



FACULTAD DE INGENIERÍA Y CIENCIAS AGROPECUARIAS

**DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN  
DE CONSULTORIOS Y CENTROS MÉDICOS**

Trabajo de Titulación presentado en conformidad a los requisitos  
establecidos para optar por el título de:  
Ingeniero en Sistemas de Computación e Informática

Profesora Guía:  
Ing. Mayrita Valle

Autor:  
Xavier Alejandro Dávila Egred

Año  
2012

### **DECLARACIÓN DEL PROFESOR GUÍA**

“Declaro haber dirigido este trabajo a través de reuniones periódicas con el estudiante, orientando sus conocimientos para un adecuado desarrollo del tema escogido, y dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación.”

---

Mayrita Valle  
Ing. Sistemas, MBA  
C.I.: 60258302-3

### **DECLARACIÓN DE AUTORÍA DEL ESTUDIANTE**

“Declaro que este trabajo es original, de mi autoría, que se han citado las fuentes correspondientes y que en su ejecución se respetaron las disposiciones legales que protegen los derechos de autor vigentes”.

---

Xavier Alejandro Dávila Egred

C.I.: 171791986-2

## **AGRADECIMIENTO**

A mis padres Miriam y Xavier por ser ejemplo en mi vida y guiarme durante mi proceso de desarrollo.

A mi hermana Gabriela por acompañarme durante todos estos años y apoyarme en todos mis proyectos.

A mi esposa Fernanda por su apoyo incondicional en estos últimos años que hemos compartido.

## **DEDICATORIA**

A Xavier Dávila mi padre, por todos los años que ha estado a mi lado, por ser un gran ejemplo de responsabilidad, valores y lucha, por enseñarme a enfrentar todas las adversidades sin dejarme caer.

## RESUMEN

El trabajo de titulación desarrollado surge en respuesta a la necesidad que tienen los Centros Médicos Integrales de mejorar y automatizar los procesos de atención y administración de pacientes que actualmente se realiza manual e ineficientemente.

El objetivo principal de este trabajo de titulación es poder brindar una solución efectiva y automatizada a través de un sistema informático que puede ser distribuido como producto o servicio; con ello se facilitará a los consultorios en la centralización de administración de citas médicas, información de pacientes y médicos y manejo de fichas médicas.

El trabajo de titulación está distribuido en cuatro capítulos.

El capítulo uno se realiza un estudio del entorno actual de los Centros Médicos Integrales “*Centro de Especialidades Médicas Tumbaco*” y el “*Centro Médico Integral San Juan de Cumbayá*”, que nos permita apreciar el funcionamiento actual y las necesidades mantienen actualmente. Se realiza el estudio metodológico y la selección de la metodología a seguir.

El capítulo dos y tres muestra la aplicación proceso metodológico seleccionado donde se documentan las fases de desarrollo de software: análisis, diseño, implementación y pruebas.

El capítulo cuatro se emiten las conclusiones y recomendaciones sobre el proyecto desarrollado.

## ABSTRACT

This thesis project is originated in response to the need of Integral Medical Centers to improve and automate the processes of patient management currently done manually and inefficiently.

The main objective of this thesis project is to provide an effective and automated software solution that can be distributed as a product or service; this will help Integral Medical Centers with the centralization of medical appointments, patient and medical information and health records management.

This thesis project is divided into four chapters.

Chapter one is a study of the current environment of Integral Medical Centers “*Centro de Especialidades Médicas Tumbaco*” and “*Centro Médico Integral San Juan de Cumbayá*” providing the knowledge of their current business procedures and needs. Methodological study was performed and the methodology selection also on this chapter.

Chapter two and three shows the application of the selected methodology with the software development phases: analysis, design, development and tests.

Chapter four contains the conclusions and recommendations of the developed project.

## ÍNDICE

<b>1</b>	<b>CAPÍTULO I. ANÁLISIS DEL ENTORNO .....</b>	<b>1</b>
1.1	ESTUDIO DEL CASO .....	1
1.1.1	Justificación del Proyecto .....	2
1.1.2	Descripción del Problema .....	3
1.1.2.1	Reserva de Citas Médicas .....	3
1.1.2.2	Registro de Ficha Médica .....	4
1.2	ESPECIFICACIÓN DE METODOLOGÍA DE DESARROLLO .....	5
1.2.1	Metodologías Tradicionales .....	5
1.2.1.1	Proceso Unificado Racional (RUP) .....	6
1.2.1.2	Microsoft Solution Framework (MSF).....	8
1.2.2	Metodologías Ágiles.....	10
1.2.2.1	Programación Extrema “Extreme Programming” (XP) .....	12
1.2.2.2	Scrum.....	14
1.2.2.3	Microsoft Solution Framework (MSF).....	15
1.2.3	Comparación entre Metodologías Tradicionales y Ágiles .....	18
1.2.4	Selección de Metodología.....	19
<b>2</b>	<b>CAPÍTULO II. ANÁLISIS Y DISEÑO DEL SISTEMA.....</b>	<b>21</b>
2.1	ANÁLISIS DEL SISTEMA.....	21
2.1.1	Modelos de Casos de Uso.....	21
2.1.1.1	Lista de Actores del Sistema.....	21
2.1.1.2	Modelos de Caso de Uso: Modelo de Negocio.....	23
2.1.1.3	Modelos de Caso de Uso: Gestor de Administración.....	24
2.1.1.4	Modelos de Caso de Uso: Gestor de Negocio.....	31
2.2	DISEÑO DEL SISTEMA .....	40
2.2.1	Infraestructura Tecnológica .....	43
2.2.2	Diagrama de Clases .....	44
2.2.3	Base de Datos .....	46
2.2.4	Diagrama de Actividades .....	49
2.2.5	Diagrama de Secuencia.....	53
2.2.6	Diseño de Interfaces de Usuario.....	56
<b>3</b>	<b>CAPÍTULO III. IMPLEMENTACIÓN Y PRUEBAS .....</b>	<b>58</b>
3.1	IMPLEMENTACIÓN .....	58
3.1.1	Estándares de Programación .....	58

3.1.1.1	Convenciones de Código, Nomenclatura y Estilos .....	58
3.1.1.2	Convenciones de Nomenclatura de Base de Datos.....	60
3.1.2	Invocación de Métodos entre Capas del Sistema.....	62
3.1.3	Diseño de Pantallas del Sistema .....	65
3.1.3.1	Pantalla de Ingreso al Sistema.....	65
3.1.3.2	Pantalla Principal .....	66
3.1.3.3	Pantalla de Administración de Médico .....	67
3.1.3.4	Pantalla de Administración de Consultorio.....	67
3.1.3.5	Pantalla de Administración de Citas.....	68
3.1.3.6	Pantalla de Creación y Actualización de Citas .....	69
3.2	PRUEBAS .....	70
3.2.1	Pruebas Unitarias .....	75
3.2.2	Pruebas Funcionales .....	82
<b>4</b>	<b>CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES .....</b>	<b>87</b>
4.1	CONCLUSIONES.....	87
4.2	RECOMENDACIONES .....	90
	<b>Referencias .....</b>	<b>92</b>
	<b>Anexos .....</b>	<b>94</b>

# 1 CAPÍTULO I. ANÁLISIS DEL ENTORNO

## 1.1 ESTUDIO DEL CASO

El presente proyecto tiene como objetivo proveer una solución informática sobre una plataforma Web que apoye los procesos de administración y captación de pacientes en Centros Médicos Integrales.

Los Centros Médicos Integrales son empresas públicas y privadas que proveen servicios médicos y odontológicos con profesionales en diferentes áreas de la salud. Estos servicios se proveen mediante visitas programadas y de urgencia en cada una de las especialidades médicas que ofrezcan los Centro Médico Integrales.

Para el desarrollo de este proyecto se ha realizado un levantamiento de requerimientos en dos Centros Médicos Integrales que en la actualidad mantienen sus procedimientos de forma manual y buscan a través de una solución informática brindar un servicio de mejor calidad a sus usuarios; los Centros Médicos Integrales con los que se realiza este proyecto presentan similares características en su funcionamiento y servicios que ofrecen:

- Proveen servicios médicos en varias especialidades.
- Proveen servicios de atención a través de visitas programadas mediante la reserva de citas para atención en cada consultorio.
- Proveen servicios de atención hospitalaria para urgencias médicas menores las cuales pueden ser atendidas entre uno y dos días.
- No cuentan con un sistema informático que ayude a automatizar la atención de clientes y fichas médicas.

Los Centros Médicos integrales seleccionados como referencia para este proyecto son el “Centro de Especialidades Médicas Tumbaco” y el “Centro Médico Integral San Juan de Cumbayá”, los cuales ofrecen atención dentro de sus consultorios de especialidades 9 horas diarias con una gran afluencia de usuarios.

Estos Centros Médicos Integrales buscan mejorar la atención a sus pacientes y la automatización de los procesos que se manejan manualmente; dos de los procesos que generan una demanda de trabajo adicional y constantes problemas dentro de la administración general es la administración de pacientes y la reserva de citas médicas, generando malestar entre los pacientes que acuden diariamente a los Centros Médicos Integrales.

### **1.1.1 Justificación del Proyecto**

Los Consultorios y Centros Médicos que fueron caso de estudio, manejan los registros de pacientes de forma manual dificultando la obtención de la información al momento de que esta sea requerida generando un incorrecto control y seguimiento de pacientes.

Los médicos que conforman los Consultorios y Centros Médicos Integrales buscan mejorar continuamente la atención a los pacientes y sus procesos internos generando la necesidad de implantar un Sistema Informático que sirva de apoyo en el trabajo de atención a pacientes y automatización de procesos internos.

El sistema Medical Center.Net busca solventar y automatizar los procedimientos y almacenamiento de información necesaria para la administración y atención de clientes; de esta forma la información de pacientes y el proceso de atención se centralizará en una base de información común agilizando y brindando un mejor servicio a los clientes de los Consultorios y Centros Médicos Integrales.

### **1.1.2 Descripción del Problema**

A continuación se presenta la situación general de los Consultorios y Centros Médicos Integrales en cuanto al proceso de atención a pacientes y el manejo de información de los mismos.

#### **1.1.2.1 Reserva de Citas Médicas**

Un paciente que desea recibir atención médica en un Consultorio Médico o dentro de alguna especialidad dentro de un Centro Médico debe previamente haber reservado una cita médica y tener asignado un turno para poder ser atendido.

Actualmente la reserva de citas se la hace de dos formas:

- El paciente asiste al Consultorio o Centro Médico Integral y solicita un turno con la asistente del Consultorio o en la recepción del Centro Médico.
- El paciente a través de una llamada telefónica realiza la reserva de una cita médica con el consultorio que desea o a través del Call Center del Centro Médico Integral.

Para el proceso de cancelación y cambio de horario de citas médicas, el paciente debe utilizar los mismos canales de comunicación utilizados para la reserva de citas.

En el caso de los Consultorios y Centros Médicos Integrales no se cuenta con una base de datos centralizada y disponible en la cual se almacene la información de las citas médicas y los pacientes. El almacenamiento y seguimiento de las citas médicas con pacientes se las hace a través de agendas físicas u hojas electrónicas las cuales son manejadas por las asistentes de cada consultorio médico y en algunos casos por el médico del consultorio.

### 1.1.2.2 Registro de Ficha Médica

Una ficha médica es el resultado de la recopilación de datos provenientes de las exploraciones que se le realizan a un paciente. Está compuesta por la anamnesis, es decir, el interrogatorio sistemático realizado a un paciente desde que se inicia la asistencia, hasta los informes y diagnósticos de las exploraciones que se puedan realizar por el psicólogo tratante o los servicios hospitalarios.

El registro de la ficha médica, dependiendo del Consultorio o Centro Médico Integral no tiene un patrón común, cada uno lo maneja de acuerdo a la conveniencia de cada Consultorio. El registro de las fichas médicas se realiza normalmente cuando el paciente asiste por primera vez a un Consultorio o Centro Médico Integral.

Las formas más comunes de registro de Fichas Médicas son:

1. Se abre una Ficha Médica para cada consultorio. Si un Centro Médico tiene varios consultorios, cada uno maneja la Ficha Médica de sus pacientes, de esta forma la información de un paciente no se distribuye ni está disponible para otros consultorios dentro del Centro Médico en caso de ser necesario.
2. Se abre una Ficha Médica general en el Centro Médico la cual es común para todos los consultorios de un Centro Médico y está disponible para todos los consultorios cuando esta se requiera. Cada vez que un médico requiere una Ficha Médica de un paciente se debe solicitar la misma en un archivo físico donde se encuentran almacenadas alfabéticamente y después de su utilización devolverla para que vuelva a ser almacenada en el archivo.

Por la naturaleza del registro actual de las Ficha Médica, no pueden estar disponibles en más de un consultorio a la vez y tampoco estar disponibles en el momento que se requiera para el médico y paciente.

## **1.2 ESPECIFICACIÓN DE METODOLOGÍA DE DESARROLLO**

Las metodologías de desarrollo de software son un grupo de pautas, técnicas y patrones para el desarrollo de software.

De acuerdo con la filosofía de desarrollo de software se puede clasificar a las metodologías en metodologías tradicionales y metodologías ágiles.

El éxito y la calidad de un producto de software depende en gran parte de la metodología que se trabajó para el proceso de desarrollo de software, sea éste tradicional o ágil, con la cual se maximice el potencial de los recursos elevando la calidad del producto con los recursos y tiempos establecidos en las fases de desarrollo de software.

Para la selección de la metodología de desarrollo, se analizará a continuación cada clasificación.

### **1.2.1 Metodologías Tradicionales**

Las metodologías tradicionales se enfocan en una planificación definida. Esta planificación se realiza esperando que los resultados de cada una de sus etapas sean determinantes y predecibles debido a que cada etapa de estas metodologías es definida previamente.

Este tipo de metodologías son efectivas y necesarias en proyectos con número considerable de recursos y tiempo, sin embargo cualquier modificación representa costos elevados en tiempo y recursos, convirtiéndose en un riesgo cuando un proyecto necesita de mucha flexibilidad.

Las metodologías tradicionales (formales) se centran en documentación, planificación y procesos. (Plantillas, técnicas de administración, revisiones, etc.), a continuación se detalla RUP uno de los métodos más usados dentro de los métodos tradicionales.

### **1.2.1.1 Proceso Unificado Racional (RUP)**

RUP es una metodología formal mediante el cual se asigna tareas y responsabilidades para el desarrollo de sistemas. Tiene como objetivo asegurar el desarrollo de software con alta calidad, garantizando la satisfacción total del cliente en tiempo y dinero. Es guiado por casos de uso, centrado en la arquitectura y utiliza UML como lenguaje de notación.

Es proceso dirigido por casos de uso, este avanza a través de una serie de flujos de trabajo (requisitos, análisis, diseño, implementación y pruebas) que parten de los casos de uso; está centrado en la arquitectura y es iterativo e incremental. Además cubre el ciclo de vida de desarrollo de un proyecto y toma en cuenta las mejores prácticas a utilizar en un modelo de desarrollo de software (Brito, 2009, p. 40)

A continuación se muestran estas prácticas:

- Desarrollo de software en forma iterativa.
- Manejo de requerimientos.
- Utiliza arquitectura basada en componentes.
- Modela el software visualmente.
- Verifica la calidad del software.
- Controla los cambios.

La metodología RUP tiene 4 fases en el ciclo de vida del desarrollo de software:

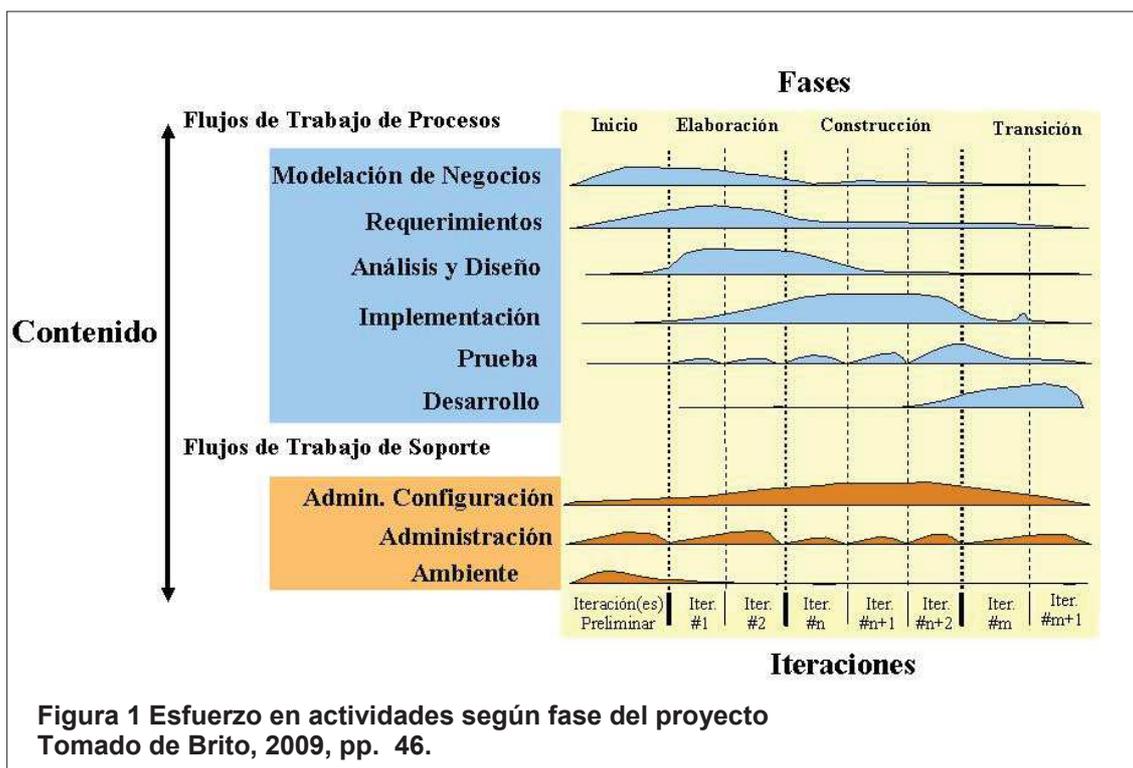
- Concepción
- Elaboración
- Construcción
- Transición

### **Ventajas**

- Cada fase contiene una etapa en la que se permite realizar cambios en los objetivos.
- Metodología adaptable a proyectos de innovación.
- Está compuesta de pasos intuitivos dentro del desarrollo de software.

### **Desventajas**

- La evaluación de riesgos es compleja
- Excesiva flexibilidad para algunos proyectos.
- El nivel técnico del cliente debe ser alto para definir el alcance de un proyecto de desarrollo de software.



### 1.2.1.2 Microsoft Solution Framework (MSF)

Microsoft Solution Framework (MSF), es una metodología desarrollada por Microsoft Consulting Services en conjunto con varios grupos de negocios de Microsoft y otras fuentes de la industria (Zaihngroup, 2010). Es un modelo de procesos que combina dos modelos muy comunes en proyectos de desarrollo, el modelo en cascada y el modelo en espiral.

Esta es una metodología flexible e interrelacionada con una serie de conceptos, modelos y prácticas de uso, que controlan la planificación, el desarrollo y la gestión de proyectos tecnológicos.

MSF se centra en los modelos de proceso y de equipo dejando en un segundo plano las elecciones tecnológicas. MSF tiene las siguientes características (Brito, 2009, p. 15):

- Adaptable: es parecido a un compás, usado en cualquier parte como un mapa, del cual su uso es limitado a un específico lugar.
- Escalable: puede organizar equipos tan pequeños entre 3 o 4 personas, así como también, proyectos que requieren 50 personas o más.
- Flexible: es utilizada en el ambiente de desarrollo de cualquier cliente.
- Tecnología Agnóstica: porque puede ser usada para desarrollar soluciones basadas en cualquier tecnología. MSF se compone de varios modelos encargados de planificar las diferentes partes implicadas en el desarrollo de un proyecto: Modelo de Arquitectura del Proyecto, Modelo de Equipo, Modelo de Proceso, Modelo de Gestión del Riesgo, Modelo de Diseño de Proceso y finalmente el modelo de Aplicación.



En La Figura 2, cada rombo representa un entregable dado con la finalización de cada etapa, sin embargo cada una de estas etapas puede sufrir de modificaciones en caso de ser necesario sin hacer que esto afecte el flujo normal del proyecto, siendo similar al modelo espiral, que permite volver a etapas previas del proyecto. Existen también puntos de control específicos que permiten tener control sobre el avance del proyecto, y poder crear una

planificación clara acerca de los recursos estimados para el cumplimiento en plazos y costos estables.

Esta metodología consta de cinco etapas, en las cuales se generan entregables concretos que ayudan a resolver los requerimientos del cliente de una manera objetiva. Las cinco etapas de MSF son:

1. Visión: Etapa donde el equipo va a tener una idea clara, pero general de los objetivos, infraestructura, tecnología, arquitectura, roles y riesgos todos estos necesarios para el buen desenvolvimiento del proyecto.
2. Planificación: Etapa donde se levanta los requerimientos específicos del cliente, tomando en cuenta que MSF está abierto a recibir cambios dentro de la etapa de desarrollo.
3. Desarrollo: Etapa donde se realizan las tareas de desarrollo y codificación del producto.
4. Estabilización: Etapa donde se prueba al proyecto en ambientes simulados, que deben ser los más parecidos a los ambientes reales.
5. Implementación: Etapa donde la solución es publicada en un ambiente de producción para que cumpla con el fin por el cual fue creada.

### **1.2.2 Metodologías Ágiles**

Las metodologías ágiles aparecen como respuesta al vacío que las metodologías tradicionales no pueden cubrir, el vacío metodológico que los proyectos de desarrollo de software tienen en entornos de constante cambio donde es necesaria la reducción en los tiempos de desarrollo y recursos.

Estas metodologías están enfocadas en proyectos pequeños, convirtiéndose en una solución a la medida a la naturaleza de estos proyectos sin dejar comprometida las buenas prácticas de desarrollo de software y la calidad del producto.

Como resultado de esta nueva teoría se crea un Manifiesto Ágil (Pires Vico, 2010) cuyas principales ideas son:

- Los individuos y las interacciones entre ellos son más importantes que las herramientas y los procesos empleados.
- Es más importante crear un producto software que funcione que escribir documentación exhaustiva.
- La colaboración con el cliente debe prevalecer sobre la negociación de contratos.
- La capacidad de respuesta ante un cambio es más importante que el seguimiento estricto de un plan.

Estas metodologías ponen de relevancia que la capacidad de respuesta a un cambio es más importante que el seguimiento estricto de un plan. La flexibilidad presentada en muchas ocasiones genera una ventaja competitiva.

### **Retrasar las decisiones y Planificación Adaptativa**

Es el eje en cual gira la metodología ágil, el retrasar las decisiones tan como sea posible de manera responsable será ventajoso tanto para el cliente como para la empresa, lo cual permite siempre mantener una satisfacción en el cliente y por ende el éxito del producto, las principales ventajas de retrasar las decisiones son:

- Reduce el número de decisiones de alta inversión que se toman.
- Reduce el número de cambios necesario en el proyecto.
- Reduce el coste del cambio

La planificación adaptativa permite estar preparados para el cambio ya que lo hemos introducido en nuestro proceso de desarrollo, además hacer una planificación adaptativa consiste en tomar decisiones a lo largo del proyecto, estaremos transformando el proyecto en un conjunto de proyectos pequeños.

Esta planificación a corto plazo nos permitirá tener software disponible para nuestros clientes y además ir aprendiendo del feedback para hacer nuestra planificación más sensible, sea ante inconvenientes que aceleren o retrasen nuestro producto.

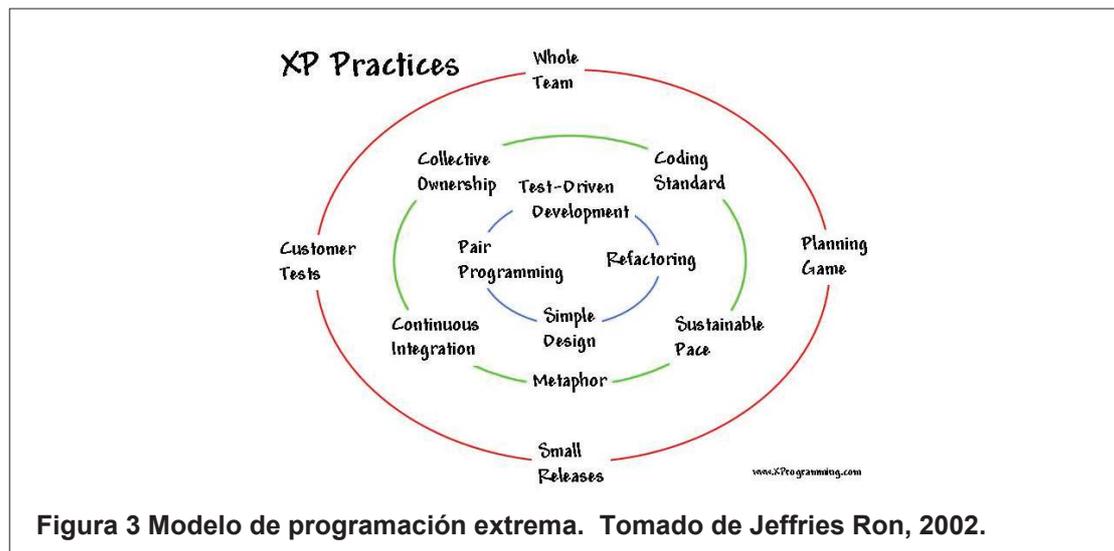
#### **1.2.2.1 Programación Extrema “Extreme Programming” (XP)**

La programación extrema es una metodología ágil que se centra en fortalecer las relaciones interpersonales para el éxito en desarrollo de software, promueven el trabajo en equipo, se preocupan por el aprendizaje de los desarrolladores, y ofrecen un buen clima de laboral.

Se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad. Es considerada la metodología referencial de los procesos ágiles de desarrollo de software.

La comunicación con el cliente es de vital importancia para identificar requerimientos y la definición del alcance del proyecto. Mientras más simple es el sistema, menos tendrá que comunicar sobre este, lo que lleva a una comunicación más completa, especialmente si se puede trabajar con un equipo reducido, característico de esta metodología.

Utiliza un técnica denominada Historias de Usuario es utilizada para especificar los requisitos del software. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales (Jeffries Ron, 2002).



## Ventajas

- Apropiado para entornos volátiles
- Modelo flexible a cambios durante sus fases de desarrollo.
- Relación constante con el cliente, la cual provee retroalimentación útil.
- La presión se encuentra distribuida en todas las fases de desarrollo, no se encuentra solamente en la fase final.

## Desventajas

- Definir el alcance del proyecto con el cliente al inicio del mismo.

- Es una metodología solamente recomendada para proyectos a corto plazo.
- No existe documentación del proyecto.

### **1.2.2.2 Scrum**

Scrum es una metodología ágil utilizada para administrar y controlar las fases de desarrollo de software.

El desarrollo de software con esta metodología se realiza de forma iterativa e incremental; cada ciclo o iteración termina con una funcionalidad del sistema. Normalmente las iteraciones pueden durar entre dos y cuatro semanas.

Scrum es utilizada como marco para otras prácticas de ingeniería de software como RUP o Extreme Programming.

Esta metodología está enfocada en la priorización de tareas en función del valor que el cliente tenga, maximizando la utilidad de lo que se desarrolla y en torno a la inversión. Es una metodología diseñada para adaptarse a cambios de requerimientos durante el desarrollo de software.

Durante el desarrollo de un proyecto se revisan y ajustan los requerimientos y prioridades en intervalos muy cortos y regulares, con el objetivo de adaptar en tiempo real el producto que se está desarrollando a las necesidades y requerimientos del cliente. Scrum busca entregar productos de software que resuelvan las necesidades y aumenten la satisfacción de los clientes.

Scrum está basado sobre una base de principios de revisión continua, innovación, adaptación y auto-gestión. Se genera expectativa y entusiasmo en los clientes al interactuar con él en cada iteración y encuentra las herramientas

con las cuales alinear los objetivos de negocio con los objetivos de su corporación. (Scrum México, 2009).

Scrum emplea las siguientes prácticas para mantener el control de un proyecto:

- Revisión de las iteraciones
- Desarrollo incremental
- Desarrollo evolutivo
- Auto-organización del equipo
- Colaboración



### 1.2.2.3 Microsoft Solution Framework (MSF)

Microsoft Solution Framework (MSF) para aplicaciones Ágiles es una variante de la metodología Microsoft Solution Framework tradicional para el trabajo en entornos en los que se pueda aplicar una metodología ágil.

Esta basada en los mismos principios que las demás metodologías ágiles:

- Potenciar todos los miembros de un equipo
- Potenciar las comunicaciones entre el equipo y el cliente
- Establecer una visión compartida de los valores de negocio del proyecto

- Asegurar una contabilización clara de las responsabilidades compartidas.
- Adaptable a los cambios.
- Aprender de las experiencias.

Esta metodología propone controlar las iteraciones a nivel de funciones que ejerce cada rol en lugar de las fases por las que tiene que pasar cada iteración. Cada fase de su predecesor tendrá uno o varios roles, con sus correspondientes actividades y entregables específicos por cada etapa (Atello 334, 2007). En la Tabla 1 se observan los roles que contempla esta metodología:

**Tabla 1. MSF Ágil Roles por Fase de Desarrollo**

Fase	Rol
Visión del Proyecto	Analista de Negocio
Planificación	Gerente de proyecto Arquitecto
Desarrollo	Desarrollador
Estabilización	Analista Funcional (Tester)
Implementación	Jefe de publicación y distribución

Microsoft Solution Framework Ágil define un grupo de tareas a realizar durante cada iteración de los roles que intervienen en un ciclo de vida de desarrollo de software incluyendo analistas de negocio, arquitectos, jefes de proyecto desarrolladores y probadores.

La Tabla 1 muestra las principales actividades asociadas a cada una de las tareas definidas.

**Tabla 2. MSF Ágil Tareas y Actividades (Spinec, 2010)**

Tareas	Actividades
Creación del Backlog del Proyecto	Capturar la visión del proyecto
Crear una solución de arquitectura	<ul style="list-style-type: none"> <li>• Crear un prototipo de arquitectura.</li> <li>• Determinar interfaces</li> <li>• Crear una arquitectura de infraestructura.</li> </ul>
Crear un escenario de uso del cliente	<ul style="list-style-type: none"> <li>• Irrumpir en los escenarios.</li> <li>• Capturar los escenarios en la herramienta de Team Foundation Server (TFS).</li> <li>• Priorizar la lista de escenarios.</li> </ul>
Crear una calidad de requisitos de servicio	<ul style="list-style-type: none"> <li>• Identificar objetivos de seguridad.</li> <li>• Identificar objetivos de rendimiento</li> <li>• Capturar los requerimientos en la herramienta de Team Foundation Server (TFS)</li> <li>• Priorizar los requerimientos</li> </ul>
Planificación por iteración	<ul style="list-style-type: none"> <li>• Dividir los escenarios y requerimientos en tareas de desarrollo y pruebas.</li> <li>• Crear estimaciones de tiempo para las tareas de desarrollo y pruebas.</li> <li>• Calendarizar y asignar tareas de desarrollo y pruebas.</li> </ul>
Desarrollo por iteración	<ul style="list-style-type: none"> <li>• Escribir código para las tareas de desarrollo.</li> <li>• Crear y actualizar pruebas unitarias para cada tarea de desarrollo.</li> <li>• Programar y asignar tareas de desarrollo y pruebas.</li> </ul>
Prueba por iteración	<ul style="list-style-type: none"> <li>• Escribir las pruebas de validación para las tareas de prueba.</li> <li>• Ejecutar los casos de prueba y llevar a cabo pruebas de exploración.</li> <li>• Reportar los errores de las incidencias encontradas.</li> </ul>
Revisión de progreso por iteración	<ul style="list-style-type: none"> <li>• Acceso al progreso de la tarea por iteración.</li> <li>• Priorizar errores identificados en la iteración.</li> <li>• Evaluar los umbrales de métricas de prueba.</li> </ul>

**Ventajas** (Chávez, 2007)

- Aplica e incentiva al trabajo en equipo y a la colaboración.
- Es útil para proyectos de pequeña y gran escala.
- Crea una disciplina de análisis de riesgos que ayuda y evoluciona con el proyecto.
- A través de las herramientas de Microsoft se proveen plantillas que ayudan al proceso de documentación.

**Desventajas**

- Por ser un modelo prescriptivo se requiere mucha documentación en todas las fases de desarrollo.
- El análisis de riesgos es necesario, pero si se lo hace muy exhaustivo puede demorar el avance del proyecto.
- Al estar basado en tecnologías Microsoft es bastante dependiente al uso de las herramientas Microsoft para aplicar la metodología.

**1.2.3 Comparación entre Metodologías Tradicionales y Ágiles**

En la Tabla 3 se compara las metodologías antes mencionadas, considerando aspectos de flexibilidad, participación del cliente dentro del proyecto, y recursos utilizados dentro de las fases de desarrollo de software.

**Tabla 3. Comparación de Metodologías Tradicionales y Ágiles (Spinec 2010)**

Metodologías Tradicionales	Metodologías Ágiles
Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo	Basadas en heurísticas provenientes de prácticas de producción de código
Cierta resistencia a los cambios	Especialmente preparados para cambios durante el proyecto
Impuestas externamente	Impuestas internamente (por el equipo)
Proceso mucho más controlado, con numerosas políticas/normas	Proceso menos controlado, con pocos principios
Grupos grandes y posiblemente distribuidos	Grupos pequeños (menores que 10) y trabajando en el mismo sitio
El cliente interactúa con el equipo de desarrollo mediante reuniones	El cliente es parte del equipo de desarrollo
Más artefactos	Pocos artefactos
Más roles	Pocos roles
La arquitectura del software es esencial y se expresa mediante modelos	Menos énfasis en la arquitectura del software
Existe un contrato prefijado	No existe contrato tradicional o al menos es bastante flexible

#### 1.2.4 Selección de Metodología

La solución a desarrollar tiene objetivos y requerimientos específicos planteados. Considerando la tabla comparativa y el análisis de algunas metodologías mencionadas anteriormente se llegó a la conclusión de que la mejor alternativa para el desarrollo de esta solución informática es el uso de la metodología Microsoft Solution Framework (MSF) Ágil, esta decisión se tomó por las siguientes razones:

- El sistema va a ser desarrollado bajo la plataforma Microsoft con las herramientas Visual Studio 2010 y SQL Server 2008. La metodología MSF recomienda el uso de estas herramientas para la aplicación de la metodología y desarrollo de software.

- La magnitud del proyecto se ajusta al tipo de proyectos que se pueden ejecutar con la metodología MSF Ágil.
- La necesidad de involucrar al cliente durante el desarrollo del proyecto.
- El soporte que se tiene sobre la implantación de la metodología por parte del proveedor.
- Las prestaciones de esta metodología en las fases de desarrollo.

## 2 CAPÍTULO II. ANÁLISIS Y DISEÑO DEL SISTEMA

### 2.1 ANÁLISIS DEL SISTEMA

Con el estudio del caso realizado durante la primera etapa, se realiza el análisis del sistema, el mismo que delimita el alcance y funcionalidad del sistema.

#### 2.1.1 Modelos de Casos de Uso

Los diagramas de casos de uso representan la interacción de los actores con el sistema en desarrollo, los casos de uso del sistema Medical Center.Net se muestran en las figuras 5, 6 y 7.

##### 2.1.1.1 Lista de Actores del Sistema

La Figura 5 muestra los actores que han sido identificados para el sistema:

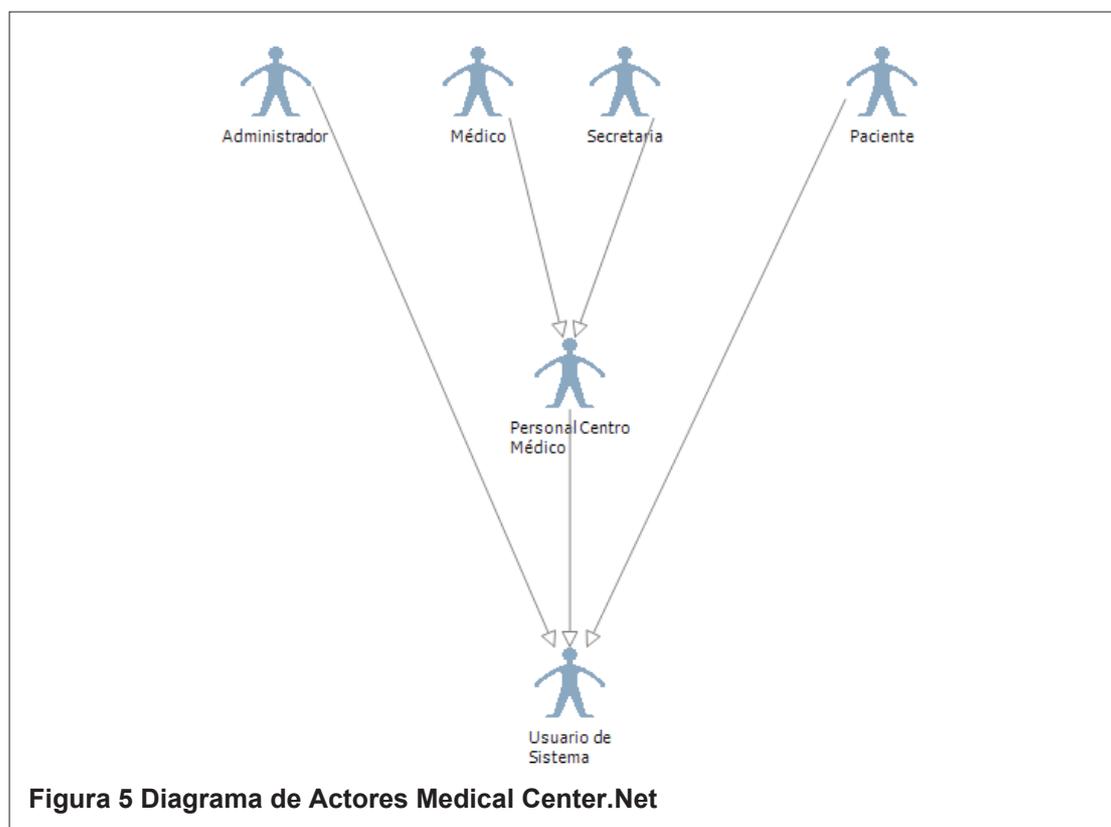
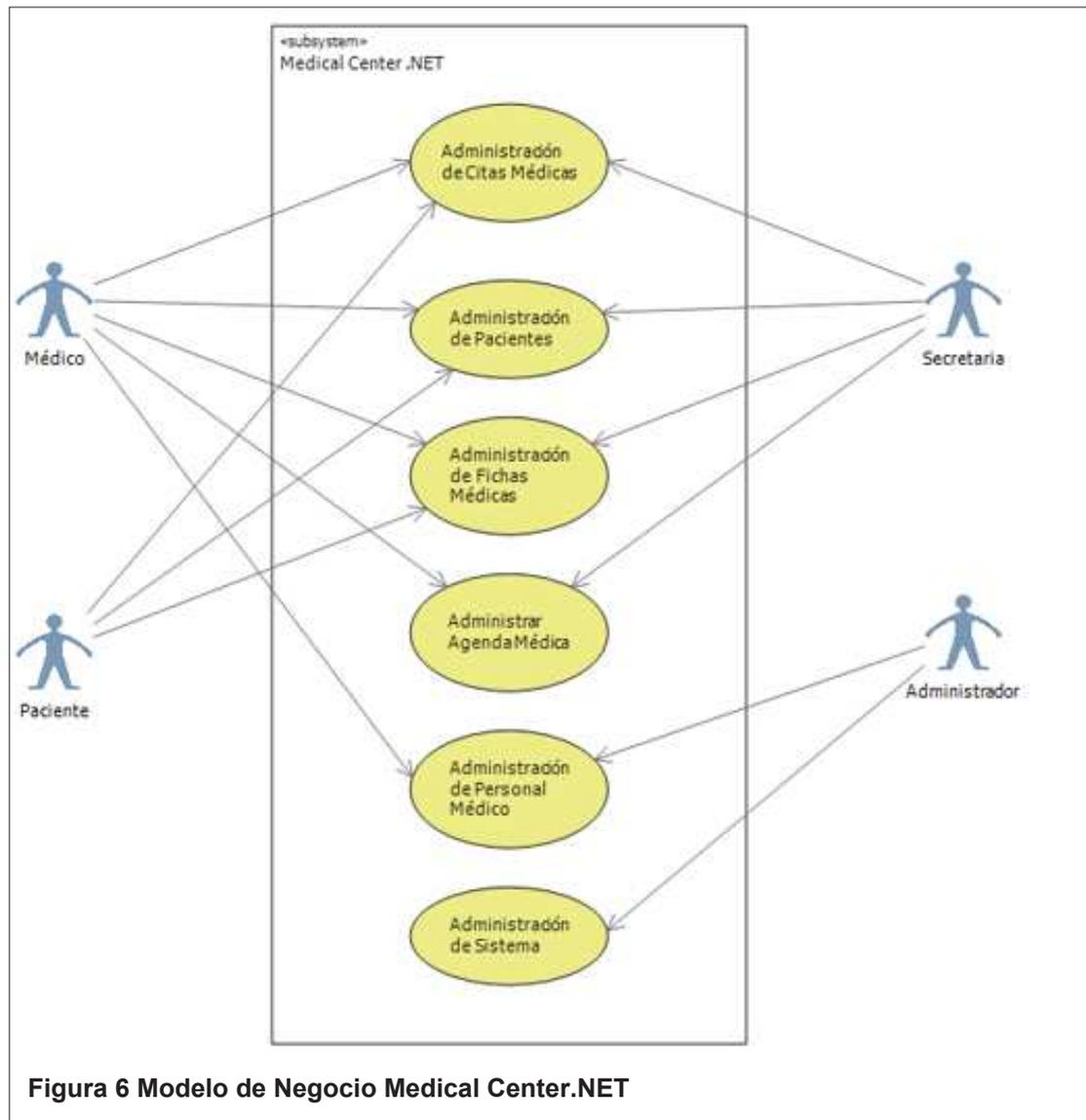


Tabla 4. Descripción de actores del sistema

Actor del Sistema	Descripción
 <p data-bbox="288 557 453 584"><b>Administrador</b></p>	<p data-bbox="523 304 1358 383">Actor que se encuentra asociado a un perfil de usuario administrador con acceso a funcionalidades administrativas y de configuración del sistema. Este actor no forma parte del personal médico del centro médico.</p> <p data-bbox="523 416 1358 465">Este actor puede realizar tareas de ingreso, actualización, borrado y consulta de:</p> <ul data-bbox="523 472 767 667" style="list-style-type: none"> <li data-bbox="523 472 767 495">• Perfiles de Usuario</li> <li data-bbox="523 501 767 524">• Usuarios</li> <li data-bbox="523 530 767 553">• Catálogos</li> <li data-bbox="523 560 767 582">• Parámetros</li> <li data-bbox="523 589 767 611">• Consultorios</li> <li data-bbox="523 618 767 640">• Personal Médico</li> <li data-bbox="523 647 767 667">• Institución Médica</li> </ul>
 <p data-bbox="328 887 429 913"><b>Médico</b></p>	<p data-bbox="523 674 1358 752">Actor que se encuentra asociado a un perfil de usuario médico con acceso a funcionalidades propias de un médico. Este actor forma parte del personal médico del centro médico.</p> <p data-bbox="523 763 1358 813">Este actor puede realizar tareas de ingreso, actualización, borrado y cancelación de:</p> <ul data-bbox="523 819 719 949" style="list-style-type: none"> <li data-bbox="523 819 719 842">• Consultorios</li> <li data-bbox="523 848 719 871">• Citas Médicas</li> <li data-bbox="523 878 719 900">• Ficha Médicas</li> <li data-bbox="523 907 719 929">• Pacientes</li> <li data-bbox="523 936 719 949">• Horarios</li> </ul>
 <p data-bbox="312 1200 437 1227"><b>Secretaria</b></p>	<p data-bbox="523 958 1358 1066">Actor que se encuentra asociado a un perfil de usuario médico con acceso a funcionalidades propias de un médico debido a que se encuentra relacionado directamente con él o ayuda con las tareas de gestión médica. Este actor forma parte del personal médico del centro médico.</p> <p data-bbox="523 1077 1358 1126">Este actor puede realizar tareas de ingreso, actualización, borrado y cancelación de:</p> <ul data-bbox="523 1133 719 1263" style="list-style-type: none"> <li data-bbox="523 1133 719 1155">• Consultorios</li> <li data-bbox="523 1162 719 1184">• Citas Médicas</li> <li data-bbox="523 1191 719 1214">• Ficha Médicas</li> <li data-bbox="523 1220 719 1243">• Pacientes</li> <li data-bbox="523 1249 719 1263">• Horarios</li> </ul>
 <p data-bbox="320 1469 429 1496"><b>Paciente</b></p>	<p data-bbox="523 1299 1358 1377">Actor que se encuentra asociado a un perfil de paciente con acceso a funcionalidades propias de un paciente. Este actor no forma parte del personal médico del centro médico.</p> <p data-bbox="523 1388 1358 1415">Este actor puede realizar tareas de ingreso, actualización y consulta:</p> <ul data-bbox="523 1422 719 1462" style="list-style-type: none"> <li data-bbox="523 1422 719 1444">• Citas Médicas</li> <li data-bbox="523 1451 719 1462">• Ficha Médicas</li> </ul>
 <p data-bbox="277 1671 464 1720"><b>Personal Centro Médico</b></p>	<p data-bbox="523 1500 1358 1550">Actor abstracto que representa al actor Médico y Secretaria. Este actor tiene las mismas funcionalidades de estos dos actores.</p>
 <p data-bbox="256 1895 485 1921"><b>Usuario de Sistema</b></p>	<p data-bbox="523 1724 1358 1774">Actor que representa de forma general a todos los actores que interactúan con el sistema.</p>

### 2.1.1.2 Modelos de Caso de Uso: Modelo de Negocio

La Figura 6 describe el proceso organizacional de mayor importancia en un Centro Medico Integral, Este proceso organizacional corresponde a la Administración de fichas médicas, pacientes, fichas médicas, agenda de médicos, personal médico y administración general.



### 2.1.1.3 Modelos de Caso de Uso: Gestor de Administración

El caso de uso de La Figura 7 muestra la gestión administrativa en la que interviene el actor Administrador.

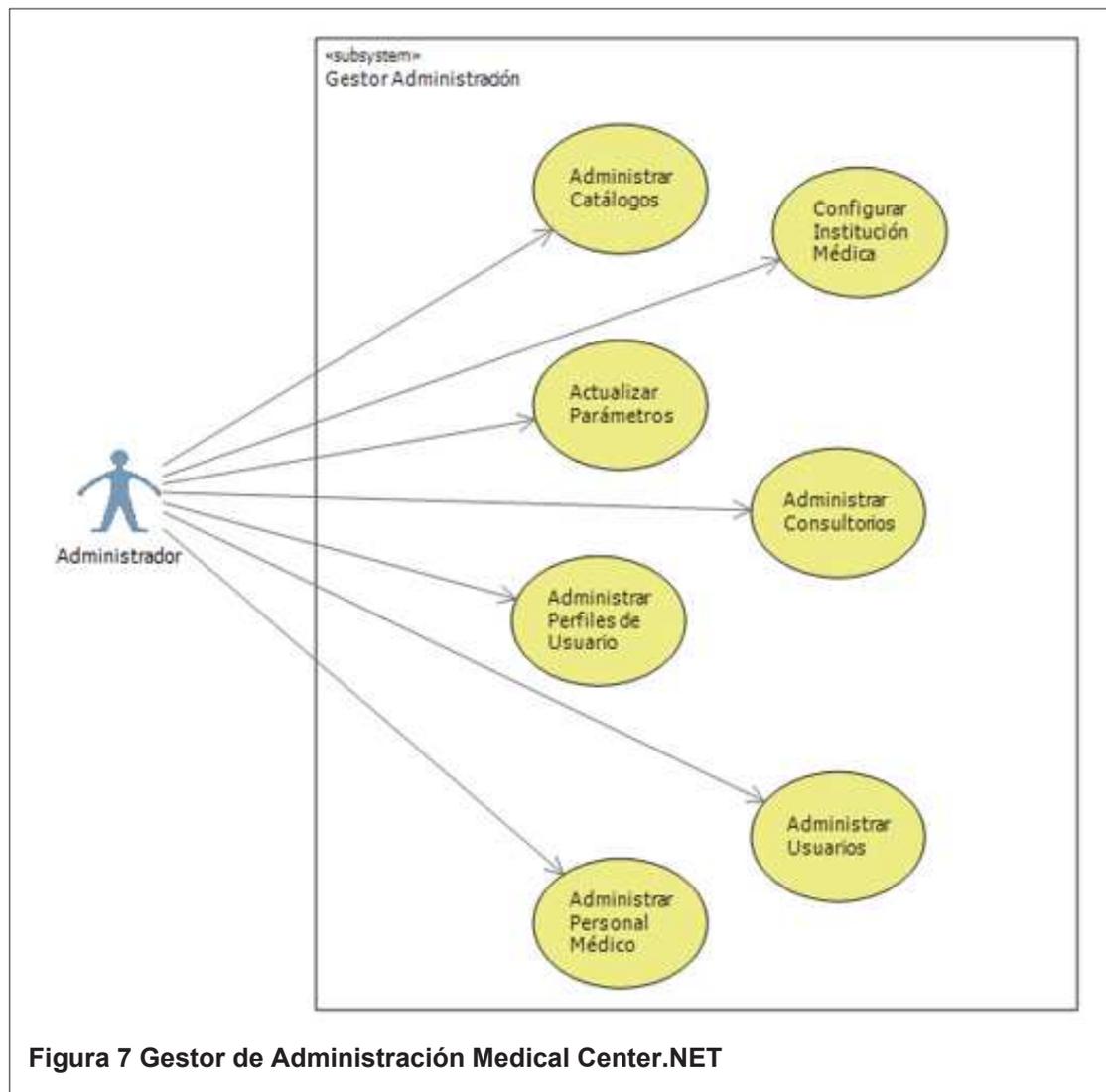


Tabla 5. Caso de Uso Administración de Catálogos

Administrar Catálogos	
<b>Descripción</b>	Permite crear, actualizar y eliminar detalles de catálogo dentro del sistema
<b>Actor</b>	Administrador
<b>Pre-condición</b>	Tener creado el catálogo
<b>Post-condición</b>	Visualizar el resultado de la acción ejecutada contra los catálogos.
<b>Flujo de Eventos</b>	
<ol style="list-style-type: none"> <li>1. El caso de uso inicial cuando el administrador del sistema ingresa al sistema, valida su usuario y contraseña en la pantalla de inicio.</li> <li>2. El sistema presenta la pantalla de administración con las funcionalidades administrativas del sistema.</li> </ol> <p><b>Subflujo Básico 1: Crear Detalle de Catálogo</b></p> <ol style="list-style-type: none"> <li>1. El usuario selecciona la opción Administrar Catálogos.</li> <li>2. El usuario selecciona la opción Crear Detalle de Catálogo.</li> <li>3. El usuario ingresa los datos del detalle de catálogo: catálogo al que pertenece, nombre del detalle, código, valor, estado, editable y orden.</li> <li>4. El usuario selecciona la opción Guardar</li> <li>5. El sistema valida los datos de entrada y verifica que no exista un detalle igual en el catálogo.</li> <li>6. El sistema registra los datos en la base de datos.</li> <li>7. El sistema muestra un mensaje con el resultado del registro</li> <li>8. Termina el caso de uso.</li> </ol> <p><b>Subflujo Básico 2: Actualizar Detalle de Catálogo</b></p> <ol style="list-style-type: none"> <li>1. El usuario selecciona la opción Administrar Catálogos.</li> <li>2. El usuario selecciona la opción Actualizar Detalle de Catálogo.</li> <li>3. El usuario busca el detalle de catálogo que desea actualizar.</li> <li>4. El usuario selecciona el detalle de catálogo que desea actualizar</li> <li>5. El usuario actualiza los datos del detalle de catálogo: catálogo al que pertenece, nombre del detalle, código, valor, estado, editable y orden.</li> <li>6. El usuario selecciona la opción Guardar</li> <li>7. El sistema valida los datos de entrada y verifica que no exista un detalle igual en el catálogo.</li> <li>8. El sistema registra los datos en la base de datos.</li> <li>9. El sistema muestra un mensaje con el resultado de la actualización</li> <li>10. Termina el caso de uso.</li> </ol> <p><b>Subflujo Básico 3: Eliminar Detalle de Catálogo</b></p> <ol style="list-style-type: none"> <li>1. El usuario selecciona la opción Administrar Catálogos.</li> <li>2. El usuario selecciona la opción Actualizar Detalle de Catálogo.</li> <li>3. El usuario busca el detalle de catálogo que desea actualizar.</li> <li>4. El usuario selecciona el detalle de catálogo que desea actualizar</li> <li>5. El usuario selecciona la opción Eliminar</li> <li>6. El sistema valida que el detalle de catálogo no esté en uso.</li> <li>7. El sistema muestra un mensaje con el resultado de la eliminación.</li> <li>8. Termina el caso de uso.</li> </ol>	
<b>Excepción</b>	<p><b>Detalle de Catálogo ya existe</b> El usuario no puede crear o actualizar un detalle de catálogo que ya se encuentre registrado en la base de datos. En este caso el sistema emitirá un error y solicitará nuevamente los datos del detalle de catálogo.</p> <p><b>Detalle de Catálogo está en uso</b> El usuario no puede eliminar un detalle de catálogo que ya se encuentre enlazado con otra información del sistema. En este caso el sistema emitirá un error y solicitará se desenlace previamente los detalles de catálogo (para evitar inconsistencia de datos)</p>

Tabla 6. Caso de Uso Actualizar Parámetro

Actualizar Parámetros	
<b>Descripción</b>	Permite actualizar parámetros generales del sistema
<b>Actor</b>	Administrador
<b>Pre-condición</b>	Tener previamente creado el parámetro
<b>Post-condición</b>	Visualizar el resultado de la actualización del parámetro
<b>Flujo de Eventos</b>	
<ol style="list-style-type: none"> <li>1. El caso de uso inicial cuando el administrador del sistema ingresa al sistema, valida su usuario y contraseña en la pantalla de inicio.</li> <li>2. El sistema presenta la pantalla de administración con las funcionalidades administrativas del sistema.</li> </ol> <p><b>Subflujo Básico 1: Actualizar Parámetro</b></p> <ol style="list-style-type: none"> <li>1. El usuario selecciona la opción Administrar Parámetros.</li> <li>2. El usuario selecciona la opción Actualizar Parámetros.</li> <li>3. El usuario busca el detalle de parámetro que desea actualizar.</li> <li>4. El usuario selecciona el parámetro que desea actualizar.</li> <li>5. El usuario actualiza los datos del parámetro: valor, tipo de dato, descripción y nombre.</li> <li>6. El usuario selecciona la opción Guardar</li> <li>7. El sistema valida los datos de entrada.</li> <li>8. El sistema registra los datos en la base de datos</li> <li>9. El sistema muestra un mensaje con el resultado del registro</li> <li>10. Termina el caso de uso.</li> </ol>	
<b>Excepción</b>	Ninguna

Tabla 7. Caso de Uso Configurar Institución Médica

Configurar Institución Médica	
<b>Descripción</b>	Permite actualizar los datos de una Institución Médica
<b>Actor</b>	Administrador
<b>Pre-condición</b>	Tener creada una institución médica por defecto
<b>Post-condición</b>	Visualizar el resultado de la actualización de la institución médica
<b>Flujo de Eventos</b>	
<ol style="list-style-type: none"> <li>1. El caso de uso inicial cuando el administrador del sistema ingresa al sistema, valida su usuario y contraseña en la pantalla de inicio.</li> <li>2. El sistema presenta la pantalla de administración con las funcionalidades administrativas del sistema.</li> </ol> <p><b>Subflujo Básico 1: Configurar Institución Médica</b></p> <ol style="list-style-type: none"> <li>1. El usuario selecciona la opción Administrar Institución Médica.</li> <li>2. El usuario actualiza los datos de la institución médica: código, nombre, descripción, dirección, teléfono, RUC, correo electrónico, fax.</li> <li>3. El usuario selecciona la opción Guardar</li> <li>4. El sistema valida los datos de entrada.</li> <li>5. El sistema registra los datos en la base de datos</li> <li>6. El sistema muestra un mensaje con el resultado del registro.</li> <li>7. Termina el caso de uso.</li> </ol>	
<b>Excepción</b>	Ninguna

Tabla 8. Caso de Uso Administrar Consultorios

Administrar Consultorios	
<b>Descripción</b>	Permite crear, actualizar y eliminar consultorios dentro del sistema
<b>Actor</b>	Administrador
<b>Pre-condición</b>	Ninguna
<b>Post-condición</b>	Visualizar el resultado de la acción ejecutada contra los consultorios.
<b>Flujo de Eventos</b>	
<ol style="list-style-type: none"> <li>1. El caso de uso inicial cuando el administrador del sistema ingresa al sistema, valida su usuario y contraseña en la pantalla de inicio.</li> <li>2. El sistema presenta la pantalla de administración con las funcionalidades administrativas del sistema.</li> </ol> <p><b>Subflujo Básico 1: Crear Consultorio</b></p> <ol style="list-style-type: none"> <li>1. El usuario selecciona la opción Administrar Consultorios.</li> <li>2. El usuario selecciona la opción Crear Consultorio.</li> <li>3. El usuario ingresa los datos del consultorio: código, grupo, correo electrónico, teléfono, fax, número.</li> <li>4. El usuario selecciona la opción Guardar</li> <li>5. El sistema valida los datos de entrada y verifica que no exista un consultorio con los mismos datos.</li> <li>6. El sistema registra los datos en la base de datos.</li> <li>7. El sistema muestra un mensaje con el resultado del registro.</li> <li>8. Termina el caso de uso.</li> </ol> <p><b>Subflujo Básico 2: Actualizar Consultorio</b></p> <ol style="list-style-type: none"> <li>1. El usuario selecciona la opción Administrar Consultorios.</li> <li>2. El usuario selecciona la opción Actualizar Consultorios.</li> <li>3. El usuario busca el consultorio que desea actualizar.</li> <li>4. El usuario selecciona el consultorio que desea actualizar</li> <li>5. El usuario actualiza los datos del consultorio: código, grupo, correo electrónico, teléfono, fax, y número.</li> <li>6. El usuario selecciona la opción Guardar</li> <li>7. El sistema valida los datos de entrada y verifica que no exista un consultorio con los mismos datos.</li> <li>8. El sistema registra los datos en la base de datos.</li> <li>9. El sistema muestra un mensaje con el resultado de la actualización.</li> <li>10. Termina el caso de uso.</li> </ol> <p><b>Subflujo Básico 3: Eliminar Consultorio</b></p> <ol style="list-style-type: none"> <li>1. El usuario selecciona la opción Administrar Consultorios.</li> <li>2. El usuario selecciona la opción Actualizar Consultorios.</li> <li>3. El usuario busca el consultorio que desea actualizar.</li> <li>4. El usuario selecciona el consultorio que desea actualizar</li> <li>5. El usuario selecciona la opción Eliminar</li> <li>6. El sistema valida que el consultorio no esté en uso.</li> <li>7. El sistema muestra un mensaje con el resultado de la eliminación.</li> <li>8. Termina el caso de uso.</li> </ol>	
<b>Excepción</b>	<p><b>Consultorio ya existe</b> El usuario no puede crear o actualizar un consultorio que ya se encuentre registrado en la base de datos. En este caso el sistema emitirá un error y solicitará nuevamente los datos del consultorio.</p> <p><b>Consultorio está en uso</b> El usuario no puede eliminar un consultorio que ya se encuentre enlazado con otra información del sistema. En este caso el sistema emitirá un error y solicitará se desenlace previamente a los médicos (para evitar inconsistencia de datos).</p>

Tabla 9. Caso de Uso Administrar Personal Médico

Administrar Personal Médico	
<b>Descripción</b>	Permite crear, actualizar y eliminar personal médico
<b>Actor</b>	Administrador
<b>Pre-condición</b>	Ninguna
<b>Post-condición</b>	Visualizar el resultado de la acción ejecutada contra el personal médico
<b>Flujo de Eventos</b>	
<ol style="list-style-type: none"> <li>1. El caso de uso inicial cuando el administrador del sistema ingresa al sistema, valida su usuario y contraseña en la pantalla de inicio.</li> <li>2. El sistema presenta la pantalla de administración con las funcionalidades administrativas del sistema.</li> </ol> <p><b>Subflujo Básico 1: Crear Personal Médico</b></p> <ol style="list-style-type: none"> <li>1. El usuario selecciona la opción Administrar Personal Médico.</li> <li>2. El usuario selecciona la opción Crear Personal Médico.</li> <li>3. El usuario ingresa los datos del personal médico: tipo e identificación, primer nombre, segundo nombre, primer apellido, segundo apellido, fecha de nacimiento, dirección de trabajo y domicilio, estado civil, correo electrónico, teléfono de casa, oficina y celular, ocupación, nacionalidad, género, código, especialización, área y título.</li> <li>4. El usuario selecciona la opción Guardar</li> <li>5. El sistema valida los datos de entrada y verifica que no exista el médico con la misma identificación.</li> <li>6. El sistema registra los datos en la base de datos.</li> <li>7. El sistema muestra un mensaje con el resultado del registro.</li> <li>8. Termina el caso de uso.</li> </ol> <p><b>Subflujo Básico 2: Actualizar Personal Médico</b></p> <ol style="list-style-type: none"> <li>1. El usuario selecciona la opción Administrar Personal Médico.</li> <li>2. El usuario selecciona la opción Actualizar Personal Médico.</li> <li>3. El usuario busca el médico que desea actualizar.</li> <li>4. El usuario selecciona el médico que desea actualizar</li> <li>5. El usuario actualiza los datos del médico: tipo e identificación, primer nombre, segundo nombre, primer apellido, segundo apellido, fecha de nacimiento, dirección de trabajo y domicilio, estado civil, correo electrónico, teléfono de casa, oficina y celular, ocupación, nacionalidad, género, código, especialización, área y título.</li> <li>6. El usuario selecciona la opción Guardar</li> <li>7. El sistema valida los datos de entrada y verifica que no exista un médico con la misma identificación.</li> <li>8. El sistema registra los datos en la base de datos.</li> <li>9. El sistema muestra un mensaje con el resultado de la actualización.</li> <li>10. Termina el caso de uso.</li> </ol> <p><b>Subflujo Básico 3: Eliminar Personal Médico</b></p> <ol style="list-style-type: none"> <li>1. El usuario selecciona la opción Administrar Personal Médico.</li> <li>2. El usuario selecciona la opción Actualizar Personal Médico.</li> <li>3. El usuario busca el médico que desea actualizar.</li> <li>4. El usuario selecciona el médico que desea actualizar</li> <li>5. El usuario selecciona la opción Eliminar</li> <li>6. El sistema valida que el médico no esté en uso.</li> <li>7. El sistema muestra un mensaje con el resultado de la eliminación.</li> <li>8. Termina el caso de uso.</li> </ol>	
<b>Excepción</b>	<p><b>Personal Médico ya existe</b></p> <p>El usuario no puede crear o actualizar un médico que ya se encuentre registrado en la base de datos. En este caso el sistema emitirá un error y solicitará nuevamente los datos del médico.</p>

Tabla 10. Caso de Uso Administrar Perfiles de Usuario

<b>Administrar Perfiles de Usuario</b>	
<b>Descripción</b>	Permite crear, actualizar y eliminar perfiles de usuario
<b>Actor</b>	Administrador
<b>Pre-condición</b>	Ninguna
<b>Post-condición</b>	Visualizar el resultado de la acción ejecutada contra los consultorios.
<b>Flujo de Eventos</b>	
<ol style="list-style-type: none"> <li>1. El caso de uso inicial cuando el administrador del sistema ingresa al sistema, valida su usuario y contraseña en la pantalla de inicio.</li> <li>2. El sistema presenta la pantalla de administración con las funcionalidades administrativas del sistema.</li> </ol> <p><b>Subflujo Básico 1: Crear Perfil de Usuario</b></p> <ol style="list-style-type: none"> <li>1. El usuario selecciona la opción Administrar Perfiles de Usuario.</li> <li>2. El usuario selecciona la opción Crear Perfil de Usuario.</li> <li>3. El usuario ingresa los datos del perfil de usuario: nombre, descripción y estado.</li> <li>4. El usuario asigna las funcionalidades del sistema necesarias.</li> <li>5. El usuario selecciona la opción Guardar</li> <li>6. El sistema valida los datos de entrada y verifica que no exista un perfil igual en el la base de datos.</li> <li>7. El sistema registra los datos en la base de datos.</li> <li>8. El sistema muestra un mensaje con el resultado del registro.</li> <li>9. Termina el caso de uso.</li> </ol> <p><b>Subflujo Básico 2: Actualizar Perfil de Usuario</b></p> <ol style="list-style-type: none"> <li>1. El usuario selecciona la opción Administrar Perfiles de Usuario.</li> <li>2. El usuario selecciona la opción Actualizar Perfil de Usuario.</li> <li>3. El usuario ingresa los datos del perfil de usuario: descripción y estado.</li> <li>4. El usuario asigna o remueve las funcionalidades del sistema necesarias.</li> <li>5. El usuario selecciona la opción Guardar</li> <li>6. El sistema valida los datos a Actualizar</li> <li>7. El sistema registra los datos en la base de datos.</li> <li>8. El sistema muestra un mensaje con el resultado de la actualización.</li> <li>9. Termina el caso de uso.</li> </ol> <p><b>Subflujo Básico 3: Eliminar Perfil de Usuario</b></p> <ol style="list-style-type: none"> <li>1. El usuario selecciona la opción Administrar Perfiles de Usuario.</li> <li>2. El usuario busca el Perfil de Usuario</li> <li>3. El usuario selecciona la opción Eliminar</li> <li>4. El sistema valida que el Perfil no esté en uso en uso, no se puede eliminar un perfil que ya este asociado a usuarios.</li> <li>5. El sistema muestra mensaje de resultado de la eliminación.</li> <li>6. Termina el caso de uso.</li> </ol>	
<b>Excepción</b>	<p><b>Perfil de Usuario ya existe</b></p> <p>El usuario administrador no puede crear o actualizar un perfil de usuario que tenga el mismo nombre que uno ya existe en el sistema. En este caso el sistema emitirá un error y solicitará que ingrese un nombre de perfil de usuario diferente.</p>

Tabla 11. Caso de Uso Administrar Usuarios

<b>Administrar Usuarios</b>	
<b>Descripción</b>	Permite crear, actualizar y eliminar usuarios
<b>Actor</b>	Administrador
<b>Pre-condición</b>	Ninguna
<b>Post-condición</b>	Visualizar el resultado de la acción ejecutada contra los consultorios.
<b>Flujo de Eventos</b>	
<ol style="list-style-type: none"> <li>1. El caso de uso inicial cuando el administrador del sistema ingresa al sistema, valida su usuario y contraseña en la pantalla de inicio.</li> <li>2. El sistema presenta la pantalla de administración con las funcionalidades administrativas del sistema.</li> </ol> <p><b>Subflujo Básico 1: Crear Usuarios</b></p> <ol style="list-style-type: none"> <li>1. El usuario selecciona la opción Administrar Usuarios.</li> <li>2. El usuario selecciona la opción Crear Usuario.</li> <li>3. El usuario ingresa los datos del consultorio: nombre de usuario, alias, contraseña.</li> <li>4. El usuario selecciona la opción Guardar</li> <li>5. El sistema valida los datos de entrada y verifica que no exista un usuario con los mismos datos.</li> <li>6. El sistema registra los datos en la base de datos.</li> <li>7. El sistema muestra un mensaje con el resultado del registro</li> <li>8. Termina el caso de uso.</li> </ol> <p><b>Subflujo Básico 2: Actualizar Usuarios</b></p> <ol style="list-style-type: none"> <li>1. El usuario selecciona la opción Administrar Usuarios.</li> <li>2. El usuario selecciona la opción Actualizar Usuario.</li> <li>3. El usuario busca el Usuario que desea actualizar.</li> <li>4. El usuario selecciona el Usuario que desea actualizar</li> <li>5. El usuario actualiza los datos del consultorio: alias y contraseña.</li> <li>6. El usuario selecciona la opción Guardar</li> <li>7. El sistema valida los datos de entrada y verifica que no exista un usuario con los mismos datos.</li> <li>8. El sistema registra los datos en la base de datos.</li> <li>9. El sistema muestra un mensaje con el resultado de la actualización.</li> <li>10. Termina el caso de uso.</li> </ol> <p><b>Subflujo Básico 3: Eliminar Usuarios</b></p> <ol style="list-style-type: none"> <li>1. El usuario selecciona la opción Administrar Usuarios.</li> <li>2. El usuario selecciona la opción Actualizar Usuario.</li> <li>3. El usuario busca el Usuario que desea actualizar.</li> <li>4. El usuario selecciona el Usuario que desea actualizar</li> <li>5. El usuario selecciona la opción Eliminar</li> <li>6. El sistema muestra un mensaje con el resultado de la eliminación</li> <li>7. Termina el caso de uso.</li> </ol>	
<b>Excepción</b>	<b>Usuario ya existe</b> El usuario administrador no puede crear un usuario que tenga el mismo nombre que uno ya existe en el sistema. En este caso el sistema emitirá un error y solicitará que ingrese un nombre de usuario diferente.

### 2.1.1.4 Modelos de Caso de Uso: Gestor de Negocio

El caso de uso de La Figura 8 muestra la gestión de negocio en la que intervienen los actores Médico, Paciente y Secretaria.

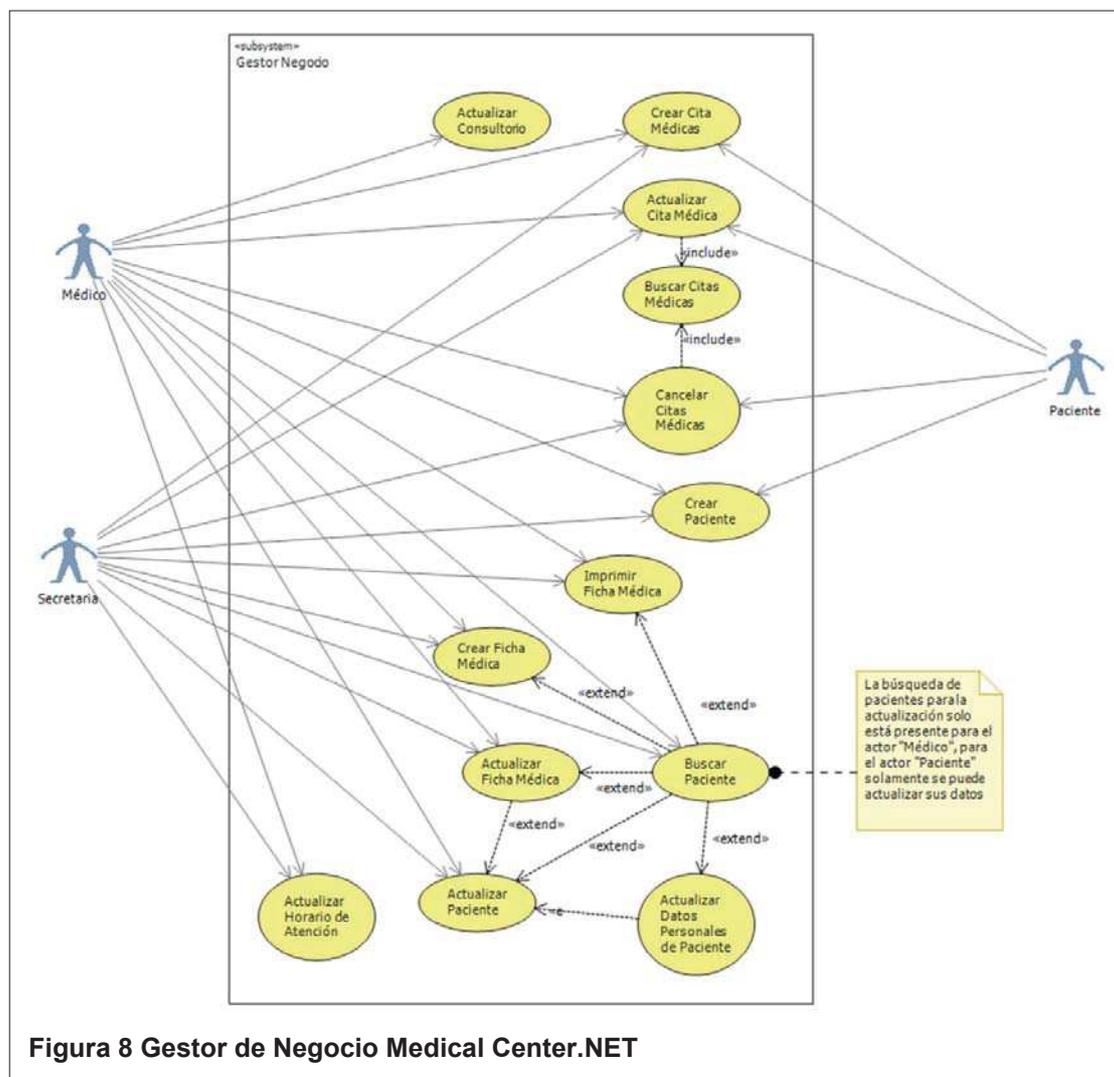


Figura 8 Gestor de Negocio Medical Center.NET

Tabla 12. Caso de Uso Actualizar Consultorios

Actualizar Consultorio	
<b>Descripción</b>	Permite actualizar información referente al consultorio
<b>Actor</b>	Médico, Secretaria
<b>Pre-condición</b>	Ninguna
<b>Post-condición</b>	Visualizar el resultado de la actualización del consultorio
<b>Flujo de Eventos</b>	
<p>1. El caso de uso inicial cuando el administrador del sistema ingresa al sistema, valida su usuario y contraseña en la pantalla de inicio.</p> <p>2. El sistema presenta la pantalla de administración con las funcionalidades administrativas del sistema.</p> <p><b>Subflujo Básico 1: Actualizar Consultorio</b></p> <p>1. El usuario selecciona la opción Administrar Consultorios.</p> <p>2. El usuario selecciona la opción Actualizar Consultorios.</p> <p>3. El usuario busca el consultorio que desea actualizar.</p> <p>4. El usuario selecciona el consultorio que desea actualizar</p> <p>5. El usuario actualiza los datos del consultorio: código, grupo, correo electrónico, teléfono, fax, y número.</p> <p>6. El usuario selecciona la opción Guardar</p> <p>7. El sistema valida los datos de entrada y verifica que no exista un consultorio con los mismos datos.</p> <p>8. El sistema registra los datos en la base de datos.</p> <p>9. El sistema muestra un mensaje con el resultado de la actualización.</p> <p>10. Termina el caso de uso.</p>	
<b>Excepción</b>	Ninguna

Tabla 13. Caso de Uso Crear Cita Médica

<b>Crear Cita Médica</b>	
<b>Descripción</b>	Permite crear citas médicas asociadas a un médico
<b>Actor</b>	Médico, Secretaria, Paciente
<b>Pre-condición</b>	Paciente y Médico deben estar registrados en el sistema
<b>Post-condición</b>	Visualizar el resultado de la creación de la cita médica
<b>Flujo de Eventos</b>	
<ol style="list-style-type: none"> <li>1. El caso de uso inicial cuando el administrador del sistema ingresa al sistema, valida su usuario y contraseña en la pantalla de inicio.</li> <li>2. El sistema presenta la pantalla de administración con las funcionalidades administrativas del sistema.</li> </ol>	
<b>Subflujo Básico 1: Crear Cita Médica</b>	
<ol style="list-style-type: none"> <li>1. El usuario selecciona la opción Médico/ Paciente.</li> <li>2. El usuario selecciona la opción Administrar Citas Médicas.</li> <li>3. El usuario selecciona la opción Nueva Cita Médica.</li> <li>4. El usuario ingresa los datos de la Cita Médica: código, grupo, correo electrónico, teléfono, fax, número.</li> <li>5. El usuario selecciona la opción Guardar</li> <li>6. El sistema valida los datos de entrada y verifica que no exista una cita previamente creada en el mismo horario</li> <li>7. El sistema registra los datos en la base de datos.</li> <li>8. El sistema muestra un mensaje de resultado de guardado.</li> <li>9. Termina el caso de uso.</li> </ol>	
<b>Excepción</b>	<p><b>Cita Médica ya existe en el horario seleccionado</b></p> <p>El usuario no puede crear o actualizar una cita médica en el mismo horario de una cita médica registrada previamente. En este caso el sistema emitirá un error y solicitará nuevamente los datos de la cita médica.</p>

Tabla 14. Caso de Uso Actualizar Cita Médica

Actualizar Cita Médica	
<b>Descripción</b>	Permite actualizar citas médicas asociadas a un médico
<b>Actor</b>	Médico, Secretaria, Paciente
<b>Pre-condición</b>	Ninguna
<b>Post-condición</b>	Visualizar el resultado de la actualización de la cita médica
<b>Flujo de Eventos</b>	
<ol style="list-style-type: none"> <li>1. El caso de uso inicial cuando el administrador del sistema ingresa al sistema, valida su usuario y contraseña en la pantalla de inicio.</li> <li>2. El sistema presenta la pantalla de administración con las funcionalidades administrativas del sistema.</li> </ol> <p><b>Subflujo Básico 1: Actualizar Cita Médica</b></p> <ol style="list-style-type: none"> <li>1. El usuario selecciona la opción Médico/ Paciente.</li> <li>2. El usuario selecciona la opción Administrar Citas Médicas.</li> <li>3. El usuario selecciona la opción Actualizar Cita Médica.</li> <li>4. El usuario busca a través del buscador de Citas Médicas la que se requiere actualizar.</li> <li>5. El usuario ingresa los datos de la Cita Médica que requiere actualizar: código, grupo, correo electrónico, teléfono, fax, número.</li> <li>6. El usuario selecciona la opción Guardar</li> <li>7. El sistema valida los datos de entrada y verifica que no exista una cita previamente creada en el mismo horario</li> <li>8. El sistema actualiza los datos en la base de datos.</li> <li>9. El sistema muestra un mensaje de resultado de actualización.</li> <li>10. Termina el caso de uso.</li> </ol>	
<b>Excepción</b>	<p><b>Cita Médica ya existe en el horario seleccionado</b></p> <p>El usuario no puede crear o actualizar una cita médica en el mismo horario de una cita médica registrada previamente. En este caso el sistema emitirá un error y solicitará nuevamente los datos de la cita médica.</p>

Tabla 15. Caso de Uso Cancelar Cita Médica

Cancelar Cita Médica	
<b>Descripción</b>	Permite cancelar citas médicas
<b>Actor</b>	Médico, Secretaria, Paciente
<b>Pre-condición</b>	Ninguna
<b>Post-condición</b>	Visualizar el resultado de la cancelación de la cita médica
<b>Flujo de Eventos</b>	
<p>1. El caso de uso inicial cuando el administrador del sistema ingresa al sistema, valida su usuario y contraseña en la pantalla de inicio.</p> <p>2. El sistema presenta la pantalla de administración con las funcionalidades administrativas del sistema.</p> <p><b>Subflujo Básico 1: Cancelar Cita Médica</b></p> <p>1. El usuario selecciona la opción Médico/ Paciente.</p> <p>2. El usuario selecciona la opción Administrar Citas Médicas.</p> <p>3. El usuario selecciona la opción Cancelar Cita Médica.</p> <p>4. El usuario busca a través del buscador de Citas Médicas la que se requiere cancelar.</p> <p>5. El usuario confirma la cancelación de la cita.</p> <p>6. El sistema muestra un mensaje de resultado de actualización.</p> <p>7. Termina el caso de uso.</p>	
<b>Excepción</b>	Ninguna

Tabla 16. Caso de Uso Actualizar Horario de Atención

<b>Actualizar Horario de Atención</b>	
<b>Descripción</b>	Permite actualizar información referente al horario de atención
<b>Actor</b>	Médico, Secretaria
<b>Pre-condición</b>	Ninguna
<b>Post-condición</b>	Visualizar el resultado de la actualización del consultorio
<b>Flujo de Eventos</b>	
<ol style="list-style-type: none"> <li>1. El caso de uso inicial cuando el administrador del sistema ingresa al sistema, valida su usuario y contraseña en la pantalla de inicio.</li> <li>2. El sistema presenta la pantalla de administración con las funcionalidades administrativas del sistema.</li> </ol> <p><b>Subflujo Básico 1: Actualizar Horario de Atención</b></p> <ol style="list-style-type: none"> <li>1. El usuario selecciona la opción Administración.</li> <li>2. El usuario selecciona la opción Actualizar Horario de Atención.</li> <li>3. El usuario ingresa los datos de horario a ingresar en el caso de agregar un nuevo detalle.</li> <li>4. El usuario selecciona y elimina el detalle del horario en el caso de eliminar un detalle existente.</li> <li>5. El sistema valida que los datos del horario no se crucen</li> <li>6. El sistema muestra un mensaje de resultado del ingreso o eliminación del detalle de horario.</li> <li>7. Termina el caso de uso</li> </ol>	
<b>Excepción</b>	Ninguna

Tabla 17. Caso de Uso Actualizar Paciente

<b>Actualizar Paciente</b>	
<b>Descripción</b>	Permite actualizar información referente al paciente
<b>Actor</b>	Médico, Secretaria, Paciente
<b>Pre-condición</b>	Ninguna
<b>Post-condición</b>	Visualizar el resultado de la actualización del paciente
<b>Flujo de Eventos</b>	
<p>3. El caso de uso inicial cuando el administrador del sistema ingresa al sistema, valida su usuario y contraseña en la pantalla de inicio.</p> <p>4. El sistema presenta la pantalla de administración con las funcionalidades administrativas del sistema.</p> <p><b>Subflujo Básico 1: Actualizar Paciente</b></p> <p>1. El usuario selecciona la opción Paciente.</p> <p>2. El usuario selecciona la opción Administrar Paciente.</p> <p>3. El usuario busca el consultorio que desea actualizar.</p> <p>4. Se selecciona el paciente que se desea actualizar. En el caso de que el usuario es de tipo Paciente, el buscador de pacientes no se mostrará</p> <p>5. El usuario ingresa los datos del paciente que requiere actualizar: código, grupo, correo electrónico, teléfono, fax, número.</p> <p>6. El usuario selecciona la opción Guardar</p> <p>7. El sistema valida que los datos del paciente estén correctos.</p> <p>8. El sistema muestra un mensaje de resultado de la actualización.</p> <p>9. Termina el caso de uso.</p>	
<b>Excepción</b>	<p><b>Paciente ya existe</b></p> <p>El usuario no puede crear o actualizar un paciente con documento de identificación de otro paciente que previamente haya sido registrado en el sistema En este caso el sistema emitirá un error y solicitará nuevamente los datos del paciente.</p>

Tabla 18. Caso de Uso Actualizar Ficha Médica

Actualizar Ficha Médica	
<b>Descripción</b>	Permite actualizar información referente a la ficha médica
<b>Actor</b>	Médico, Secretaria
<b>Pre-condición</b>	Paciente debe existir
<b>Post-condición</b>	Visualizar el resultado de la actualización de la ficha médica
<b>Flujo de Eventos</b>	
<p>1. El caso de uso inicial cuando el administrador del sistema ingresa al sistema, valida su usuario y contraseña en la pantalla de inicio.</p> <p>2. El sistema presenta la pantalla de administración con las funcionalidades administrativas del sistema.</p> <p><b>Subflujo Básico 1: Actualizar Ficha Médica</b></p> <p>1. El usuario selecciona la opción Paciente.</p> <p>2. El usuario selecciona la opción Ficha Médica</p> <p>3. Se selecciona el paciente que se desea actualizar. En el caso de que el usuario es de tipo Paciente, el buscador de pacientes no se mostrará</p> <p>4. El usuario ingresa los datos de la ficha médica que requiere actualizar: código, grupo, correo electrónico, teléfono, fax, número.</p> <p>5. El usuario selecciona la opción Guardar</p> <p>6. El sistema valida que los datos del de la ficha médica estén correctos.</p> <p>7. El sistema muestra un mensaje de resultado de la actualización.</p> <p>8. Termina el caso de uso.</p>	
<b>Excepción</b>	Ninguna

Tabla 19. Caso de Uso Imprimir Ficha Médica

Imprimir Ficha Médica	
<b>Descripción</b>	Permite imprimir la ficha médica
<b>Actor</b>	Médico, Secretaria, Paciente
<b>Pre-condición</b>	Ninguna
<b>Post-condición</b>	Ficha médica impresa
<b>Flujo de Eventos</b>	
<p>1. El caso de uso inicial cuando el administrador del sistema ingresa al sistema, valida su usuario y contraseña en la pantalla de inicio.</p> <p>2. El sistema presenta la pantalla de administración con las funcionalidades administrativas del sistema.</p> <p><b>Subflujo Básico 1: Imprimir Ficha Médica</b></p> <p>1. El usuario selecciona la opción Paciente.</p> <p>2. El usuario selecciona la opción Ficha Médica</p> <p>3. Se selecciona el paciente que se desea actualizar. En el caso de que el usuario es de tipo Paciente, el buscador de pacientes no se mostrará</p> <p>4. El usuario da selecciona la opción Imprimir Ficha Médica.</p> <p>5. Termina el caso de uso.</p>	
<b>Excepción</b>	Ninguna

Tabla 20. Caso de Uso Buscar Paciente

Buscar Paciente	
<b>Descripción</b>	Permite obtener los datos de un paciente dentro de la base de pacientes
<b>Actor</b>	Médico, Secretaria, Paciente
<b>Pre-condición</b>	Ninguna
<b>Post-condición</b>	Visualizar el resultado de la búsqueda de pacientes
<b>Flujo de Eventos</b>	
<ol style="list-style-type: none"> <li>1. El caso de uso inicial cuando el administrador del sistema ingresa al sistema, valida su usuario y contraseña en la pantalla de inicio.</li> <li>2. El sistema presenta la pantalla de administración con las funcionalidades administrativas del sistema.</li> </ol> <p><b>Subflujo Básico 1: Buscar Paciente</b></p> <ol style="list-style-type: none"> <li>1. El usuario ingresa los datos de paciente.</li> <li>2. El sistema muestra los resultados de búsqueda dado los datos ingresados.</li> <li>3. Termina el caso de uso.</li> </ol>	
<b>Excepción</b>	Ninguna

## 2.2 DISEÑO DEL SISTEMA

La arquitectura para el Sistema de administración de Consultorios y Centros Médicos Integrales está basada en el patrón de diseño en tres niveles (tres capas), el cual organiza el proyecto en tres diferentes componentes:

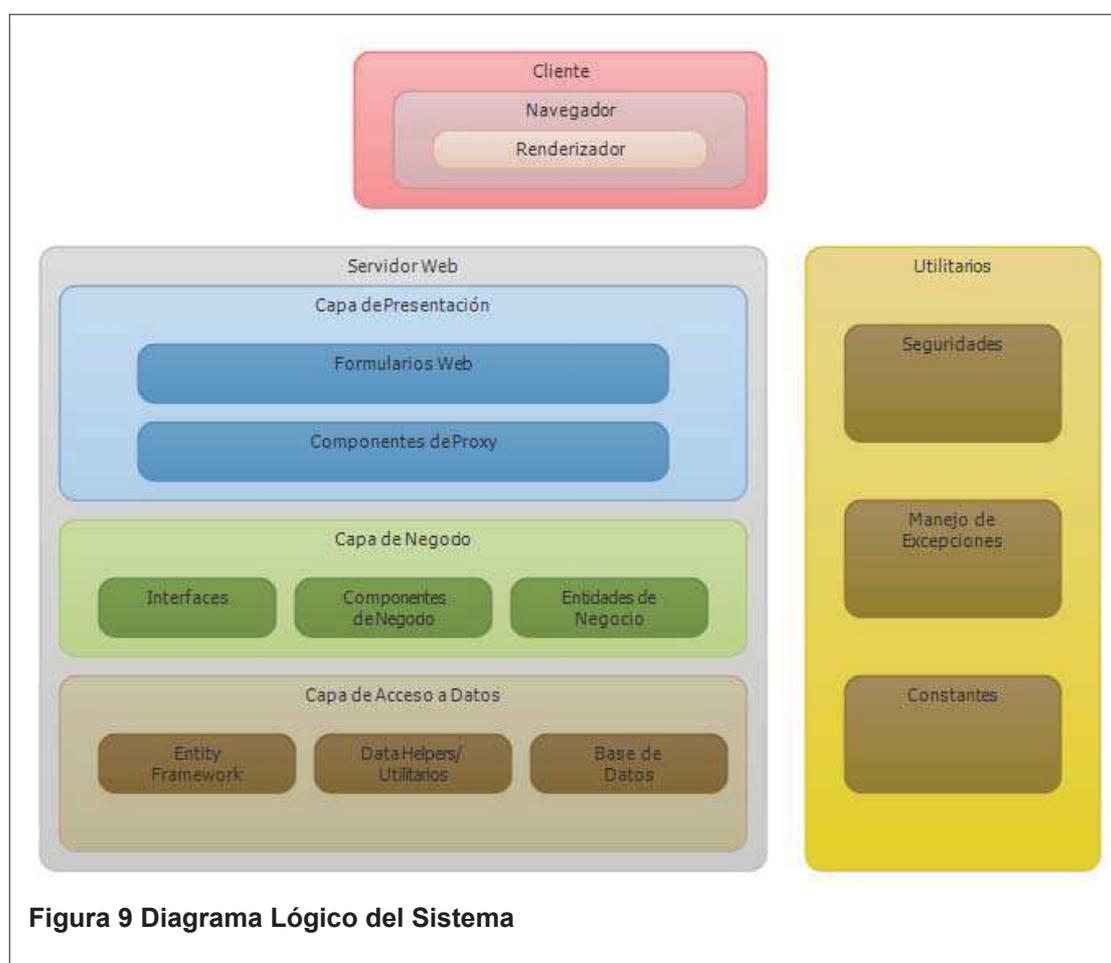
- **Capa de Presentación:** también llamada “Capa de Usuario” es la encargada de interactuar con el usuario recibiendo solicitudes y presentando resultados. (Wikipedia, 2012) Esta capa interactúa directa y únicamente con la capa de negocio a través de una sub capa “Proxy” encargada de exponer las funcionalidades implementadas en la capa de negocio hacia la de presentación. El Proxy permite cambiar la plataforma

Web a Windows e incluso la tecnología de ser necesario, haciendo que toda la lógica de negocio esté separada de la presentación haciendo flexible el uso de otras tecnologías para la implementación de la capa de presentación. Adicionalmente, el Proxy provee un mecanismo de seguridad para la protección del código fuente de la aplicación.

- **Capa de Negocio:** Contiene toda la lógica de negocio del sistema. Esta capa recibe todas las solicitudes realizadas en la capa de presentación, la procesa y envía la solicitud hacia la capa de acceso a datos. Se encarga de enviar todos los resultados provenientes de la capa de acceso a datos a la capa de presentación. La capa de negocio se encuentra implementada y dividida de acuerdo a su funcionalidad:
  - Administración
  - Seguridad
  - BackOffice
  
- **Capa de Datos:** La capa de acceso a datos está dividida en dos sub capas:
  - Entity Framework: Mapea los objetos de la base de datos (tablas, funciones y procedimientos almacenados) en clases y funciones dentro del proyecto utilizando para ello, tecnología “Entity Framework” de Microsoft.
  - Esta capa funciona como DTOs (Data Transfer Object), entidades que sirven para compartir información entre las diferentes capas del sistema con la diferencia que Entity Framework asocia directamente al DTO con la base de datos facilitando el acceso a la base de datos y el mantenimiento del sistema.

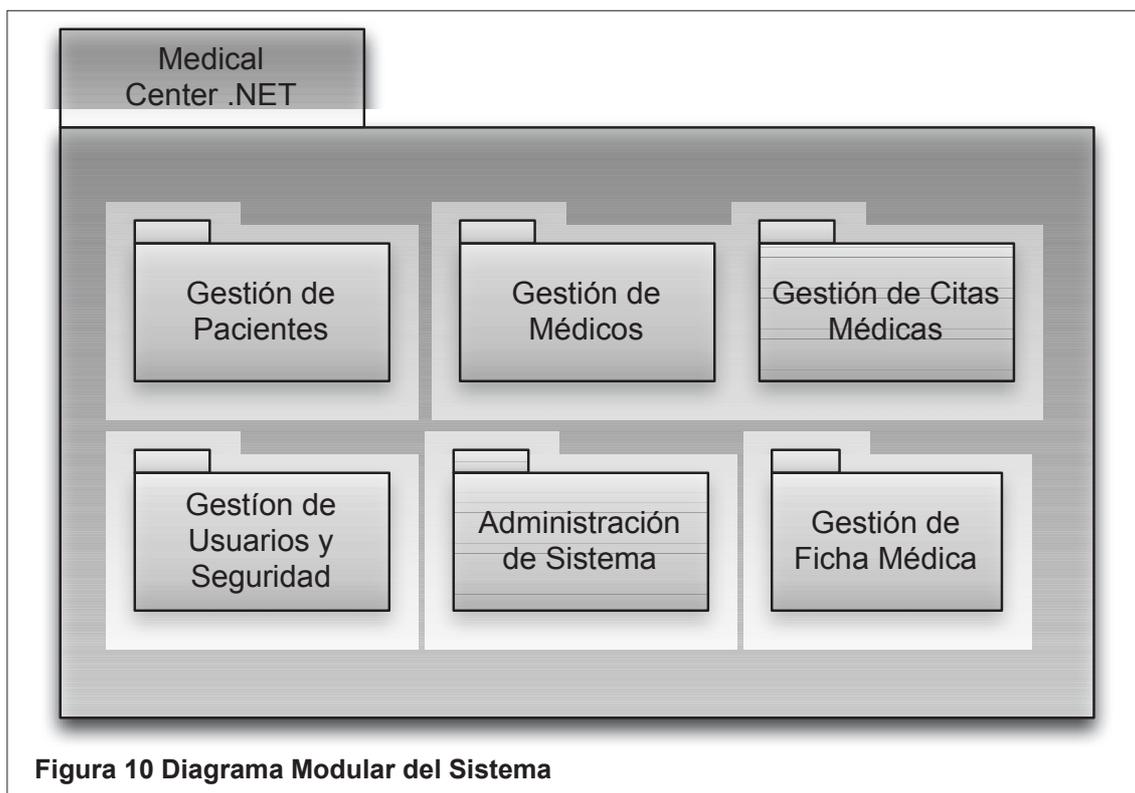
- Capa de Datos: Utiliza la capa de Entity Framework para implementar el acceso a la base de datos. Independiza el acceso a la base de datos con el resto de la aplicación, independientemente de la tecnología de Base de Datos que se vaya a utilizar.

En La Figura 9 se muestra la interacción entre capas de la arquitectura del sistema de forma lógica.



**Figura 9 Diagrama Lógico del Sistema**

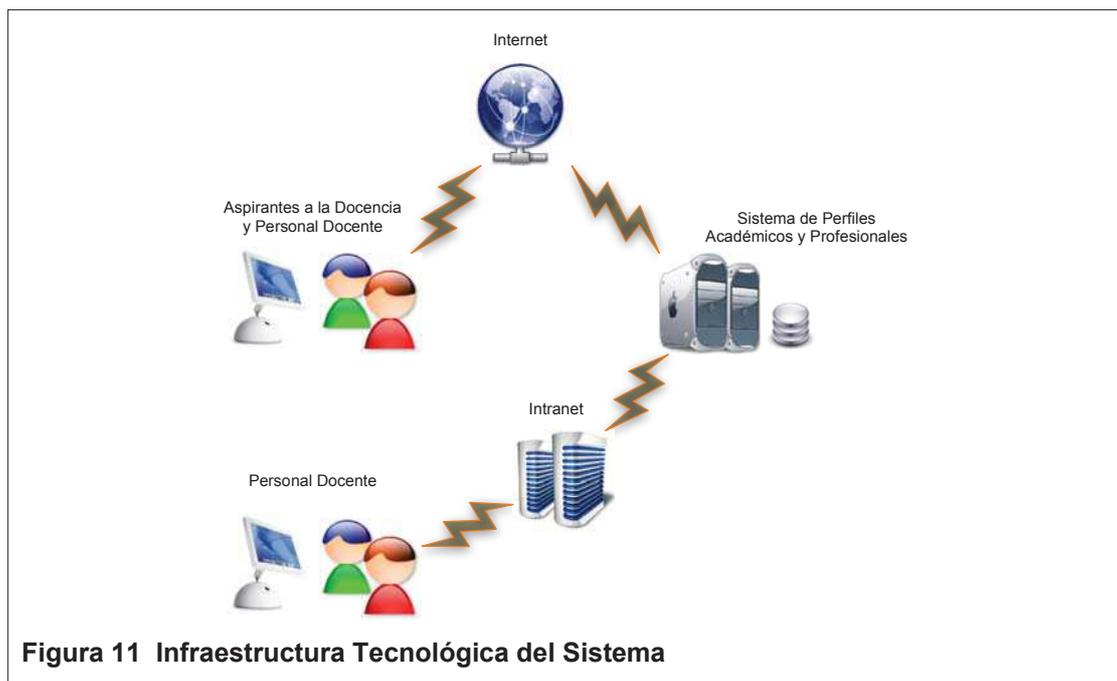
En La Figura 10 se identifican los módulos que conforman toda la funcionalidad del sistema.



### 2.2.1 Infraestructura Tecnológica

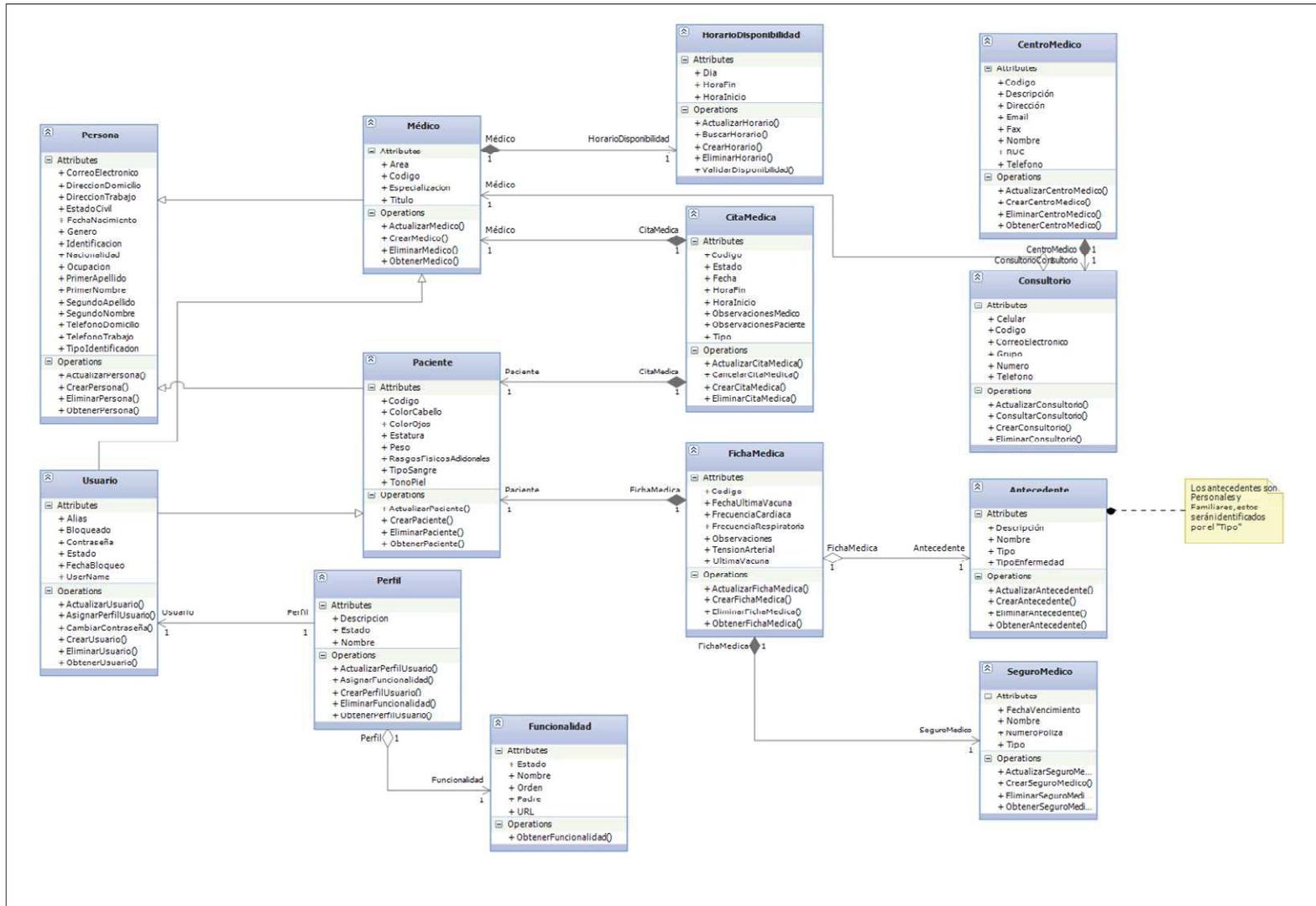
El Sistema para la Administración de Consultorios y Centros Médicos Integrales utiliza como sistema de comunicación, dos medios de acceso, según se muestra en La Figura 11:

- **Internet:** Personal Médico, administrativo y pacientes podrán acceder al sistema a través de internet utilizando un computador con comunicación al mismo.
- **Intranet:** Personal Médico y administrativo podrá acceder al sistema a través de la intranet del Centro Médico utilizando un computador conectado a la intranet de la institución.



### 2.2.2 Diagrama de Clases

La Figura 12 muestra el diagrama de clases con las clases que el sistema Medical Center.Net va a utilizar.



Los antecedentes son Personales y Familiares, estos serán identificados por el "Tipo"

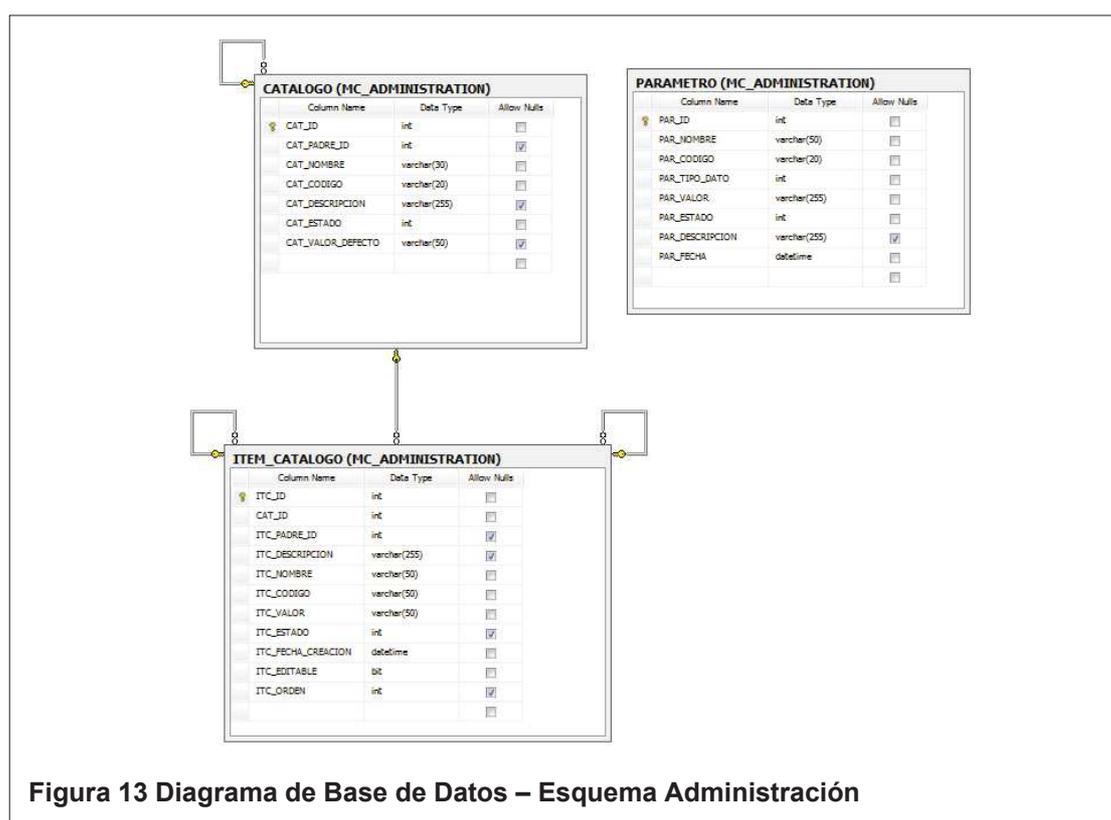
Figura 12 Diagrama de Clases

### 2.2.3 Base de Datos

La base de datos contiene toda la información que el sistema necesita para su funcionamiento (parámetros, catálogos, perfiles de seguridad, información funcional).

El acceso a las tablas de la base de datos para lectura y escritura se la hace a través de procedimientos almacenados, funciones y vistas.

Las figuras 13, 14, 15 muestran el modelo de datos del sistema Medical Center.Net, representado mediante el modelo relacional y organizado por tres esquemas generales: Administración, Negocio y Seguridad.



**Figura 13 Diagrama de Base de Datos – Esquema Administración**

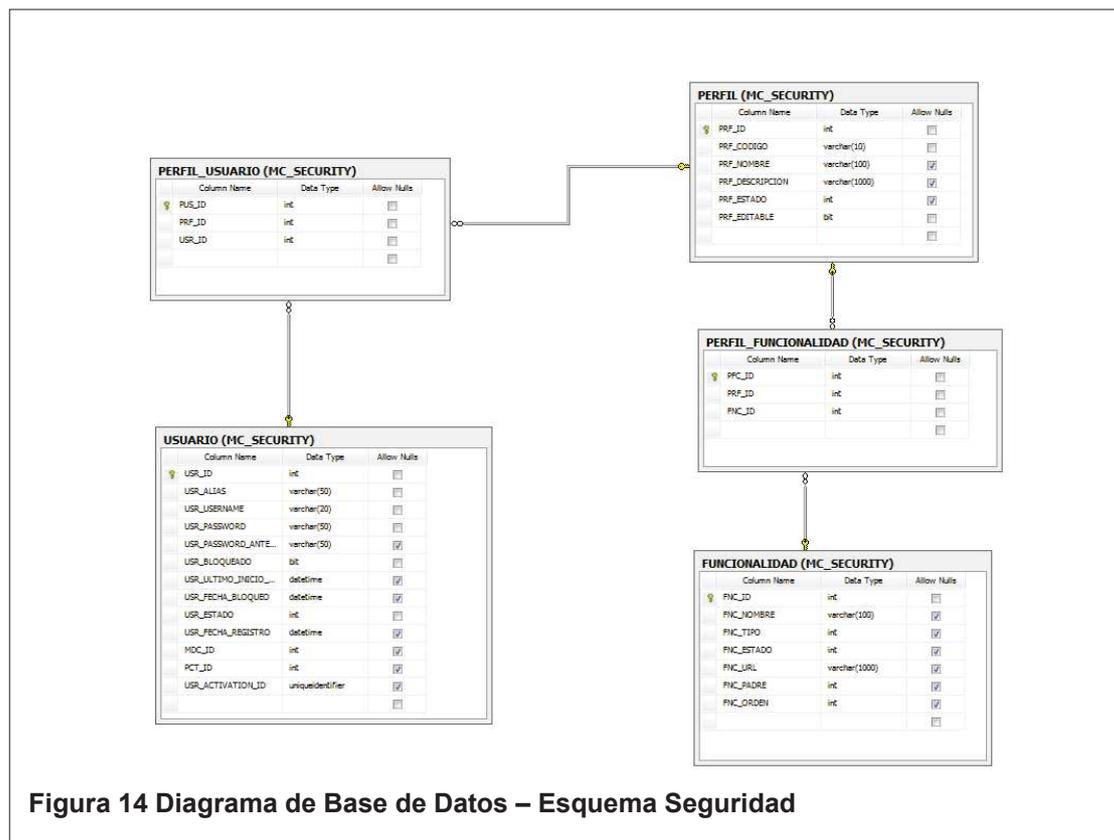


Figura 14 Diagrama de Base de Datos – Esquema Seguridad

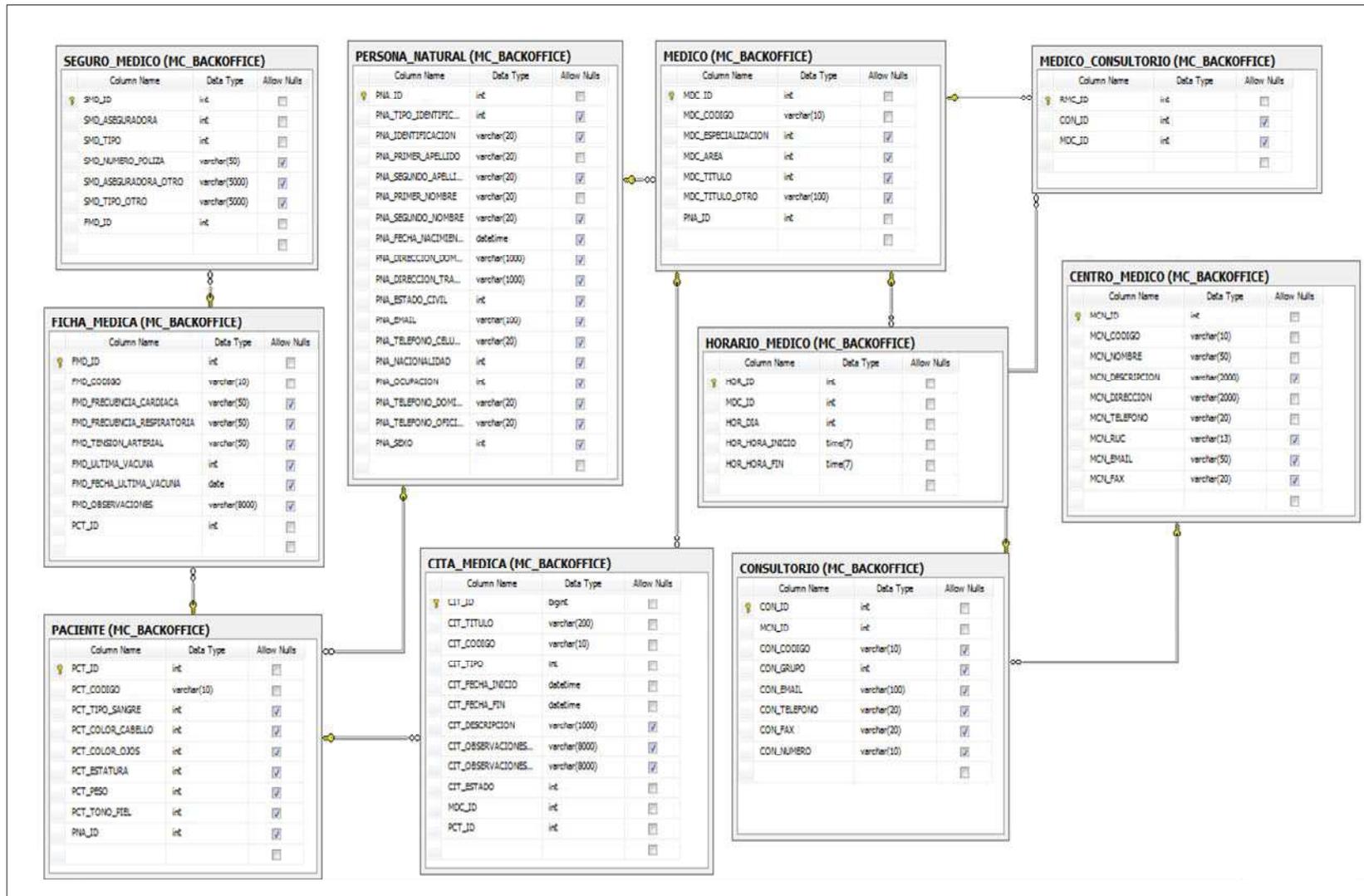
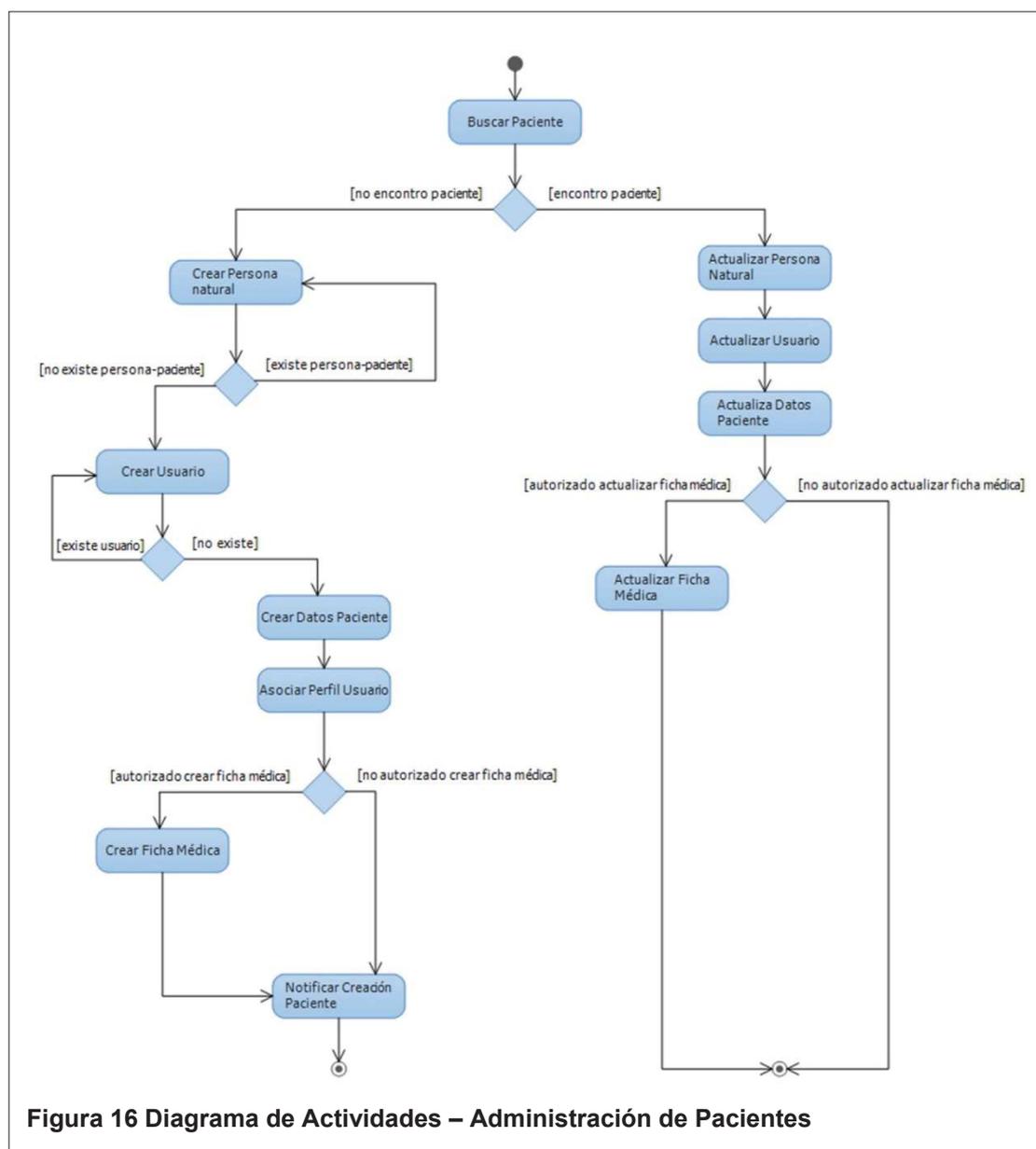


Figura 15 Infraestructura Tecnológica del Sistema

## 2.2.4 Diagrama de Actividades

Los diagramas de actividades de las figuras 16, 17, 18 y 19 muestran el flujo principal de la administración de pacientes, médicos, fichas médicas, citas médicas del sistema.



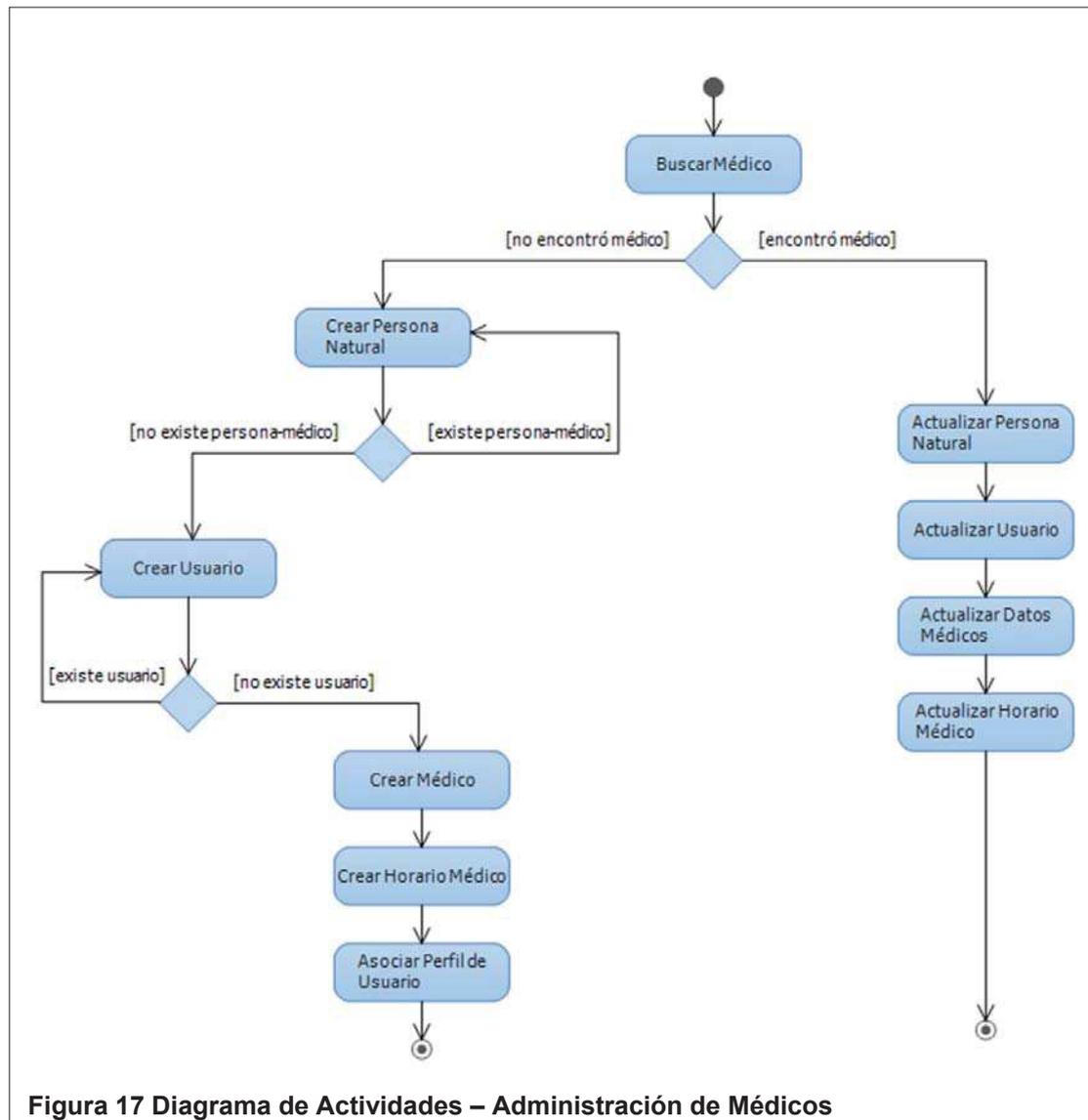


Figura 17 Diagrama de Actividades – Administración de Médicos

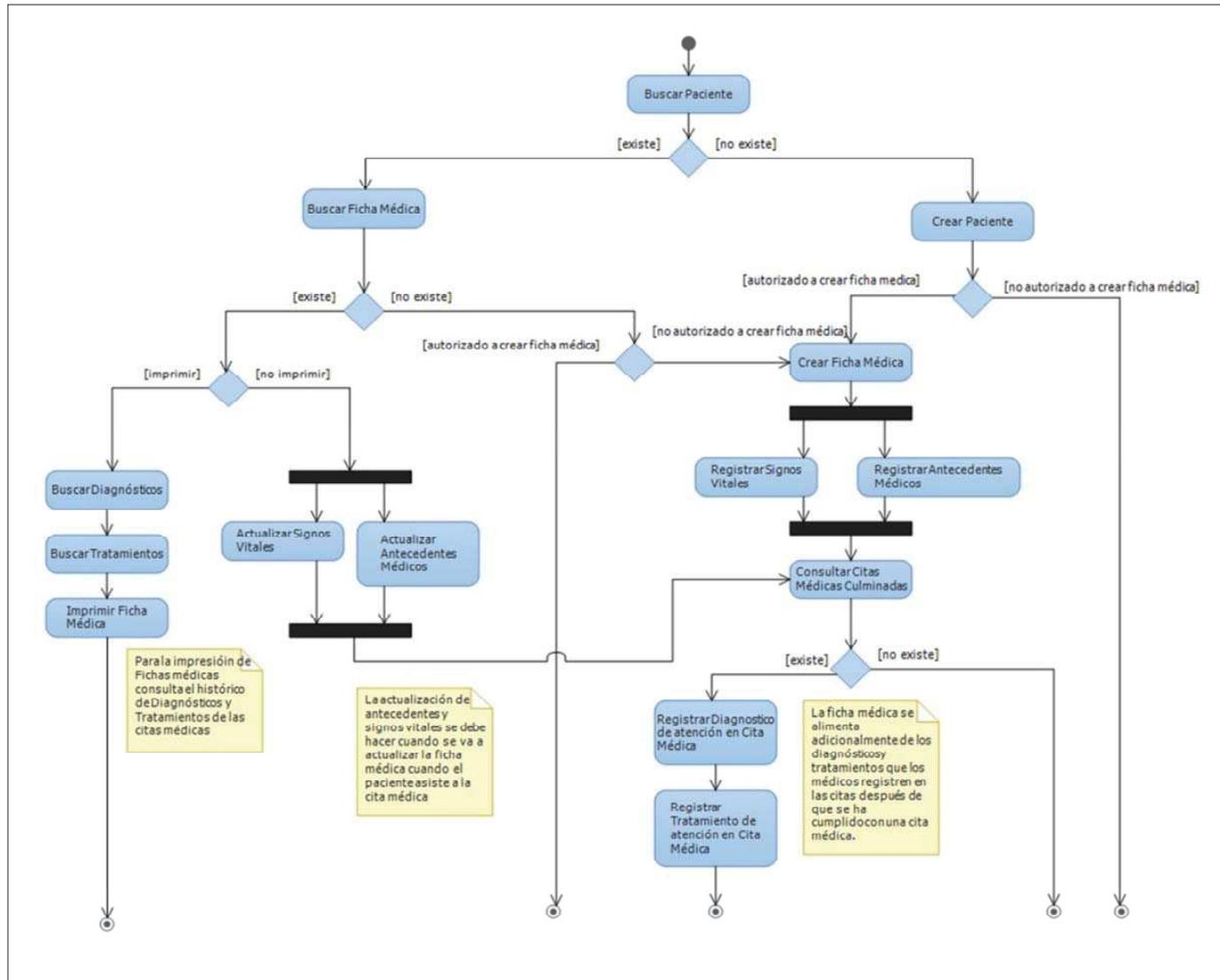


Figura 18 Diagrama de Actividades – Administrar Ficha Médica

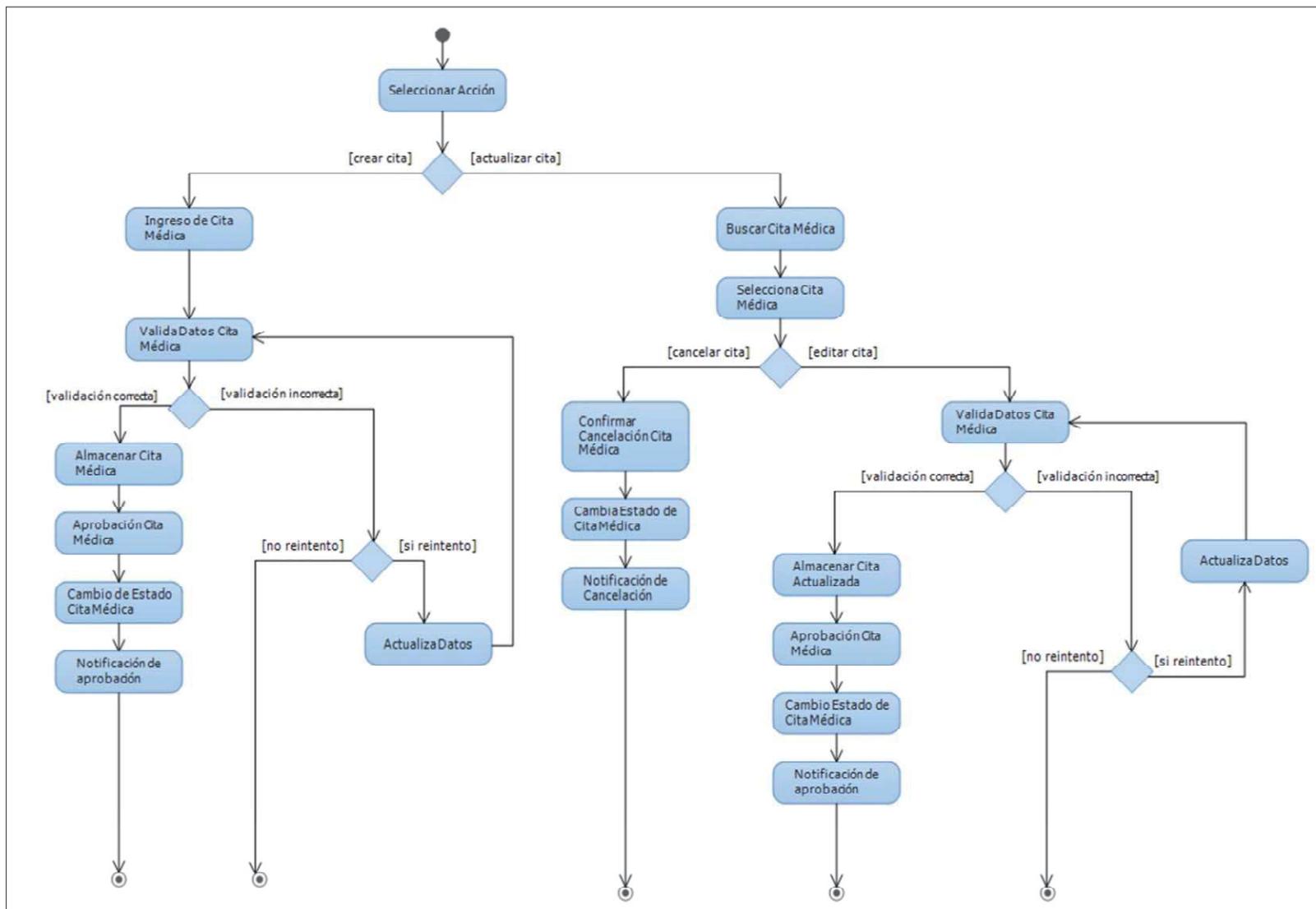


Figura 19 Diagrama de Actividades – Administrar Cita Médica

## 2.2.5 Diagrama de Secuencia

Los diagramas de secuencia de las figuras 20, 21 y 22 muestran el flujo principal de la administración de fichas médicas y citas médicas del sistema.

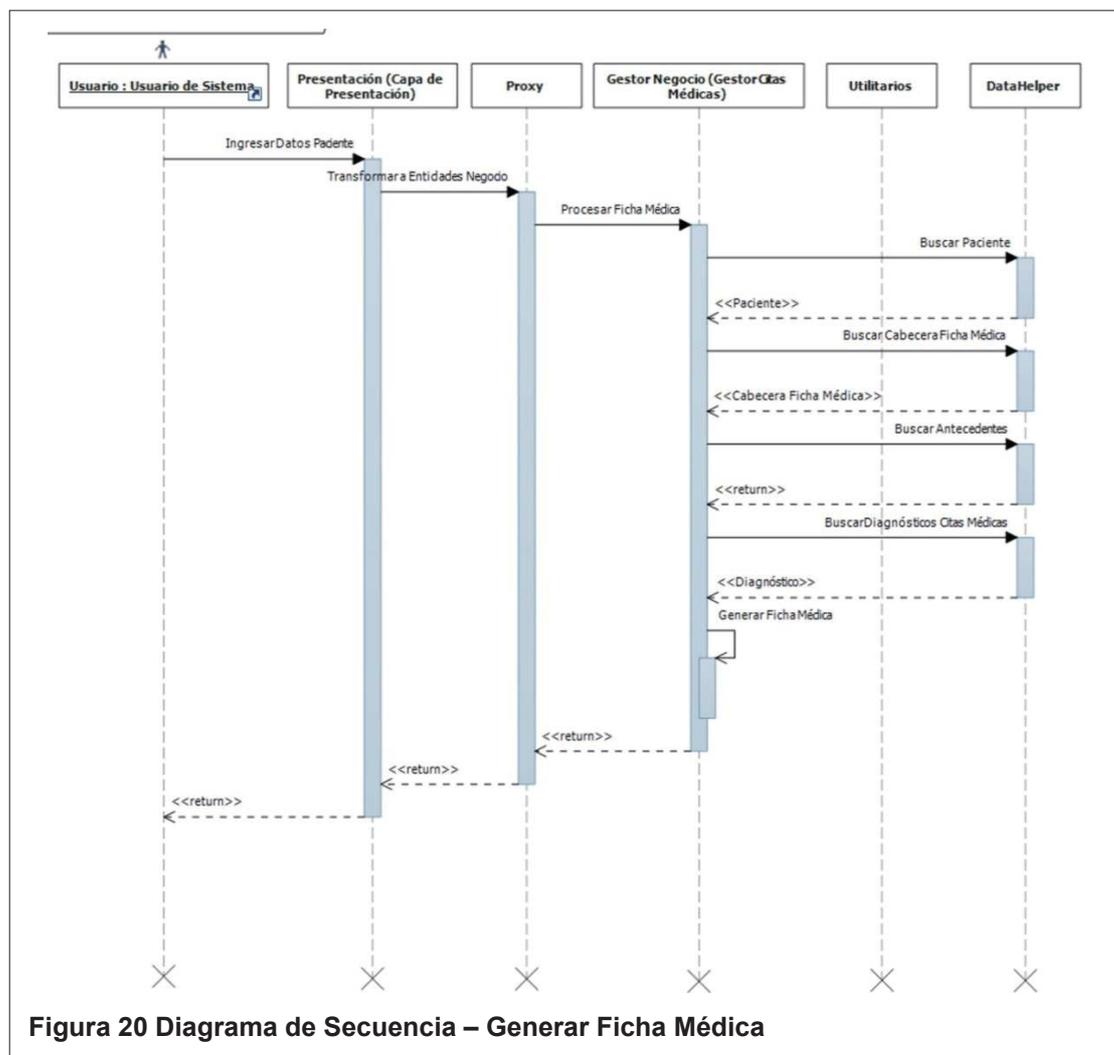


Figura 20 Diagrama de Secuencia – Generar Ficha Médica

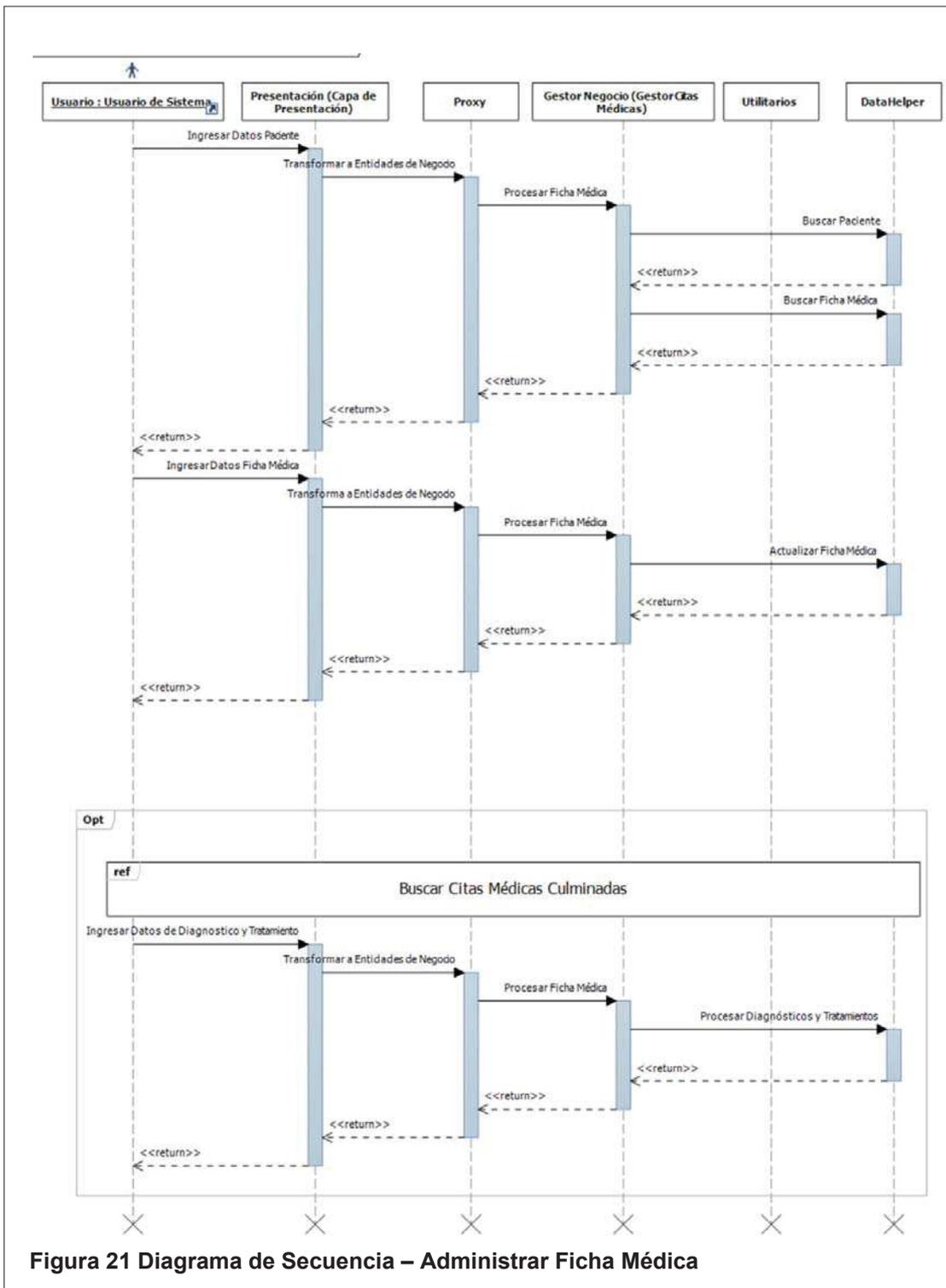


Figura 21 Diagrama de Secuencia – Administrar Ficha Médica

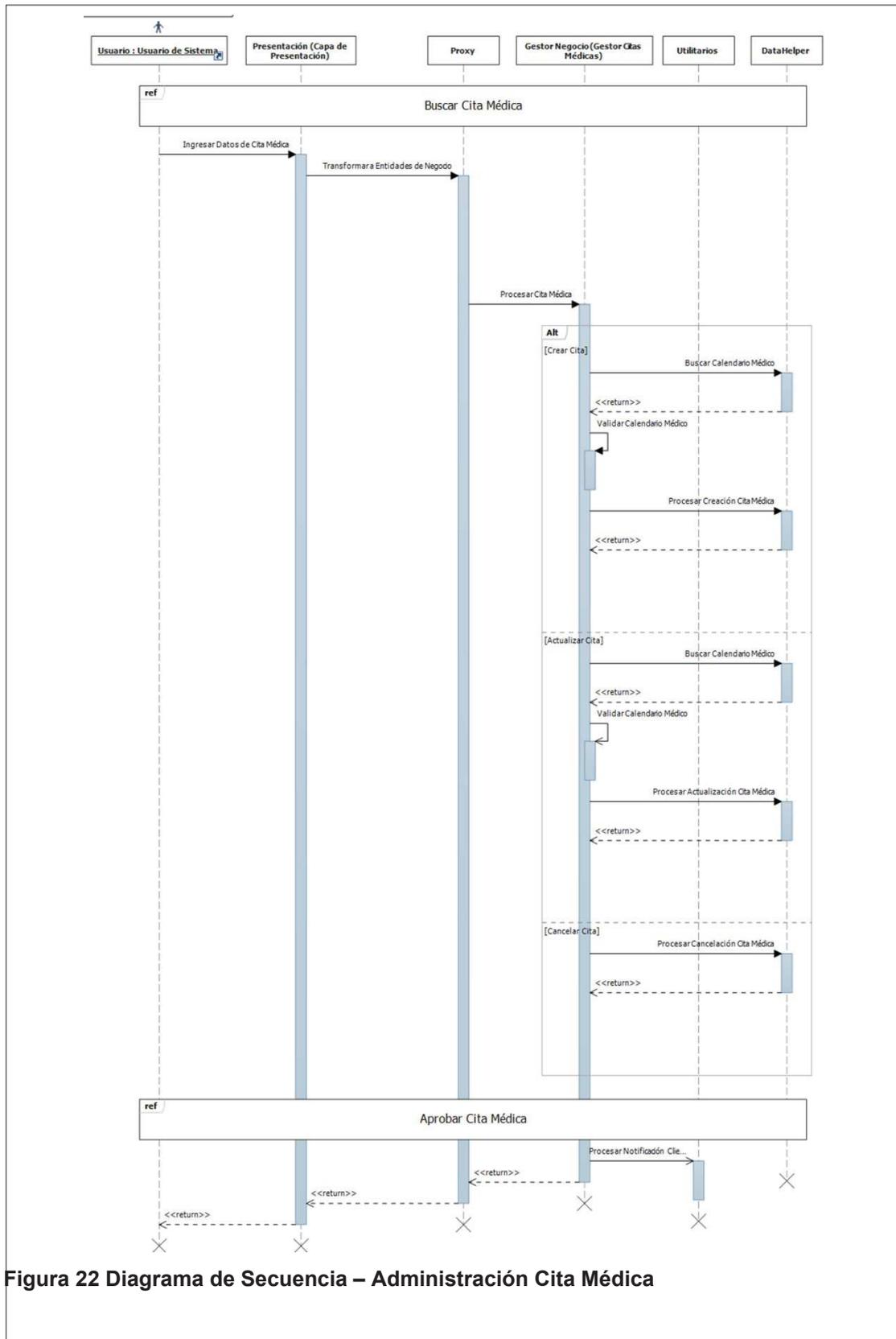


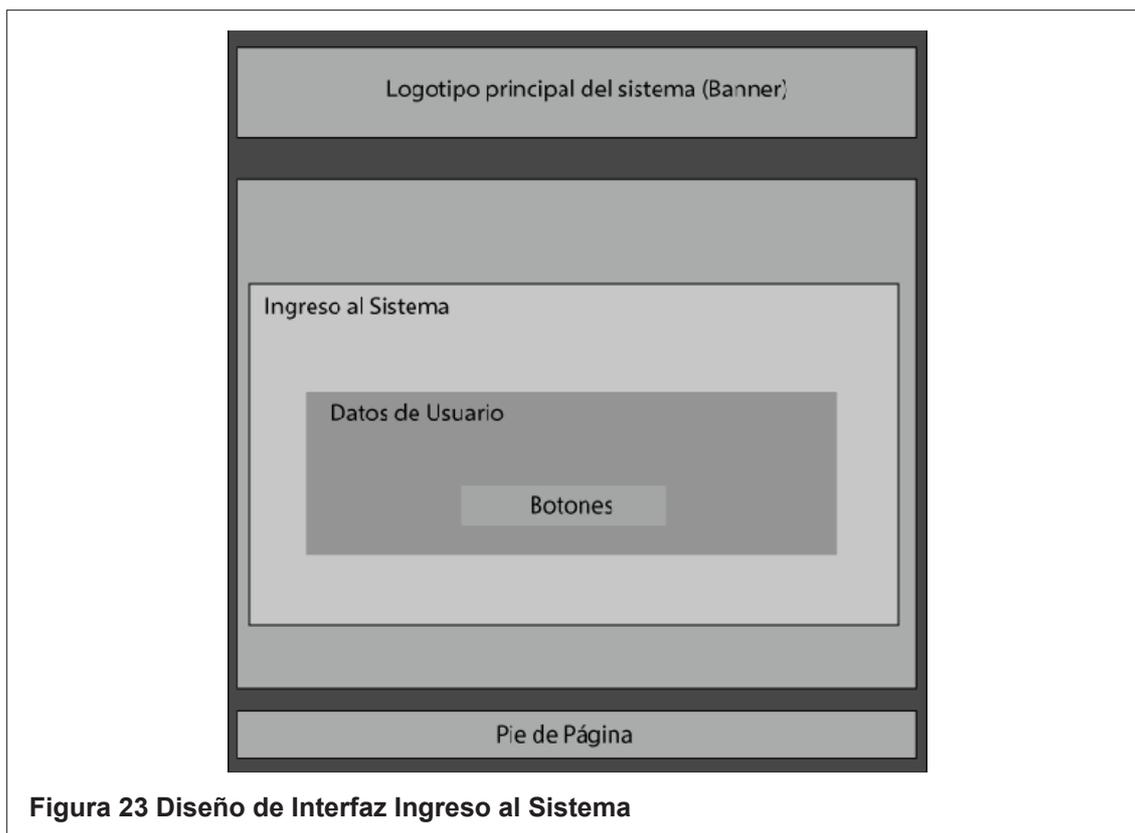
Figura 22 Diagrama de Secuencia – Administración Cita Médica

## 2.2.6 Diseño de Interfaces de Usuario

Medical Center.Net es un sistema basado en plataforma web; para que la experiencia de usuario sea fácil se ha definido un diseño estándar de las interfaces de usuario. Las figuras 23 y 24 y su descripción en las tablas 21 y 22 presentan el diseño básico sobre el cual se van a realizar las interfaces de usuario.

**Tabla 21. Ingreso al Sistema**

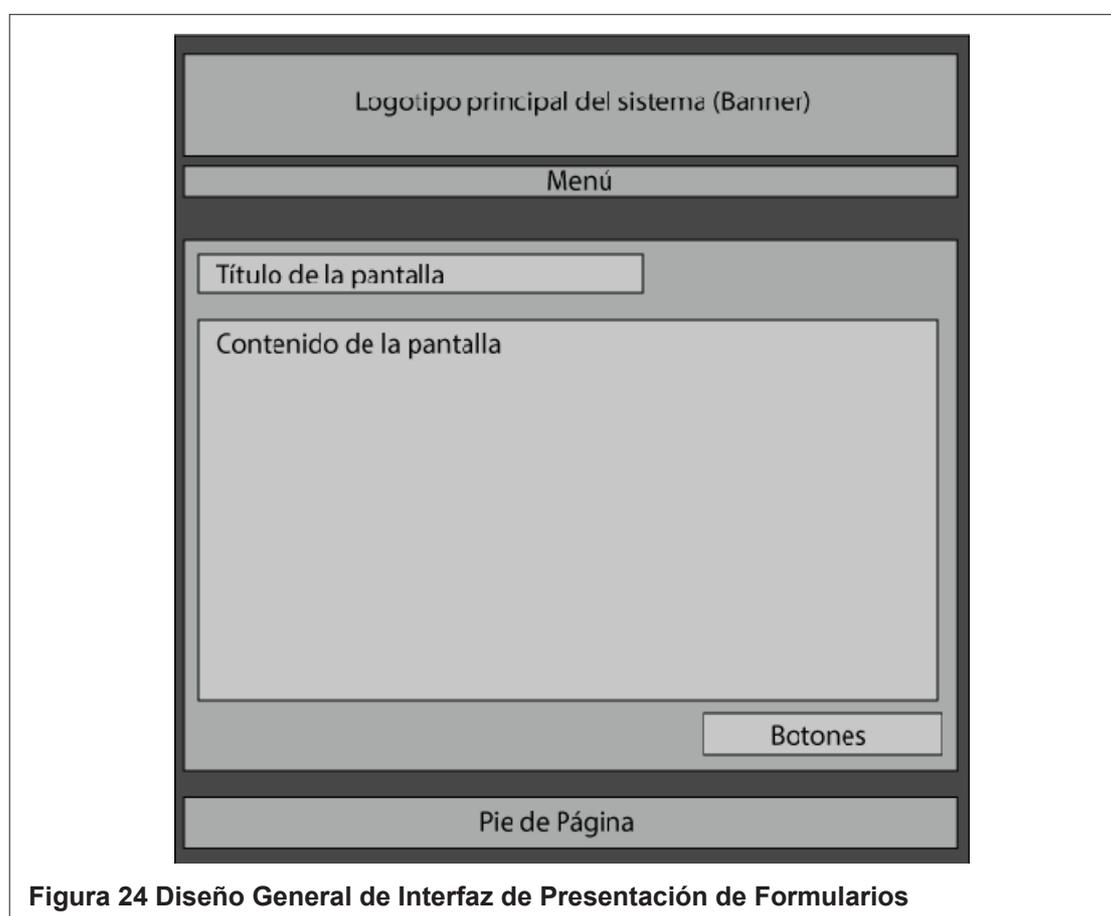
Ingreso al Sistema	
<b>Descripción</b>	Prototipo de ingreso al sistema en el cual el usuario ingresará su usuario y contraseña para ser verificado.



**Figura 23 Diseño de Interfaz Ingreso al Sistema**

**Tabla 22. Diseño General de Interfaz de Presentación de Formularios Web**

Diseño General de Interfaz de Presentación de Formularios Web	
<b>Descripción</b>	<p>Diseño de interfaz base para todos los usuarios independientemente de su perfil.</p> <p>Dependiendo de las funcionalidades del perfil se llenará la sección de menú.</p> <p>Todas las pantallas se cargarán en la sección Contenido de Pantalla.</p> <p>El diseño no varía entre pantallas para garantizar la homogeneidad y fácil uso del sistema.</p>



## 3 CAPÍTULO III. IMPLEMENTACIÓN Y PRUEBAS

### 3.1 IMPLEMENTACIÓN

Para la implementación del sistema Medical Center.Net se ha tomado en cuenta los siguientes aspectos:

- Estándares de programación
- Convenciones de Código, nomenclatura y estilos
- Convenciones de nomenclatura de Base de Datos

#### 3.1.1 Estándares de Programación

El uso de buenas prácticas de programación es fundamental en un proyecto de desarrollo de software, esto facilita el desarrollo de funcionalidades nuevas y el mantenimiento de un sistema sin importar el recurso que se emplee para estas tareas. (Lowi, 2008, p. 4-14)

El estándar de programación seleccionado proviene de una guía y mejores prácticas para lenguaje de programación C# y SQL Server creado por una empresa enfocada en la excelencia del desarrollo de software, IDesign.

##### 3.1.1.1 Convenciones de Código, Nomenclatura y Estilos

- Utilización de método de capitalización Pascal para el nombre de clases, métodos y constantes.

```
public class ClaseEjemplo
{
    const int ConstanteEjemplo = 100;
    public void MetodoEjemplo() {}
}
```

- Utilización de método de capitalización Camel para variables locales y parámetros en las firmas de los métodos.

```
void MetodoEjemplo(int enteroEjemplo)
{
int numero;
}
```

- Utilización del prefijo “I” en nombres de interfaces.

```
interface IPruebaInterface
```

- Utilización del prefijo “m\_” en nombres de miembros privados, seguidos del nombre de los miembros privados con capitalización Pascal.

```
private int m_Number;
```

- Nombrar los métodos combinando el verbo con el objeto (verbo-objeto).

```
void ObtenerPersona() {}
```

- Los Métodos que retornan valor deben ser nombrados con una descripción del valor que retornan.

```
string ObtenerPrimerNombrePersona() {}
```

- Los nombres de las variables deben ser descriptivos.
  - No se debe declarar variables con caracteres como **i** o **t**. Utilice en esos casos **index** o **temp**.
  - Evite utilizar notación Húngara como **strNombre**.

- Los nombres de las variables, métodos y clases no deben abreviarse como **num** en lugar de **number**.
- Utilice el sufijo **Exception** para objetos de tipo excepción.
- Evitar interfaces con un solo miembro y más de veinte miembros.
- Documentar en cada cabecera de método con una descripción detallada del funcionamiento del método y parámetros de entrada y salida.

```
/// <summary>
```

```
/// Método que representa un ejemplo de cómo documentar
```

```
/// </summary>
```

```
/// <param name="enteroEjemplo">parámetro de ejemplo</param>
```

```
void MetodoEjemplo(int enteroEjemplo){}
```

### 3.1.1.2 Convenciones de Nomenclatura de Base de Datos

- Utilización de esquemas de seguridad de acuerdo al módulo al que pertenece cada tabla.
  - El nombre del esquema debe estar en mayúscula.
  - Si existen palabras compuestas debe estar separada con el carácter “\_”.
  - Utilizar un prefijo con las siglas del nombre de la base de datos.

Esquema de Seguridad: **AC\_SEGURIDAD**

- Los nombres de las tablas en letras mayúsculas.

- El nombre de las tablas debe separarse con el carácter “\_” si es un nombre compuesto.
- Los nombres de las tablas en singular.
- El nombre de la tabla debe explicar la información que contiene.
- Los nombres de las columnas deben contener el prefijo de tres caracteres que simplifiquen el nombre de la tabla.

Tabla: USUARIO

Columna: **USR\_PRIMER\_NOMBRE**

- Los nombres de las columnas en letras mayúsculas.
- El nombre de las columnas debe separarse con el carácter “\_” si es un nombre compuesto.
- Los nombres de las columnas deben contener el prefijo de tres caracteres que simplifiquen el nombre de la tabla y el carácter “\_”.

Tabla: USUARIO

Columna: **USR\_PRIMER\_NOMBRE**

- Utilización de método de capitalización Pascal para el nombre de los procedimientos almacenados

### **ObtenerPersonaNatural**

- Nombrar los procesos almacenados combinando el verbo con el objeto (verbo-objeto).

### **ObtenerItemCatalogo**

### 3.1.2 Invocación de Métodos entre Capas del Sistema

La invocación de métodos entre las capas del sistema Medical Center.Net varía dependiendo la capa en la que se trabaje. Las figuras 25, 26, 27, 28 y 29 muestran un ejemplo de como se consumen los métodos de cada capa del sistema.

```

|<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceholder1" runat="Server">
|   <div class="container-admin-appointment">
|     <div id="title-header">
|       <%= Resources.WebsiteLocalizations.AdminMedicalAppointment%>
|     </div>
|     <div class="errorMessage">
|       <asp:Label ID="lblMessage" runat="server" CssClass="errorMessage"
|         meta:resourcekey="lblMessageResource1"><asp:Label>
|     </div>
|     <div class="container-controls-admin-appointment">
|       <fieldset class="containerGloboPatientSearch">
|         <legend>
|           <%= Resources.WebsiteLocalizations.GeneralData%>
|         </legend>
|         <div class="containerGloboPatientSearch">
|           <asp:LinkButton ID="LinkButton1" runat="server" Text="Add"
|            PostBackUrl="~/BackOffice/Doctor/AddUpdateMedicalAppointment.aspx">LinkButton</asp:LinkButton>
|           <div class="containerInputGeneralLeftPatientSearch">
|             <div class="containerInputGeneralPatientSearch">
|               <div class="containerLabel">
|                 <asp:Label ID="lblStartDate" runat="server"
|                   meta:resourcekey="lblStartDateResource1"></asp:Label>
|               </div>
|               <div class="containerInputControl">...</div>
|             </div>
|             <div class="containerInputGeneralPatientSearch">...</div>
|           </div>
|           <div class="containerInputGeneralRightPatientSearch">...</div>
|           <div style="width: 10%; height: 25px; float: left;">
|           </div>
|           <div>...</div>
|         </div>
|       </fieldset>
|     <div>
|       <asp:GridView ID="grvAppointmentResults" ...</asp:GridView>
|     </div>
|   </div>
| </div>
| </asp:Content>

```

**Figura 25** Capa de Presentación – Código de Front End

```

/// <summary>
/// called when Add button is clicked this is called when a mouse is clicked on open space of any day
/// </summary>
/// <param name="improperEvent"></param>
/// <returns></returns>
[System.Web.Services.WebMethod]
public static int addEvent(ImproperCalendarEvent improperEvent)
{
    try
    {
        WebMappingManager mapper = new WebMappingManager();
        int result = -2;
        CITA_MEDICA medicalAppointment = mapper.MapMedicalAppointment(improperEvent);
        medicalAppointment.MDC_ID =
        result = Proxy.ProxyMedicalAppointmentManager.CreateMedicalAppointment(medicalAppointment);

        List<int> idList = (List<int>)System.Web.HttpContext.Current.Session["idList"];

        if (idList != null)
        {
            idList.Add(result);
        }

        return result;
    }
    catch (Exception ex)
    {
        Util.ManageException(ex, Constant.EXCEPTION_GENERAL_POLICY);
        return -2;
    }
}
}

```

**Figura 26** Capa Presentación – Invocación de Proxy

```

/// <summary>
/// Proxy de manejo de citas médicas
/// </summary>
public static IMedicalAppointment ProxyMedicalAppointmentManager
{
    get
    {
        IMedicalAppointment proxy = (IMedicalAppointment)new MedicalAppointment();
        return proxy;
    }
}
}

```

**Figura 27** Capa de Proxy – Declaración de Proxy

```

public interface IMedicalAppointment
{
    /// <summary>
    /// Actualiza una cita médica
    /// </summary>
    /// <param name="appointment">Cita médica</param>
    /// <returns>resultado de creación</returns>
    int UpdateMedicalAppointment(CITA_MEDICA appointment);

    /// <summary>
    /// Crea una cita médica
    /// </summary>
    /// <param name="appointment">Cita médica</param>
    /// <returns>resultado de creación</returns>
    int CreateMedicalAppointment(CITA_MEDICA appointment);
}

```

**Figura 28** Capa Interfaz – Declaración de Interfaz

```

    /// <summary>
    /// Crea una cita médica
    /// </summary>
    /// <param name="appointment">Cita médica</param>
    /// <returns>resultado de creación</returns>
    public int CreateMedicalAppointment(CITA_MEDICA appointment)
    {
        try
        {
            ObjectParameter insertResult = new ObjectParameter("o_resultado", typeof(Int32));

            context.InsertMedicalAppointment(insertResult, appointment.CIT_TITULO, appointment.CIT_CODIGO,
                appointment.CIT_TIPO, appointment.CIT_FECHA_INICIO, appointment.CIT_FECHA_FIN,
                appointment.CIT_DESCRIPCION, appointment.CIT_OBSERVACIONES_PACIENTE, appointment.CIT_OBSERVACIONES_MEDI
                appointment.CIT_ESTADO, appointment.MDC_ID, appointment.PCT_ID);
            if (Convert.ToInt32(insertResult.Value.ToString()) > 0)
                return Convert.ToInt32(insertResult.Value);

            return -2;
        }
        catch (Exception ex)
        {
            Util.ManageException(ex, Constant.EXCEPTION_GENERAL_POLICY);
            return -2;
        }
    }
}

```

**Figura 29** Capa de Negocio –Invocación de Eventos de Entity Framework

La Figura 30 muestra las entidades de Entity Framework que han sido mapeadas desde la base de datos a la capa de acceso a datos del sistema y han sido convertidas en entidades de negocio.

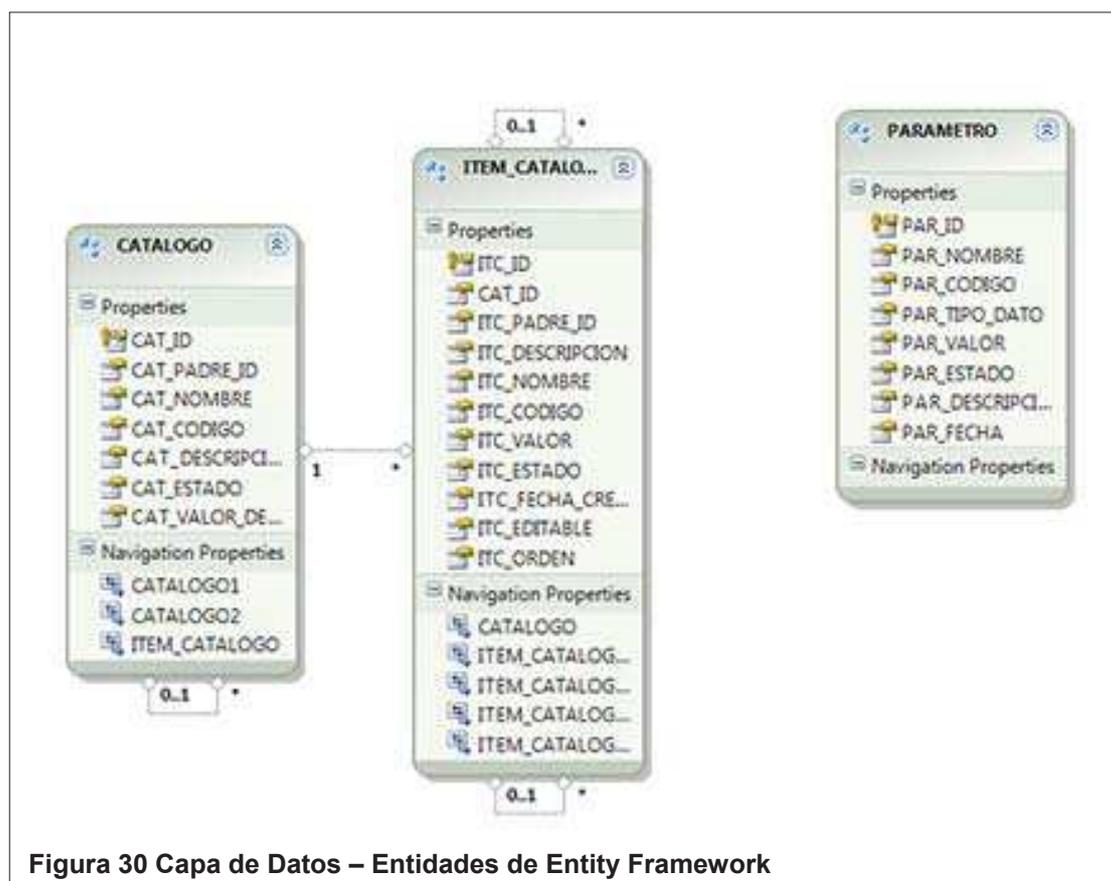


Figura 30 Capa de Datos – Entidades de Entity Framework

### 3.1.3 Diseño de Pantallas del Sistema

Las pantallas del sistema se diseñaron en base a las especificaciones dadas en la sección de interfaces. Fueron diseñadas con la herramienta seleccionada para el desarrollo del sistema Visual Studio 2010. A continuación se muestran las principales pantallas del sistema con sus respectivas especificaciones.

#### 3.1.3.1 Pantalla de Ingreso al Sistema

La pantalla de ingreso al sistema tiene la función de permitir al usuario validar sus credenciales para poder acceder al sistema o direccionar a la pantalla de Registro de Pacientes en el Sistema.

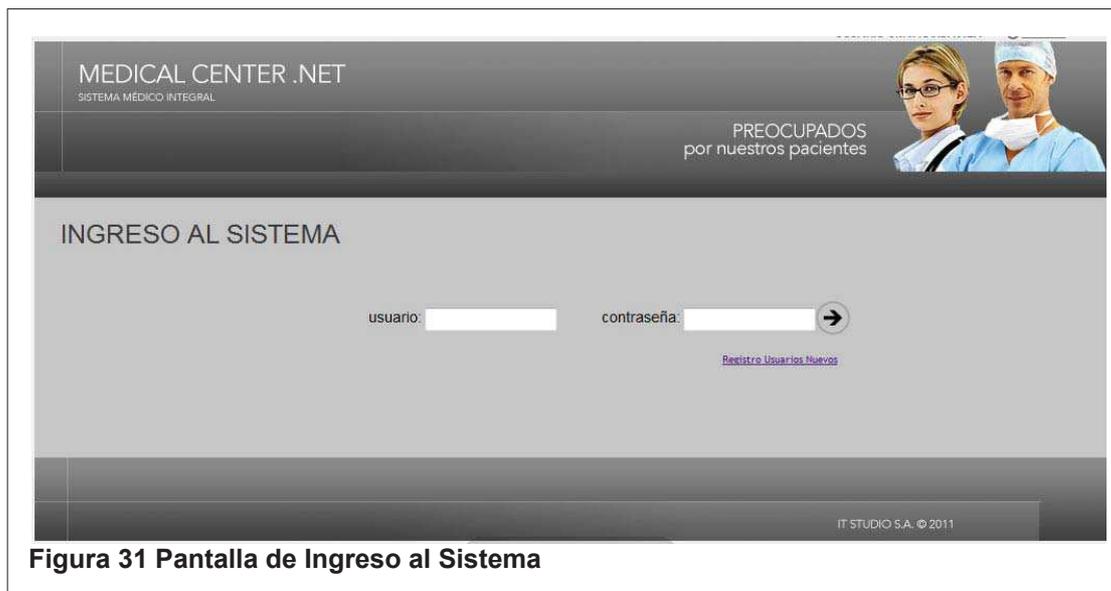


Figura 31 Pantalla de Ingreso al Sistema

### 3.1.3.2 Pantalla Principal

La pantalla principal tiene la función de mostrar al usuario las opciones del menú disponibles, esta interfaz se muestra únicamente cuando la validación de credenciales en la pantalla de Ingreso al Sistema fue satisfactoria.



Figura 32 Pantalla Principal

### 3.1.3.3 Pantalla de Administración de Médico

La Pantalla de Administración de Médico tiene la función de ingresar y actualizar los datos generales de un médico usuario del sistema.

Esta pantalla esta diseñada para usuarios de tipo Médico y Administrador.

MEDICAL CENTER .NET  
SISTEMA MÉDICO INTEGRAL

USUARIO UNIVACIXDAVILA LOGOFF

PREOCUPADOS por nuestros pacientes

Inicio Administración Médico Paciente Seguridad Contactenos

## ADMINISTRACIÓN DE MÉDICO

**DATOS GENERALES**

Tipo de Identificación	Cédula	Identificación	0104704572
Primer Nombre	Jose	Segundo Nombre	Ricardo
Primer Apellido	Naranjo	Segundo Apellido	Izquierdo
Nacionalidad	Ecuatoriana	Estado Civil	Soltero
Género	Masculino	Fecha de Nacimiento	3/8/1985
Teléfono Casa	2891234	Ocupación	Abogado
Teléfono Trabajo	2891102	Celular	087472754
Dirección Domicilio	23th street	Correo Electrónico	negro.xavier@gmail.com
		Dirección Trabajo	24th street

**DATOS DE MÉDICO**

Código	MED-0001	Área	Pediatria
Especialización	Cirujía	Otro	
Título	Médico		

IT STUDIO S.A. © 2011

Figura 33 Pantalla Administración de Médico

### 3.1.3.4 Pantalla de Administración de Consultorio

La pantalla de Administración de Consultorio tiene la función de permitir al usuario crear y actualizar los datos de un consultorio registrado en el sistema.

Esta Pantalla esta diseñada para usuarios de tipo Médico y Administrador.

USUARIO UNIVACUAVILA LOGOFF

MEDICAL CENTER .NET  
SISTEMA MÉDICO INTEGRAL

PREOCUPADOS por nuestros pacientes

Inicio Administración Médico Paciente Seguridad Contáctenos

### ADMINISTRACIÓN DE CONSULTORIO

**DATOS DE CONSULTORIO**

Código de Oficina	<input type="text" value="PDT-001"/>	Fax	<input type="text" value="2899222 ext 211"/>
Nº de Oficina	<input type="text" value="001"/>	Grupo	<input type="text" value="Pediatria"/>
Teléfono	<input type="text" value="2890877"/>		
Email	<input type="text" value="pediatria@axis.com"/>		

IT STUDIO S.A. © 2011

**Figura 34 Pantalla de Administración de Consultorio**

### 3.1.3.5 Pantalla de Administración de Citas

La pantalla de Administración de Citas tiene la función de permitir al usuario médico de visualizar su calendario de citas, filtrar por mes, semana y día, abrir detalles e invocar el control para agregar, editar, cancelar e ingresar resultados de una cita médica.

The screenshot shows the 'ADMINISTRACIÓN DE CITAS MÉDICAS' (Medical Appointment Management) interface. At the top, there is a header for 'MEDICAL CENTER .NET SISTEMA MEDICO INTEGRAL' and a slogan 'PREOCUPADOS por nuestros pacientes' with an image of medical staff. A navigation menu includes 'Inicio', 'Administración', 'Médico', 'Paciente', 'Seguridad', and 'Contáctenos'. The main content area displays a calendar for July 2012. The calendar has tabs for 'MES', 'SEMANA', and 'DIA', and a 'HOY' button. The calendar grid shows days from 1 to 31. Two appointments are visible: '6p CITA MEDICA TEMP' on Friday, July 13th, and '6p Evento Exp' on Saturday, July 14th. The interface is clean and professional, using a grey and white color scheme.

**Figura 34 Pantalla de Administración de Citas Médicas**

### 3.1.3.6 Pantalla de Creación y Actualización de Citas

La pantalla de creación y actualización de citas se muestra como un popup al momento de dar clic sobre una cita medica en el administrador de citas o sobre un día del calendario, dependiendo de donde se dio el clic se muestra la pantalla para creación o actualización de datos de una cita médica.

**MEDICAL CENTER .NET**  
SISTEMA MÉDICO INTEGRAL

PREOCUPADOS por nuestros pacientes

Inicio Administración Médico Paciente Seguridad Contáctenos

**Evento Médico**

Código:

Fecha Inicio:

Título:

Título:

Fecha Fin:

Paciente:

Estado:

Descripción:

Observaciones Médico:

Observaciones Paciente:

29	30	31	1	2	3	4
5	6	7	8	9	10	11

**Figura 35 Pantalla Creación y Actualización de Citas**

### 3.2 PRUEBAS

El propósito del plan de pruebas, es obtener información para la planificación, control, mantenimiento y aseguramiento del sistema desarrollado.

La metodología Microsoft Solution Framework (MSF) recomienda dentro de la fase de pruebas, la ejecución de pruebas unitarias y funcionales.

Las pruebas unitarias del Sistema se realizaron de acuerdo a la arquitectura de la aplicación utilizando NUnit como herramienta recomendada en lenguajes de programación de Microsoft.Net.

Las pruebas unitarias se realizaron de la capa de Proxy hacia atrás, es decir se incluyen todas las capas de la arquitectura excepto los formularios de

presentación e interacción con el usuario, las cuales se verificarán con las pruebas de aceptación de usuario.

El plan general de pruebas que se detalla en las tablas 23, 24, 25 y 26 incluye:

- Prueba de conexión y consistencia de datos
- Prueba de funcionamiento
- Prueba de interfaces de usuario
- Prueba de rendimiento
- Prueba de seguridad y control de acceso
- Prueba de instalación

Tabla 23. Prueba de Conexión y Consistencia de Datos

Prueba de Conexión y Consistencia de Datos	
<b>Objetivo</b>	<p>Verificar:</p> <ul style="list-style-type: none"> <li>• Acceso a la base de datos de cada uno de los métodos sea correcta.</li> <li>• Todas las conexiones a la base de datos se cierren después de cumplir su ciclo.</li> <li>• Datos ingresados en la base de datos estén consistentes por los enviados desde la capa de presentación.</li> </ul>
<b>Técnica</b>	<ul style="list-style-type: none"> <li>• La verificación de acceso a la base de datos se realiza ejecutando pruebas unitarias de NUnit.</li> <li>• La verificación de cierre de conexiones de la Base de Datos se realiza ejecutando pruebas unitarias de NUnit sobre la capa de Datos y se monitorea las conexiones abiertas y cerradas del motor de Base de Datos SQL Server.</li> <li>• La verificación de la consistencia de datos se realiza ejecutando las pruebas unitarias de NUnit con datos correctos e incorrectos simulando desconexión y timeouts para verificar que los procedimientos de reverso, control de errores y validación de datos controlen que los datos almacenados en la Base de Datos sea consistente.</li> </ul> <p>La técnica de esta prueba se realiza para todos los métodos de consulta, inserción, actualización y eliminación.</p>
<b>Criterios de Éxito</b>	<ul style="list-style-type: none"> <li>• Todos los métodos con acceso a la base de datos acceden correctamente a la misma.</li> <li>• El sistema no deja conexiones de base de datos abiertas.</li> <li>• No existen datos inconsistentes, los procesos de rollback funcionan correctamente en caso de un error.</li> </ul>

Tabla 24. Prueba de Funcionamiento

Prueba de Funcionamiento	
<b>Objetivo</b>	Verificar que el sistema cumpla con los requerimientos del cliente.
<b>Técnica</b>	<p>Ejecutar casos de prueba funcional para cada pantalla del sistema verificando que cada una cumpla con la especificación.</p> <p>Con la ejecución de los casos de prueba funcional se validará:</p> <ul style="list-style-type: none"> <li>• Los resultados con la interacción del sistema sean los esperados utilizando datos válidos y no válidos.</li> <li>• Se muestren los mensajes informativos y de error apropiados.</li> <li>• Validaciones de datos cuando la información proporcionada no es válida.</li> </ul>
<b>Criterios de Éxito</b>	La ejecución de los casos de prueba produzcan los resultados esperados.

Tabla 25. Prueba de Interfaces de Usuario

Prueba de Interfaces de Usuario	
<b>Objetivo</b>	<p>Verificar:</p> <ul style="list-style-type: none"> <li>• El cumplimiento del estándar de diseño en todas las interfaces de usuario.</li> <li>• Navegabilidad del Sistema sea amigable con el usuario.</li> <li>• Controles interactivos (botones, enlaces, cajas de texto, combos, etc.) funcionen y presenten la información adecuada.</li> </ul>
<b>Técnica</b>	<p>Verificar que todas las interfaces de usuario estén utilizando la misma hoja de estilo estándar de todo el sistema.</p> <p>Ejecutar los casos de prueba funcional para cada interfaz de usuario para comprobar que los controles interactivos están funcionando correctamente a la interacción con el usuario.</p>
<b>Criterios de Éxito</b>	Todos los controles interactivos y diseño de interfaces cumplan con el estándar establecido.

Tabla 26. Prueba de Rendimiento

Prueba de Rendimiento	
<b>Objetivo</b>	Medir el tiempo de respuesta del sistema y evaluar si el mismo es aceptable.
<b>Técnica</b>	Ejecutar las pruebas de simulación de Visual Studio 2010 para obtener los reportes de posibles cuellos de botella y optimización de código que puedan ocasionar un bajo rendimiento de la aplicación. Ejecutar los casos de prueba funcional dentro de una intranet e internet para obtener reportes simulando un ambiente de producción.
<b>Criterios de Éxito</b>	Tiempo de respuesta promedio aceptable en todas las funcionalidades del sistema.

Tabla 27. Prueba de Seguridad y Control de Acceso

Prueba de Seguridad y Control de Acceso	
<b>Objetivo</b>	Certificar que el acceso de los usuarios al sistema concuerde con el perfil de usuario asociado e identificar las posibles vulnerabilidades del sistema.
<b>Técnica</b>	Certificar que todos los formularios tengan un control de sesión de usuario. Verificar que la conexión a la base de datos sea segura. Verificar que la utilización de los perfiles de usuario se aplique en toda la aplicación. Verificar que el único acceso al sistema sea a través de la página principal. Ejecutar casos de prueba unitaria para comprobar que el sistema no esté vulnerable a SQL Injection.
<b>Criterios de Éxito</b>	No se debe acceder a ninguna pantalla del sistema si previamente no se ha realizado una validación de usuario y contraseña. No se debe mostrar funcionalidades adicionales ni menos funcionalidades a las asociadas al perfil de usuario. El sistema no debe ejecutar sentencias de SQL Injection para vulnerar la información del sistema.

Tabla 28. Prueba de Seguridad y Control de Acceso

Prueba de Instalación	
<b>Objetivo</b>	Comprobar que el procedimiento de instalación de cómo resultado el funcionamiento correcto de la aplicación.
<b>Técnica</b>	Comprobar que los ambientes donde van a ser instalados tengan el software base requerido para el correcto funcionamiento de la aplicación. Instalar el sistema utilizando el manual de instalación siguiendo el paso a paso. Realizar la prueba de instalación en un ambiente que tuvo una instalación previa existente del sistema (reinstalación), y una prueba de instalación en un ambiente limpio (sin ninguna instalación).
<b>Criterios de Éxito</b>	Instalación y reinstalación realizada con el manual de usuario tiene resultados satisfactorios.

### 3.2.1 Pruebas Unitarias

La Metodología Microsoft Solution Framework para desarrollo Ágil recomienda ejecutar pruebas unitarias.

Para el desarrollo de estas pruebas Unitarias se automatizarán las pruebas unitarias utilizando NUnit.

NUnit es una herramienta de pruebas unitarias para todos los lenguajes de la plataforma.Net de Microsoft; esta herramienta está desarrollada en su totalidad en lenguaje C# y por tal razón se aprovechan todos los beneficios del Framework de Microsoft.

Para la explicación de como se llevaron a cabo las pruebas unitarias del sistema se utilizarán las funcionalidades principales los módulos de Seguridad y Negocio.

La ejecución de las pruebas Unitarias se realizó desde la subcapa de Proxy perteneciente a la capa de Presentación.

La Figura 36 muestra como ejemplo, las pruebas unitarias del módulo de seguridad. Los casos de prueba verificados en este ejemplo fueron:

- Validar credenciales de usuario.
- Cambiar contraseña de usuario.

Se ejecutaron dos tipos de prueba, un caso de prueba con datos de usuario y contraseña válidos y un caso de prueba con datos incorrectos (un usuario no existente en el sistema).

```
[TestFixture]
public class Medical_Center_Security
{
    /// <summary>
    /// Unit-Test Login
    /// </summary>
    [Test]
    public void TestCanUserLogin()
    {
        USUARIO user = Proxy.ProxyUserManager.GetUser("xdavila", "xdavila");
        if (user != null)
        {
            Assert.Greater(user.USR_ID, 0, "User Logged In Correct");
        }
        else
        {
            Assert.Fail("User Not logged In");
        }
    }

    |
    /// <summary>
    /// Unit-Test Change Password
    /// </summary>
    [Test]
    public void TestCanChangePassword()
    {
        bool resultUpdate = Proxy.ProxyUserManager.ChangePassword(1, "password");
        if (resultUpdate)
        {
            Assert.IsTrue(resultUpdate, "Password is changed");
        }
        else
        {
            Assert.Fail("Password not changed");
        }
    }
}
```

Figura 36 Pruebas Unitarias del Módulo de Seguridad

La Figura 37 muestra los resultados de las pruebas de seguridad que se realizaron para el caso de prueba con datos válidos y para el caso de prueba con datos no válidos.

- **Resultado de validación de credenciales de usuario:** el usuario y la contraseña ingresados fueron válidos y el sistema retornó el objeto de sesión que permite acceder al sistema a un usuario.
- **Resultado de cambio de contraseña de usuario:** los datos de usuario, contraseña y contraseña nueva para realizar el cambio fueron válidos y el sistema retornó un mensaje satisfactorio de cambio de contraseña.



**Figura 37 Resultados Pruebas Unitarias del Módulo de Seguridad**

La Figura 38 muestra los resultados de las pruebas de seguridad que se realizaron con datos de usuarios no existentes y datos de contraseña errados.

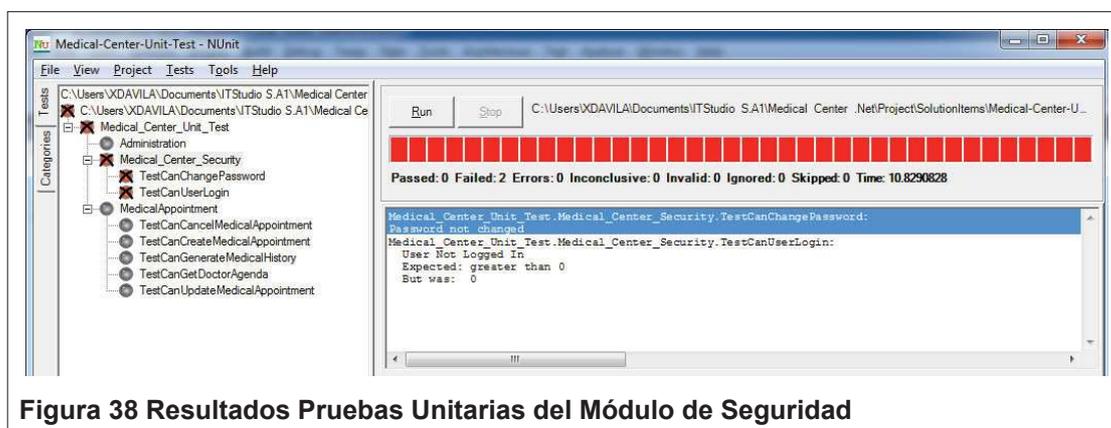
- **Resultado de validación de credenciales de usuario:** Se probaron los siguientes casos:
  - a) El usuario ingresado es válido y la contraseña no corresponde al usuario.
  - b) El usuario ingresado no existe en el sistema.

El sistema retornó un error de credenciales inválidas.

- **Resultado de cambio de contraseña de usuario:** Se probaron los siguientes casos:

- a) La contraseña nueva es igual a la contraseña actual.
- b) La contraseña no cumple con el mínimo requerimiento de caracteres.

El sistema retornó un error de validación de contraseña.



**Figura 38 Resultados Pruebas Unitarias del Módulo de Seguridad**

La Figura 39 y 40 muestran como ejemplo, las pruebas unitarias de las funcionalidades más importantes del módulo de negocio. Los casos de prueba verificados en este ejemplo fueron:

- Crear cita médica.
- Actualizar cita médica.
- Cancelar cita médica
- Obtener agenda de médico
- Generar Ficha Médica

Se ejecutaron dos tipos de prueba, un caso de prueba con datos para la gestión de citas y ficha médica válidos y caso de prueba con datos no válidos.

```

[TestFixture]
public class MedicalAppointment
{
    Medical_Center_Common commonManager = new Medical_Center_Common();
    /// <summary>
    /// Unit-Test Create Medical Appointment
    /// </summary>
    [Test]
    public void TestCanCreateMedicalAppointment()
    {
        CITA_MEDICA medicalAppointment = commonManager.GetMedicalAppointment();
        int result = ITSTUDIO.PX_Business.Proxy.ProxyMedicalAppointmentManager.CreateMedicalAppointment(medicalAppointment);
        if (result > 0)
        {
            Assert.Greater(result, 0, "Medical Appointment was created");
        }
        else
        {
            Assert.Fail("Medical Appointment was not created");
        }
    }

    /// <summary>
    /// Unit-Test Update Medical Appointment
    /// </summary>
    [Test]
    public void TestCanUpdateMedicalAppointment()
    {
        CITA_MEDICA medicalAppointment = commonManager.GetMedicalAppointment();
        int result = ITSTUDIO.PX_Business.Proxy.ProxyMedicalAppointmentManager.UpdateMedicalAppointment(medicalAppointment);
        if (result > 0)
        {
            Assert.Greater(result, 0, "Medical Appointment was updated");
        }
        else
        {
            Assert.Fail("Medical Appointment was not updated");
        }
    }

    /// <summary>
    /// Unit-Test Cancel Medical Appointment
    /// </summary>
    [Test]
    public void TestCanCancelMedicalAppointment()
    {
        CITA_MEDICA medicalAppointment = commonManager.GetMedicalAppointment();
        int result = ITSTUDIO.PX_Business.Proxy.ProxyMedicalAppointmentManager.CancelMedicalMedicalAppointment(medicalAppointment);
        if (result > 0)
        {
            Assert.Greater(result, 0, "Medical Appointment was canceled");
        }
        else
        {
            Assert.Fail("Medical Appointment was not canceled");
        }
    }
}

```

**Figura 39 Pruebas Unitarias de Citas Médicas**

```

/// <summary>
/// Unit-Test Get Doctor Agenda
/// </summary>
[Test]
public void TestCanGetDoctorAgenda()
{
    int doctorIdTest = 2;
    List<CITA_MEDICA> doctorAgenda = ITSTUDIO.PX_Business.Proxy.ProxyMedicalAppointmentManager.GetMedicalCenterAppointments(doctorIdTest);
    if (doctorAgenda != null)
    {
        Assert.NotNull(doctorAgenda, "Doctor Agenda Returned Correct");
    }
    else
    {
        Assert.Fail("Medical Appointment was not updated");
    }
}

/// <summary>
/// Unit-Test Generate Medical History
/// </summary>
[Test]
public void TestCanGenerateMedicalHistory()
{
    int doctorIdTest = 2;
    int patientIdTest = 4;
    MedicalHistory medicalHistory = new MedicalHistory();

    List<CITA_MEDICA> doctorAgenda = ITSTUDIO.PX_Business.Proxy.ProxyMedicalAppointmentManager.GetMedicalCenterAppointments(doctorIdTest);
    medicalHistory = commonManager.GenerateMedicalHistory(doctorIdTest, patientIdTest);
    if (medicalHistory != null)
    {
        Assert.NotNull(medicalHistory, "Medical History Generated Correct");
    }
    else
    {
        Assert.Fail("Medical History was not generated");
    }
}

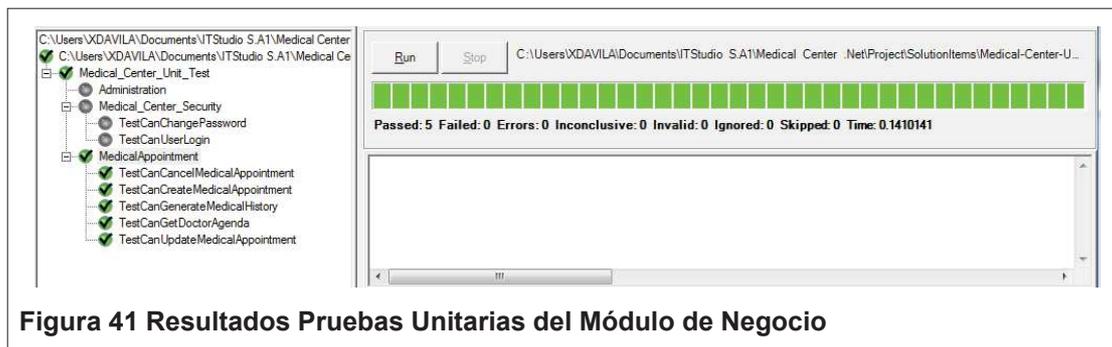
```

**Figura 40 Pruebas Unitarias de Agenda y Ficha Médica**

La Figura 41 muestra los resultados de las pruebas de negocio que se realizaron para el caso de prueba con datos válidos y para el caso de prueba con datos no válidos.

- **Resultado de creación de cita médica:** los datos de cita médica, paciente y médico ingresados fueron válidos y el sistema retornó el resultado de la creación de la cita médica.
- **Resultado de actualización de cita médica:** los datos de actualización de la cita médica creada previamente fueron válidos y el sistema retorno el resultado de la actualización de la cita médica.
- **Resultado de cancelación de cita médica:** la solicitud de cancelación de la cita médica creada previamente fue correcta y el sistema retornó el resultado de la cancelación de la cita médica.

- **Resultado de obtención de agenda médica:** los datos del médico para obtener la agenda médica fueron válidos y el sistema retornó la agenda del médico.
- **Resultado de generación de ficha médica:** los datos del paciente para obtener la ficha médica fueron válidos y el sistema retornó la ficha médica del paciente.

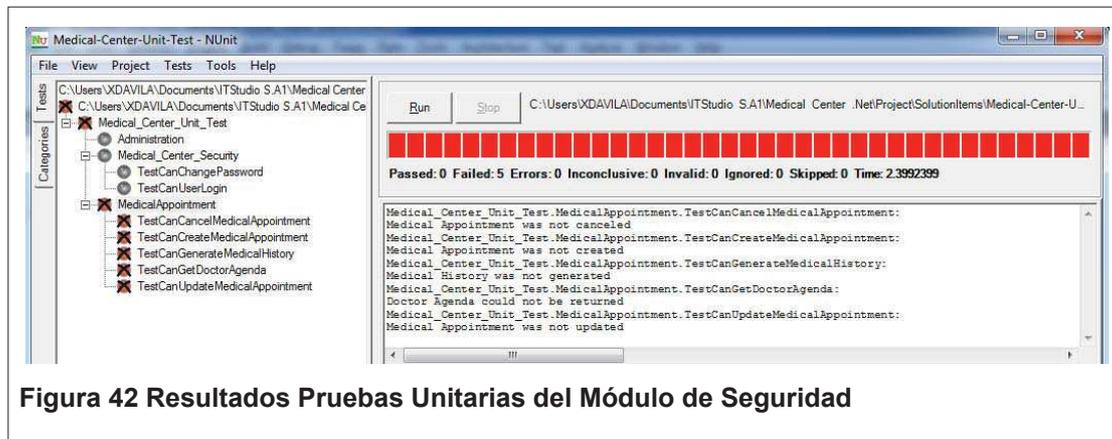


**Figura 41 Resultados Pruebas Unitarias del Módulo de Negocio**

La Figura 42 muestra los resultados de las pruebas de negocio que se realizaron con datos no válidos y datos que generen respuestas de error.

- **Resultado de creación de cita médica:** Se probó la creación de una cita médica con una hora fuera del horario del médico. El sistema retornó un error de cita médica fuera del horario de médico.
- **Resultado de actualización de cita médica:** Se probó la actualización de una cita médica con el horario de la cita igual a la de otra cita existente. El sistema retornó un error de otra cita existente.
- **Resultado de cancelación de cita médica:** Se probó la cancelación de una cita médica que no existe. El sistema retornó un error de cita no existente.
- **Resultado de obtención de agenda médica:** Se probó la obtención de una agenda médica de un médico que no existe. El sistema retornó un error de médico no existente.

- **Resultado de generación de ficha médica:** Se probó la generación de una ficha médica de un paciente que no existe. El sistema retornó un error de paciente no existente y ficha no existente.



**Figura 42 Resultados Pruebas Unitarias del Módulo de Seguridad**

### 3.2.2 Pruebas Funcionales

Las pruebas funcionales del sistema Medical Center.Net se realizaron con un equipo de analistas funcionales que validaron que las funcionalidades del sistema se ajusten a los requerimientos levantados con los usuarios de los Centros Médicos Integrales.

Los analistas funcionales con ayuda de los casos de uso, validaron la funcionalidad del sistema.

En la tabla 29 se describen las características revisadas por los analistas funcionales.

Tabla 29. Pruebas Funcionales

Pruebas Funcionales	
<b>Distribución de pantallas</b>	Cada formulario debe ajustarse al prototipo en el cual se define la ubicación de banner, títulos, menús y botones.
<b>Imágenes del sistema</b>	Cada formulario debe mostrar las imágenes del sistema acorde a la funcionalidad en la que se encuentra. Las imágenes de botones deben ser coherentes a la acción que los mismos ejecutan.
<b>Mensajes informativos y de error</b>	Cada formulario debe mostrar los mensajes coherentes a la acción que se está ejecutando. Cuando se presenta un error los mensajes deben presentarse de forma amigable y comprensible para el usuario.
<b>Carga de Pantallas</b>	Cada formulario debe cargarse correctamente, información de catálogos en pantallas requeridas debe cargarse junto con el formulario.
<b>Validación de Datos</b>	Cada formulario cuenta con controles de validación en sus campos. Los campos mandatorios deben presentarse con un asterisco y los mensajes deben mostrarse amigablemente al usuario cuando ocurre un error de validación.
<b>Coherencia de Datos</b>	La información ingresada en las pantallas de ingreso, actualización y eliminación debe manipular correctamente la información. Información ingresada actualizada y eliminada es validada en los buscadores y consulta de información manipulada en las distintas funcionalidades del sistema.

Los resultados de incidencias presentados por los analistas funcionales se clasificaron en tres tipos:

- **Incidencia Nivel Bajo:** Son incidencias de forma que no detienen el flujo del sistema y no detienen el proceso de pruebas funcionales. Por ejemplo errores en imágenes, mensajes de notificación, faltas ortográficas y textos.

- **Incidencia Nivel Medio:** Son incidencias que han generado errores en el sistema sin embargo no detienen el proceso de pruebas funcionales. Por ejemplo se ingresan datos al sistema y los mismos se almacenan correctamente pero la página muestra un error.
- **Incidencia nivel Alto:** Son incidencias que han generado errores del sistema que detienen el proceso de pruebas funcionales y no permiten continuar con el flujo del sistema. Por ejemplo excepciones no controladas que derivan en pantallas de error y no se ejecutan las peticiones realizadas.

La tabla 30 muestra el resultado global por módulo de las pruebas funcionales ejecutadas. En esta tabla se pueden identificar los siguientes resultados:

- **Casos de Prueba Ejecutados:** Corresponden a la cantidad de casos de prueba que los analistas funcionales ejecutaron para la validación de cumplimiento de requerimientos basándose en las características generales definidas para las pruebas.
- **Incidencias Bajas Reportadas:** Corresponden a la cantidad de incidencias en categoría baja que se reportaron durante la ejecución de los casos de prueba definidos para cada módulo.
- **Incidencias Medias Reportadas:** Corresponden a la cantidad de incidencias en categoría media que se reportaron durante la ejecución de los casos de prueba definidos para cada módulo.
- **Incidencias Altas Reportadas:** Corresponden a la cantidad de incidencias en categoría alta que se reportaron durante la ejecución de los casos de prueba definidos para cada módulo.

- **Casos de Prueba que Cumple con Requerimientos:** Corresponden a la cantidad de casos de prueba ejecutados que cumplieron con la especificación del requerimiento.

Los resultados de la tabla 30 muestran que después de la ejecución de las pruebas funcionales, el porcentaje de incidencias medias y altas es bajo y que el cumplimiento de requerimientos es total, el sistema cumple con las necesidades del cliente.

Las incidencias reportadas se corrigieron posteriores a la entrega del informe de resultados de pruebas funcionales.

**Tabla 30. Resultados de Pruebas Funcionales**

<b>Resultados de Pruebas Funcionales</b>					
<b>Módulo Probado</b>	<b>Casos de Prueba Ejecutadas</b>	<b>Incidencias Bajas Reportadas</b>	<b>Incidencias Medias Reportadas</b>	<b>Incidencias Altas Reportadas</b>	<b>Casos de Prueba que Cumple con Requerimiento</b>
<b>Administración</b>	26	2	0	0	26
<b>Negocio</b>	72	8	3	2	72
<b>Seguridad</b>	15	1	1	0	15

## 4 CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES

### 4.1 CONCLUSIONES

A partir del proceso de investigación y desarrollo de este proyecto de desarrollo de software se concluye:

- Los Centros Médicos Integrales que fueron caso de estudio “*Centro de Especialidades Médicas Tumbaco*” y el “*Centro Médico Integral San Juan de Cumbayá*” para el desarrollo de este proyecto no tienen un proceso organizado y definido de atención y administración de pacientes y Personal Médico. Cada funcionario tiene una visión diferente de los procesos internos de un Centro Médico Integral. A través de consensos entre los funcionarios de los Centros Médicos Integrales se definieron los procesos principales que fueron automatizados dentro del sistema Medical Center.Net. Esta definición de procesos ayudaron a identificar a los actores del sistema la interacción que tienen dentro del mismo. Esto facilitó la elaboración de diagramas de clases y secuencia utilizados para la implementación del sistema Medical Center.Net.
- La automatización de los procesos administrativos y de atención al cliente en Centros Médicos Integrales erradicaron el problema de almacenamiento manual y movilización de fichas médicas desde el archivo a un consultorio, la información se encuentra centralizada dentro del sistema Medical Center.Net y puede ser utilizada de manera concurrente por varios usuarios dentro de un Centro Médico Integral.
- La metodología de desarrollo de software Microsoft Solution Framework MSF para aplicaciones ágiles se ajusta a la medida al desarrollo de sistemas de software de las características de este proyecto de tesis. Los diagramas de casos de uso, secuencia y actividades generados en la aplicación de la Microsoft Solution Framework en este proyecto facilitan el

entendimiento general del negocio y necesidades que el sistema debe satisfacer. En función de ello se facilitó el involucramiento y comunicación con el cliente en la evolución y cambios de requerimientos presentados durante la fase de desarrollo y la generación de casos de prueba para los analistas funcionales.

- Aplicar la metodología Microsoft Solution Framework MSF para aplicaciones ágiles con herramientas de análisis, diseño, implementación y pruebas de Microsoft (Visual Studio 2010, SQL Server 2005 y Visual Studio Team Foundation Server) han generado un resultado favorable en el cumplimiento de la especificación de requerimientos y lineamientos planteados. Estos lineamientos definidos en la fase de diseño son controlados con ayuda de Microsoft Team Foundation Server que garantiza que el desarrollo del sistema esté alineado con el diseño del sistema.
- La administración de recursos durante las fases de desarrollo de este proyecto se realizó a través la definición de roles, asignación de tareas y cronogramas, todas ellas alineadas con el cumplimiento de requerimientos y alcance del proyecto. La utilización de la plantilla de Microsoft Solution Framework (MSF) para aplicaciones ágiles dentro de Visual Studio Team Foundation Server ayudó a gestionar los recursos para el cumplimiento de los objetivos de este proyecto, de esta forma los recursos pueden interactuar con el programa para conocer sus tareas asignadas, reportar sus avances y controlar el cumplimiento de requerimientos y alcance del proyecto. La información que se provee al sistema ayudo al gerente del proyecto a tomar decisiones en cambios a las funcionalidades y asignación de tareas para optimizar los recursos utilizados en el desarrollo de este proyecto.
- Definir de estándares de programación y nomenclatura estándar previo al desarrollo del sistema, ayudó a los programadores a estar alineados con

una forma común de codificar y dejar el código fuente comprensible para cualquier equipo que sea responsable de dar mantenimiento al sistema Medical Center.Net.

- La automatización de casos de pruebas unitarias durante la fase de pruebas del sistema ayudó considerablemente a detectar defectos que no pudieron ser identificados en la fase de desarrollo reduciendo el tiempo de mantenimiento correctivo, permitiendo a los analistas funcionales poder ejecutar los ciclos de pruebas funcionales sin interrupciones y garantizar la calidad del producto final. Parte de la automatización de las pruebas unitarias incluye la historia y objetivos de cada funcionalidad que se está automatizando a través de la prueba. Estas pruebas quedan registradas y serán ayuda para futuros mantenimientos donde se requiera verificar el impacto de un cambio dentro del sistema.
- El sistema Medical Center.Net está dirigido a un segmento de mercado en el cual los usuarios tienen un conocimiento limitado de manejo de sistemas médicos. Proveer una buena experiencia de usuario con una navegabilidad sencilla y común es fundamental para que los usuarios puedan cumplir con el objetivo de uso de este sistema. A través de una interfaz gráfica genérica común en todos los formularios del sistema y la utilización de recursos multimedia que faciliten la interacción del usuario con el sistema, se logró que los usuarios se sientan familiarizados en cualquier funcionalidad del sistema.
- Las tendencias tecnológicas obligan a los sistemas a interactuar y funcionar con distintos tipos de dispositivos móviles. La arquitectura del sistema Medical Center.Net en tres capas (presentación, negocio y datos) está diseñada con esa visión. El bajo impacto en mantenimiento y alta seguridad por la independencia de sus capas, la escalabilidad en clientes y recursos de servidor y la flexibilidad de su capa de presentación a

interfaces de usuario para dispositivos móviles, formularios Windows, etc. facilitan la adaptabilidad del sistema a las tendencias tecnológicas.

- El desarrollo de este proyecto de tesis bajo la tutoría académica brindada en la materia de Metodología de Titulación, prepara al estudiante para aplicar la teoría aprendida durante la carrera de Ingeniería en Sistemas e Informática de manera correcta, logrando que esta experiencia sirva de base y experiencia para proyectos futuros.

## **4.2 RECOMENDACIONES**

A partir del proceso de investigación y desarrollo de este proyecto de desarrollo de software, se presenta las siguientes recomendaciones:

- Previo al inicio del desarrollo de un proyecto de software en un Centro Médico Integral, se debe recomendar la optimización de todos los procedimientos de negocio para su posterior automatización. Esto generará que la automatización de los procesos dentro de un Centro Médico Integral se realice de forma eficiente y en un menor plazo.
- Una vez finalizada la fase de análisis y diseño se recomienda formalizar por escrito los documentos de análisis de requerimientos y alcance del sistema para que el cliente y el dueño del producto estén de acuerdo con el alcance del sistema y se pueda llegar a un acuerdo mutuo y satisfacción del cliente al momento de la entrega del mismo.
- Se recomienda capacitar al personal del centro de cómputo de cada Centro Médico Integral o a las personas encargadas de administrar el sistema dentro de la institución para que puedan proveer un correcto mantenimiento y soporte del sistema.

- La tendencia actual de desarrollo de software, obliga a que los sistemas de software estén disponibles en dispositivos móviles, por esta razón se recomienda en futuras versiones añadir compatibilidad completa con dispositivos móviles más utilizados en el mercado.
- Dada la confidencialidad de las Fichas Médicas se recomienda incorporar todas las seguridades necesarias físicas donde va a estar instalado el sistema adicional a la implementación y configuración de todas las seguridades de acceso y control del sistema.
- Se recomienda crear un proceso de respaldo periódico de información, con el objetivo de prevenir la pérdida de información y garantizar la disponibilidad del sistema ante cualquier desastre que pueda presentarse durante el día a día.
- Se recomienda incluir a personal del Centro Médico dentro de todas las fases del desarrollo del sistema para poder efectuar una abstracción correcta de requerimientos que posteriormente satisfagan las necesidades del negocio y cumplan con los requerimientos planteados.
- Se debe considerar un tiempo considerable para pruebas unitarias y funcionales en las cuales se pueda incluir al cliente para la aceptación y recepción de recomendaciones que se puedan tener previo a la entrega del sistema.
- Se recomienda que la Universidad promueva el uso de un calendario en el cual se comprometa al alumno y al profesor a reunirse periódicamente durante el desarrollo de la tesis con el fin de incentivar la culminación del proyecto de tesis.

## REFERENCIAS

### Libros:

- BRITO, K., (2009). *Selección de metodologías de desarrollo para aplicaciones web en la facultad de Informática de la Universidad de Cienfuegos*. Cuba: Universidad Cienfuegos.
- CWALINA, K., ABRAMS, B., (2008). *Framework Design Guidelines (2nd ed.)*. USA: Addison-Wesley
- DAVIDSON, L., (2008). *Pro SQL Server 2008 Relational Database Design and Implementation*. USA: Apress.
- FIGUEROA, R., (2009). *Metodologías Tradicionales vs. Metodologías Ágiles*. Ecuador, (paper)
- HOTEK, M., (2009). *Exam 70-432: Microsoft SQL Server 2008 – Implementation and Maintenance*. USA: Microsoft Press.
- JONES, A., FREEMAN, A., (2010). *Visual C# 2010 Recipes*. USA: Apress.
- KLEIN, S., (2010). *Pro Entity Framework 4.0*. USA: Apress.
- LOWY, J, (2008), *C# Coding Standard*. USA: IDesign
- MILLETT, S., (2010). *Professional ASP.NET Design Patterns*. USA: Wiley Publishing Inc.
- NEWTON, K., (2007). *The Definitive Guide to the Microsoft Enterprise Library*. USA: Apress.
- TENNY, L., HIRANI, Z., (2010). *Entity Framework 4.0 Recipes – A Problem Solution Approach*. USA: Apress.
- WATSON, B., (2010). *C# 4.0 How-To*. USA: Person Education Inc.
- WOODY, L., (2010). *MSF for Agile Software Development v5.0 – Process Guidance*. USA: Microsoft Press.
- ZAIHNGROUP, (2010), *Microsoft Solution Framework*. Recuperado el 12 de febrero de 2012

**Documentos de Internet:**

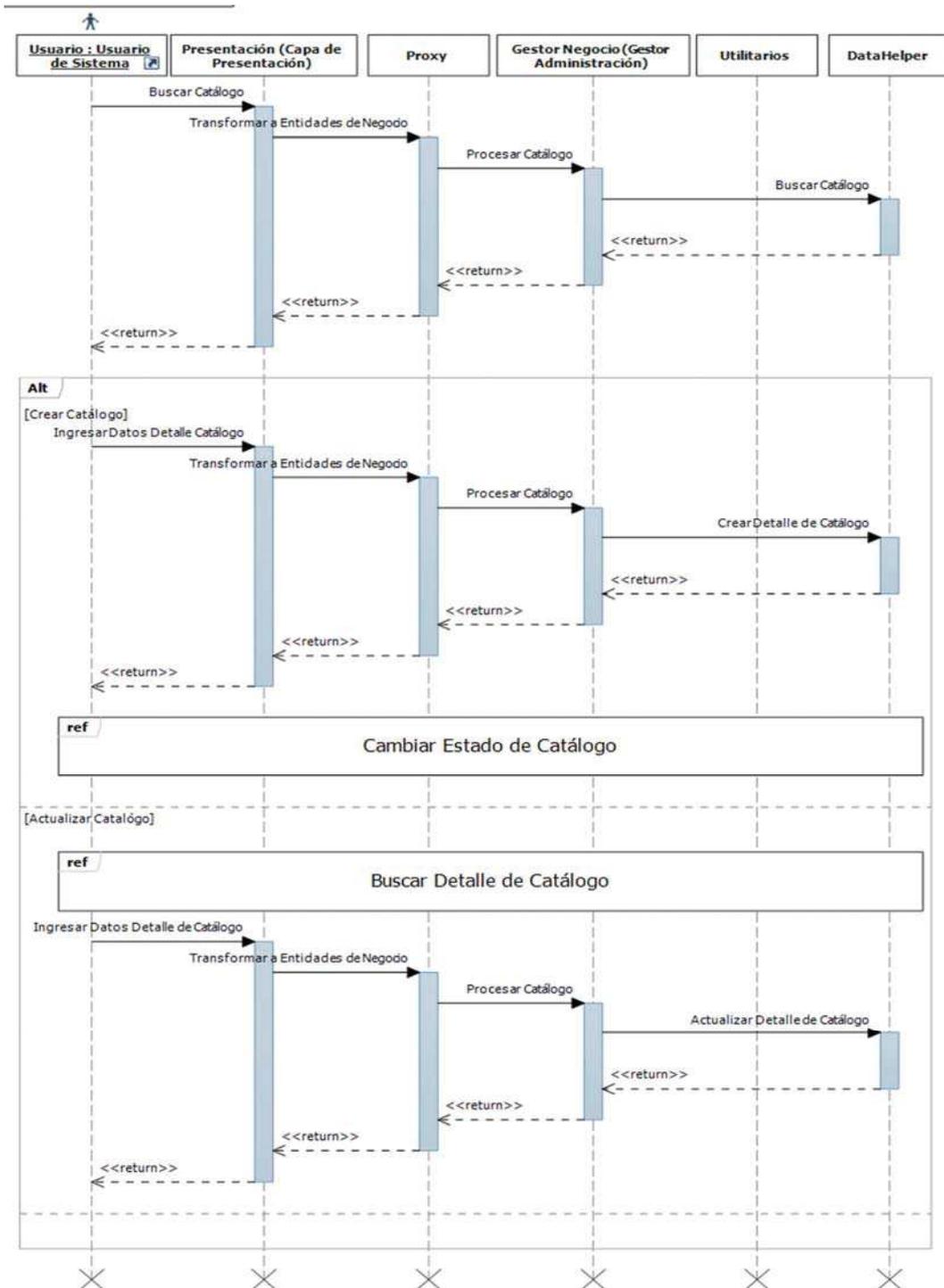
- ATELLO334, (2007), *Introducción a la Ingeniería de Software*. Recuperado de <http://atello334.blogspot.es/tags/MSF/>
- CHÁVEZ, A, (2007), *Microsoft Solution Framework (MSF)*. Recuperado el 15 de febrero de 2012, de <http://achavez334.blogspot.es>
- JEFFRIES, R, (2002). *What is Extreme Programming*. Recuperado de: <http://xprogramming.com>
- MSF for Agile Software Development v5.0 (2010). Recuperado de: <http://msdn.microsoft.com/en-us/library/dd380647.aspx>
- PIRES, D, (2010). *Manifiesto Ágil*, UCLA, Recuperado de: <http://www.manifiestoagile.com>
- SPINEC, (2010), *Metodologías Ágiles*. Recuperado de [http://www.spinec.org/wp-content/metodologiasagiles\\_01.pdf](http://www.spinec.org/wp-content/metodologiasagiles_01.pdf)
- STECNOLOGIA, (2011), *Proceso de Desenvolvimento Ágil Através da Metodologia SCRUM*. Recuperado de: <http://www.sstecnologia.com.br>
- WIKIPEDIA, (2012), *Programación por Capas*. Recuperado el 6 de enero de 2011, de [http://es.wikipedia.org/wiki/Programación\\_por\\_capas](http://es.wikipedia.org/wiki/Programación_por_capas)

# ANEXOS

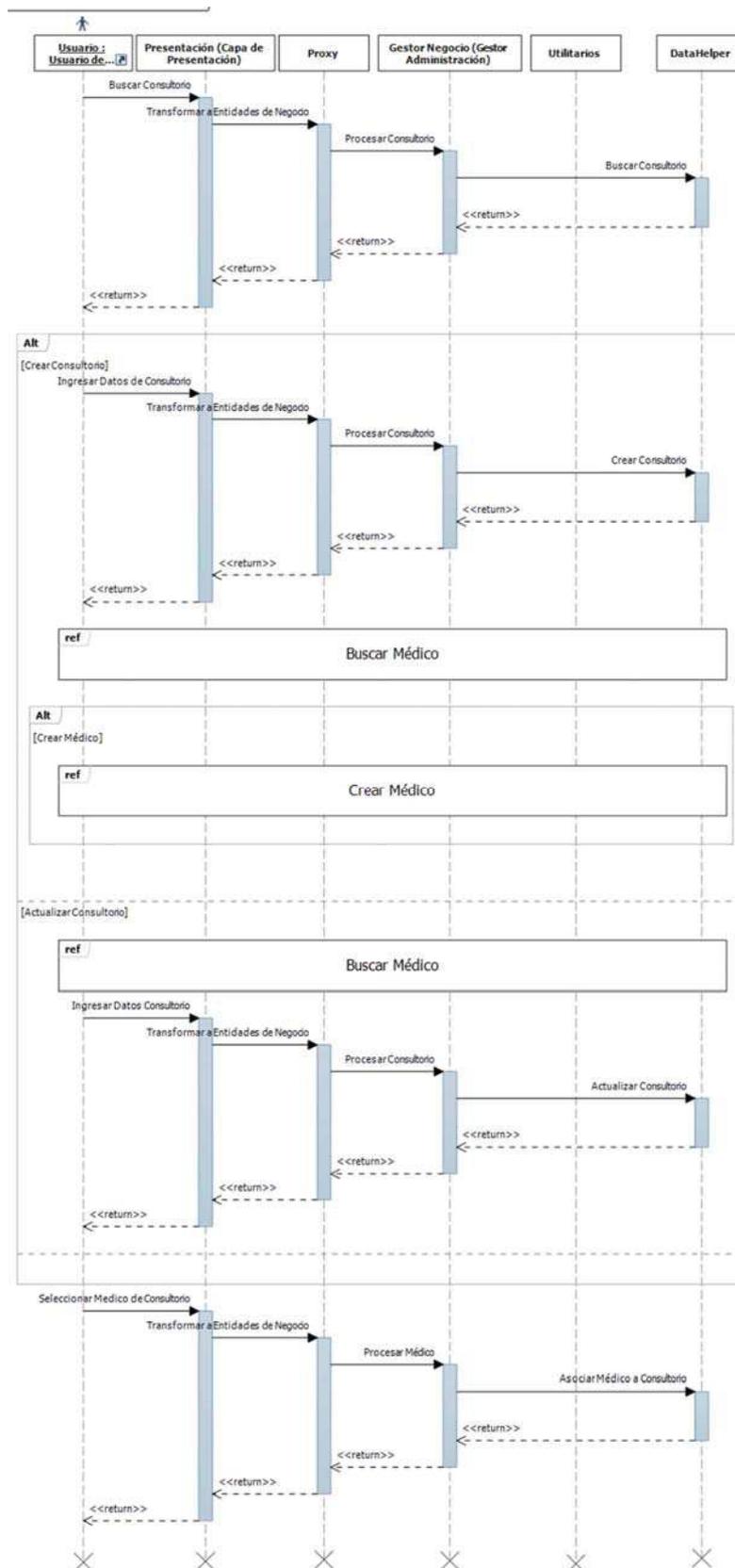
# ANEXO 1

## Diagrama de Secuencia

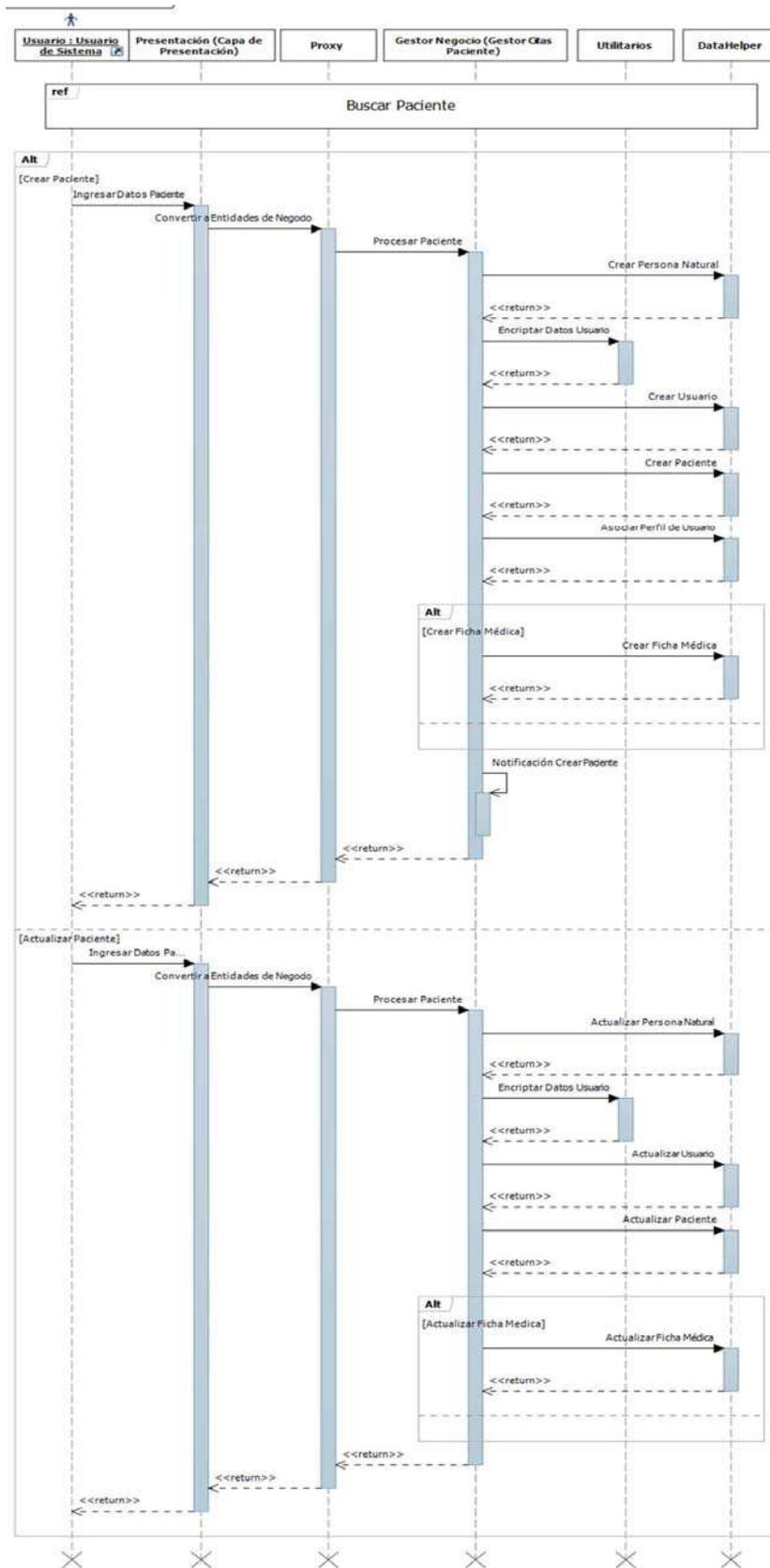
### 1. Administración de Catálogos



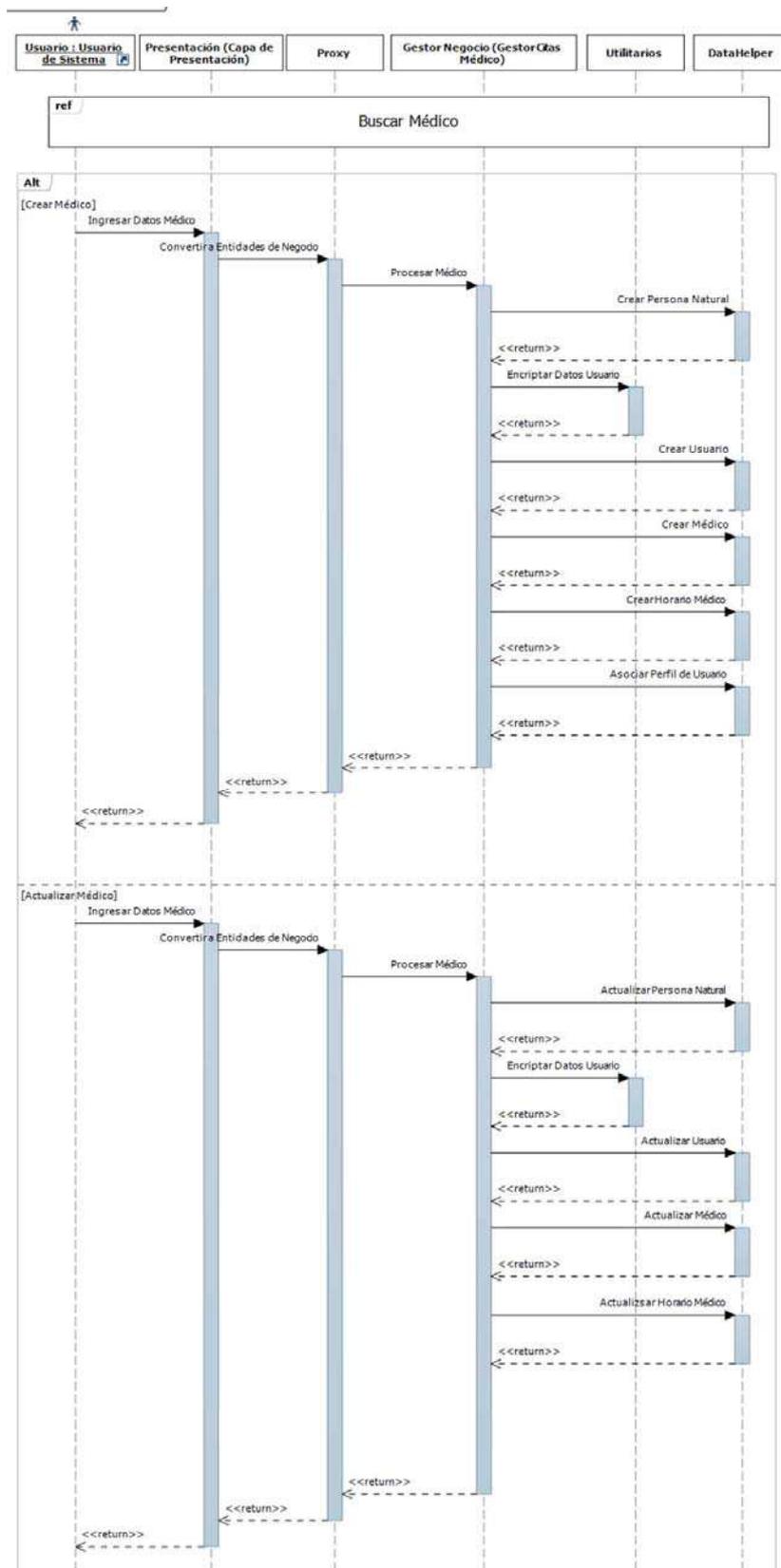
## 2. Administración de Catálogos



### 3. Administración de Pacientes



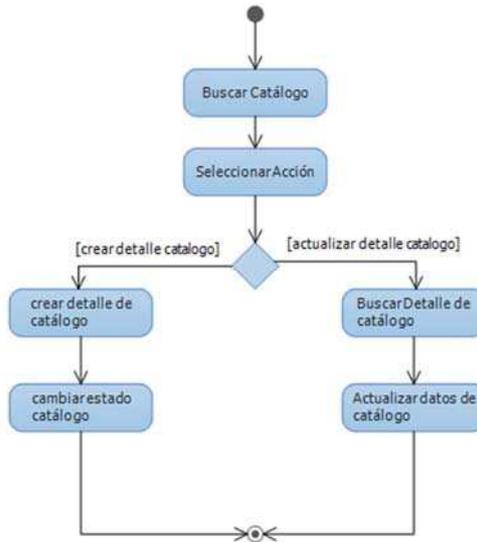
## 4. Administración de Médicos



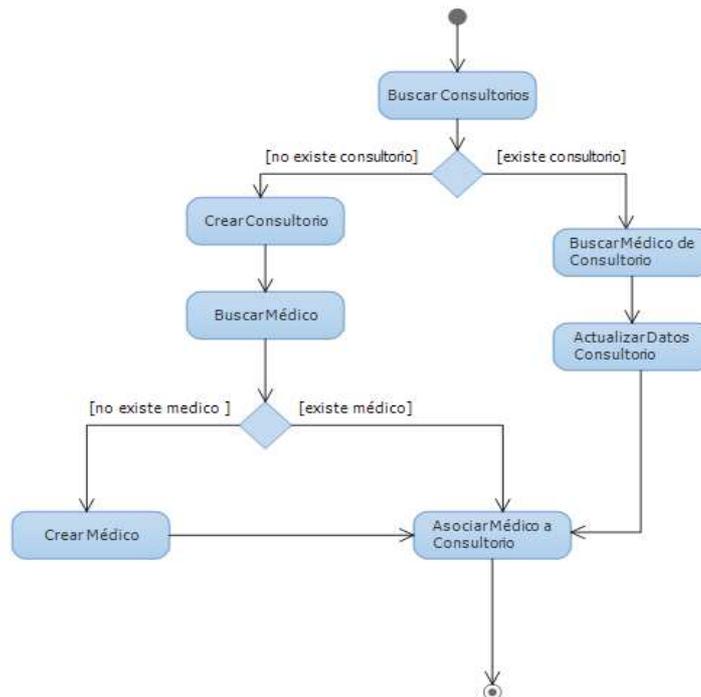
## ANEXO 2

### Diagramas de Actividades

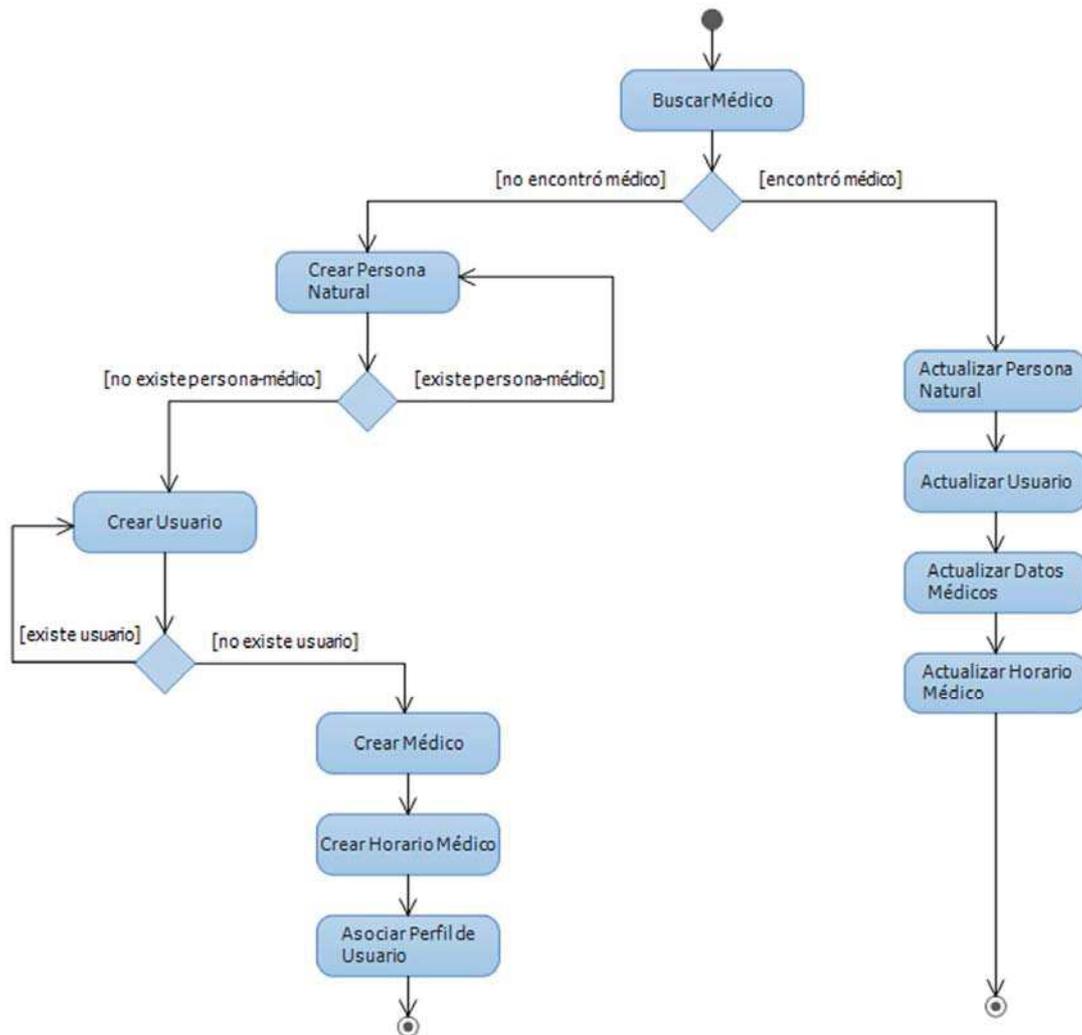
#### 1. Administración de Catálogos



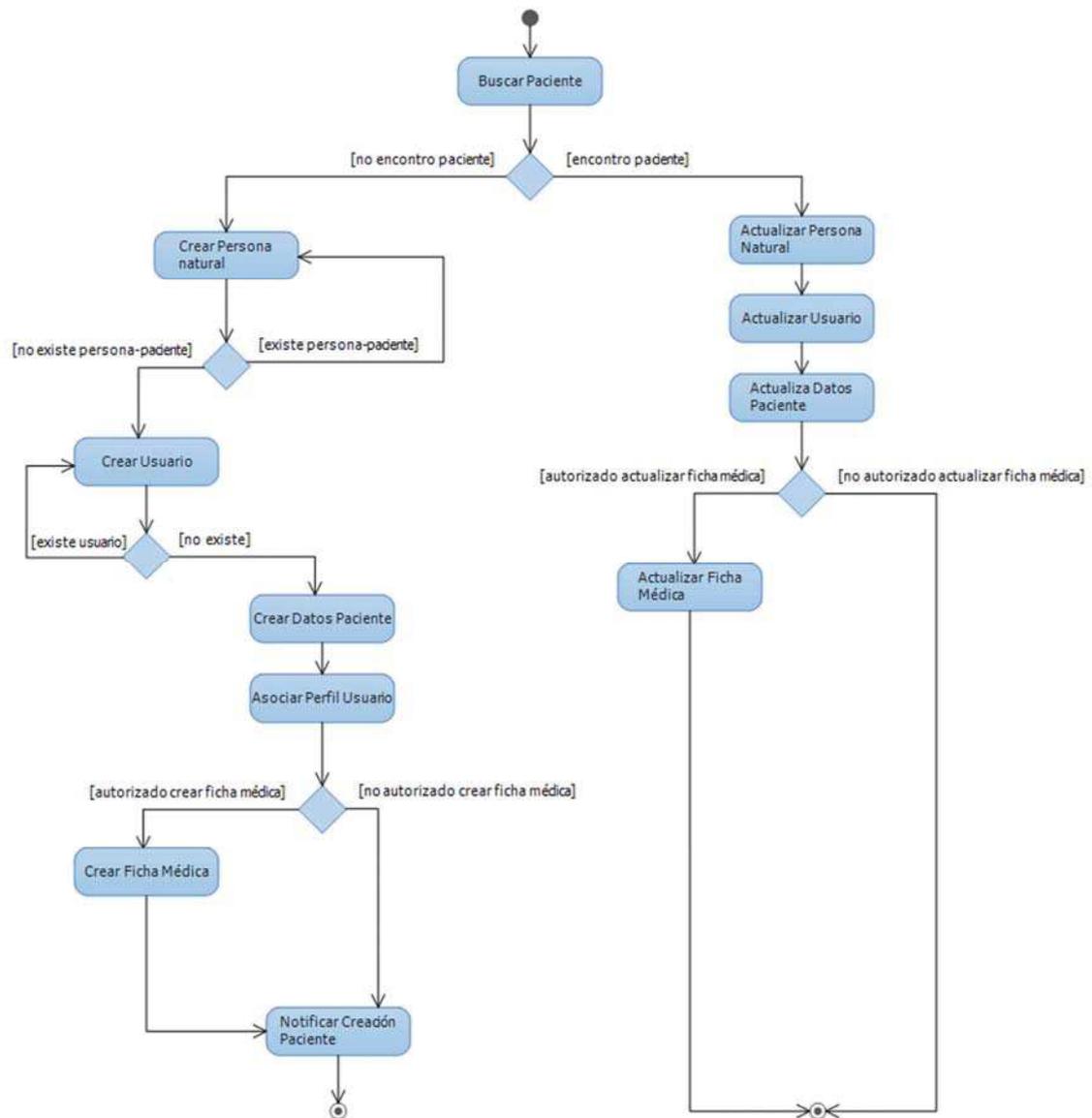
#### 2. Administración de Consultorios



### 3. Administración de Médico



## 4. Administración de Pacientes



## ANEXO 3

### Manual de Instalación Medical Center.Net

#### A. Introducción

Este manual provee las instrucciones que se deben seguir para la instalación del sistema Medical Center.Net en un servidor.

#### B. Prerequisitos

La instalación del sistema Medical Center.Net se debe realizar en un servidor que cumpla con los siguientes requerimientos de hardware:

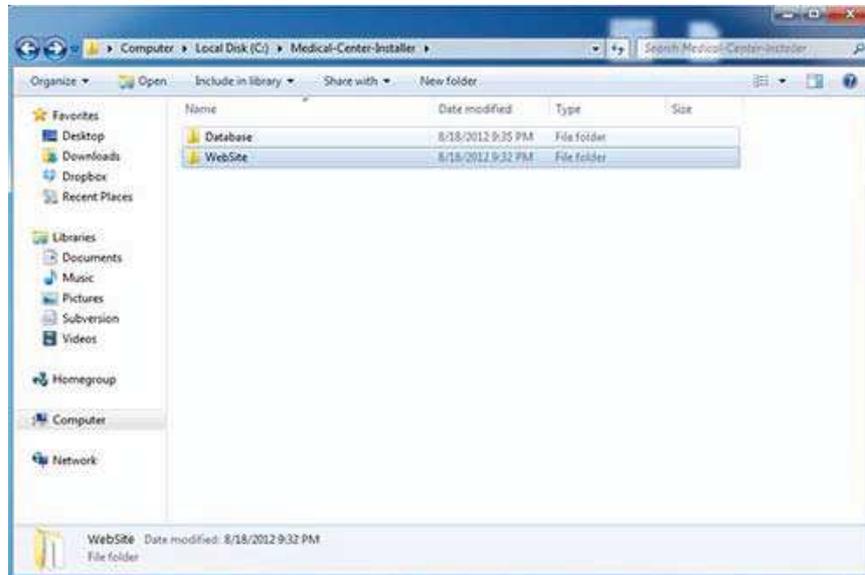
- Procesador Quad core 2,8 (o superior).
- Mínimo 2 Gbs de memoria RAM (recomendado 4 gbs).
- 10 gigas de espacio libre mínimo.

El servidor debe tener pre instalado los siguientes paquetes de software base:

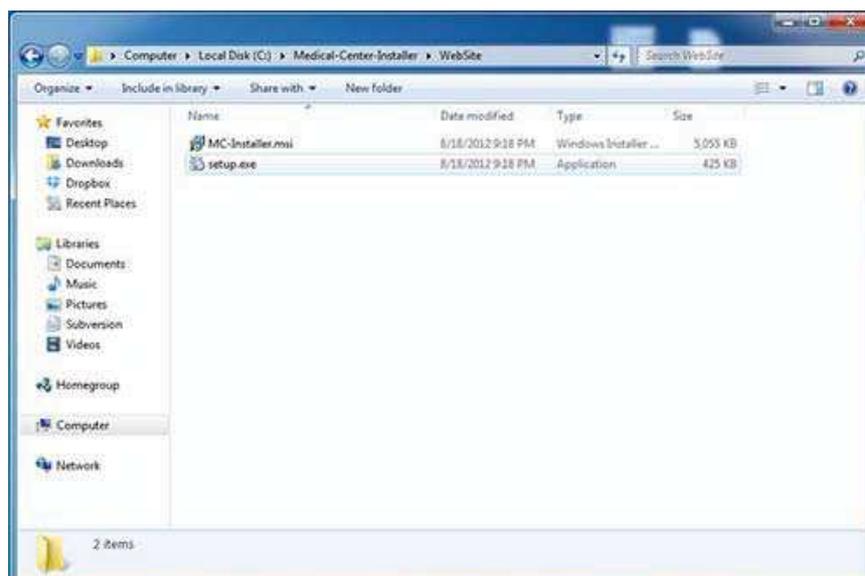
- Sistema Operativo Windows Server 2008 (o superior).
- Internet Information Server 7 (o superior).
- SQL Server 2005 Estándar (o superior).

## C. Instalación

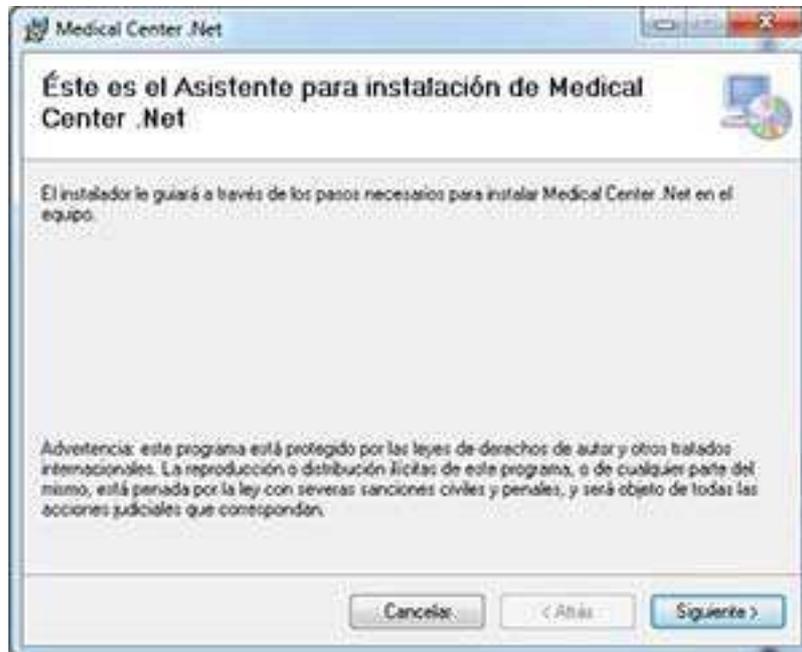
- i. Ubicar la carpeta del instalador.



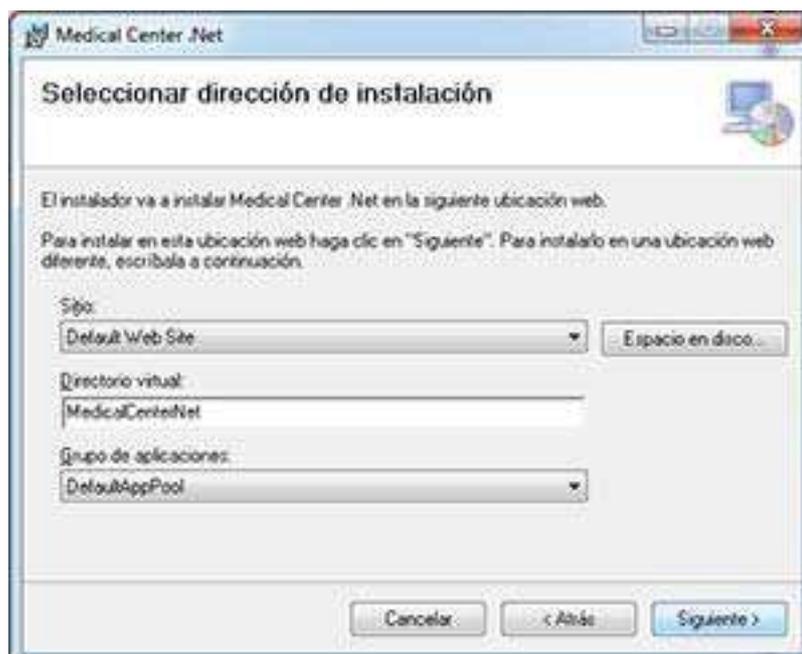
- ii. Ingresar a la carpeta **WebSite** y dar doble clic sobre el archivo **setup.exe**. Este archivo iniciará el proceso de instalación del sitio Web del sistema Medical Center.Net



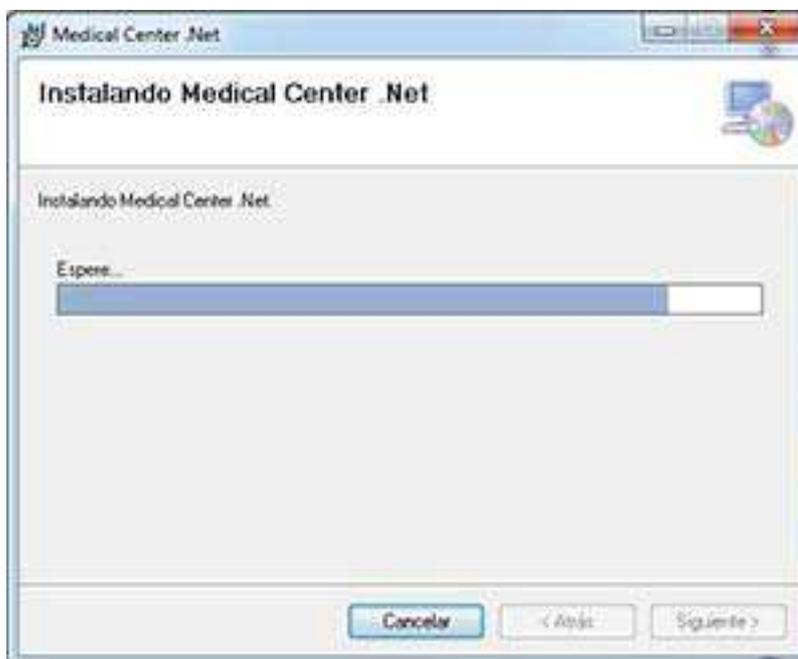
- iii. En la pantalla de bienvenida del instalador Medical Center.Net dar clic en **Siguiente**.



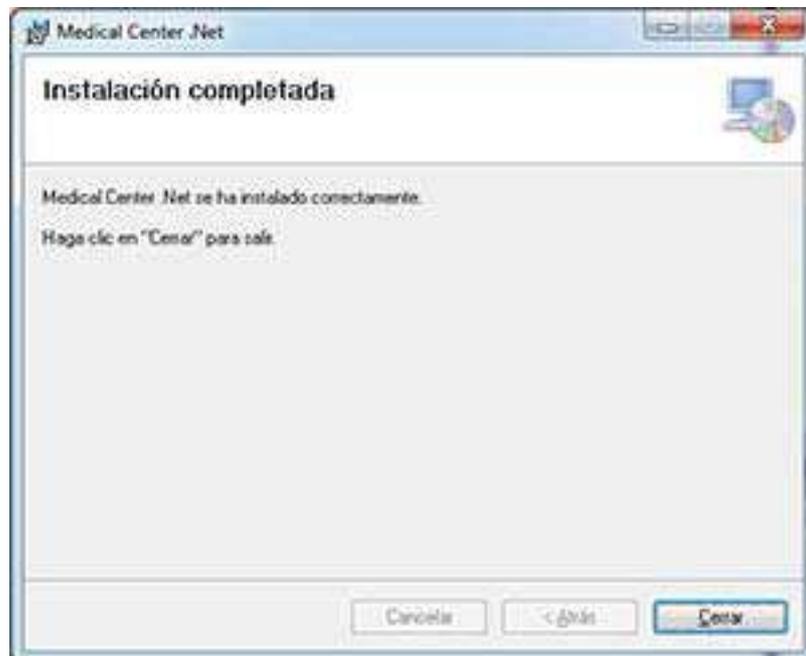
- iv. Seleccione la dirección, directorio virtual y grupo de aplicaciones que desea utilizar para el sistema. (Puede dejar la configuración por defecto si requiere). Dar clic en **Siguiente**.



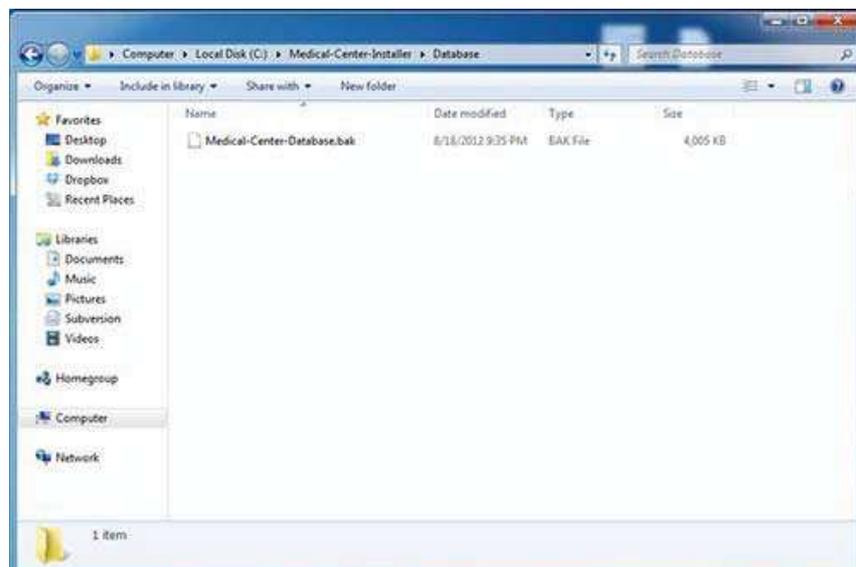
- v. Dar clic en siguiente en la pantalla de confirmación. El sistema empezará con el proceso de copia de archivos, creación del sitio web y configuración general.



vi. Finalizada la instalación dar clic en **Cerrar**.



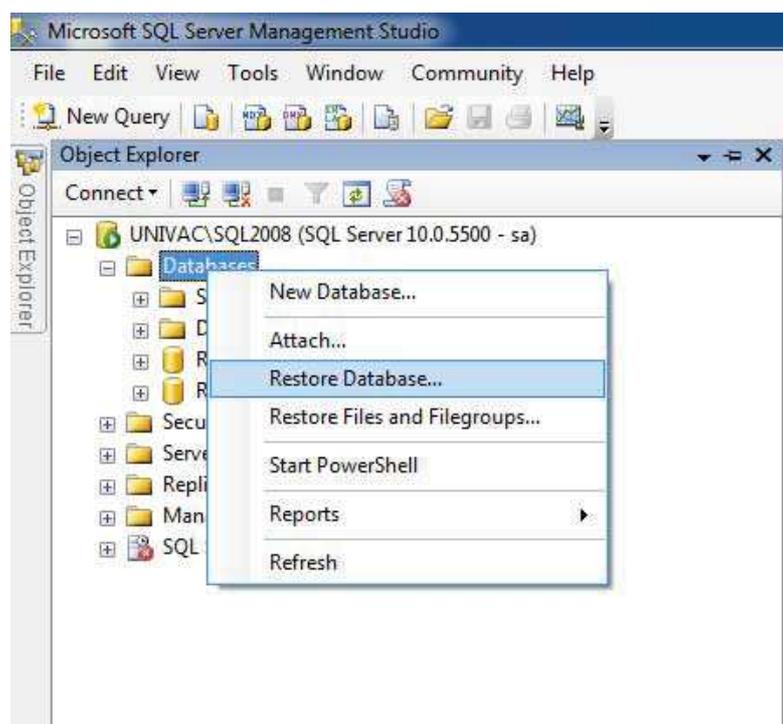
vii. Ingresar a la carpeta **Database** y ubicar el archivo **Medical-Center-Database.bak**.



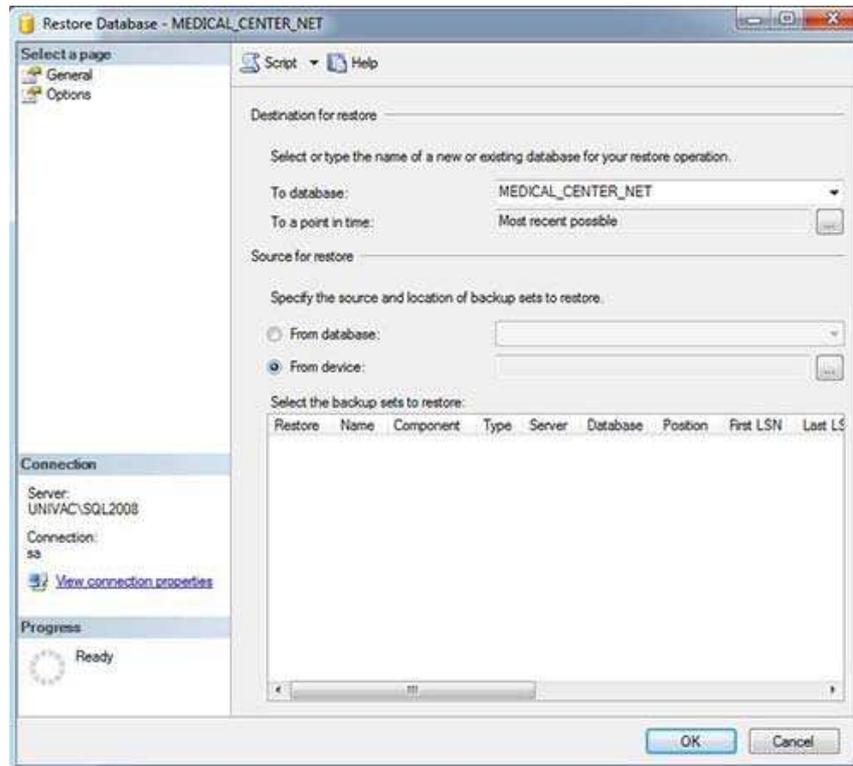
- viii. Abrir **SQL Server Management Studio** e ingresar al motor de base de datos con las credenciales de usuario y contraseña.



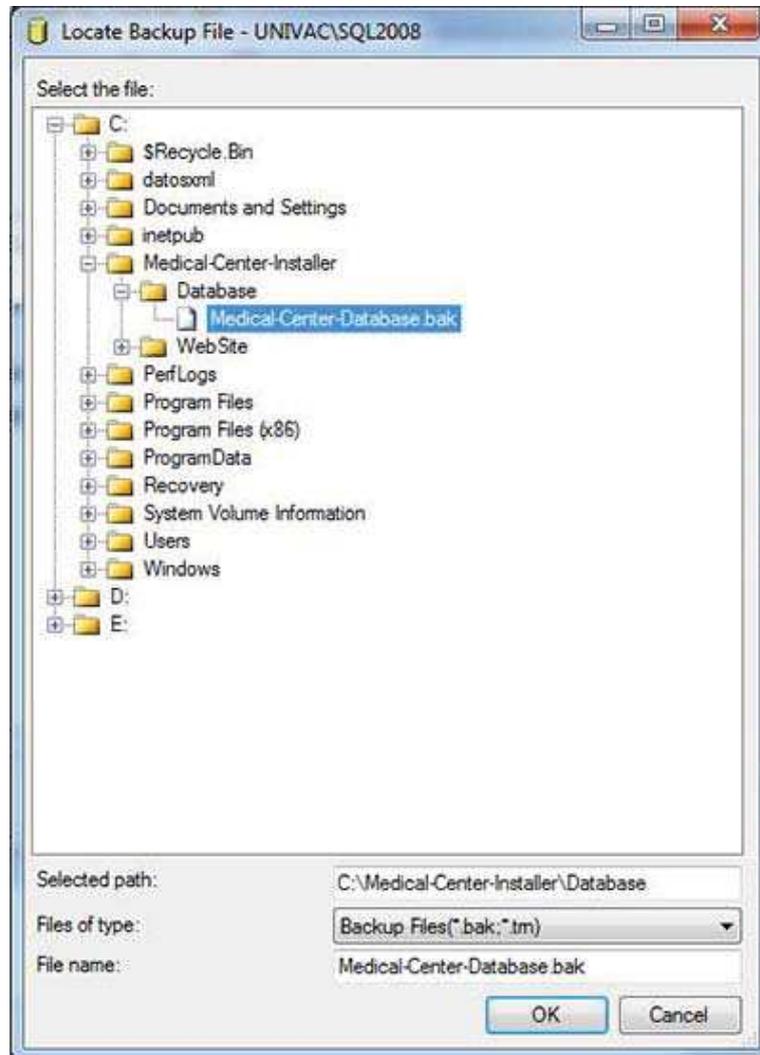
- ix. Damos clic derecho sobre la carpeta **Databases (Base de Datos)** y seleccionamos la opción **Restore Database (Restaurar Base de Datos)**.



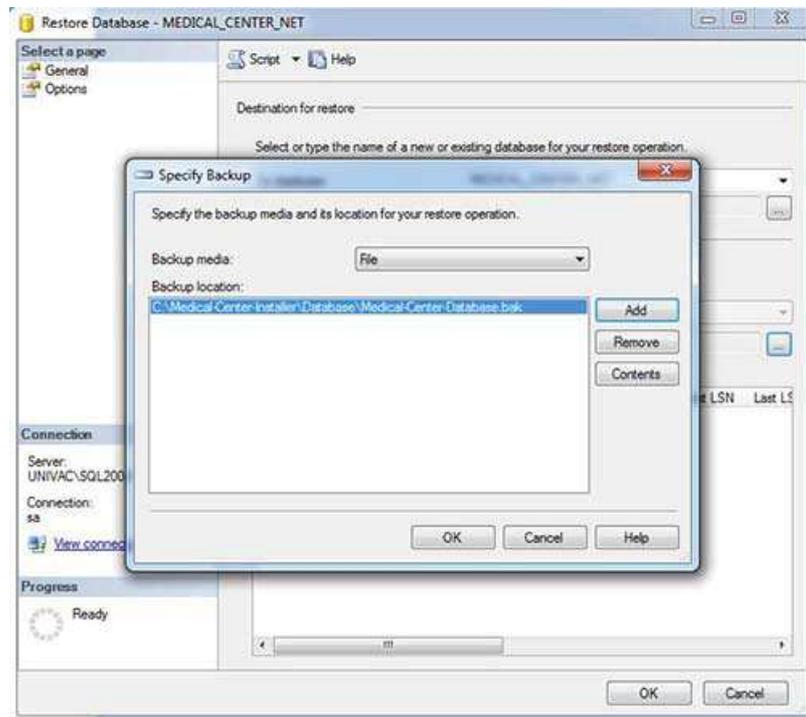
- x. En la opción **To Database (a la base de datos)** se ingresa el nombre **MEDICAL\_CENTER\_NET** y en la opción **Source of restore** seleccionamos la opción **From Device** y damos clic en el botón para seleccionar el dispositivo o respaldo.



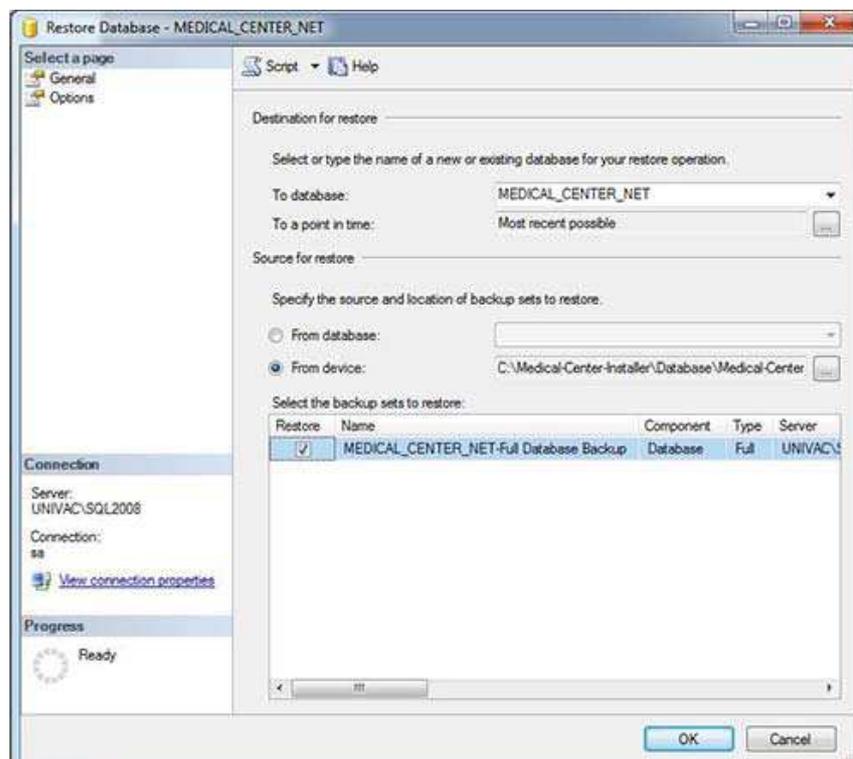
- xi. Ubicar el archivo de respaldo en la carpeta del instalador y damos click en **Ok**.



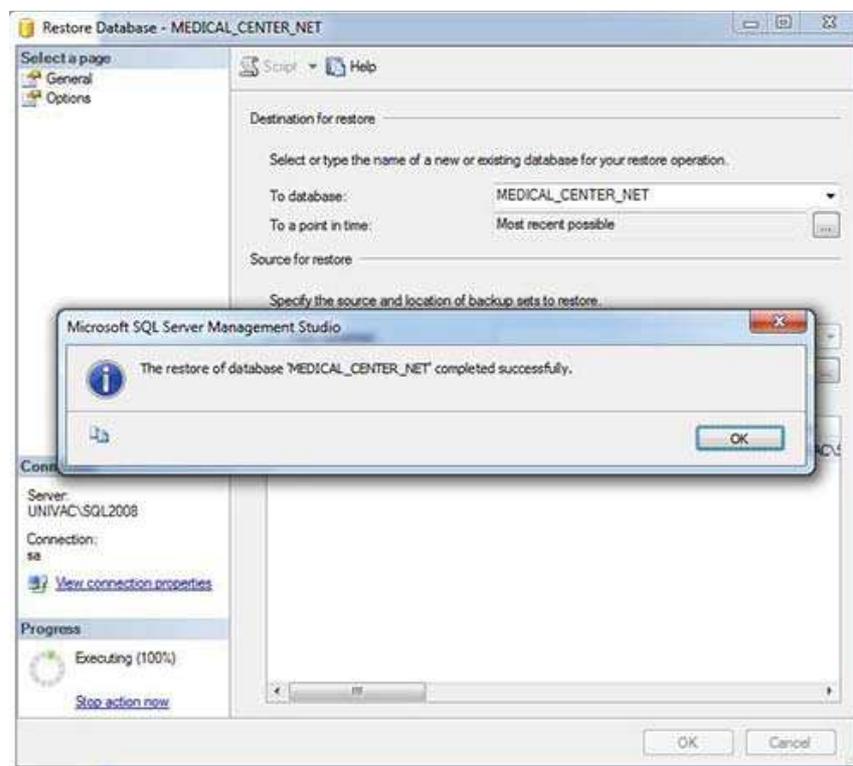
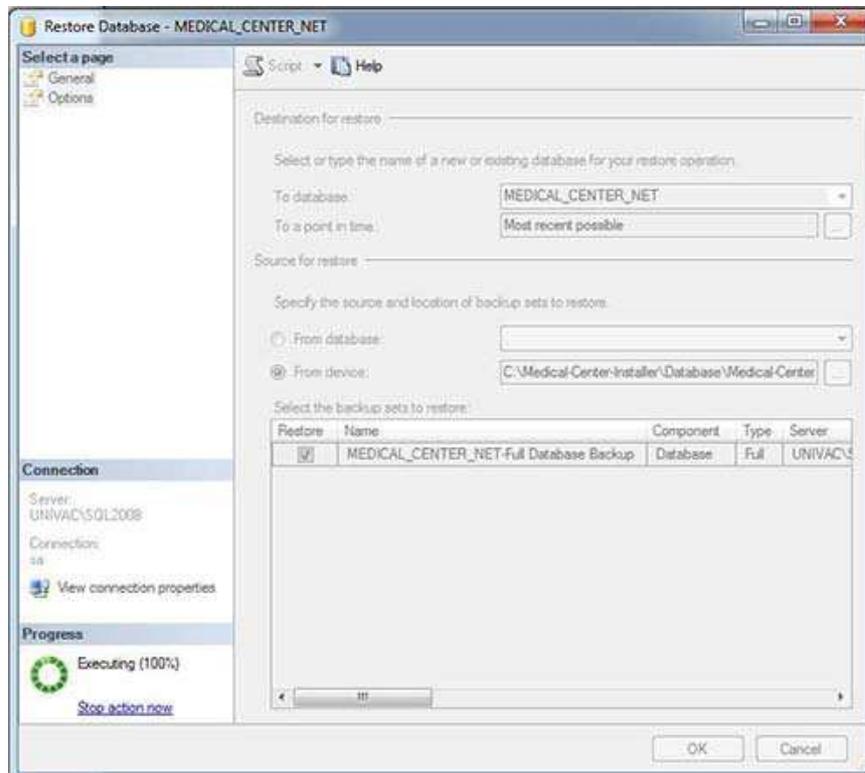
xii. Dar click en **Ok** en la pantalla de especificación de respaldo



xiii. Seleccionar el respaldo seleccionado previamente y dar clic en **Ok**.



- xiv. El respaldo de la base de datos se cargará en el motor SQL Server, una vez cargado el respaldo dar clic en **Ok**.



xv. Ubicar la carpeta donde se encuentra publicado el sistema:  
“C:\inetpub\wwwroot\[nombreSitioMedicalCenter]” y abrir el archivo  
**web.config**

xvi. Modificar el archivo con los datos de la base de datos Nombre de instancia, usuario y contraseña del motor de base de datos, para permitir al sistema poder conectarse con la base de datos. Guardar el archivo.

```
<connectionStrings>
  <add name="MEDICAL_CENTER_DB_CON" connectionString="metadata=res://*/Administration.csdl|res://*/Administration.ssdl|res://*/Administration.msl;
  provider=System.Data.SqlClient;provider connection string="Data Source=ENIAC\SQL2008;Initial Catalog=MEDICAL_CENTER_NET;Persist Security
  Info=True;User ID=sa;Password=password;MultipleActiveResultSets=True";"
  providerName="System.Data.EntityClient" />
  <add name="MEDICAL_CENTER_DB_CON_EXP" connectionString="Data Source=ENIAC\SQL2008;Initial Catalog=MEDICAL_CENTER_NET;Integrated Security=False;
  Persist Security Info=True;User ID=sa;Password=password"
  providerName="System.Data.SqlClient" />
  <add name="MEDICAL_CENTER_BK_DB_CON" connectionString="metadata=res://*/BackOffice.csdl|res://*/BackOffice.ssdl|res://*/BackOffice.msl;provider=
  System.Data.SqlClient;provider connection string="Data Source=ENIAC\SQL2008;Initial Catalog=MEDICAL_CENTER_NET;Persist Security Info=True;
  User ID=sa;Password=password;MultipleActiveResultSets=True;Application Name=EntityFramework";"
  providerName="System.Data.EntityClient" />
</connectionStrings>
```

xvii. Para ejecutar el aplicativo acceder desde cualquier navegador a la URL:  
**Referencia de hipervínculo no válida. servidor]/MedicalCenterNet**

