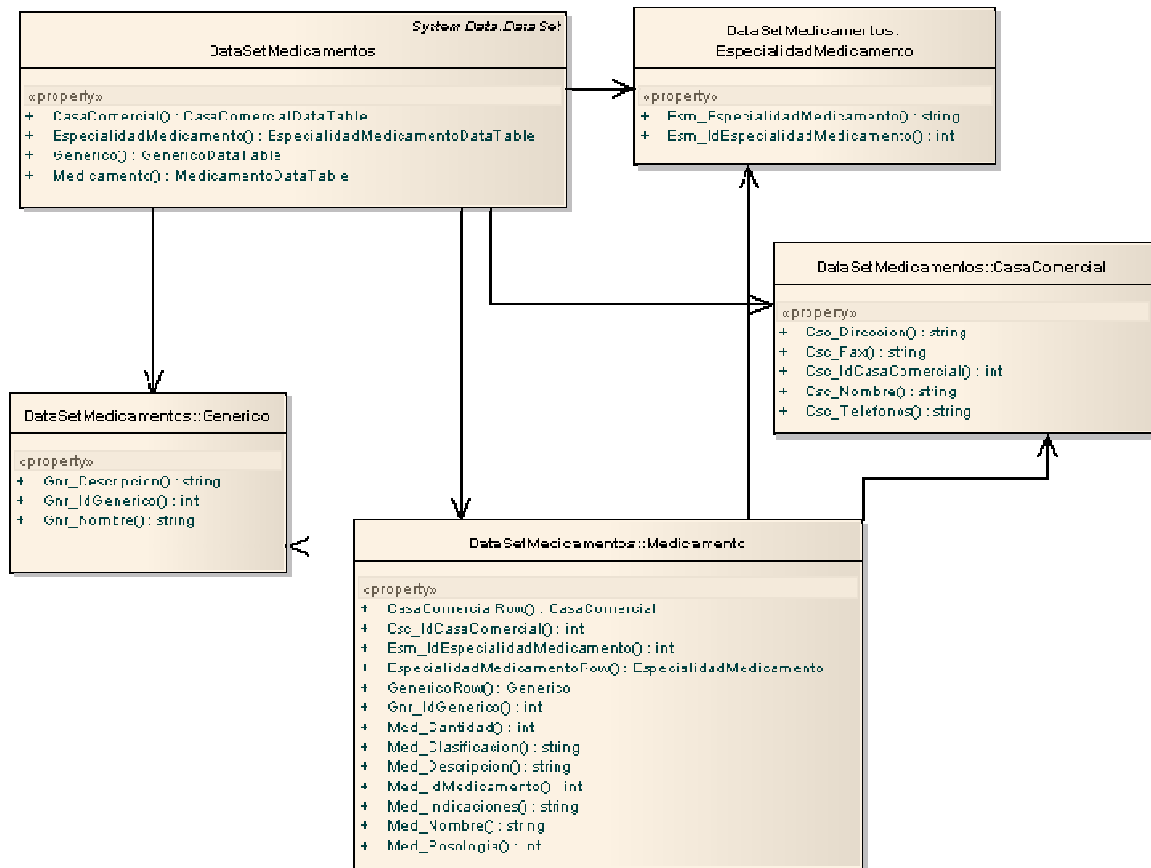


Figura 3.1.2.7 Estructura de la Capa Entidades Sección 7

La Figura 3.1.2.8 muestra el DataSet Órganos Internos IESS que contiene los DataTable's de boca, genitales, cuello, abdomen, osteo muscular y tórax; con el cual podemos manipular los datos del Informe del Órganos Internos del IESS.



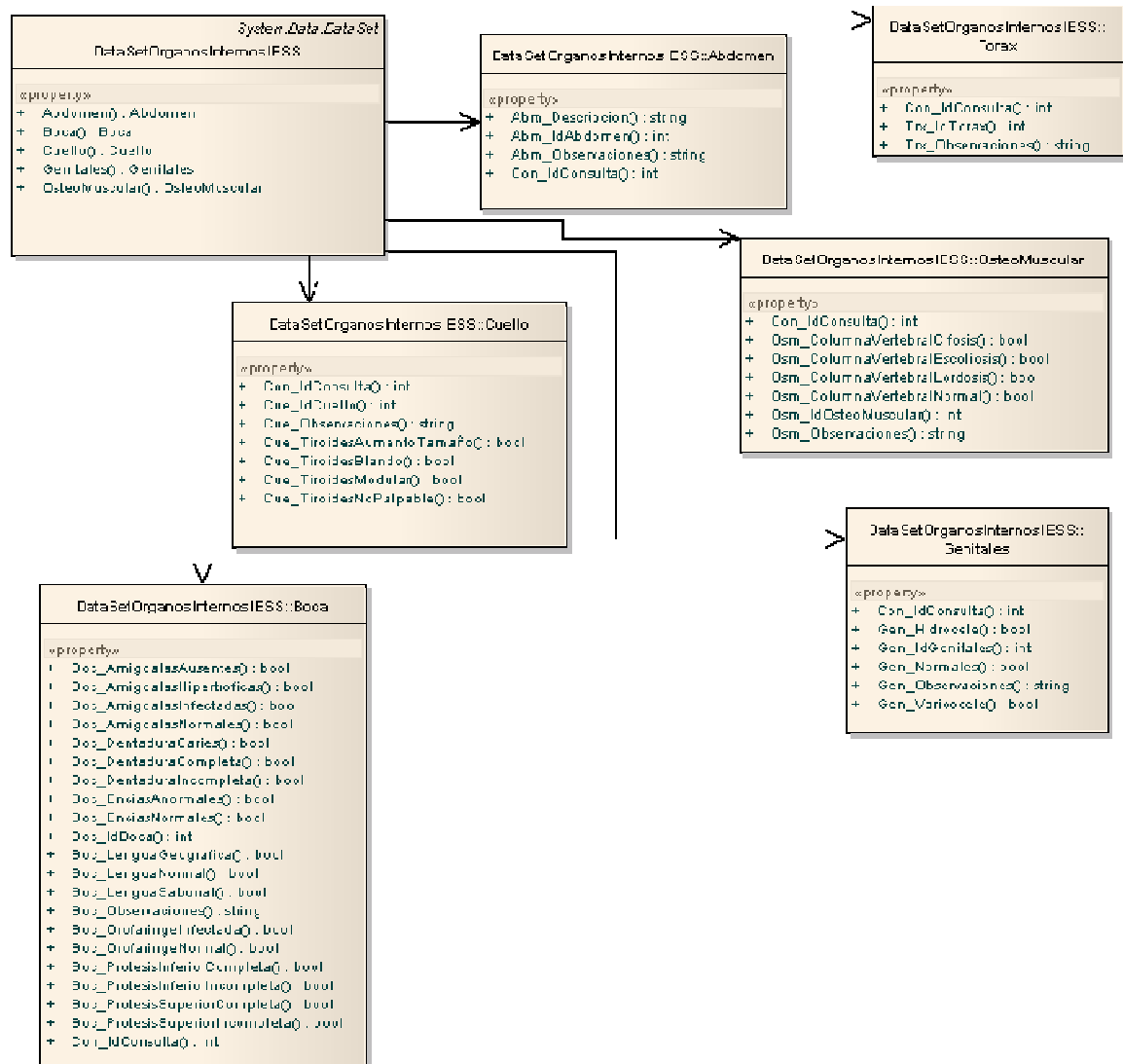


Figura 3.1.2.8 Estructura de la Capa Entidades Sección 8

La Figura 3.1.2.9 muestra el DataSet Paciente que contiene los DataTable's de paciente, historia clínica, plan médico, antecedente personal, antecedente familiar, parentesco y patología; con el cual podemos manipular los datos del Paciente.

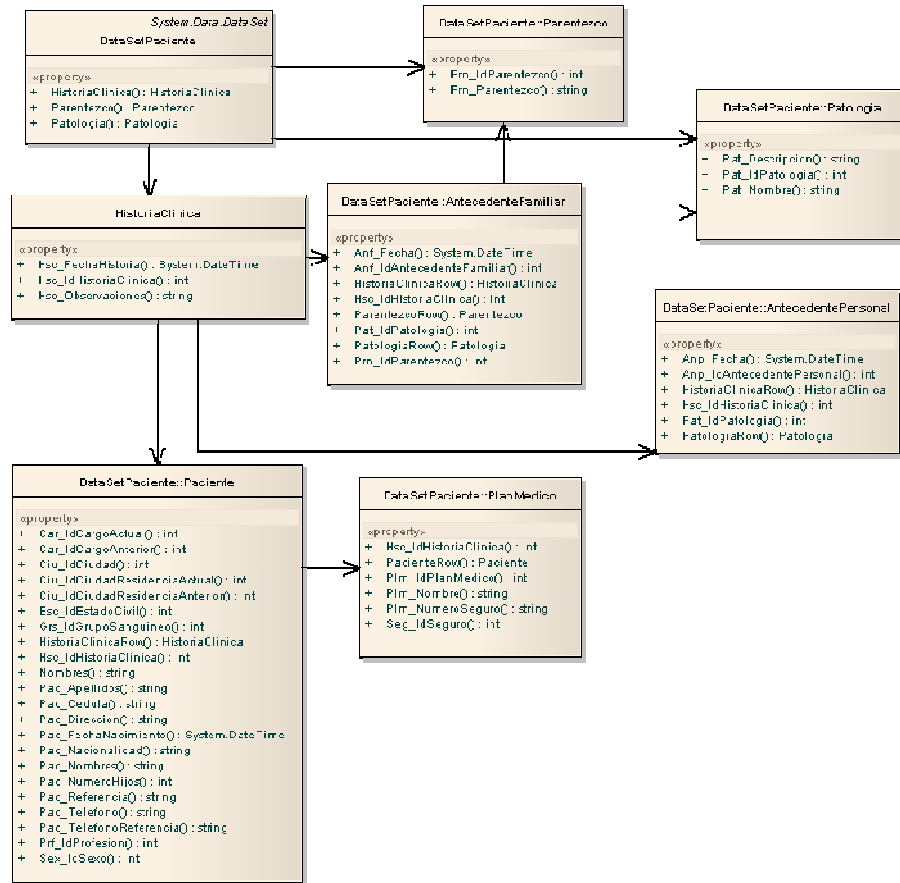


Figura 3.1.2.9 Estructura de la Capa Entidades Sección 9

La Figura 3.1.2.10 muestra el DataSet Receta que contiene los DataTable's de receta y receta medicamento; con el cual podemos manipular los datos de la Receta.

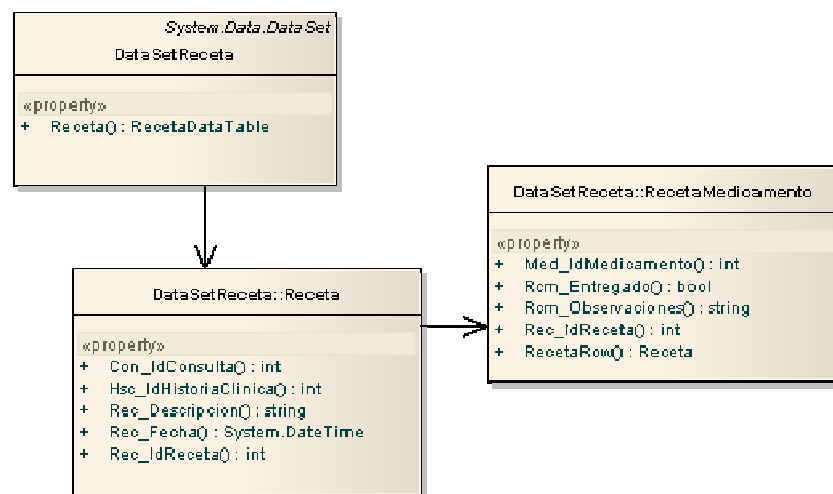
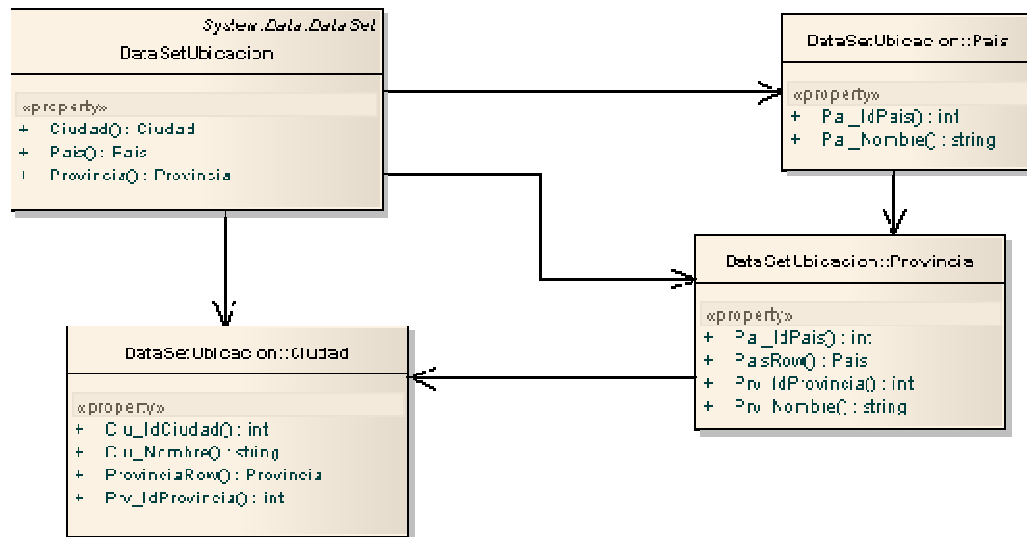


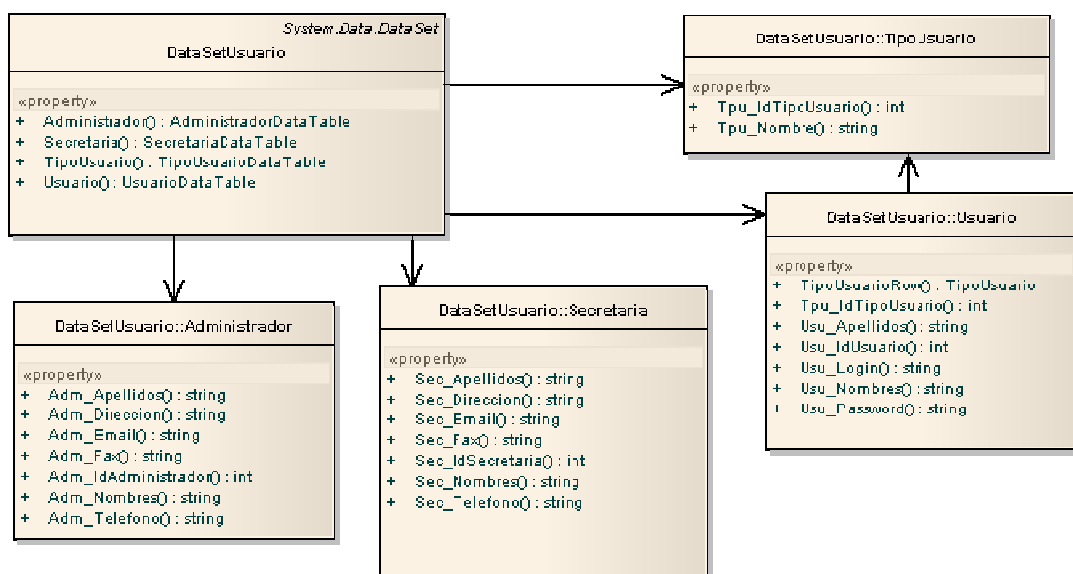
Figura 3.1.2.10 Estructura de la Capa Entidades Sección 10

La Figura 3.1.2.11 muestra el DataSet Ubicación que contiene los DataTable's de país, provincia y ciudad; con el cual podemos manipular los datos del Informe de la ubicación geográfica del paciente.



**Figura 3.1.2.11 Estructura de la Capa Entidades Sección 11**

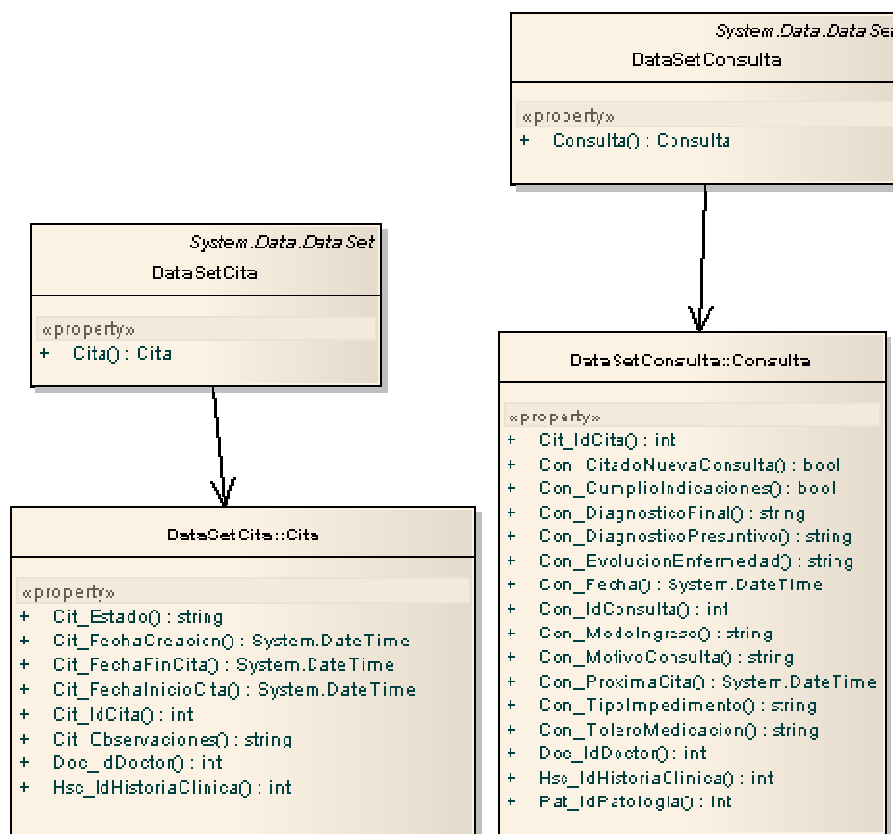
La Figura 3.1.2.12 muestra el DataSet Usuario que contiene los DataTable's de usuario, tipo de usuario, administrador y secretaria; con el cual podemos manipular los datos de los usuarios.



**Figura 3.1.2.12 Estructura de la Capa Entidades Sección 12**

La Figura 3.1.2.13 muestra el DataSet Cita que contiene el DataTable de cita; con el cual podemos manipular los datos de la Cita.

En la misma figura se puede observar el DataSet Consulta que contiene el DataTable de consulta; con el cual podemos manipular los datos de la Consulta.



**Figura 3.1.2.13 Estructura de la Capa Entidades Sección 13**

### 3.1.2.2 Lógica

La Figura 3.1.2.14 muestra los métodos creados y agrupados según la necesidad del negocio.

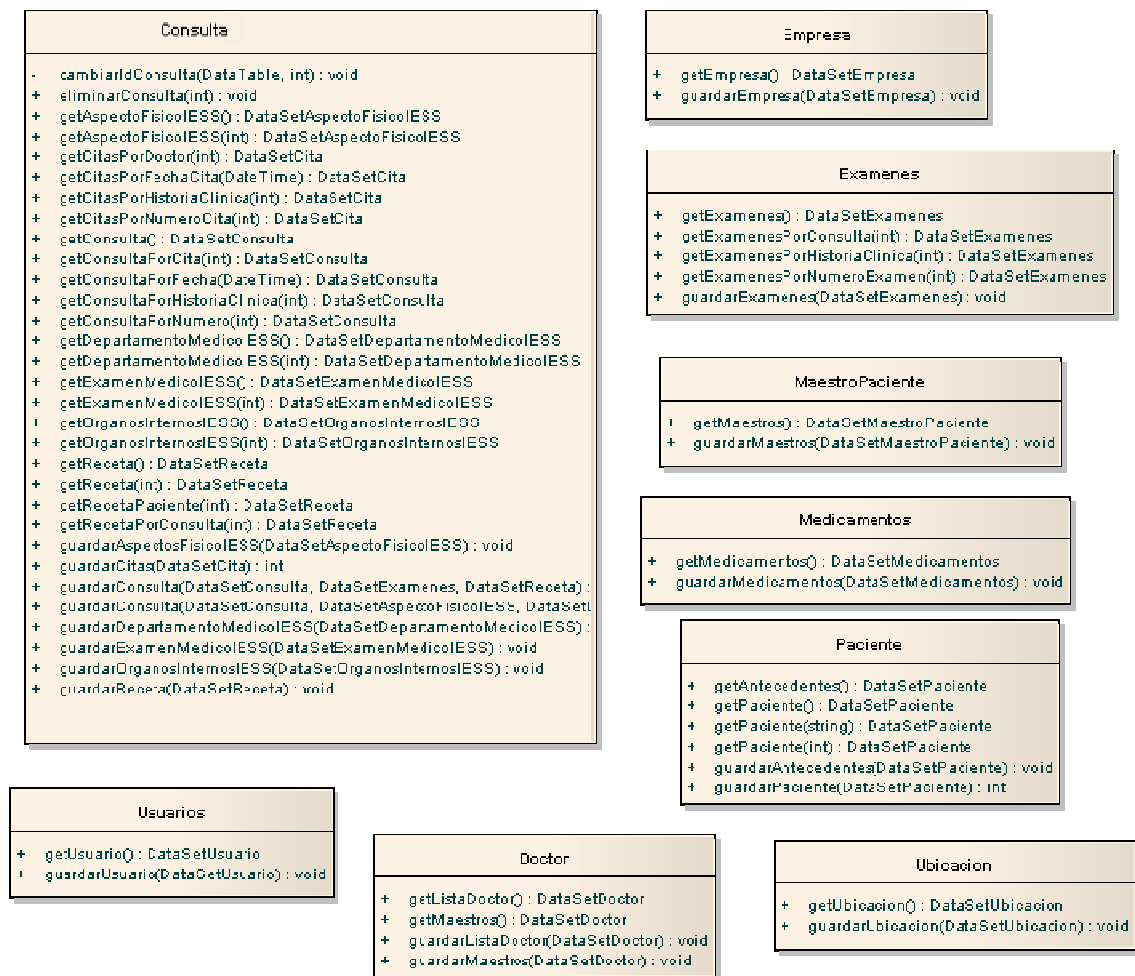


Figura 3.1.2.14 Estructura de la Capa de Lógica

### 3.1.2.3 Servicios

La Figura 3.1.2.15 muestra los Webservice's que maneja el sistema para la utilización de los métodos definidos en la capa de Lógica.

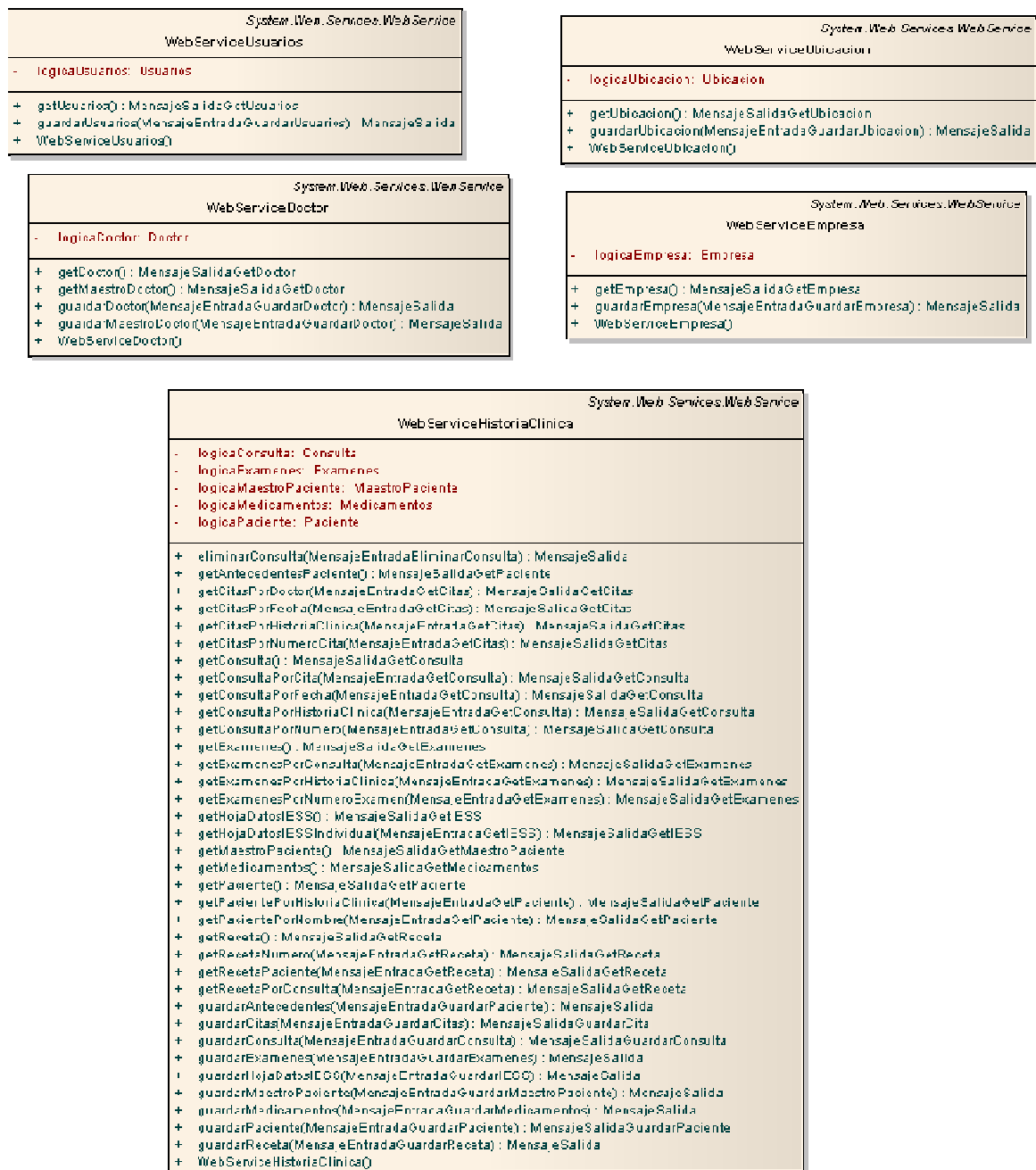
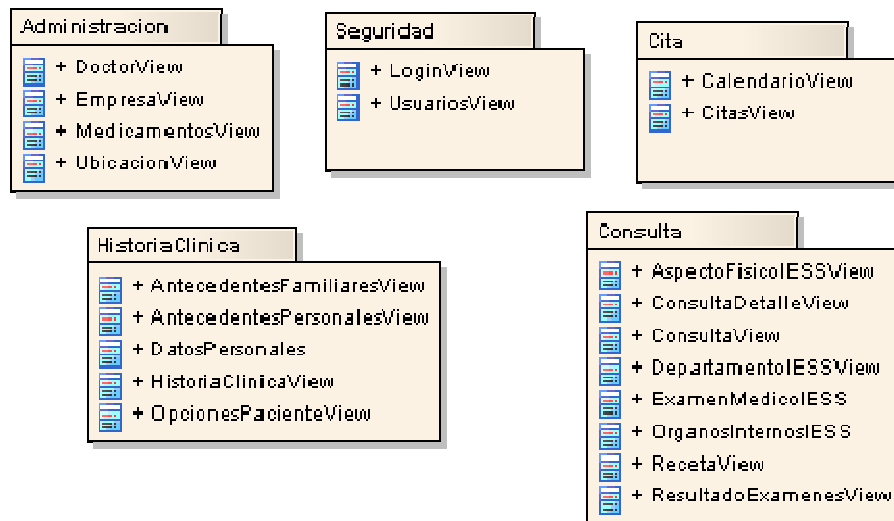


Figura 3.1.2.15 Estructura de la Capa de Servicios

### 3.1.2.4 GUI

La Figura 3.1.2.16 muestra los formularios creados para la interacción con el usuario.



**Figura 3.1.2.16 Estructura de la Capa de GUI**

## 3.2 PRUEBAS

### 3.2.1 ESCENARIO: GESTIONAR LAS HISTORIAS CLÍNICAS: REGISTRAR PACIENTES

En este escenario se debe ingresar todos los datos personales del paciente.

Para cada paciente es obligatorio ingresar una consulta.

En caso de un error en el ingreso de datos, el sistema debe presentar un mensaje avisando al usuario el error cometido.

#### 3.2.1.1 Gestionar la Historia Clínica: Registrar Pacientes – Caso de Prueba 1: Ingresar Incorrectamente un Paciente

##### 3.2.1.1.1 Precondiciones:

El usuario debe tener perfil de médico para realizar la gestión de Registro de Pacientes.

##### 3.2.1.1.2 Entrada:

1. El usuario accede a “Historia Clínica” → “Paciente” → “Datos Personales”
2. El usuario presiona el botón <Nuevo>.
3. El sistema desactiva los botones <Nuevo>, <Modificar>, <Eliminar> e <Imprimir> y activa los botones <Guardar> y <Cancelar>.

4. El usuario deja en blanco algunos campos de los datos personales del paciente.
5. El usuario presiona el botón <Guardar>.

#### *3.2.1.1.3 Resultado Esperado:*

1. El sistema verifica internamente que todos los campos hayan sido ingresados.
2. El sistema muestra un mensaje de error indicando que todos los campos son mandatorios.

### **3.2.1.2 Gestionar la Historia Clínica: Registrar Pacientes – Caso de Prueba 2: Ingresar Correctamente un Paciente**

#### *3.2.1.2.1 Precondiciones:*

El usuario debe tener perfil de médico para realizar la gestión de Registro de Pacientes.

#### *3.2.1.2.2 Entrada:*

1. El usuario accede a “Historia Clínica” → “Paciente” → “Datos Personales”
2. El usuario presiona el botón <Nuevo>.
3. El sistema desactiva los botones <Nuevo>, <Modificar>, <Eliminar> e <Imprimir> y activa los botones <Guardar> y <Cancelar>.
4. El usuario ingresa todos los campos de los datos personales del paciente.
5. El usuario presiona el botón <Guardar>.

#### *3.2.1.2.3 Resultado Esperado:*

1. El sistema verifica internamente que todos los campos hayan sido ingresados.
2. El sistema muestra un mensaje indicando que los datos han sido guardados con éxito.
3. El sistema desactiva los botones <Guardar> y <Cancelar> y activa los botones <Nuevo>, <Modificar>, <Eliminar> e <Imprimir>.



### **3.2.1.3 Gestionar la Historia Clínica: Registrar Pacientes – Caso de Prueba 3: Modificar Incorrectamente un Paciente**

#### *3.2.1.3.1 Precondiciones:*

El usuario debe tener perfil de médico para realizar la gestión de Registro de Pacientes.

El paciente que se desea modificar debe estar previamente ingresado.

#### *3.2.1.3.2 Entrada:*

1. El usuario accede a “Historia Clínica” → “Paciente” → “Datos Personales”
2. El usuario ingresa el nombre del paciente y presiona el botón <Buscar>.
3. El sistema desactiva el botón <Buscar >.
4. El sistema muestra una grilla con datos que concuerden con el parámetro de búsqueda.
5. El usuario selecciona el paciente que desea y el sistema activa la pantalla en donde se puede observar los datos personales del paciente.
6. El usuario presiona el botón <Modificar>.
7. El sistema desactiva los botones <Nuevo>, <Modificar>, <Eliminar> e <Imprimir> y activa los botones <Guardar> y <Cancelar>.
8. El usuario modifica los datos personales del paciente.
9. El usuario presiona el botón <Guardar>.

#### *3.2.1.3.3 Resultado Esperado:*

1. El sistema verifica internamente que todos los campos hayan sido ingresados correctamente.
2. El sistema muestra un mensaje de error indicando que los datos no se han guardado.

### **3.2.1.4 Gestionar la Historia Clínica: Registrar Pacientes – Caso de Prueba 4: Modificar Correctamente un Paciente**

#### *3.2.1.4.1 Precondiciones:*

El usuario debe tener perfil de médico para realizar la gestión de Registro de Pacientes.

El paciente que se desea modificar debe estar previamente ingresado.

#### *3.2.1.4.2 Entrada:*

1. El usuario accede a “Historia Clínica” → “Paciente” → “Datos Personales”
2. El usuario ingresa el nombre del paciente y presiona el botón <Buscar>.
3. El sistema desactiva el botón <Buscar >.
4. El sistema muestra una grilla con datos que concuerden con el parámetro de búsqueda.
5. El usuario selecciona el paciente que desea y el sistema activa la pantalla en donde se puede observar los datos personales del paciente.
6. El usuario presiona el botón <Modificar>.
7. El sistema desactiva los botones <Nuevo>, <Modificar>, <Eliminar> e <Imprimir> y activa los botones <Guardar> y <Cancelar>.
8. El usuario modifica los datos personales del paciente.
9. El usuario presiona el botón <Guardar>.

#### *3.2.1.4.3 Resultado Esperado:*

1. El sistema verifica internamente que todos los datos hayan sido ingresados correctamente.
2. El sistema muestra un mensaje indicando que los datos han sido guardados con éxito.
3. El sistema desactiva los botones <Guardar> y <Cancelar> y activa los botones <Nuevo>, <Modificar>, <Eliminar> e <Imprimir>.

### **3.2.1.5 Gestionar la Historia Clínica: Registrar Pacientes – Caso de Prueba 5: Eliminar Correctamente un Paciente**

#### *3.2.1.5.1 Precondiciones:*

El usuario debe tener perfil de médico para realizar la gestión de Registro de Pacientes.

El paciente que se desea eliminar debe estar previamente ingresado.

#### *3.2.1.5.2 Entrada:*

1. El usuario accede a “Historia Clínica” → “Paciente” → “Datos Personales”
2. El usuario ingresa el nombre del paciente y presiona el botón <Buscar>.
3. El sistema desactiva el botón <Buscar >.
4. El sistema muestra una grilla con datos que concuerden con el parámetro de búsqueda.
5. El usuario selecciona el paciente que desea eliminar y el sistema activa la pantalla en donde se puede observar los datos personales del paciente.
6. El usuario presiona el botón <Eliminar>.
7. El sistema muestra un mensaje que pregunta si está seguro de eliminar el registro actual.
8. El usuario presiona el botón <Si>.

#### *3.2.1.5.3 Resultado Esperado:*

1. El sistema elimina el registro seleccionado.
2. El sistema muestra un mensaje indicando que el paciente ha sido eliminado con éxito.

## 3.2.1.5.4 Resumen de Pruebas

	Datos de Entrada	Resultado Esperado	Resultado Obtenido
	<b>CASO DE PRUEBA 1: INGRESAR INCORRECTAMENTE UN PACIENTE</b>	<b>ITERACIÓN 1</b>	
Nombres: osifj23983		Falla	Exitosa
Apellidos: 928fjjfoe		Falla	Exitosa
Dirección: Idife eife		Éxito	Exitosa
Teléfono: Iskdfje		Éxito	Fallida
Referencia: Idifem irfe		Éxito	Exitosa
Teléfono de Referencia: Ilsiejf		Éxito	Fallida
Fecha de Nacimiento: 01/01/1955		Éxito	Exitosa
Ciudad: Quito		Éxito	Exitosa
Nacionalidad: Ecuatoriana		Éxito	Exitosa
Ciudad de Residencia Actual: Quito		Éxito	Exitosa
Ciudad de Residencia Anterior: Ambato		Éxito	Exitosa
Sexo: Masculino		Éxito	Exitosa
Estado Civil: Casado		Éxito	Exitosa
Seguro: IESS		Éxito	Exitosa
Plan Médico: (Vacío)		Éxito	Fallida
Número de Seguro: (Vacío)		Éxito	Fallida
<b>ITERACIÓN 2</b>			
Nombres: osifj23983		Falla	Exitosa
Apellidos: 928fjjfoe		Falla	Exitosa
Dirección: Idife eife		Éxito	Exitosa
Teléfono: Iskdfje		Falla	Exitosa
Referencia: Idifem irfe		Éxito	Exitosa
Teléfono de Referencia: Ilsiejf		Falla	Exitosa
Fecha de Nacimiento: 01/01/1955		Éxito	Exitosa
Ciudad: Quito		Éxito	Exitosa
Nacionalidad: Ecuatoriana		Éxito	Exitosa
Ciudad de Residencia Actual: Quito	Éxito	Exitosa	
Ciudad de Residencia Anterior: Ambato	Éxito	Exitosa	
Sexo: Masculino	Éxito	Exitosa	
Estado Civil: Casado	Éxito	Exitosa	
Seguro: IESS	Éxito	Exitosa	
Plan Médico: (Vacío)	Falla	Exitosa	
Número de Seguro: (Vacío)	Falla	Exitosa	

<b>CASO DE PRUEBA 2: INGRESAR CORRECTAMENTE UN PACIENTE</b>	<b>Datos de Entrada</b>	<b>Resultado Esperado</b>	<b>Resultado Obtenido</b>
	Nombres: Edwin	Éxito	Exitosa
	Apellidos: Villacís	Éxito	Exitosa
	Dirección: Pucará y República	Éxito	Exitosa
	Teléfono: 099876567	Éxito	Exitosa
	Referencia: Ingrid Reyes	Éxito	Exitosa
	Teléfono de Referencia: 098456768	Éxito	Exitosa
	Fecha de Nacimiento: 01/01/1955	Éxito	Exitosa
	Ciudad: Quito	Éxito	Exitosa
	Nacionalidad: Ecuatoriana	Éxito	Exitosa
	Ciudad de Residencia Actual: Quito	Éxito	Exitosa
	Ciudad de Residencia Anterior: Ambato	Éxito	Exitosa
	Sexo: Masculino	Éxito	Exitosa
	Estado Civil: Casado	Éxito	Exitosa
	Seguro: IESS	Éxito	Exitosa
	Plan Médico: Público	Éxito	Exitosa
	Número de Seguro: 237692	Éxito	Exitosa
<b>CASO DE PRUEBA 3: MODIFICAR INCORRECTAMENTE UN PACIENTE</b>	<b>Datos de Entrada</b>	<b>Resultado Esperado</b>	<b>Resultado Obtenido</b>
	Nombres: Edwin	Éxito	Exitosa
	Apellidos: Villacís	Éxito	Exitosa
	Dirección: Pucará y República	Éxito	Exitosa
	Teléfono: IOEFIFIE	Falla	Exitosa
	Referencia: Ingrid Reyes	Éxito	Exitosa
	Teléfono de Referencia: AOEFKFO	Falla	Exitosa
	Fecha de Nacimiento: 01/01/1955	Éxito	Exitosa
	Ciudad: Quito	Éxito	Exitosa
	Nacionalidad: Ecuatoriana	Éxito	Exitosa
	Ciudad de Residencia Actual: Quito	Éxito	Exitosa
	Ciudad de Residencia Anterior: Ambato	Éxito	Exitosa
	Sexo: Masculino	Éxito	Exitosa
	Estado Civil: Casado	Éxito	Exitosa
	Seguro: IESS	Éxito	Exitosa
	Plan Médico: Público	Éxito	Exitosa
	Número de Seguro: ALOIEF	Falla	Exitosa

<b>CASO DE PRUEBA 4: MODIFICAR CORRECTAMENTE UN PACIENTE</b>	<b>Datos de Entrada</b>	<b>Resultado Esperado</b>	<b>Resultado Obtenido</b>
	Nombres: Edwin	Éxito	Exitosa
	Apellidos: Villacís	Éxito	Exitosa
	Dirección: Pucará y República	Éxito	Exitosa
	Teléfono: 099876567	Éxito	Exitosa
	Referencia: Ingrid Reyes	Éxito	Exitosa
	Teléfono de Referencia: 098456768	Éxito	Exitosa
	Fecha de Nacimiento: 01/01/1955	Éxito	Exitosa
	Ciudad: Quito	Éxito	Exitosa
	Nacionalidad: Ecuatoriana	Éxito	Exitosa
	Ciudad de Residencia Actual: Quito	Éxito	Exitosa
	Ciudad de Residencia Anterior: Ambato	Éxito	Exitosa
	Sexo: Masculino	Éxito	Exitosa
	Estado Civil: Casado	Éxito	Exitosa
	Seguro: IESS	Éxito	Exitosa
	Plan Médico: Público	Éxito	Exitosa
	Número de Seguro: 098765	Éxito	Exitosa
<b>CASO DE PRUEBA 5: ELIMINAR CORRECTAMENTE UN PACIENTE</b>	<b>Datos de Entrada</b>	<b>Resultado Esperado</b>	<b>Resultado Obtenido</b>
	Nombres: Edwin	Éxito	Exitosa
	Apellidos: Villacís	Éxito	Exitosa
	Dirección: Pucará y República	Éxito	Exitosa
	Teléfono: 099876567	Éxito	Exitosa
	Referencia: Ingrid Reyes	Éxito	Exitosa
	Teléfono de Referencia: 098456768	Éxito	Exitosa
	Fecha de Nacimiento: 01/01/1955	Éxito	Exitosa
	Ciudad: Quito	Éxito	Exitosa
	Nacionalidad: Ecuatoriana	Éxito	Exitosa
	Ciudad de Residencia Actual: Quito	Éxito	Exitosa
	Ciudad de Residencia Anterior: Ambato	Éxito	Exitosa
	Sexo: Masculino	Éxito	Exitosa
	Estado Civil: Casado	Éxito	Exitosa
	Seguro: IESS	Éxito	Exitosa
	Plan Médico: Público	Éxito	Exitosa
	Número de Seguro: 098765	Éxito	Exitosa

**Tabla 3.2.1.1 Pruebas del Escenario Gestionar la Historia Clínica: Registrar Pacientes**

### **3.2.2 ESCENARIO: GESTIONAR LAS CONSULTAS: REGISTRAR CONSULTA**

En este escenario hay que verificar que el empleado haya reservado una cita.

Para cada paciente y para cada entrevista con el médico se registrará una consulta.

Para cada paciente nuevo, debe crearse una historia clínica y una consulta.

En caso de un error en el ingreso de datos, el sistema debe presentar un mensaje avisando al usuario el error cometido.

#### **3.2.2.1 Gestionar las Consultas: Registrar Consulta – Caso de Prueba 1: Ingresar Incorrectamente una Consulta**

##### *3.2.2.1.1 Precondiciones:*

El usuario debe tener perfil de médico para realizar la gestión de Registro de Consulta.

##### *3.2.2.1.2 Entrada:*

1. El usuario accede a “Historia Clínica” → “Servicios” → “Consulta”
2. El usuario presiona el botón <Nuevo>.
3. El sistema desactiva los botones <Buscar>, <Nuevo>, <Modificar>, <Eliminar> e <Imprimir> y activa los botones <Guardar> y <Cancelar>.
4. El sistema activa la pantalla en donde se puede ingresar los datos de la consulta.
5. El usuario ingresa el paciente, médico y fecha, deja en blanco los demás campos y presiona el botón <Guardar>.

##### *3.2.2.1.3 Resultado Esperado:*

1. El sistema verifica internamente que todos los campos hayan sido ingresados.
2. El sistema muestra un mensaje de error indicando que todos los campos son mandatorios.

### **3.2.2.2 Gestionar las Consultas: Registrar Consulta – Caso de Prueba 1: Ingresar Correctamente una Consulta**

#### *3.2.2.2.1 Precondiciones:*

El usuario debe tener perfil de médico para realizar la gestión de Registro de Consultas.

#### *3.2.2.2.2 Entrada:*

1. El usuario accede a “Historia Clínica” → “Servicios” → “Consulta”
2. El usuario presiona el botón <Buscar Cita> o <Buscar por Número>.
3. El sistema desactiva los botones <Buscar Cita>, <Buscar por Número>, <Nuevo>, <Modificar>, <Eliminar> e <Imprimir> y activa los botones <Guardar> y <Cancelar>.
4. El sistema activa la pantalla en donde se puede ingresar los datos de la consulta.
5. El usuario ingresa todos los campos de la consulta y presiona el botón <Guardar>.

#### *3.2.2.2.3 Resultado Esperado:*

1. El sistema verifica internamente que todos los campos hayan sido ingresados.
2. El sistema muestra un mensaje indicando que los datos han sido guardados con éxito.
3. El sistema desactiva los botones <Guardar> y <Cancelar> y activa los botones <Buscar Cita>, <Buscar por Número>, <Nuevo>, <Modificar>, <Eliminar> e <Imprimir>.

### **3.2.2.3 Gestionar las Consultas: Registrar Consulta – Caso de Prueba 1: Modificar Incorrectamente una Consulta**

#### *3.2.2.3.1 Precondiciones:*

El usuario debe tener perfil de médico para realizar la gestión de Registro de Consulta.



La consulta que se desea modificar debe estar previamente ingresada.

#### *3.2.2.3.2 Entrada:*

1. El usuario accede a “Historia Clínica” → “Servicios” → “Consulta”
2. El usuario presiona el botón <Buscar Cita> o <Buscar por Número>.
3. El sistema desactiva los botones <Buscar Cita>, <Buscar por Número>, <Nuevo>, <Modificar>, <Eliminar> e <Imprimir> y activa los botones <Guardar> y <Cancelar>.
4. El sistema activa la pantalla en donde se puede ingresar los datos de la consulta.
5. El usuario modifica los datos de la consulta y presiona el botón <Guardar>.

#### *3.2.2.3.3 Resultado Esperado:*

1. El sistema verifica internamente que todos los campos hayan sido ingresados.
2. El sistema muestra un mensaje de error indicando que todos los campos son mandatorios.

### **3.2.2.4 Gestionar las Consultas: Registrar Consulta – Caso de Prueba 1: Modificar Correctamente una Consulta**

#### *3.2.2.4.1 Precondiciones:*

El usuario debe tener perfil de médico para realizar la gestión de Registro de Consultas.

La Consulta que se desea modificar debe estar previamente ingresada.

#### *3.2.2.4.2 Entrada:*

1. El usuario accede a “Historia Clínica” → “Servicios” → “Consulta”
2. El usuario presiona el botón <Buscar Cita> o <Buscar por Número>.
3. El sistema desactiva los botones <Buscar Cita>, <Buscar por Número>, <Nuevo>, <Modificar>, <Eliminar> e <Imprimir> y activa los botones <Guardar> y <Cancelar>.

4. El sistema activa la pantalla en donde se puede ingresar los datos de la consulta.
5. El usuario ingresa todos los campos de la consulta y presiona el botón <Guardar>.

#### *3.2.2.4.3 Resultado Esperado:*

1. El sistema verifica internamente que todos los campos hayan sido ingresados.
2. El sistema muestra un mensaje indicando que los datos han sido guardados con éxito.
3. El sistema desactiva los botones <Guardar> y <Cancelar> y activa los botones <Buscar Cita>, <Buscar por Número>, <Nuevo>, <Modificar>, <Eliminar> e <Imprimir>.

### **3.2.2.5 Gestionar las Consultas: Registrar Consulta – Caso de Prueba 1: Eliminar Correctamente una Consulta**

#### *3.2.2.5.1 Precondiciones:*

El usuario debe tener perfil de médico para realizar la gestión de Registro de Consultas.

La Consulta que se desea eliminar debe estar previamente ingresada.

#### *3.2.2.5.2 Entrada:*

1. El usuario accede a “Historia Clínica” → “Servicios” → “Consulta”
2. El usuario presiona el botón <Buscar Cita> o <Buscar por Número>.
3. El sistema desactiva los botones <Buscar Cita>, <Buscar por Número>, <Nuevo>, <Modificar>, <Eliminar> e <Imprimir> y activa los botones <Guardar> y <Cancelar>.
4. El sistema activa la pantalla en donde se puede observar los datos de la consulta.
5. El usuario presiona el botón <Eliminar>.
6. El sistema muestra un mensaje que pregunta si está seguro de eliminar el registro actual.

7. El usuario presiona el botón <Si>.

3.2.2.5.3 *Resultado Esperado:*

1. El sistema elimina el registro seleccionado.
2. El sistema muestra un mensaje indicando que la consulta ha sido eliminada con éxito.

3.2.2.5.4 *Resumen de Pruebas*

	Datos de Entrada	Resultado Esperado	Resultado Obtenido
	<b>CASO DE PRUEBA 1: INGRESAR INCORRECTAMENTE UNA CONSULTA</b>	<b>ITERACION 1</b>	
Cita:		Falla	Exitosa
Número Consulta: 0		Éxito	Exitosa
Doctor: César Cisneros		Éxito	Exitosa
Paciente: Edwin Villacís		Éxito	Exitosa
Fecha: 01/01/2008		Éxito	Exitosa
Motivo de la Consulta: Gripe		Éxito	Exitosa
Toleró Medicación: (Vacío)		Éxito	Fallida
Evolución de la Enfermedad: (Vacío)		Éxito	Fallida
Cumplió Indicaciones: (Vacío)		Éxito	Exitosa
Tipo de Impedimento: sljewo		Éxito	Exitosa
Diagnóstico Presuntivo: (Vacío)		Éxito	Fallida
Diagnóstico Final: (Vacío)		Éxito	Fallida
Citado para Nueva Cita: Verdadero		Éxito	Exitosa
Próxima Cita: 31/12/2007		Falla	Exitosa
<b>ITERACION 2</b>			
Cita:		Falla	Exitosa
Número Consulta: 0		Éxito	Exitosa
Doctor: César Cisneros		Éxito	Exitosa
Paciente: Edwin Villacís		Éxito	Exitosa
Fecha: 01/01/2008		Éxito	Exitosa
Motivo de la Consulta: Gripe		Éxito	Exitosa
Toleró Medicación: (Vacío)		Falla	Exitosa
Evolución de la Enfermedad: (Vacío)		Falla	Exitosa
Cumplió Indicaciones: (Vacío)		Éxito	Exitosa
Tipo de Impedimento: sljewo		Éxito	Exitosa
Diagnóstico Presuntivo: (Vacío)		Falla	Exitosa
Diagnóstico Final: (Vacío)	Falla	Exitosa	
Citado para Nueva Cita: Verdadero	Éxito	Exitosa	
Próxima Cita: 31/12/2007	Falla	Exitosa	

<b>CASO DE PRUEBA 2: INGRESAR CORRECTAMENTE UNA CONSULTA</b>	<b>Datos de Entrada</b>	<b>Resultado Esperado</b>	<b>Resultado Obtenido</b>
	Cita: 1	Éxito	Exitosa
	Número Consulta: 0	Éxito	Exitosa
	Doctor: César Cisneros	Éxito	Exitosa
	Paciente: Edwin Villacís	Éxito	Exitosa
	Fecha: 01/01/2008	Éxito	Exitosa
	Motivo de la Consulta: Gripe	Éxito	Exitosa
	Toleró Medicación: No tiene alergias	Éxito	Exitosa
	Evolución de la Enfermedad: Mejorando	Éxito	Exitosa
	Cumplió Indicaciones: Verdadero	Éxito	Exitosa
	Tipo de Impedimento: Ninguno	Éxito	Exitosa
	Diagnóstico Presuntivo: Gripe	Éxito	Exitosa
	Diagnóstico Final: Gripe	Éxito	Exitosa
	Citado para Nueva Cita: (Vacío)	Éxito	Exitosa
	Próxima Cita: (Bloqueado)	Éxito	Exitosa
<b>CASO DE PRUEBA 3: MODIFICAR INCORRECTAMENTE UNA CONSULTA</b>	<b>Datos de Entrada</b>	<b>Resultado Esperado</b>	<b>Resultado Obtenido</b>
	Cita: 1	Éxito	Exitosa
	Número Consulta: 0	Éxito	Exitosa
	Doctor: César Cisneros	Éxito	Exitosa
	Paciente: Edwin Villacís	Éxito	Exitosa
	Fecha: 01/01/2008	Éxito	Exitosa
	Motivo de la Consulta: Gripe	Éxito	Exitosa
	Toleró Medicación: No tiene alergias	Éxito	Exitosa
	Evolución de la Enfermedad: Mejorando	Éxito	Exitosa
	Cumplió Indicaciones: Verdadero	Éxito	Exitosa
	Tipo de Impedimento: Ninguno	Éxito	Exitosa
	Diagnóstico Presuntivo: Gripe	Éxito	Exitosa
	Diagnóstico Final: Gripe	Éxito	Exitosa
	Citado para Nueva Cita: Verdadero	Éxito	Exitosa
	Próxima Cita: 12/12/2007	Falla	Exitosa

	Datos de Entrada	Resultado Esperado	Resultado Obtenido
	<b>CASO DE PRUEBA 4: MODIFICAR CORRECTAMENTE UNA CONSULTA</b>	Cita: 1	Éxito
Número Consulta: 0		Éxito	Exitosa
Doctor: César Cisneros		Éxito	Exitosa
Paciente: Edwin Villacís		Éxito	Exitosa
Fecha: 01/01/2008		Éxito	Exitosa
Motivo de la Consulta: Gripe		Éxito	Exitosa
Toleró Medicación: No tiene alergias		Éxito	Exitosa
Evolución de la Enfermedad: Mejorando		Éxito	Exitosa
Cumplió Indicaciones: Verdadero		Éxito	Exitosa
Tipo de Impedimento: Ninguno		Éxito	Exitosa
Diagnóstico Presuntivo: Gripe		Éxito	Exitosa
Diagnóstico Final: Gripe		Éxito	Exitosa
Citado para Nueva Cita: Verdadero		Éxito	Exitosa
Próxima Cita: 03/01/2008		Éxito	Exitosa
	Datos de Entrada	Resultado Esperado	Resultado Obtenido
	<b>CASO DE PRUEBA 5: ELIMINAR CORRECTAMENTE UNA CONSULTA</b>	Cita: 1	Éxito
Número Consulta: 0		Éxito	Exitosa
Doctor: César Cisneros		Éxito	Exitosa
Paciente: Edwin Villacís		Éxito	Exitosa
Fecha: 01/01/2008		Éxito	Exitosa
Motivo de la Consulta: Gripe		Éxito	Exitosa
Toleró Medicación: No tiene alergias		Éxito	Exitosa
Evolución de la Enfermedad: Mejorando		Éxito	Exitosa
Cumplió Indicaciones: Verdadero		Éxito	Exitosa
Tipo de Impedimento: Ninguno		Éxito	Exitosa
Diagnóstico Presuntivo: Gripe		Éxito	Exitosa
Diagnóstico Final: Gripe		Éxito	Exitosa
Citado para Nueva Cita: Verdadero		Éxito	Exitosa
Próxima Cita: 03/01/2008		Éxito	Exitosa

**Tabla 3.2.2.1 Pruebas del Escenario Gestionar las Consultas: Registrar Consulta**

*La descripción completa de las pruebas se encuentra disponible en el Anexo 4*

## 3.3 EVALUACIÓN

### 3.3.1 CARACTERÍSTICA: FUNCIONALIDAD

#### 3.3.1.1 Subcaracterística: Seguridad de Acceso

##### 3.2.2.5.5 Métrica interna de Seguridad De Acceso

Nombre de la métrica	Propósito de la métrica	Método de aplicación	Medición, fórmula y cálculo de datos	Interpretación de los valores medidos	Tipo de escala de métrica	Tipo de medida	Entradas para medición	Referente ISO/IEC 12207 SLCP	Usuarios seleccionados
<b>Prevención en el mal uso de datos</b>	¿Cuán completa es la implementación en la prevención del mal uso de datos?	Contar el número de instancias implementadas para la prevención del mal uso de datos como se especifica y comparar con el número de instancias / accesos especificados en los requisitos con capacidad de alterar / destruir los datos.	$X = A / B$ A = número de instancias implementadas para la prevención del mal uso de datos como se especifica confirmado en la revisión B = número de instancias de operaciones / accesos identificadas en los requerimientos con capacidad de alterar / destruir datos. Nota: considere niveles de seguridad cuando use esta métrica	$0 \leq X \leq 1$  El más cercano a 1. Es el mejor	Absoluto	X = contable/ contable  A = contable B = contable	Requerimientos  Especificación  Diseño  Código Fuente  Reporte de revisión	6.5 Validación  6.6 Revisión colectiva	Desarrolladores

Tabla 3.3.1.1 Métrica Interna de Seguridad de Acceso

### 3.3.1.2 Subcaracterística: Cumplimiento de la Funcionalidad

#### 3.2.2.5.6 Métrica interna de Cumplimiento de la Funcionalidad

Nombre de la métrica	Propósito de la métrica	Método de aplicación	Medición, fórmula y cálculo de datos	Interpretación de los valores medidos	Tipo de escala de métrica	Tipo de medida	Entradas para medición	Referente ISO/IEC 12207 SLCP	Usuarios seleccionados
<b>Cumplimiento de la Funcionalidad</b>	¿Cuán dócil es la funcionalidad del producto al aplicar regulaciones, estándares y convenciones?	Contar el número de detalles que se han reunido y que requieren cumplimiento y comparar con el número de detalles que requieren cumplimiento como en la especificación	$X = A / B$ A = número de ítems implementados correctamente relacionados con el cumplimiento de funcionalidad confirmado en la evaluación B = número total de ítems de cumplimiento.	$0 \leq X \leq 1$ El más cercano a 1. Es el mejor	Absoluto	X = contable/ contable A = contable B = contable	Especificación de cumplimiento y relación de estándares, convenciones o regulaciones. Diseño Código Fuente Reporte de Revisión.	Verificación Revisión colectiva	Analistas Desarrolladores

**Tabla 3.3.1.2 Métrica Interna de Cumplimiento de la Funcionalidad**

### 3.3.2 CARACTERÍSTICA: USABILIDAD

#### 3.3.2.1 Subcaracterística: Capacidad para ser entendido

##### 3.2.2.5.7 Métrica interna de Capacidad para ser entendido

Nombre de la métrica	Propósito de la métrica	Método de aplicación	Medición, fórmula y cálculo de datos	Interpretación de los valores medidos	Tipo de escala de métrica	Tipo de medida	Entradas para medición	Referente ISO/IEC 12207 SLCP	Usuarios seleccionados
<b>Funciones Evidentes</b>	¿Qué proporción de las funciones del producto son evidentes al usuario?	Contar el número de funciones que son evidentes al usuario y comparar con el número total de funciones.	$X = A / B$  A = número de funciones (o tipo de funciones) evidentes al usuario  B = número total de funciones (o tipo de funciones).	$0 \leq X \leq 1$  El límite a 1 es el mejor.	Absoluto	X = contable/ contable  A = contable B = contable	Especificación de requerimientos  Diseño  Reporte de revisión	Verificación  Revisión colectiva	Desarrolladores  Analistas

Tabla 3.3.2.1 Métrica Interna de Capacidad para ser entendido



### 3.3.2.2 Subcaracterística: Operabilidad

#### 3.2.2.5.8 Métrica interna de Operabilidad

Nombre de la métrica	Propósito de la métrica	Método de aplicación	Medición, fórmula y cálculo de datos	Interpretación de los valores medidos	Tipo de escala de métrica	Tipo de medida	Entradas para medición	Referente ISO/IEC 12207 SLCP	Usuarios seleccionados
<b>Claridad del mensaje</b>	¿Qué proporción del mensaje es auto explicativo?	Contar el número de mensajes implementados con explicaciones claras y comparar con el número total de mensajes implementados.	X=A/B  A=Número de mensajes llevados a cabo con explicaciones claras.  B= Número de mensajes llevados a cabo	0 <= X <= 1  El más cercano a 1, el más claro.	Absoluto	X=contable / A= contable B= contable	La especificación de Requisitos Diseño Informe de revisión	Comprobación Revisión colectiva	Diseñadores Analistas

**Tabla 3.3.2.2 Métrica Interna de Operabilidad**

### 3.3.3 CARACTERÍSTICA: EFICIENCIA

#### 3.3.3.1 Subcaracterística: Comportamiento del Tiempo

##### 3.2.2.5.9 Métrica Externa de comportamiento del tiempo

Nombre Métrica	Propósito de la Métrica	Método de Aplicación	La medición, la fórmula y los datos de los elementos computables.	Interpretación del valor medido	Tipo de escala métrica	Tipo de Medida	Medida de Entrada	Referencia ISO/IEC 12207 SLCP	Audiencias de Meta
<b>Rendimiento</b>	Cuántas tareas pueden ser realizadas exitosamente sobre un periodo de tiempo dado?	Calibre cada tarea según la prioridad intencional dada. Empiece varias tareas del trabajo. Mida el tiempo que toma para la tarea moderada para completar su funcionamiento. Guarde un registro de cada esfuerzo.	$X = A / T$ A = Número de tareas completadas T = Observación del periodo de tiempo	$0 < X$ El más largo es el mejor.	Ratio	X = contable / contable A= contable T= tiempo segundos	Reporte de Test  Reporte de operación del tiempo mostrado transcurrido	5.3 Sys./Sw. Integration 5.3 Qualification testing 5.4 Operation 5.5 Maintenance	Usuario  Desarrollador  Maintainer  SQA

Tabla 3.3.3.1 Métrica Externa de Comportamiento del Tiempo

### 3.3.3.2 Subcaracterística: Tiempo de Respuesta

#### 3.2.2.5.10 Métrica Externa de comportamiento del tiempo

Nombre Métrica	Propósito de la Métrica	Método de Aplicación	La medición, la fórmula y los datos de los elementos computables.	Interpretación del valor medido	Tipo de escala métrica	Tipo de Medida	Medida de Entrada	Referencia ISO/IEC 12207 SLCP	Audiencias de Meta
<b>Tiempo de Respuesta</b>	<p>Cuál es el tiempo que toma completar una tarea específica?</p> <p>¿Cuánto tiempo toma antes de la contestación del sistema a una operación especificado?</p>	<p>Empiece una tarea especificada. Mida el tiempo que toma para la muestra para completar su funcionamiento. Guarde un registro de cada intento.</p>	<p><math>T = (\text{Tiempo de resultado obtenido}) - (\text{tiempo de la entrada de comando terminado})</math></p>	<p><math>0 &lt; T</math> Lo más cercano es mejor.</p>	Proporción	<p>T = contable – contable</p> <p>T= contable segundos</p>	<p>Reporte de Test</p> <p>reporte de operación del tiempo mostrado transcurrido</p>	<p>5.3 Sys./Sw. Integration</p> <p>5.3 Qualification testing</p> <p>5.4 Operation</p> <p>5.5 Maintenance</p>	<p>Usuario</p> <p>Desarrollador</p> <p>Maintainer</p> <p>SQA</p>

**Tabla 3.3.3.2 Métrica Externa de comportamiento del Tiempo**

### 3.3.4 VALORAR DE RESULTADOS

Esta etapa de la evaluación se refiere a la sumarización de los datos obtenidos de las métricas aplicadas al caso de estudio

Métrica	Subcaracterística	Valor de la Métrica	Valor de Subcaracterística	Porcentaje de la Característica
<b>Funcionalidad</b>	Prevención en el mal uso de datos	1	0,86	<b>80,5 %</b>
	Cumplimiento de la Funcionalidad	1	0,75	
<b>Usabilidad</b>	Funciones Evidentes	1	0,75	<b>85,5 %</b>
	Claridad del mensaje	1	0,96	
<b>Eficiencia</b>	Rendimiento	0	0,30	<b>27 %</b>
	Tiempo de Respuesta		0,24	

**Tabla 3.3.4.1 Sumarización de Valores Obtenidos**

### 3.3.5 CONCLUSIÓN DE LA EVALUACIÓN

La conclusión de la evaluación contiene la representación gráfica de los resultados de las medidas y su análisis respectivo.

### 3.3.6 ANÁLISIS DE RESULTADOS

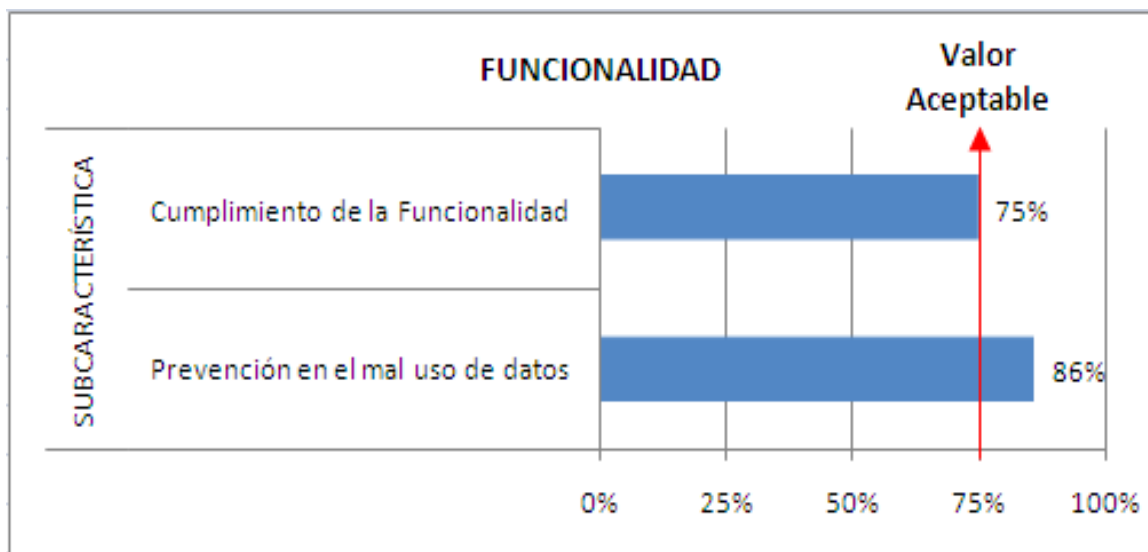


Tabla 3.3.6.1 Análisis de Resultados, característica Funcionalidad

Como se puede apreciar las subcaracterísticas de **Funcionalidad**, cumplen el valor aceptable de la calidad, presentando una alta seguridad para evitar el mal uso del sistema y/o datos. De igual manera el producto de software cumple con los estándares, convenciones y regulaciones en relación a funcionalidad.

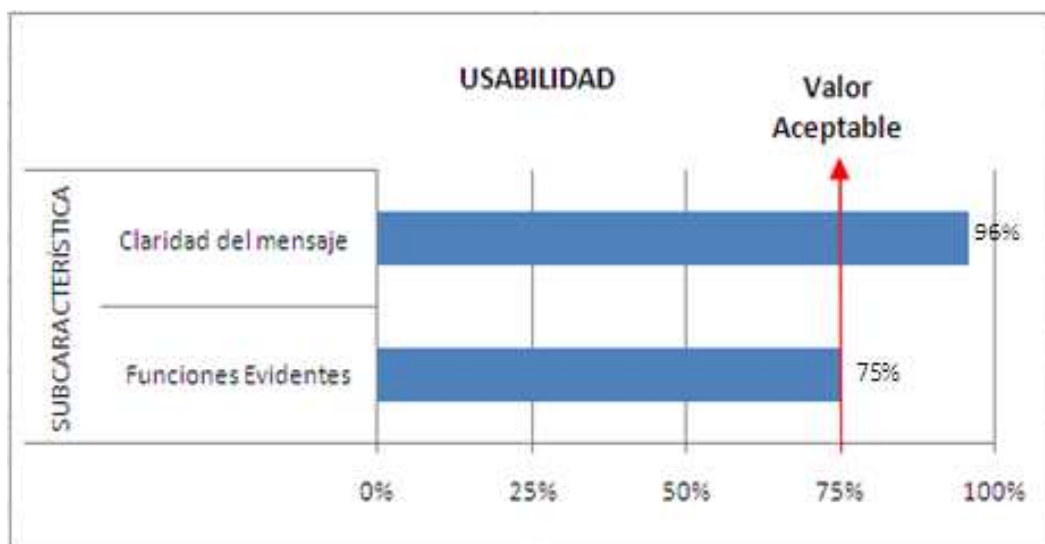
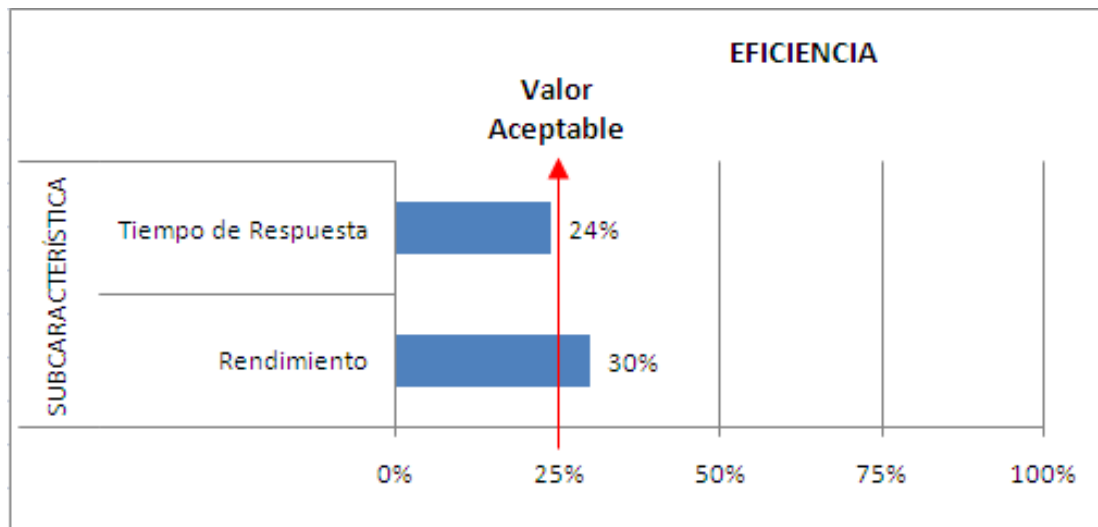


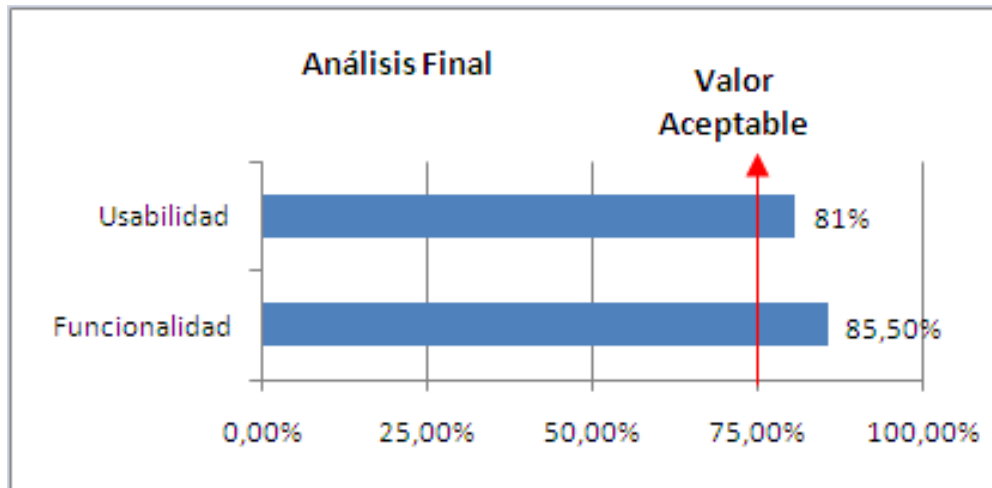
Tabla 3.3.6.2 Análisis de Resultados, característica Usabilidad

Mediante el gráfico se puede observar que las subcaracterísticas de **Usabilidad** cumplen el nivel aceptable de calidad, es decir, los usuarios pueden operar, controlar el producto de software y pueden entenderlo para realizar tareas particulares.



**Tabla 3.3.6.3 Análisis de Resultados, característica Eficiencia**

El valor aceptable de la Eficiencia está entre el 0 y el 25%, porque mientras menos se tarde en responder el sistema mejor es su rendimiento. Como se puede apreciar en el gráfico, la subcaracterística de **Eficiencia** referida al Rendimiento sobrepasa el valor aceptable de calidad, por lo tanto, es posible que algunas de las tareas realizadas no se ejecuten exitosamente en un periodo de tiempo. Con respecto a la subcaracterística de tiempo de respuesta, se encuentra en un valor aceptable, lo que significa que se obtienen los resultados en el tiempo esperado o moderado.



**Tabla 3.3.6.4 Análisis Final de Resultados**

El gráfico representa la sumarización de los datos obtenidos de las características Funcionalidad y Usabilidad, se puede observar que las características de Funcionalidad y Usabilidad sobrepasan el valor aceptable de calidad. Por lo tanto el código fuente del producto de software evaluado con respecto a la Funcionalidad satisface los requisitos funcionales, de estándares y de seguridad. De manera similar la Usabilidad del producto de software refleja que tiene la facilidad de ser entendido y de ser operado.

## **CAPITULO IV**

### **CONCLUSIONES Y RECOMENDACIONES**

#### **4.1 CONCLUSIONES**

- La metodología Microsoft Solution Framework for Agile toma las mejores prácticas en el proceso de desarrollo de software y las integra en distintos modelos, principios y guías, lo cual nos permitió dividir el desarrollo en fases, facilitando así la comprensión y cumplimiento de los objetivos planteados. En cada fase se realizaron iteraciones que permitieron encontrar y corregir errores, de tal forma que la iteración final cumpla con los requerimientos establecidos durante la fase de análisis del sistema. La utilización de esta metodología facilita la asignación de cada miembro del equipo las funciones y las actividades que se debían cumplir durante el desarrollo del sistema.
- La tecnología .Net Framework de Microsoft combinado con el IDE Visual Studio 2008 permite realizar desarrollos con tiempos más eficientes, facilita la creación de librerías que pueden ser reutilizadas en todo el sistema. Sus funcionalidades nos ayudaron a desarrollar el sistema de manera ágil y sencilla, utilizando menor número de líneas de código, reduciendo la complejidad del mantenimiento que normalmente conlleva más tiempo, tanto para la comprensión como para las modificaciones que se necesitan efectuar para cumplir con los nuevos objetivos o requerimientos.
- El sistema desarrollado cumple con las necesidades del cliente permitiendo la generación de historias clínicas con la información de los antecedentes, exámenes y diagnósticos realizados a los pacientes, facilitando así el trabajo del médico porque puede acceder a esta información de manera rápida y oportuna. De igual forma, el médico puede gestionar eficientemente las citas que tiene con cada paciente, sin correr el riesgo de que se crucen o hayan equivocaciones en los nombres u horarios seleccionados para el control médico. Además, durante las consultas el



médico podrá prescribir a sus pacientes el medicamento adecuado y puede verificar el stock del mismo. Toda esta información se registra en el sistema permitiendo al médico generar reportes resumidos con su correspondiente estadística.

- Trabajar con una arquitectura orientada a servicios, nos permitió separar la interfaz de usuario de la lógica del negocio teniendo un sistema distribuido; además de centralizar en el servidor la interacción con la base de datos y exponer en servicios solamente lo necesario a la interfaz de usuario y/u otra aplicación que lo necesite. Al separar en capas el sistema mejora su desempeño y el mantenimiento del mismo resulta más fácil porque los cambios se realizan en la capa que lo necesite y no en todo el sistema; así mismo se puede separar el desarrollo en varios equipos de trabajo.
- Es importante que todas las aplicaciones tengan una fase de evaluación en la cual se pueda determinar si su funcionalidad cumple con los requisitos establecidos y que se hayan controlado todos los posibles errores. En cada iteración se puede utilizar una estrategia de pruebas que permita encontrar errores, corregirlos y mejorar el rendimiento del sistema. La estrategia utilizada nos ayudó a controlar los errores en cada una de las capas del sistema mejorando así su desempeño y usabilidad.
- Los conocimientos adquiridos durante las clases, investigaciones, laboratorios y trabajos desarrollados en el transcurso de la carrera forman las bases necesarias que nos han permitido realizar el proyecto de titulación, lo aprendido nos ayudó a comprender y elaborar las tareas que se exigen en cada una de las fases que constituyen nuestro sistema, además ha facilitado el cumplimiento y satisfacción de los objetivos planteados y requerimientos solicitados por el usuario.
- Al utilizar Smart Client Software Factory, varias tareas comunes en el desarrollo de Aplicaciones Windows se simplificaron porque este framework incluye varios patrones de diseño a utilizar, enfocándonos solo en el desarrollo de la lógica del negocio y no en actividades comunes a este tipo de sistemas; aunque al inicio tardamos en comprender la utilización del framework, logramos disminuir el tiempo que toma construir una aplicación desde cero.

## 4.2 RECOMENDACIONES

- Es recomendable utilizar herramientas case en el desarrollo de un sistema porque permiten realizar los artefactos de una manera más rápida y confiable; centrándose solo en el desarrollo de la solución y no en el trabajo mecánico. Se recomienda investigar nuevas herramientas case que se encuentren en el mercado para determinar cuál es la mejor elección.
- Se recomienda utilizar un solo lenguaje de programación y arquitectura en el desarrollo del sistema porque permite homogenizar la codificación, de tal forma que si el sistema requiere mantenimiento o añadir otras funcionalidades por terceras personas, les sea más sencillo y rápido comprender que hace el sistema y cómo realizar los cambios sin necesidad de conocer varios lenguajes y arquitecturas.
- Se recomienda utilizar Smart Client Software Factory para la creación de Aplicaciones Windows con el Microsoft .Net Framework, el cual permite implementar las mejores prácticas y patrones de diseño construyendo Aplicaciones compuestas de varias partes complejas con el Composite UI Application Block, que es un Framework desarrollado por Microsoft Patterns and Practices que permiten la construcción de Smart Clients (Clientes Inteligentes; Aplicaciones Windows principalmente).
- Se recomienda basarse en las necesidades del cliente y en la usabilidad del sistema cuando se desarrolla una aplicación, y utilizar las últimas innovaciones o actualizaciones de tecnología cuando sea aconsejable y no por pretensión.
- Es recomendable que en todo sistema se defina las iteraciones que va a tener, porque en cada iteración se determina el alcance del sistema, se pueden corregir errores, mejorar la funcionalidad y rendimiento del mismo, además ayudan a mantener las versiones de la aplicación de tal forma que se puede regresar a una versión anterior u observar la dimensión de los cambios realizados.
- Se recomienda conocer el ambiente en el que se desenvuelve el usuario e interactuar con el mismo para obtener mejores resultados y evitar posibles errores o malas interpretaciones en los requerimientos, de esta forma se

puede asegurar que la solución informática propuesta va a satisfacer sus necesidades e inclusive superar las expectativas.

- Se recomienda investigar más de una sola metodología de software porque las necesidades de un sistema a otro varían; de cada metodología se deben recoger las mejores prácticas para que el proceso de desarrollo se adapte a los requerimientos.
- Para la implementación del sistema es recomendable utilizar la versión Express de Microsoft SQL Server 2005 que es gratuita, al no requerir licencias se reducen los costos, sin embargo posee sus limitaciones que son 4 GB máximo en la base de datos, no permite hacer clusters ni replicaciones y solo permite la implementación de reportes más no el IDE de desarrollo.

## **BIBLIOGRAFÍA**

### **LIBROS Y MANUALES**

- [1] BOULTER, Mark; MICROSOFT Patterns & Practices. Smart Client Architecture and Design Guide. Microsoft Corporation. 2004
- [2] MICROSOFT. Application Architecture for .NET: Designing Applications and Services. Microsoft Corporation. 2002
- [3] Seguro Social. Formularios de Exámenes Médicos requeridos.

### **ARTÍCULOS**

- [4] MATTERN, Thomas; WOODS, Dan. Enterprise SOA: Designing IT for Business Innovation. O'Reilly. Abril 2006

### **DIRECCIONES ELECTRÓNICAS**

- [5] Composite UI Application Block.  
<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnpag2/html/CAB.asp>
- [6] LETELIER TORRES, Patricio. Desarrollo de Software Orientado a Objeto usando UML.  
<http://www.dsic.upv.es/~uml>
- [7] Introducción a Rational Unified Process (RUP)  
<https://pid.dsic.upv.es/C1/Material/Documentos%20Disponibles/Introducci%C3%B3n%20a%20RUP.ppt> 2008.
- [8] Ciclo de vida de un proyecto XP  
<http://oness.sourceforge.net/proyecto/html/ch05s02.html> 2008.
- [9] Microsoft Solution Framework v.4 Agile (MSF)

- <http://download.microsoft.com/download/4/4/E/44E1B331-E509-4D10-A9E3-B60640A3A403/20051206-ARC-BA.ppt> 2008.
- [10] Power Designer's Architecture  
[http://www.aisintl.com/case/user\\_groups/PowerDesigner/library/sd\\_architecture.html](http://www.aisintl.com/case/user_groups/PowerDesigner/library/sd_architecture.html) 2008.
- [11] Microsoft Office Visio 2007  
[http://download.microsoft.com/download/8/2/7/82717b95-b244-4f2e-8d31-5580adc84471/Refresh\\_Visio2007\\_Datasheet\\_readyforweb.doc](http://download.microsoft.com/download/8/2/7/82717b95-b244-4f2e-8d31-5580adc84471/Refresh_Visio2007_Datasheet_readyforweb.doc) 2008.
- [12] Microsoft SQL Server  
<http://www.microsoft.com/spain/sql/productinfo/overview/what-is-sql-server.mspx>. 2008.
- [13] Arquitectura de un Manejador de Bases de Datos  
[http://www.bd.cesma.usb.ve/ci5313/docs/taller1\\_Edna.pdf](http://www.bd.cesma.usb.ve/ci5313/docs/taller1_Edna.pdf) 2008.
- [14] Comparing SQL Server 2005 and Oracle  
<http://www.microsoft.com/sql/prodinfo/compare/oracle> 2008.
- [15] What's The Top IDE?  
<http://www.internetnews.com/dev-news/article.php/3616626> 2008.
- [16] ¿Qué es Java Enterprise?  
<http://www.jtech.ua.es/j2ee/2006-2007/jee.html> 2008.
- [17] Metodología Microsoft Solution Framework: Ejemplo.  
<http://www.microsoft.com/latam/technet/fases/msf.asp> 2008
- [18] Microsoft Solution Framework  
<http://www.geocities.com/trescapas/msf.htm> 2008
- [19] Microsoft TechNet Latinoamérica

<http://www.microsoft.com/latam/technet/fases/msf.asp>

2008

## **ANEXOS**

### **Anexo 1: Descripción de Escenarios**

*(Anexo Digital)*

### **Anexo 2: Descripción de Roles**

*(Anexo Digital)*

### **Anexo 3: Descripción de Diccionario de Datos**

*(Anexo Digital)*

### **Anexo 4: Descripción de Pruebas**

*(Anexo Digital)*

### **Anexo 5: Manual de Instalación**

*(Anexo Digital)*

### **Anexo 6: Manual del Usuario**

*(Anexo Digital)*