

Diseño y Modelación de un Proyecto de Software Utilizando el lenguaje UML

INTRODUCCION

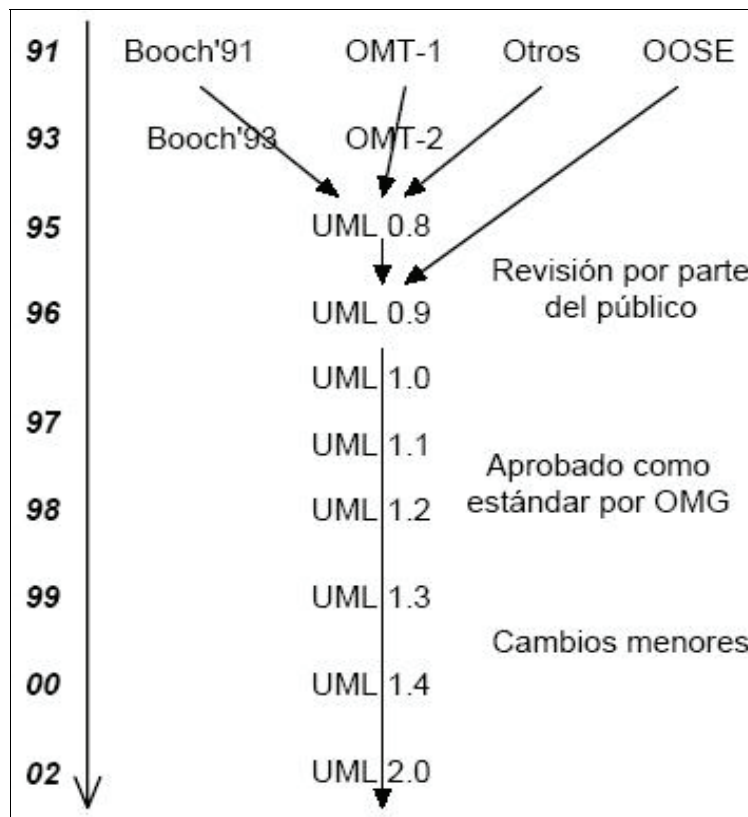
Desde los inicios de la informática se han estado utilizando distintas formas de representar los diseños de una manera más bien personal o con algún modelo gráfico, La falta de estandarización en la representación gráfica de un modelo impedía que los diseños gráficos realizados se pudieran compartir fácilmente entre distintos diseñadores, con este objetivo se creo el Lenguaje Unificado de Modelado (UML: Unified Modeling Language).

UML es el lenguaje de modelado de sistemas de software más conocido en la actualidad; es el estándar internacional aprobado por la OMG[1](Object Managment Group), consorcio creado en 1989 responsable de la creación, desarrollo y revisión de especificaciones para la industrial del software.

UML son un grupo de especificaciones de notación orientadas a Objeto, las cuales están compuesta por distintos diagramas, que representan las diferentes etapas del desarrollo de un proyecto de software. Este trabajo se centra en un Sistema de Control de Citas Médicas. Se han usados varios de los diagramas de UML, de modo que se muestre el uso de los mismos, enfocado desde una perspectiva práctica.

DESCRIPCION

El lenguaje UML comenzó a gestarse en octubre de 1994, cuando Rumbaugh se unió a la compañía Rational fundada por Booch (dos reputados investigadores en el área de metodología del software). El objetivo de ambos era unificar dos métodos que habían desarrollado: el método Booch y el OMT (Object Modelling Tool). El primer borrador apareció en octubre de 1995. En esa misma época otro reputado investigador, Jacobson, se unió a Rational y se incluyeron ideas suyas. Estas tres personas son conocidas como los "tres amigos". Además, este lenguaje se abrió a la colaboración de otras empresas para que aportaran sus ideas. Todas estas colaboraciones condujeron a la definición de la primera versión de UML.



1. Modelado: es el diseño de un software antes de su codificación, es la visualización de lo que se quiere construir.

Esta primera versión se ofreció a un grupo de trabajo para convertirlo en 1997 en un estándar del OMG. Este grupo gestiona estándares relacionados con la tecnología orientada a objetos (metodologías, bases de datos objetuales, CORBA, etc.), propuso una serie de modificaciones y una nueva versión de UML (la 1.1), que fue adoptada por el OMG como estándar en noviembre de 1997. Desde aquella versión han habido varias revisiones que gestiona la OMG Revision Task Force. La última versión aprobada es la UML 2.0

superstructure. En estos momentos se está desarrollando actualizaciones a esta versión en la que se incluirán cambios importantes (principalmente añadir nuevos diagramas).

OBJETIVOS GENERALES

- Desarrollar el diseño y modelación de un Sistema de Control de Citas Médicas utilizando el lenguaje UML.
- Impulsar el acercamiento hacia una nueva manera de entender el diseño de software basado en UML.

OBJETIVOS ESPECIFICOS

- Estudiar el lenguaje de Modelado UML.
- Desarrollar por completo el diseño de un proyecto de software con el fin de comprender todo el proceso.
- Identificar en el diseño del proyecto los distintos tipos de diagramas que existen como son los:
 - Diagramas de clases
 - Casos de usos
 - Paquetes
 - Diagramas de interacción y secuencia, y los diagramas de transición de estados
- Aplicar patrones de diseño modernos para la construcción de una aplicación de software utilizando para ello la herramienta Rational Rose.
- Mostrar como UML crea un protocolo de comunicación estándar entre los desarrolladores.

ALCANCE

El trabajo presentado a continuación es un estudio sobre el Lenguaje de Modelado que abarca desde la definición de sus conceptos hasta su aplicación en un ejemplo práctico, en el mismo veremos como UML nos permite experimentar y visualizar un sistema que aun no ha sido codificado.

Este trabajo contiene la siguiente documentación:

- Diseño de Sistema utilizando UML
 - Historia del UML
 - Que es UML
 - Bloques de Construcción UML
 - Elementos Estructurales
 - Elementos de comportamiento
 - Elementos de agrupación
 - Elementos de anotación
 - Relaciones
 - Diagramas
- Caso Practico de un Diseño de Software utilizando UML (Sistema de Control de Citas Medicas)
 - Definición de los requerimientos del sistema.
 - Los diagramas de casos y subcasos de uso.
 - La descripción de los casos de uso.
 - Diagrama de Estructura Estática (de Clases).
 - Diagrama de Interacción.

Este trabajo solamente incluye la codificación del modulo de paciente, con el fin de mostrar como se lleva a un lenguaje particular el diseño que se ha realizado en UML.

JUSTIFICACION

Standish Group, CHAOS Report nos muestra en su estudio del 2002 que el 26% de los proyectos de software son exitosos, lo que quiere decir que el 74% fallan. La razón básica por la que fallan los proyectos se determina en la etapa de análisis y diseño del sistema.

Entendiendo lo anterior, podemos decir que es necesario y obligatorio el mejorar la calidad del desarrollo de software y para esto debemos adoptar procedimientos, metodologías y herramientas que permitan una estandarización en la ingeniería de software, esto es precisamente lo que ofrecen los lenguajes de modelado de software, un lenguaje común que permite el crear una disciplina con estándares como existe en la ingeniería civil, ingeniería eléctrica, etc.

Siendo UML el estándar internacional para el modelado hemos decidido el desarrollar este tema para este proyecto, veamos algunos de los beneficios que ofrece UML:

- Contaremos con un mejor entendimiento del riesgo del proyecto antes de construir el sistema
- Mejores tiempos totales de desarrollo (de 50% o mas)
- Podremos especificar la estructura y el comportamiento del sistema y comunicarlo a todos los integrantes del proyecto

- Se documentarán las decisiones de la arquitectura del proyecto
- Se obtendrá el “plano” del sistema
- Mejor soporte a la planeación y al control del proyecto
- Un aumento en la calidad del desarrollo
- Reducción en los costos económicos

Estas son algunas de las razones por la cual es necesario adoptar UML como lenguaje de modelado, otra razón importante es el hecho de que muchas compañías a la hora de contratar servicios de desarrollo exigen que el lenguaje de modelado utilizado sea UML.

METODOLOGÍA

Tarea 1. Documentación: En esta etapa se realizarán consultas bibliográficas relacionadas con el análisis y diseño de sistemas de información con UML, a los fines de elaborar un manual de UML con sus diagramas, definición y ejemplos.

Tarea 2. Análisis de requerimientos: En esta etapa se busca la necesidad del usuario y la forma en que se va a presentar la solución.

Actividades:

- Identificar Casos de Uso del sistema
- Dar detalle a los casos de uso descritos
- Definir una interfaz inicial del sistema
- Desarrollar el Diagramas necesarios
- Desarrollar Diccionario de Datos

Tarea 3. Diseño del sistema: en esta etapa se define una subdivisión del sistema por funciones y la forma de comunicación para su interacción.

Actividades:

- Identificar la arquitectura del sistema
 1. Definir los componentes del sistema
 2. Refinar los casos de uso (textualmente y en diagrama)

Tarea 4. Diseño detallado: en esta etapa se adecuará el análisis a las características específicas del software.

Actividades:

- Agregar detalles de implementación al modelo del mundo
- Desarrollar el modelo de interfaz
- Desarrollar los modelos de control, persistencia y comunicación

Medios y Materiales a utilizar:

- Hardware
 - Computador Pentium bajo Windows XP.
- Software
 - Rational Rose(Software para el modelado)

Historia del UML

La notación UML se deriva y unifica las tres metodologías de análisis y diseño Orientada a Objeto más extendidas:

- Metodología de *Grady Booch* para la descripción de conjuntos de objetos y sus relaciones.
- Técnica de modelado orientada a objetos de *James Rumbaugh* (OMT: Object-Modeling Technique).
- Aproximación de *Ivar Jacobson* (OOSE: Object- Oriented Software Engineering) mediante la metodología de casos de uso (*use case*).

El desarrollo de UML comenzó a finales de 1994 cuando *Grady Booch* y *Jim Rumbaugh* de *Rational Software Corporation* empezaron a unificar sus métodos. A finales de 1995, *Ivar Jacobson* y su compañía *Objectory* se incorporaron a *Rational* en su unificación, aportando el método OOSE.

De las tres metodologías de partida, las de *Booch* y *Rumbaugh* pueden ser descritas como centradas en objetos, ya que sus aproximaciones se enfocan hacia el modelado de los objetos que componen el sistema, su relación y colaboración. Por otro lado, la metodología de *Jacobson* es más centrada a usuario, ya que todo en su método se deriva de los escenarios de uso. UML se ha ido fomentando y aceptando como estándar desde el OMG que es también el origen de CORBA, el estándar líder en la industria para la programación de objetos distribuidos. En 1997 UML 1.1 fue aprobada por la OMG convirtiéndose en la notación estándar de facto para el análisis y el diseño orientado a objetos.

Qué es UML?

UML es el primer método en publicar un meta-modelo en su propia notación, incluyendo la notación para la mayoría de la información de requisitos, análisis y diseño. Se trata pues de un meta-modelo auto-referencial (cualquier lenguaje de modelado de propósito general debería ser capaz de modelarse a sí mismo).

UML es un lenguaje estándar que sirve para escribir los *planos del software*, puede utilizarse para visualizar, especificar, construir y documentar todos los artefactos que componen un sistema con gran cantidad de software. UML puede usarse para modelar desde sistemas de información hasta aplicaciones distribuidas basadas en Web, pasando por sistemas empujados de tiempo real.

UML es solamente un lenguaje por lo que es sólo una parte de un método de desarrollo software, es independiente del proceso aunque para que sea óptimo debe usarse en un proceso dirigido por casos de uso, centrado en la arquitectura, iterativo e incremental.

UML es un lenguaje por que proporciona un vocabulario y las reglas para utilizarlo, además es un lenguaje de modelado lo que significa que el vocabulario y las reglas se utilizan para la representación conceptual y física del sistema.

UML es un lenguaje que nos ayuda a interpretar grandes sistemas mediante gráficos o mediante texto obteniendo modelos explícitos que ayudan a la comunicación durante el desarrollo ya que al ser estándar, los modelos podrán ser interpretados por personas que no participaron en su diseño (e incluso por herramientas) sin ninguna ambigüedad. En este contexto, UML sirve para *especificar*, modelos concretos, no ambiguos y completos.

Debido a su estandarización y su definición completa no ambigua, y aunque no sea un lenguaje de programación, UML se puede conectar de manera directa a lenguajes de programación como Java, C++ o Visual Basic, esta correspondencia permite lo que se denomina como ingeniería directa (obtener el código fuente partiendo de los modelos) pero además es posible reconstruir un modelo en UML partiendo de la implementación, o sea, la ingeniería inversa.

UML proporciona la capacidad de modelar actividades de planificación de proyectos y de sus versiones, expresar requisitos y las pruebas sobre el sistema, representar todos sus detalles así como la propia arquitectura. Mediante estas capacidades se obtiene una documentación que es valida durante todo el ciclo de vida de un proyecto.

El lenguaje UML se compone de tres elementos básicos, los bloques de construcción, las reglas y algunos mecanismos comunes. Estos elementos interactúan entre sí para dar a UML el carácter de completitud y no-ambigüedad que antes comentábamos.

Los **bloques de construcción** se dividen en tres partes:

- **Elementos**, que son las abstracciones de primer nivel.
- **Relaciones**, que unen a los elementos entre sí.
- **Diagramas**, que son agrupaciones de elementos.

Existen cuatro tipos de elementos en UML, dependiendo del uso que se haga de ellos:

- *Elementos estructurales.*
- *Elementos de comportamiento.*
- *Elementos de agrupación*
- *Elementos de anotación.*

Las relaciones, a su vez se dividen para abarcar las posibles interacciones entre elementos que se nos pueden presentar a la hora de modelar usando UML, estas son: *relaciones de dependencia*, *relaciones de asociación*, *relaciones de generalización* y *relaciones de realización*.

Se utilizan diferentes diagramas dependiendo de qué, nos interese representar en cada momento, para dar diferentes perspectivas de un mismo problema, para ajustar el nivel de detalle..., por esta razón UML soporta un gran numero de diagramas diferentes aunque, en la practica, sólo se utilicen un pequeño número de combinaciones.

UML proporciona un conjunto de reglas que dictan las pautas a la hora de realizar asociaciones entre objetos para poder obtener modelos bien formados, estas son reglas semánticas que afectan a los **nombres**, al **alcance** de dichos nombres, a la **visibilidad** de estos nombres por otros, a la **integridad** de unos elementos con otros y a la **ejecución**, o sea la vista dinámica del sistema.

UML proporciona una serie de mecanismos comunes que sirven para que cada persona o entidad adapte el lenguaje a sus necesidades, pero dentro de un marco ordenado y siguiendo unas ciertas reglas para que en el trasfondo de la adaptación no se pierda la semántica propia de UML. Dentro de estos mecanismos están las **especificaciones**, que proporcionan la explicación textual de la sintaxis y semántica de los bloques de construcción.

Otro mecanismo es el de los **adornos** que sirven para conferir a los modelos de más semántica, los adornos son elementos secundarios ya que proporcionan más nivel de detalle, que quizá en un primer momento no sea conveniente descubrir. Las **divisiones comunes** permiten que los modelos se dividan al menos en un

par de formas diferentes para facilitar la comprensión desde distintos puntos de vista, en primer lugar tenemos la división entre clase y objeto (clase es una abstracción y objeto es una manifestación de esa abstracción), en segundo lugar tenemos la división interfaz / implementación donde la interfaz presenta un contrato (algo que se va a cumplir de una determinada manera) mientras que la implementación es la manera en que se cumple dicho contrato.


Por ultimo, los **mecanismos de extensibilidad** que UML proporciona sirven para evitar posibles problemas que puedan surgir debido a la necesidad de poder representar ciertos matices, por esta razón UML incluye los *estereotipos*, para poder extender el vocabulario con nuevos bloques de construcción, los *valores etiquetados*, para extender las propiedades un bloque, y las *restricciones*, para extender la semántica. De esta manera UML es un lenguaje estándar "abierto-cerrado" siendo posible extender el lenguaje de manera controlada.

Elementos Estructurales

Los elementos estructurales en UML, es su mayoría, son las partes estáticas del modelo y representan cosas que son conceptuales o materiales.


Clases

Una clase es una descripción de un conjunto de objetos que comparten los mismos atributos, operaciones, relaciones y semántica. Una clase implementa una o más interfaces. Gráficamente se representa como un rectángulo que incluye su nombre, sus atributos y sus operaciones.

Clase	 <pre> classDiagram class Tostadora { +noBandejas +color +tostar() } </pre>	Describe un conjunto de objetos que comparten los mismos atributos, métodos, relaciones y semántica. Las clases implementan una o más interfaces.
-------	--	---


Interfaz

Una interfaz es una colección de operaciones que especifican un servicio de una determinada clase o componente. Una interfaz describe el comportamiento visible externamente de ese elemento, puede mostrar el comportamiento completo o sólo una parte del mismo. Una interfaz describe un conjunto de especificaciones de operaciones (o sea su signatura) pero nunca su implementación. Se representa con un círculo, , y rara vez se encuentra aislada sino que más bien conectada a la clase o componente que realiza.

Interfaz	 <p>Interfaz</p>	Agrupación de métodos u operaciones que especifican un servicio de una clase o componente, describiendo su comportamiento, completo o parcial, externamente visible. UML permite emplear un círculo para representar las interfaces, aunque lo más normal es emplear la clase con el nombre en cursiva.
----------	---	---


Colaboración

Define una interacción y es una sociedad de roles y otros elementos que colaboran para proporcionar un comportamiento cooperativo mayor que la suma de los comportamientos de sus elementos. Las colaboraciones tienen una dimensión tanto estructural como de comportamiento. Una misma clase puede participar en diferentes colaboraciones. Las colaboraciones representan la implementación de patrones que forman un sistema. Se representa mediante una elipse con borde discontinuo.

Colaboración	 Colaboracion	Define una interacción entre elementos que cooperan para proporcionar un comportamiento mayor que la suma de los comportamientos de sus elementos.
--------------	---	--

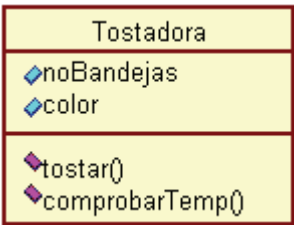
Casos de Uso

Un caso de uso es la descripción de un conjunto de acciones que un sistema ejecuta y que produce un determinado resultado que es de interés para un actor particular. Un caso de uso se utiliza para organizar los aspectos del comportamiento en un modelo. Un caso de uso es realizado por una colaboración. Se representa como en la figura 6, una elipse con borde continuo.

Caso de uso	 Caso de Uso	Describe un conjunto de secuencias de acciones que un sistema ejecuta, para producir un resultado observable de interés. Se emplea para estructurar los aspectos de comportamiento de un modelo.
-------------	--	--

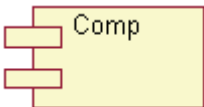
Clase Activa

Es una clase cuyos objetos tienen uno o más procesos o hilos de ejecución por lo y tanto pueden dar lugar a actividades de control. Una clase activa es igual que una clase, excepto que sus objetos representan elementos cuyo comportamiento es concurrente con otros elementos. Se representa igual que una clase, pero con líneas más gruesas

Clase activa		Se trata de una clase, en la que existe procesos o hilos de ejecución concurrentes con otros elementos. Las líneas del contorno son más gruesas que en la clase "normal"
--------------	---	--

Componentes

Un componente es una parte física y reemplazable de un sistema que conforma con un conjunto de interfaces y proporciona la implementación de dicho conjunto. Un componente representa típicamente el empaquetamiento físico de diferentes elementos lógicos, como clases, interfaces y colaboraciones.

Componente	 Comp	Parte física y por tanto reemplazable de un modelo, que agrupa un conjunto de interfaces, archivos de código fuente, clases, colaboraciones y proporciona la implementación de dichos elementos.
------------	---	--

Nodos

Un nodo es un elemento físico que existe en tiempo de ejecución y representa un recurso computacional que, por lo general, dispone de algo de memoria y, con frecuencia, de capacidad de procesamiento. Un conjunto de componentes puede residir en un nodo.

Nodo		Elemento físico que existe en tiempo de ejecución y representa un recurso computacional con capacidad de procesar.
------	---	--

Estos siete elementos vistos son los elementos estructurales básico que se pueden incluir en un modelo UML. Existen variaciones sobre estos elementos básicos, tales como actores, señales, utilidades (tipos de clases), procesos e hilos (tipos de clases activas) y aplicaciones, documentos, archivos, bibliotecas, páginas y tablas (tipos de componentes).

Elementos de comportamiento

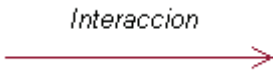
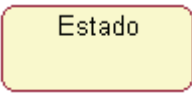
Los elementos de comportamiento son las partes dinámicas de un modelo. Se podría decir que son los verbos de un modelo y representan el comportamiento en el tiempo y en el espacio. Los principales elementos son los dos que siguen.

Interacción

Es un comportamiento que comprende un conjunto de mensajes intercambiados entre un conjunto de objetos, dentro de un contexto particular para conseguir un propósito específico. Una interacción involucra otros muchos elementos, incluyendo mensajes, secuencias de acción (comportamiento invocado por un objeto) y enlaces (conexiones entre objetos). La representación de un mensaje es una flecha dirigida que normalmente con el nombre de la operación.

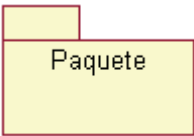
Maquinas de estados

Es un comportamiento que especifica las secuencias de estados por las que van pasando los objetos o las interacciones durante su vida en respuesta a eventos, junto con las respuestas a esos eventos. Una maquina de estados involucra otros elementos como son estados, transiciones (flujo de un estado a otro), eventos (que disparan una transición) y actividades (respuesta de una transición)

Elementos de comportamiento	Interacción		Comprende un conjunto de mensajes que se intercambian entre un conjunto de objetos, para cumplir un objetivo específico.
	Máquinas de estados		Especifica la secuencia de estados por los que pasa un objeto o una interacción, en respuesta a eventos.

Elementos de agrupación

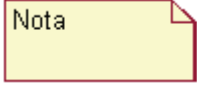
Forman la parte organizativa de los modelos UML. El principal elemento de agrupación es el **paquete**, que es un mecanismo de propósito general para organizar elementos en grupos. Los elementos estructurales, los elementos de comportamiento, incluso los propios elementos de agrupación se pueden incluir en un paquete. Un paquete es puramente conceptual (sólo existe en tiempo de desarrollo). Gráficamente se representa como una carpeta conteniendo normalmente su nombre y, a veces, su contenido.

Elementos de agrupación	Paquete		Se emplea para organizar otros elementos en grupos.
-------------------------	---------	---	---

Elementos de anotación

Los elementos de anotación son las partes explicativas de los modelos UML. Son comentarios que se pueden aplicar para describir, clasificar y hacer observaciones sobre cualquier elemento de un modelo.

El tipo principal de anotación es la **nota** que simplemente es un símbolo para mostrar restricciones y comentarios junto a un elemento o un conjunto de elementos.


Elementos de notación	Nota		Partes explicativa de UML, que puede describir textualmente cualquier aspecto del modelo
-----------------------	------	---	--

Relaciones

Existen cuatro tipos de relaciones entre los elementos de un modelo UML. *Dependencia*, *asociación*, *generalización* y *realización*, estas se describen a continuación:


Dependencia

Es una relación semántica entre dos elementos en la cual un cambio a un elemento (el elemento independiente) puede afectar a la semántica del otro elemento (elemento dependiente). Se representa como una línea discontinua, posiblemente dirigida, que a veces incluye una etiqueta.

Dependencia		Es una relación entre dos elementos, tal que un cambio en uno puede afectar al otro.
-------------	---	--


Asociación

Es una relación estructural que describe un conjunto de enlaces, los cuales son conexiones entre objetos. La agregación es un tipo especial de asociación y representa una relación estructural entre un todo y sus partes. La asociación se representa con una línea continua, posiblemente dirigida, que a veces incluye una etiqueta. A menudo se incluyen otros adornos para indicar la multiplicidad y roles de los objetos involucrados.

Asociación		Es una relación estructural que resume un conjunto de enlaces que son conexiones entre objetos.
------------	---	---


Generalización

Es una relación de especialización / generalización en la cual los objetos del elemento especializado (el hijo) pueden sustituir a los objetos del elemento general (el padre). De esta forma, el hijo comparte la estructura y el comportamiento del padre. Gráficamente, la generalización se representa con una línea con punta de flecha vacía.

Generalización		Es una relación en la que el elemento generalizado puede ser substituido por cualquiera de los elementos hijos, ya que comparten su estructura y comportamiento.
----------------	---	--

Realización

Es una relación semántica entre clasificadores, donde un clasificador especifica un contrato que otro clasificador garantiza que cumplirá. Se pueden encontrar relaciones de realización en dos sitios: entre interfaces y las clases y componentes que las realizan, y entre los casos de uso y las colaboraciones que los realizan. La realización se representa como una mezcla entre la generalización y la dependencia, esto es, una línea discontinua con una punta de flecha vacía .

Realización		Es una relación que implica que la parte realizante cumple con una serie de especificaciones propuestas por la clase realizada (interfaces).
-------------	---	--

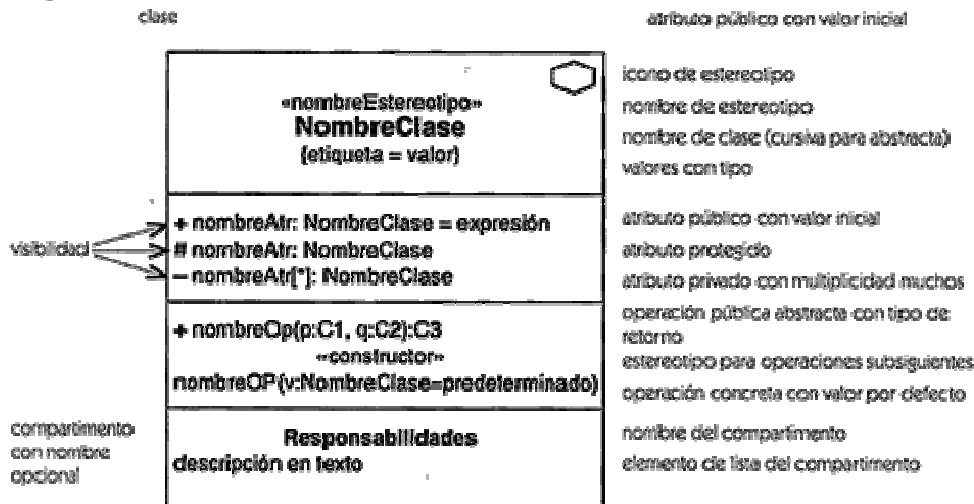
Diagramas

Los diagramas se utilizan para representar diferentes perspectivas de un sistema de forma que un diagrama es una proyección del mismo. UML proporciona un amplio conjunto de diagramas que normalmente se usan en pequeños subconjuntos para poder representar las cinco vistas principales de la arquitectura de un sistema.

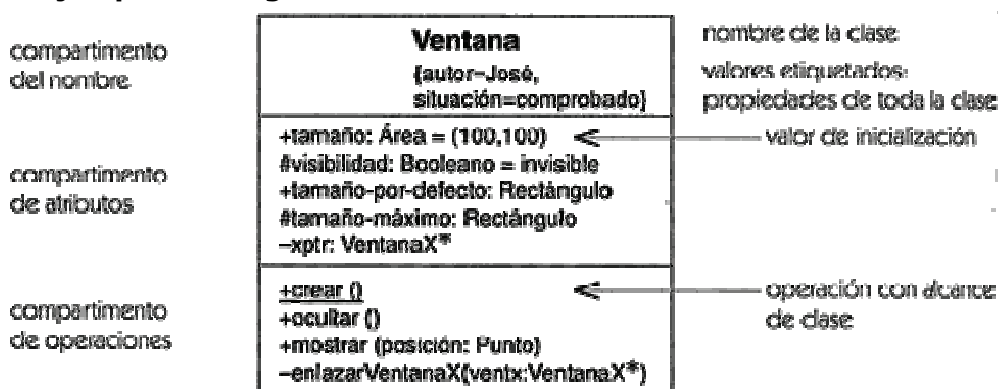
Diagramas de Clases

Muestran un conjunto de clases, interfaces y colaboraciones, así como sus relaciones. Estos diagramas son los más comunes en el modelado de sistemas orientados a objetos y cubren la vista de diseño estática o la vista de procesos estática (sí incluyen clases activas).

Diagrama de Clases

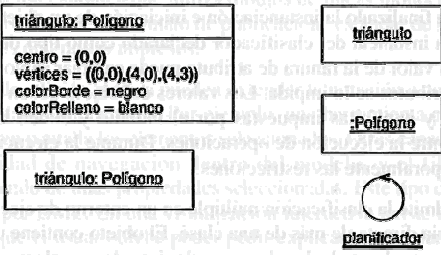


Ejemplo de Diagrama de Clases



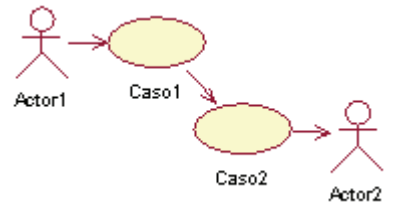
Diagramas de Objetos

Muestran un conjunto de objetos y sus relaciones, son como fotos instantáneas de los diagramas de clases y cubren la vista de diseño estática o la vista de procesos estática desde la perspectiva de casos reales o prototípicos.

Objetos	<p>Diagrama de Objetos</p>  <p>Diagrama de Objetos que muestra tres objetos de la clase <u>Polígono</u>: un triángulo con atributos (centro = (0,0), vértices = ((0,0),(4,0),(4,3)), colorBorde = negro, colorRelleno = blanco), otro triángulo, y un polígono. Se muestra también un planificador que genera el triángulo.</p>	<p>Análogo al diagrama de clases, muestra un conjunto de objetos y sus relaciones, pero a modo de vista instantánea de instancias de una clase en el tiempo.</p>
---------	---	--

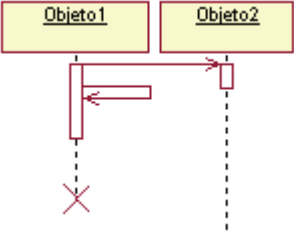
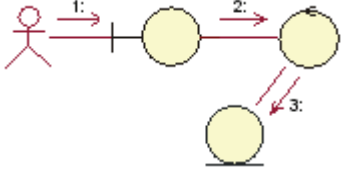
Diagramas de Casos de Usos

Muestran un conjunto de casos de uso y actores (tipo especial de clases) y sus relaciones. Cubren la vista estática de los casos de uso y son especialmente importantes para el modelado y organización del comportamiento.

Casos de Uso	 <p>Diagrama de Casos de Uso que muestra dos actores (Actor1 y Actor2) y dos casos de uso (Caso1 y Caso2). Actor1 está conectado a Caso1, y Caso1 está conectado a Caso2, que a su vez está conectado a Actor2.</p>	<p>Muestra un conjunto de casos de uso, los actores implicados y sus relaciones. Son diagramas fundamentales en el modelado y organización del sistema.</p>
--------------	---	---

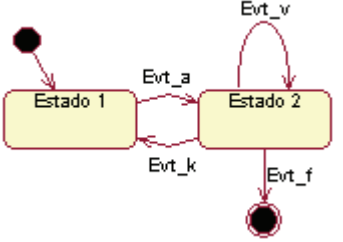
Diagramas de Secuencia y de Colaboración

Tanto los diagramas de secuencia como los diagramas de colaboración son un tipo de diagramas de interacción. Constan de un conjunto de objetos y sus relaciones, incluyendo los mensajes que se pueden enviar unos objetos a otros. Cubren la vista dinámica del sistema. Los diagramas de secuencia enfatizan el ordenamiento temporal de los mensajes mientras que los diagramas de colaboración muestran la organización estructural de los objetos que envían y reciben mensajes. Los diagramas de secuencia se pueden convertir en diagramas de colaboración sin pérdida de información, lo mismo ocurren en sentido opuesto.

Secuencia		Son diagramas de interacción, muestran un conjunto de objetos y sus relaciones, así como los mensajes que se intercambian entre ellos. Cubren la vista dinámica del sistema. El diagrama de secuencia resalta la ordenación temporal de los mensajes, mientras que el de colaboración resalta la organización estructural de los objetos, ambos siendo equivalentes o isomorfos.
Colaboración		En el diagrama de colaboración de la figura de la izquierda, se puede ver que los elementos gráficos no son cajas rectangulares, como cabría esperar, y en su lugar encontramos sus versiones adornadas. Estas versiones tienen como finalidad evidenciar un rol específico del objeto siendo modelado. En la figura encontramos de izquierda a derecha y de arriba abajo un Actor, una Interfaz, un Control (modela un comportamiento) y una Instancia (modela un objeto de dato).

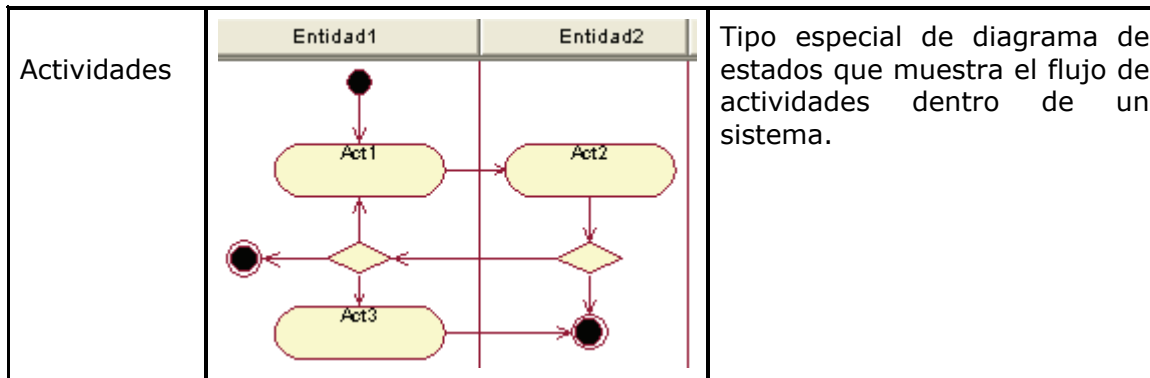
Diagramas de Estados

Muestran una máquina de estados compuesta por estados, transiciones, eventos y actividades. Estos diagramas cubren la vista dinámica de un sistema y son muy importantes a la hora de modelar el comportamiento de una interfaz, clase o colaboración.

Estados		Muestra una máquina de estados, con sus estados, transiciones, eventos y actividades. Cubren la vista dinámica de un sistema. Modelan comportamientos reactivos en base a eventos.
---------	---	--

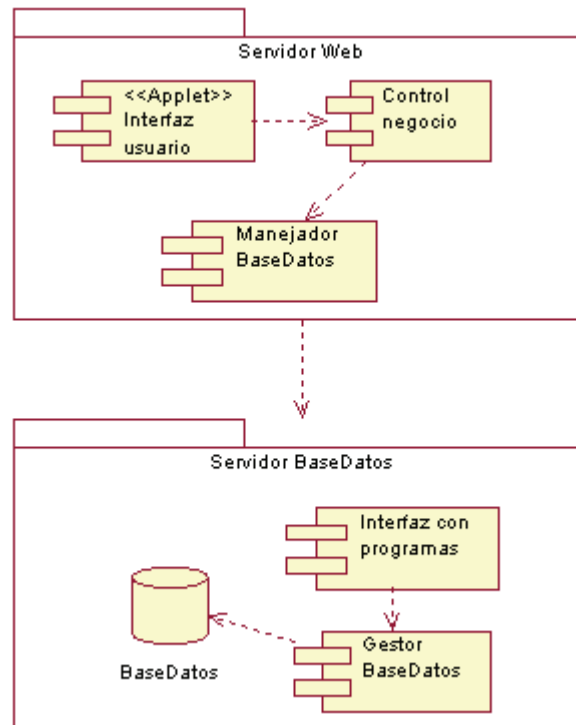
Diagramas de Actividades

Son un tipo especial de diagramas de estados que se centra en mostrar el flujo de actividades dentro de un sistema. Los diagramas de actividades cubren la parte dinámica de un sistema y se utilizan para modelar el funcionamiento de un sistema resaltando el flujo de control entre objetos.



Diagramas de Componentes

Muestra la organización y las dependencias entre un conjunto de componentes. Cubren la vista de la implementación estática y se relacionan con los diagramas de clases ya que en un componente suele tener una o más clases, interfaces o colaboraciones



Diagramas de Despliegue

Representan la configuración de los nodos de procesamiento en tiempo de ejecución y los componentes que residen en ellos. Muestran la vista de despliegue estática de una arquitectura y se relacionan con los componentes ya que, por lo común, los nodos contienen uno o más componentes.

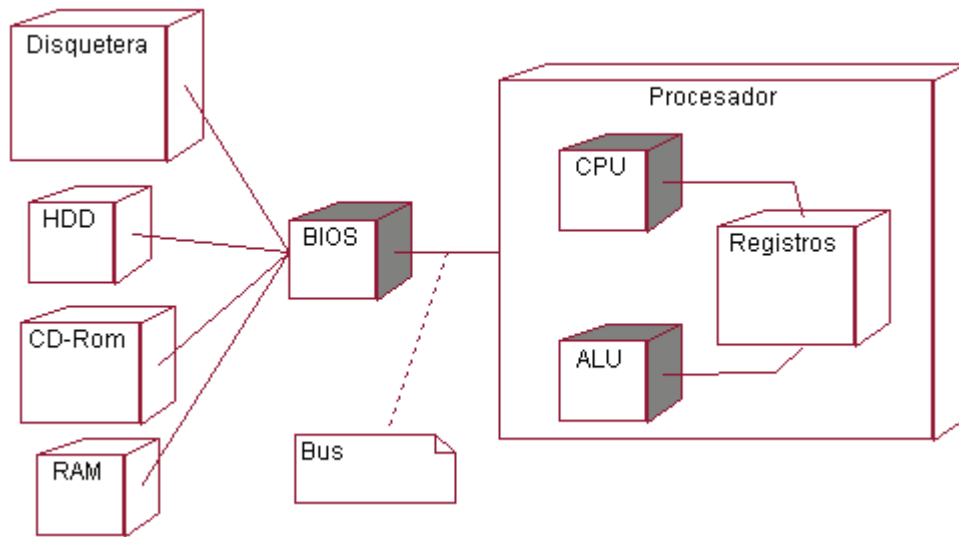


Diagrama de Despliegue

Arquitectura

El desarrollo de un sistema con gran cantidad de software requiere que este sea visto desde diferentes perspectivas. Diferentes usuarios (usuario final, analistas, desarrolladores, integradores, jefes de proyecto...) siguen diferentes actividades en diferentes momentos del ciclo de vida del proyecto, lo que da lugar a las diferentes vistas del proyecto, dependiendo de qué interés más en cada instante de tiempo.

La arquitectura es el conjunto de decisiones significativas sobre:

- La organización del sistema
- Selección de elementos estructurales y sus interfaces a través de los cuales se constituye el sistema.
- El Comportamiento, como se especifica las colaboraciones entre esos componentes.
- Composición de los elementos estructurales y de comportamiento en subsistemas progresivamente más grandes.
- El estilo arquitectónico que guía esta organización: elementos estáticos y dinámicos y sus interfaces, sus colaboraciones y su composición.

La una arquitectura que no debe centrarse únicamente en la estructura y en el comportamiento, sino que abarque temas como el uso, funcionalidad, rendimiento, capacidad de adaptación, reutilización, capacidad para ser comprendida, restricciones, compromisos entre alternativas, así como aspectos estéticos. Para ello se sugiere una arquitectura que permita describir mejor los sistemas desde diferentes vistas, donde cada una de ellas es una proyección de la organización y la estructura centrada en un aspecto particular del sistema.

La *vista de casos de uso* comprende la descripción del comportamiento del sistema tal y como es percibido por los usuarios finales, analistas y encargados de las pruebas y se utilizan los diagramas de casos de uso para capturar los aspectos estáticos mientras que los dinámicos son representados por diagramas de interacción, estados y actividades.

La *vista de diseño* comprende las clases, interfaces y colaboraciones que forman el vocabulario del problema y de la solución. Esta vista soporta principalmente los requisitos funcionales del sistema, o sea, los servicios que el sistema debe proporcionar. Los aspectos estáticos se representan mediante diagramas de clases y objetos y los aspectos dinámicos con diagramas de interacción, estados y actividades.

La *vista de procesos* comprende los hilos y procesos que forman mecanismos de sincronización y concurrencia del sistema cubriendo el funcionamiento, capacidad de crecimiento y el rendimiento del sistema. Con UML, los aspectos estáticos y dinámicos se representan igual que en la vista de diseño, pero con el énfasis que aportan las clases activas, las cuales representan los procesos y los hilos.

La *Vista de implementación* comprende los componentes y los archivos que un sistema utiliza para ensamblar y hacer disponible el sistema físico. Se ocupa principalmente de la gestión de configuraciones de las distintas versiones del sistema. Los aspectos estáticos se capturan con los diagramas de componentes y los aspectos dinámicos con los diagramas de interacción, estados y actividades.

La *vista de despliegue* de un sistema contiene los nodos que forman la topología hardware sobre la que se ejecuta el sistema. Se preocupa principalmente de la distribución, entrega e instalación de las partes que constituyen el sistema. Los aspectos estáticos de esta vista se representan mediante los diagramas de despliegue y los aspectos dinámicos con diagramas de interacción, estados y actividades

Ciclo de Vida

Se entiende por *ciclo de vida de un proyecto software* a todas las etapas por las que pasa un proyecto, desde la concepción de la idea que hace surgir la necesidad de diseñar un sistema software, pasando por el análisis, desarrollo, implantación y mantenimiento del mismo y hasta que finalmente muere por ser sustituido por otro sistema.

Aunque UML es bastante independiente del proceso, para obtener el máximo rendimiento de UML se debería considerar un proceso que fuese:

Dirigido por los *casos de uso*, o sea, que los casos de uso sean un artefacto básico para establecer el comportamiento del deseado del sistema, para validar la arquitectura, para las pruebas y para la comunicación entre las personas involucradas en el proyecto.

Centrado en la *arquitectura* de modo que sea el artefacto básico para conceptuar, construir, gestionar y hacer evolucionar el sistema.

Un proceso *iterativo*, que es aquel que involucra la gestión del flujo de ejecutables del sistema, e *incremental*, que es aquel donde cada nueva versión corrige defectos de la anterior e incorpora nueva funcionalidad. Un proceso iterativo e incremental se denomina *dirigido por el riesgo*, lo que significa que cada nueva versión se ataca y reducen los riesgos más significativos para el éxito del proyecto.

Este proceso, dirigido a los casos de uso, centrado en la arquitectura, iterativo e incremental puede descomponerse en fases, donde cada fase es el intervalo de tiempo entre dos hitos importantes del proceso, cuando se cumplen los objetivos bien definidos, se completan los artefactos y se toman decisiones sobre si pasar o no a la siguiente fase.

En el ciclo de vida de un proyecto software existen cuatro fases. La **iniciación**, que es cuando la idea inicial está lo suficientemente fundada para poder garantizar la entrada en la fase de **elaboración**, esta fase es cuando se produce la definición de la arquitectura y la visión del producto. En esta fase se deben determinar los requisitos del sistema y las pruebas sobre el mismo.

Posteriormente se pasa a la fase de **construcción**, que es cuando se pasa de la base arquitectónica ejecutable hasta su disponibilidad para los usuarios, en esta fase se reexaminan los requisitos y las pruebas que ha de soportar. La **transición**, cuarta fase del proceso, que es cuando el software se pone en mano de los usuarios. Raramente el proceso del software termina en la etapa de transición, incluso durante esta fase el proyecto es continuamente reexaminado y mejorado erradicando errores y añadiendo nuevas funcionalidades no contempladas.

Un elemento que distingue a este proceso y afecta a las cuatro fases es una **iteración**, que es un conjunto bien definido de actividades, con un plan y unos criterios de evaluación, que acaban en una versión del producto, bien interna o externa.

Caso Práctico
Requerimientos

No	Descripción
Consultas / Informes	
R01	Informe Record de pacientes
R02	Informe Citas por fecha
R03	Informe Citas por paciente por fecha

No	Descripción
Almacenamiento	
R04	Datos de Pacientes: C_PNOMBRE, C_SNOMBRE, C_PAPELIDO, C_SAPELLIDO, C_SEXO, D_FNAC, C_CEDULA, C_TELEFONO, C_COMPANIA, C_TELCOMPANIA, D_FREGISTRO
R05	Datos de Citas: C_MOTIVO, N_IDCITA, D_FREGISTRO, D_FCITA, C_HCITA, M_NOTA, C_ESTATUS, C_CEDULA.
R06	Datos Encabezado del Records: N_IDRECORD, C_CEDULA y D_FREGISTRO
R07	Datos Detalles del Record: N_IDRECORD, N_IDDETALLERECORD, C_TRATAMIENTOMEDICO, N_IDENFERMEDADESPACIENTE, N_IDMEDICAMENTOSPACIENTE, N_IDALERGIASPACIENTE y M_NOTA
R08	Datos por enfermedades de paciente: N_IDENFERMEDADESPACIENTE, N_IDENFERMEDAD y M_NOTA
R09	Datos por Medicamentos que toma el paciente: N_IDMEDICAMENTOSPACIENTE, N_IDMEDICAMENTO y M_NOTA
R10	Datos por Alergias que padece el paciente: N_IDALERGIASPACIENTE, N_IDALERGIA y M_NOTA
R11	Datos de Enfermedades: N_IDENFERMEDAD y C_ENFERMEDAD
R12	Datos de Medicamentos: N_IDMEDICAMENTO y C_MEDICAMENTO
R13	Datos de Alergias: N_IDALERGIA y C_ALERGIA

No	Descripción
Procesamiento	
R14	Calculo de Edad del Paciente: ((Fecha del Sistema - D_FNAC) / 365))

Diagramas de Casos de Uso



Sistema de Gestión de Pacientes Caso de Uso 1

Manejo de Pacientes

Manejo de Citas

Manejo de Record

Manejo de
Enfermedades

Manejo de
Medicamentos

Manejo de Alergias

Sistema de Gestión de Pacientes

Caso de Uso 1



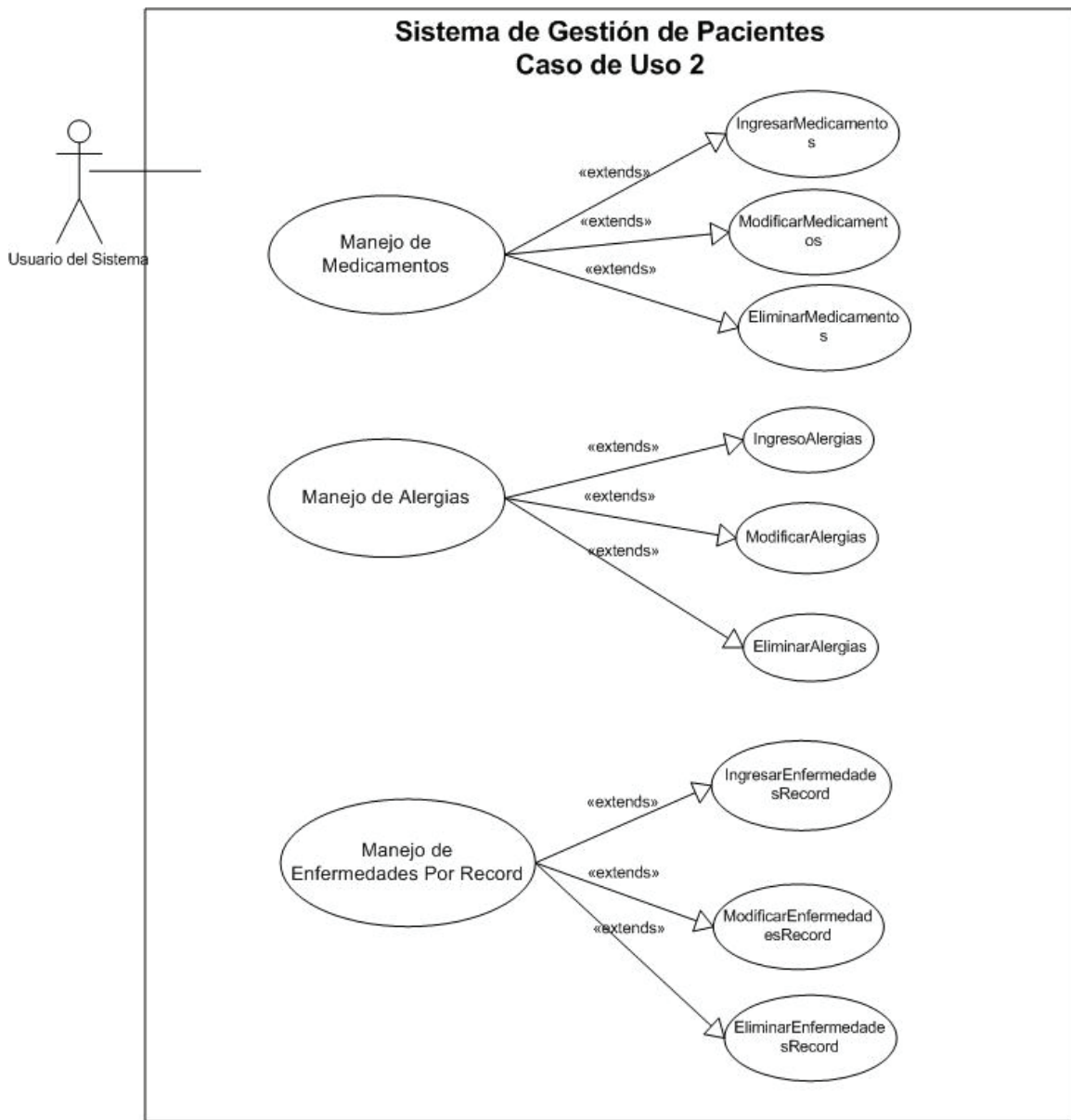
Usuario del Sistema

Manejo de
Enfermedades Por Record

Manejo de
Medicamentos Por Record

Manejo de Alergias
Por Record

Imprimir





Descripción de Casos de Uso

Nombre:	Manejo de Pacientes
Alias:	
Actores:	Usuario del Sistema, Cliente
Función:	Permitir el mantenimiento del catalogo de pacientes.
	<p>El Usuario del Sistema puede registrar pacientes nuevos, ingresando sus datos. El sistema debe validar:</p> <ol style="list-style-type: none">1. Que se ingrese una cédula.2. Que se ingrese el primer nombre y el primer apellido.3. Se asigne un Sexo.4. Se ingrese la fecha de nacimiento del paciente.5. Se ingrese un teléfono de contacto.6. Se ingrese la fecha de registro, esta será tomada de la fecha del sistema.
Descripción:	También es posible modificar o Eliminar un paciente.
Referencias:	

Nombre:	Manejo de Citas
Alias:	
Actores:	Usuario del Sistema, Cliente
Función:	Permitir el mantenimiento del catalogo de citas.
	<p>El Usuario del Sistema puede registrar nuevas citas, ingresando sus datos. El sistema debe validar:</p> <ol style="list-style-type: none">1. Que se ingrese un motivo de la cita.2. Que se ingrese un código para la cita, es generado por el sistema.3. Se ingrese una fecha de registro, esta será tomada del sistema...4. Se ingrese la fecha en que se realizará la cita.5. Se ingrese la hora de la cita.6. Se ingrese la cédula del paciente.7. Se ingrese el estatus de la cita, por defecto "abierta"
Descripción:	También es posible modificar el registro de un paciente o eliminarlo.
Referencias:	

Nombre:	Manejo de Records
Alias:	
Actores:	Usuario del Sistema, Cliente
Función:	Permitir el mantenimiento del catalogo de Records Médicos.
	<p>El Usuario del Sistema puede registrar el records médicos, ingresando sus datos. El sistema debe validar:</p> <ol style="list-style-type: none">1. Se genere un número de record automático.2. Se ingrese un numero de cédula de paciente.3. Se ingrese una fecha de registro, esta fecha es generada por el sistema.4. Se indica si el paciente esta en tratamiento medico.5. Se ingrese un comentario.
Descripción:	También es posible modificar o Eliminar un Record Medico.
Referencias:	

Nombre:	Manejo de Enfermedades
Alias:	
Actores:	Usuario del Sistema, Cliente
Función:	Permitir el mantenimiento del catalogo de enfermedades.
Descripción:	El Usuario del Sistema puede registrar enfermedades en el catalogo de enfermedades. El sistema debe validar: 1. Se genere un número de enfermedad automático. 2. Se ingrese un nombre de enfermedad.
Referencias:	También es posible modificar o eliminar una enfermedad.

Nombre:	Manejo de Medicamentos
Alias:	
Actores:	Usuario del Sistema, Cliente
Función:	Permitir el mantenimiento del catalogo de Medicamentos.
Descripción:	El Usuario del Sistema puede registrar medicamentos en el catalogo de medicamentos. El sistema debe validar: 1. Se genere un número de medicamento automático. 2. Se ingrese un nombre del medicamento.
Referencias:	También es posible modificar o eliminar un medicamento.

Nombre:	Manejo de Alergias
Alias:	
Actores:	Usuario del Sistema, Cliente
Función:	Permitir el mantenimiento del catalogo de alergias.
Descripción:	El Usuario del Sistema puede registrar nuevas alergias en el catalogo. El sistema debe validar: 1. Se genere un número de alergia automático. 2. Se ingrese un nombre de alergia.
Referencias:	También es posible modificar o eliminar una alergia.

Nombre:	Manejo de Enfermedades por Record
Alias:	
Actores:	Usuario del Sistema, Cliente
Función:	Permitir el mantenimiento de enfermedades por Record
Descripción:	El usuario del Sistema puede crear y asociar enfermedades con el record medico de un paciente. Puede modificar y eliminar sus datos.
Referencias:	

Nombre:	Manejo de Medicamentos Por Record
Alias:	
Actores:	Usuario del Sistema, Cliente
Función:	Permitir el mantenimiento de medicamentos por Record
Descripción:	El usuario del sistema puede crear y asociar el uso de medicamento con el record medico de un paciente. Puede modificar y eliminar sus datos.
Referencias:	

Nombre:	Manejo de Alergias Por Record
Alias:	
Actores:	Usuario del Sistema, Cliente
Función:	Permitir el mantenimiento de alergias por Record
Descripción:	El usuario del sistema puede crear y asociar alergias con el record medico de un paciente. Puede modificar y eliminar sus datos.
Referencias:	

Nombre:	Imprimir
Alias:	
Actores:	Usuario del Sistema, Cliente
Función:	Permitir generar Consultas o Informes Impresos.
Descripción:	Permite consultar: <ul style="list-style-type: none"> • Records Médicos. • Citas por intervalo de fechas. • Citas por intervalo de fecha de un paciente.
Referencias:	

Descripción de los Sub Casos de Uso

Nombre:	Manejo de Pacientes - IngresarPaciente
Alias:	
Actores:	Usuario del Sistema, Cliente
Función:	Permite la creación de nuevos pacientes
Descripción:	El usuario del sistema puede crear nuevos pacientes, ingresando todos sus datos requeridos, el sistema debe validar: <ol style="list-style-type: none"> 1. Que se ingrese una cédula. 2. Que se ingrese el primer nombre y el primer apellido. 3. Se asigne un Sexo. 4. Se ingrese la fecha de nacimiento del paciente. 5. Se ingrese un teléfono de contacto. 6. Se ingrese la fecha de registro, esta será tomada de la fecha del sistema.
Referencias:	

Nombre:	Manejo de Pacientes - ModificarPaciente
Alias:	
Actores:	Usuario del Sistema, Cliente
Función:	Permite la modificación de pacientes existentes.
Descripción:	El usuario del sistema puede modificar datos del pacientes, el sistema debe validar que la cédula exista:
Referencias:	

Nombre:	Manejo de Pacientes - EliminarPaciente
Alias:	
Actores:	Usuario del Sistema, Cliente
Función:	Permite eliminar un paciente existente.
Descripción:	El usuario del sistema puede eliminar pacientes, el sistema debe validar que la cédula exista y que el paciente no tenga citas pendientes. En este caso se debe eliminar primero la cita pendiente.
Referencias:	

Nombre:	Manejo de Citas - IngresarCita
Alias:	
Actores:	Usuario del Sistema, Cliente
Función:	Permite la creación de nuevas citas
Descripción:	<p>El usuario del sistema puede crear nuevas citas, ingresando todos sus datos requeridos, el sistema debe validar:</p> <ol style="list-style-type: none"> 1. Que se ingrese un motivo de la cita. 2. Que se ingrese un código para la cita, es generado por el sistema. 3. Se ingrese una fecha de registro, esta será tomada del sistema.. 4. Se ingrese la fecha en que se realizará la cita. 5. Se ingrese la hora de la cita. 6. Se ingrese la cédula del paciente. 7. Se ingrese el estatus de la cita, por defecto "abierta"
Referencias:	

Nombre:	Manejo de Citas - ModificarCita
Alias:	
Actores:	Usuario del Sistema, Cliente
Función:	Permite la modificación los datos de una cita existentes.
Descripción:	El usuario del sistema puede modificar datos de una cita.
Referencias:	

Nombre:	Manejo de Citas - EliminarCita
Alias:	
Actores:	Usuario del Sistema, Cliente
Función:	Permite eliminar una cita.
Descripción:	El usuario del sistema puede eliminar una cita, el sistema debe validar que la cita esta cerrada. En este caso se debe cambiar el estatus a cerrada antes de eliminar la cita.
Referencias:	

Nombre:	Manejo de Record - IngresarRecord
Alias:	
Actores:	Usuario del Sistema, Cliente
Función:	Permite la creación de nuevos record médicos
Descripción:	<p>El usuario del sistema puede crear nuevos record, ingresando todos sus datos requeridos, el sistema debe validar:</p> <ol style="list-style-type: none"> 1. Se genere un número de record automático. 2. Se ingrese un número de cédula para el paciente. 3. Se ingrese una fecha de registro, esta fecha es generada por el sistema.
Referencias:	

Nombre:	Manejo de Record - ModificarPecord
Alias:	
Actores:	Usuario del Sistema, Cliente
Función:	Permite la modificación los datos de records existentes.
Descripción:	El usuario del sistema puede modificar datos de un record.
Referencias:	

Nombre:	Manejo de Record - EliminarPecord
Alias:	
Actores:	Usuario del Sistema, Cliente
Función:	Permite eliminar un record.
Descripción:	El usuario del sistema puede eliminar un record.
Referencias:	

Nombre:	Manejo de Enfermedades - IngresarEnfermedades
Alias:	
Actores:	Usuario del Sistema, Cliente
Función:	Permite la creación de enfermedades en el catalogo
Descripción:	El usuario del sistema puede crear nuevos record, ingresando todos sus datos requeridos, el sistema debe validar: <ol style="list-style-type: none"> 1. Se genere un número de record automático. 2. Se ingrese un nombre de enfermedad.
Referencias:	

Nombre:	Manejo de Enfermedades - ModificarEnfermedades
Alias:	
Actores:	Usuario del Sistema, Cliente
Función:	Permite la modificación del nombre de una enfermedad.
Descripción:	El usuario del sistema puede modificar el nombre de una enfermedad existente.
Referencias:	

Nombre:	Manejo de Enfermedades - EliminarEnfermedades
Alias:	
Actores:	Usuario del Sistema, Cliente
Función:	Permite eliminar una enfermedad.
Descripción:	El usuario del sistema puede eliminar una enfermedad. El sistema debe validar que esta enfermedad no este en algún record medico. De ser el caso debe eliminarla antes de los record médicos.
Referencias:	

Nombre:	Manejo de Medicamentos - IngresarMedicamentos
Alias:	
Actores:	Usuario del Sistema, Cliente
Función:	Permite la creación un nuevo medicamento en el catalogo
Descripción:	El usuario del sistema puede crear nuevos record, ingresando todos sus datos requeridos, el sistema debe validar: <ol style="list-style-type: none"> 1. Se genere un número de record automático 2. Se ingrese un nombre del medicamento.
Referencias:	

Nombre:	Manejo de Medicamentos - ModificarMedicamentos
Alias:	
Actores:	Usuario del Sistema, Cliente
Función:	Permite la modificación del nombre de un medicamento.
Descripción:	El usuario del sistema puede modificar el nombre de un medicamento existente.
Referencias:	

Nombre:	Manejo de Medicamentos - EliminarMedicamentos
Alias:	
Actores:	Usuario del Sistema, Cliente
Función:	Permite eliminar un medicamento.
Descripción:	El usuario del sistema puede eliminar un medicamento. El sistema debe validar que el medicamento no este en algún record medico. De ser el caso debe eliminarla antes de los record médicos.
Referencias:	

Nombre:	Manejo de Alergias - IngresarAlergias
Alias:	
Actores:	Usuario del Sistema, Cliente
Función:	Permite la creación una nueva alergia en el catalogo
Descripción:	El usuario del sistema puede crear una nueva alergia, ingresando todos sus datos requeridos, el sistema debe validar: 1. Se genere un número de record automático 2. Se ingrese un nombre de alergia.
Referencias:	

Nombre:	Manejo de Alergias - ModificarAlergias
Alias:	
Actores:	Usuario del Sistema, Cliente
Función:	Permite la modificación del nombre de una alergia.
Descripción:	El usuario del sistema puede modificar el nombre de una alergia existente.
Referencias:	

Nombre:	Manejo de Alergias - EliminarAlergias
Alias:	
Actores:	Usuario del Sistema, Cliente
Función:	Permite eliminar una alergia.
Descripción:	El usuario del sistema puede eliminar una alergia. El sistema debe validar que la alergia no este en algún record medico. De ser el caso debe eliminarla antes de los record médicos.
Referencias:	

Nombre:	Manejo de Enfermedades Por Record - IngresarEnfermedadesRecord
Alias:	
Actores:	Usuario del Sistema, Cliente
Función:	Permite la creación una nueva alergia en el catalogo
Descripción:	El usuario del Sistema puede crear y asociar enfermedades con el record medico de un paciente. El sistema debe validar: 1. Se genere un número de record automático 2. Se ingrese una enfermedad.
Referencias:	

Nombre:	Manejo de Enfermedades Por Record - ModificarEnfermedadesRecord
Alias:	
Actores:	Usuario del Sistema, Cliente
Función:	Permite la modificación del nombre de la enfermedad o el comentario.
Descripción:	El usuario del sistema puede modificar el nombre de la enfermedad asociada o el comentario sobre ella.
Referencias:	

Nombre:	Manejo de Enfermedades Por Record - EliminarEnfermedadesRecord
Alias:	
Actores:	Usuario del Sistema, Cliente
Función:	Permite eliminar una enfermedad asociada.
Descripción:	El usuario del sistema puede eliminar una enfermedad asociada.
Referencias:	

Nombre:	Manejo de Pacientes - IngresarMedicamentosRecord
Alias:	
Actores:	Usuario del Sistema, Cliente
Función:	Permite la creación un nuevo medicamento asociado a un record medico.
Descripción:	El usuario del Sistema puede crear y asociar medicamentos que esta tomando un paciente con su record medico. El sistema debe validar: 1. Se genere un número de record automático 2. Se ingrese un medicamento.
Referencias:	

Nombre:	Manejo de Medicamentos Por Record - ModificarMedicamentosRecord
Alias:	
Actores:	Usuario del Sistema, Cliente
Función:	Permite la modificación del nombre de un medicamento asociado o el comentario.
Descripción:	El usuario del sistema puede modificar el nombre del medicamento asociada o el comentario sobre el.
Referencias:	

Nombre:	Manejo de Medicamentos Por Record - EliminarMedicamentosRecord
Alias:	
Actores:	Usuario del Sistema, Cliente
Función:	Permite eliminar un medicamento asociado.
Descripción:	El usuario del sistema puede eliminar un medicamento asociada.
Referencias:	

Nombre:	Manejo de Alergias Por Record - IngresoAlergiasRecord
Alias:	
Actores:	Usuario del Sistema, Cliente
Función:	Permite la creación una nueva alergia asociada a un record medico.
Descripción:	El usuario del Sistema puede crear y asociar alergia que padece un paciente con su record medico. El sistema debe validar: <ol style="list-style-type: none"> 1. Se genere un número de record automático 2. Se ingrese una alergia.
Referencias:	

Nombre:	Manejo de Alergias Por Record - ModificarAlergiasRecord
Alias:	
Actores:	Usuario del Sistema, Cliente
Función:	Permite la modificación del nombre de una alergia asociada o el comentario.
Descripción:	El usuario del sistema puede modificar el nombre de una alergia asociada o el comentario sobre el.
Referencias:	

Nombre:	Manejo de Alergias Por Record - EliminarAlergiasRecord
Alias:	
Actores:	Usuario del Sistema, Cliente
Función:	Permite eliminar una alergia asociado.
Descripción:	El usuario del sistema puede eliminar una alergia asociada.
Referencias:	

Nombre:	Imprimir - Record de pacientes
Alias:	
Actores:	Usuario del Sistema, Cliente
Función:	Permite la consulta o informe del record medico de un paciente.
Descripción:	El usuario del sistema puede visualizar el record medico de un paciente o puede imprimirlo.
Referencias:	

Nombre:	Imprimir - Citas por fecha
Alias:	
Actores:	Usuario del Sistema, Cliente
Función:	Permite la consulta o informe de las citas medicas reservadas para un intervalo de fechas.
Descripción:	El usuario del sistema puede visualizar las citas medicas reservadas en un intervalo de fechas.
Referencias:	

Nombre:	Imprimir - Citas por paciente por fecha
Alias:	
Actores:	Usuario del Sistema, Cliente
Función:	Permite la consulta o informe de las citas medicas reservadas por un paciente en un intervalo de fechas.
Descripción:	Permite la consulta o informe de las citas medicas reservadas por un paciente en un intervalo de fechas.
Referencias:	

Flujo Básico y Flujo alterno de los Sub Casos de Uso

Flujo Básico	
Nombre del Caso de Uso:	Manejo de Pacientes - IngresarPaciente
Alias:	
Actores:	Usuario del Sistema, Cliente
Usuario del Sistema	Respuesta del Sistema
1. Digita cedula	
	2. Buscar paciente por cedula
3. Digita los datos	
	5. Valida datos: <ul style="list-style-type: none"> 1. Que se ingrese una cédula. 2. Que se ingrese el primer nombre y el primer apellido. 3. Se asigne un Sexo. 4. Se ingrese la fecha de nacimiento del paciente. 5. Se ingrese un teléfono de contacto. 6. Se ingrese la fecha de registro, esta será tomada de la fecha del sistema.
6. Clic en el botón grabar	
	7. Graba datos en el sistema.
Flujo alternos	
Nombre del Caso de Uso:	Manejo de Pacientes – IngresarPaciente
Alias:	
Actores:	Usuario del Sistema, Cliente
Usuario del Sistema	Respuesta del Sistema
línea 1	Si el paciente ya existe lo informa y muestra datos.
línea 3	Si algún dato obligatorio esta vacío lo indica.

Flujo Básico	
Nombre del Caso de Uso:	Manejo de Pacientes - ModificarPaciente
Alias:	
Actores:	Usuario del Sistema, Cliente
Usuario del Sistema	Respuesta del Sistema
1. Digita cedula	
	2. Buscar paciente por cedula
3. Digita los datos	
	5. Valida datos: <ul style="list-style-type: none"> 1. Que se ingrese una cédula. 2. Que se ingrese el primer nombre y el primer apellido. 3. Se asigne un Sexo. 4. Se ingrese la fecha de nacimiento del paciente. 5. Se ingrese un teléfono de contacto. 6. Se ingrese la fecha de registro, esta será tomada de la fecha del sistema.
6. Clic en el botón grabar	
	7. Graba datos en el sistema.
Flujo alternos	
Nombre del Caso de Uso:	Manejo de Pacientes - ModificarPaciente
Alias:	
Actores:	Usuario del Sistema, Cliente
Usuario del Sistema	Respuesta del Sistema
línea 1	Si el paciente no existe lo informa.
línea 3	Si algún dato obligatorio esta vacío lo indica.

Flujo Básico	
Nombre del Caso de Uso:	Manejo de Pacientes - EliminarPaciente
Alias:	
Actores:	Usuario del Sistema, Cliente
Usuario del Sistema	Respuesta del Sistema
1. Digita cedula	
	2. Buscar paciente por cedula
3. Clic en el botón borrar.	
	4. elimina los datos en el sistema.
Flujo alternos	
Nombre del Caso de Uso:	Manejo de Pacientes - EliminarPaciente
Alias:	
Actores:	Usuario del Sistema, Cliente
Usuario del Sistema	Respuesta del Sistema
línea 1	Si el paciente no existe lo informa.

Flujo Básico	
Nombre del Caso de Uso:	Manejo de Citas - IngresarCita
Alias:	
Actores:	Usuario del Sistema, Cliente
Usuario del Sistema	Respuesta del Sistema
1. Digita los datos	
	2. Valida datos: <ol style="list-style-type: none"> 1. Que se ingrese un motivo de la cita. 2. Que se ingrese un código para la cita, es generado por el sistema. 3. Se ingrese una fecha de registro, esta será tomada del sistema... 4. Se ingrese la fecha en que se realizará la cita. 5. Se ingrese la hora de la cita. 6. Se ingrese la cédula del paciente. 7. Se ingrese el estatus de la cita, por defecto "abierta"
3. Clic en el botón grabar	
	4. Graba datos en el sistema.
Flujo alternos	
Nombre del Caso de Uso:	Manejo de Citas - IngresarCita
Alias:	
Actores:	Usuario del Sistema, Cliente
Usuario del Sistema	Respuesta del Sistema
línea 4	El sistema valida que no exista otra cita en esa fecha y hora. El sistema valida que el paciente exista. Si algún dato obligatorio esta vacío lo indica.
línea 6	Si aparece algún error en la grabación lo informa y cancela la operación.

Flujo Básico	
Nombre del Caso de Uso:	Manejo de Citas – ModificarCita
Alias:	
Actores:	Usuario del Sistema, Cliente
Usuario del Sistema	Respuesta del Sistema
1. El usuario selecciona la opción modificar cita.	
	2. El sistema captura la selección.
	3. Muestra la pantalla para modificar datos.
4. Ingresa el código de la cita que desea modificar.	
	5. Busca la cita.
6. Modifica los datos.	
	7. Valida nuevos datos.
8. Clic en el botón grabar.	
	9. Graba datos en el sistema.
Flujo alternos	
Nombre del Caso de Uso:	Manejo de Citas – ModificarCita
Alias:	
Actores:	Usuario del Sistema, Cliente
Usuario del Sistema	Respuesta del Sistema
línea 4	Si la cita no existe lo informa y permite intentarlo de nuevo.
Línea 6	Si algún dato obligatorio esta vacío lo indica.
Línea 8	Si aparece algún error en la modificación lo informa y cancela la operación.

Flujo Básico	
Nombre del Caso de Uso:	Manejo de Citas - EliminarCita
Alias:	
Actores:	Usuario del Sistema, Cliente
Usuario del Sistema	Respuesta del Sistema
1. El usuario selecciona la opción eliminar cita.	
	2. El sistema captura la selección.
	3. Muestra la pantalla para eliminar datos.
4. Ingresa el código de la cita que desea eliminar.	
	5. Busca la cita.
6. Clic en el botón borrar.	
	7. elimina los datos en el sistema.
Flujo alternos	
Nombre del Caso de Uso:	Manejo de Citas - EliminarCita
Alias:	
Actores:	Usuario del Sistema, Cliente
Usuario del Sistema	Respuesta del Sistema
línea 4	Si la cita no existe lo informa y permite intentarlo de nuevo. Si la cita esta abierta no permite eliminarla. Lo informa.
línea 6	Si aparece algún error en la eliminación lo informa y cancela la operación.

Flujo Básico	
Nombre del Caso de Uso:	Manejo de Record – IngresarRecord
Alias:	
Actores:	Usuario del Sistema, Cliente
Usuario del Sistema	Respuesta del Sistema
1. El usuario selecciona la opción ingresar record.	
	2. El sistema captura la selección.
	3. Muestra la pantalla de captura de datos.
4. Digita los datos los datos del record.	
	5. Valida datos del nuevo record.
6. Clic en el botón grabar	
	7. Graba datos en el sistema.
Flujo alternos	
Nombre del Caso de Uso:	Manejo de Record – IngresarRecord
Alias:	
Actores:	Usuario del Sistema, Cliente
Usuario del Sistema	Respuesta del Sistema
línea 4	El sistema valida que el paciente no tenga ya un record creado. De ser el caso lo Informa. El sistema valida que el paciente exista. Informa. Si algún dato obligatorio esta vacío lo indica.
Línea 6	Si aparece algún error en la grabación lo informa y cancela la operación.

Flujo Básico	
Nombre del Caso de Uso:	Manejo de Record - ModificarPecord
Alias:	
Actores:	Usuario del Sistema, Cliente
Usuario del Sistema	Respuesta del Sistema
1. El usuario selecciona la opción modificar record.	
	2. El sistema captura la selección.
	3. Muestra la pantalla para modificar datos.
4. Ingresa el código del record que desea modificar.	
	5. Busca el record.
6. Modifica los datos.	
	7. Valida nuevos datos.
8. Clic en el botón grabar.	
	9. Graba datos en el sistema.
Flujo alternos	
Nombre del Caso de Uso:	Manejo de Record - ModificarPecord
Alias:	
Actores:	Usuario del Sistema, Cliente
Usuario del Sistema	Respuesta del Sistema
línea 4	Si el record no existe lo informa y permite intentarlo de nuevo.
línea 6	Si algún dato obligatorio esta vacío lo indica.
línea 8	Si aparece algún error en la modificación lo informa y cancela la operación.

Flujo Básico	
Nombre del Caso de Uso:	Manejo de Record – EliminarPecord
Alias:	
Actores:	Usuario del Sistema, Cliente
Usuario del Sistema	Respuesta del Sistema
1. El usuario selecciona la opción eliminar record.	
	2. El sistema captura la selección.
	3. Muestra la pantalla para eliminar datos.
4. Ingresa el código del record que desea eliminar.	
	5. Busca el record.
6. Clic en el botón borrar.	
	7. elimina los datos en el sistema.
Flujo alternos	
Nombre del Caso de Uso:	Manejo de Record – EliminarPecord
Alias:	
Actores:	Usuario del Sistema, Cliente
Usuario del Sistema	Respuesta del Sistema
línea 4	Si el record no existe lo informa y permite intentarlo de nuevo.
Línea 6	Si aparece algún error en la eliminación lo informa y cancela la operación.

Flujo Básico	
Nombre del Caso de Uso:	Manejo de Enfermedades - IngresarEnfermedades
Alias:	
Actores:	Usuario del Sistema, Cliente
Usuario del Sistema	Respuesta del Sistema
1. El usuario selecciona la opción ingresar enfermedad.	
	2. El sistema captura la selección.
	3. Muestra la pantalla de captura de datos.
4. Digita los datos los datos de la enfermedad.	
	5. Valida datos de la nueva enfermedad.
6. Clic en el botón grabar	
	7. Graba datos en el sistema.
Flujo alternos	
Nombre del Caso de Uso:	Manejo de Enfermedades - IngresarEnfermedades
Alias:	
Actores:	Usuario del Sistema, Cliente
Usuario del Sistema	Respuesta del Sistema
línea 4	Si algún dato obligatorio esta vacío lo indica.
línea 6	Si aparece algún error en la grabación lo informa y cancela la operación.

Flujo Básico	
Nombre del Caso de Uso:	Manejo de Enfermedades - ModificarEnfermedades
Alias:	
Actores:	Usuario del Sistema, Cliente
Usuario del Sistema	Respuesta del Sistema
1. El usuario selecciona la opción modificar enfermedad.	
	2. El sistema captura la selección.
	3. Muestra la pantalla para modificar datos.
4. Ingresar el código de la enfermedad que desea modificar.	
	5. Busca la enfermedad.
6. Modifica los datos.	
	7. Valida nuevos datos.
8. Clic en el botón grabar.	
	9. Graba datos en el sistema.
Flujo alternos	
Nombre del Caso de Uso:	Manejo de Enfermedades - ModificarEnfermedades
Alias:	
Actores:	Usuario del Sistema, Cliente
Usuario del Sistema	Respuesta del Sistema
línea 4	Si la enfermedad no existe lo informa y permite intentarlo de nuevo.
línea 6	Si algún dato obligatorio esta vacío lo indica.
línea 8	Si aparece algún error en la modificación lo informa y cancela la operación.

Flujo Básico	
Nombre del Caso de Uso:	Manejo de Enfermedades – EliminarEnfermedades
Alias:	
Actores:	Usuario del Sistema, Cliente
Usuario del Sistema	Respuesta del Sistema
1. El usuario selecciona la opción eliminar enfermedad.	
	2. El sistema captura la selección.
	3. Muestra la pantalla para eliminar datos.
4. Ingresa el código de la enfermedad que desea eliminar.	
	5. Busca la enfermedad.
6. Clic en el botón borrar.	
	7. elimina los datos en el sistema.

Flujo alternos	
Nombre del Caso de Uso:	Manejo de Enfermedades – EliminarEnfermedades
Alias:	
Actores:	Usuario del Sistema, Cliente
Usuario del Sistema	Respuesta del Sistema
línea 4	Si la enfermedad no existe lo informa y permite intentarlo de nuevo. El sistema valida que la enfermedad no este en algún record medico. De ser el caso debe eliminarla antes de los record médicos. El sistema Informa.
Línea 6	Si aparece algún error en la eliminación lo informa y cancela la operación.

Flujo Básico	
Nombre del Caso de Uso:	Manejo de Medicamentos - IngresarMedicamentos
Alias:	
Actores:	Usuario del Sistema, Cliente
Usuario del Sistema	Respuesta del Sistema
1. El usuario selecciona la opción ingresar Medicamento.	
	2. El sistema captura la selección.
	3. Muestra la pantalla de captura de datos.
4. Digita los datos los datos del Medicamento.	
	5. Valida datos del nuevo Medicamento.
6. Clic en el botón grabar	
	7. Graba datos en el sistema.

Flujo alternos	
Nombre del Caso de Uso:	Manejo de Medicamentos - IngresarMedicamentos
Alias:	
Actores:	Usuario del Sistema, Cliente
Usuario del Sistema	Respuesta del Sistema
línea 4	Si algún dato obligatorio esta vacío lo indica.
línea 6	Si aparece algún error en la grabación lo informa y cancela la operación.

Flujo Básico	
Nombre del Caso de Uso:	Manejo de Medicamentos - ModificarMedicamentos
Alias:	
Actores:	Usuario del Sistema, Cliente
Usuario del Sistema	Respuesta del Sistema
1. El usuario selecciona la opción modificar Medicamento.	
	2. El sistema captura la selección.
	3. Muestra la pantalla para modificar datos.
4. Ingresar el código del Medicamento que desea modificar.	
	5. Busca el Medicamento.
6. Modifica los datos.	
	7. Valida nuevos datos.
8. Clic en el botón grabar.	
	9. Graba datos en el sistema.
Flujo alternos	
Nombre del Caso de Uso:	Manejo de Medicamentos - ModificarMedicamentos
Alias:	
Actores:	Usuario del Sistema, Cliente
Usuario del Sistema	Respuesta del Sistema
línea 4	Si el Medicamento no existe lo informa y permite intentarlo de nuevo.
línea 6	Si algún dato obligatorio esta vacío lo indica.
línea 8	Si aparece algún error en la modificación lo informa y cancela la operación.

Flujo Básico	
Nombre del Caso de Uso:	Manejo de Medicamentos - EliminarMedicamentos
Alias:	
Actores:	Usuario del Sistema, Cliente
Usuario del Sistema	Respuesta del Sistema
1. El usuario selecciona la opción eliminar Medicamento.	
	2. El sistema captura la selección.
	3. Muestra la pantalla para eliminar datos.
4. Ingresar el código del Medicamento que desea eliminar.	
	5. Busca el Medicamento.
6. Clic en el botón borrar.	
	7. elimina los datos en el sistema.

Flujo alternos	
Nombre del Caso de Uso:	Manejo de Medicamentos - EliminarMedicamentos
Alias:	
Actores:	Usuario del Sistema, Cliente
Usuario del Sistema	Respuesta del Sistema
línea 4	Si el Medicamento no existe lo informa y permite intentarlo de nuevo. El sistema valida que el medicamento no este en algún record medico. De ser el caso debe eliminarla antes de los record médicos. El sistema Informa.
línea 6	Si aparece algún error en la eliminación lo informa y cancela la operación.

Flujo Básico	
Nombre del Caso de Uso:	Manejo de Alergias – IngresarAlergias
Alias:	
Actores:	Usuario del Sistema, Cliente
Usuario del Sistema	Respuesta del Sistema
1. El usuario selecciona la opción ingresar alergia.	
	2. El sistema captura la selección.
	3. Muestra la pantalla de captura de datos.
4. Digita los datos los datos de la alergia.	
	5. Valida datos de la nueva alergia.
6. Clic en el botón grabar	
	7. Graba datos en el sistema.

Flujo alternos	
Nombre del Caso de Uso:	Manejo de Alergias – Ingresaralergias
Alias:	
Actores:	Usuario del Sistema, Cliente
Usuario del Sistema	Respuesta del Sistema
línea 4	Si algún dato obligatorio esta vacío lo indica.
Línea 6	Si aparece algún error en la grabación lo informa y cancela la operación.

Flujo Básico	
Nombre del Caso de Uso:	Manejo de Alergias - ModificarAlergias
Alias:	
Actores:	Usuario del Sistema, Cliente
Usuario del Sistema	Respuesta del Sistema
1. El usuario selecciona la opción modificar alergia.	
	2. El sistema captura la selección.
	3. Muestra la pantalla para modificar datos.
4. Ingresa el código de la alergia que desea modificar.	
	5. Busca la alergia.
6. Modifica los datos.	
	7. Valida nuevos datos.
8. Clic en el botón grabar.	
	9. Graba datos en el sistema.
Flujo alternos	
Nombre del Caso de Uso:	Manejo de Alergias - ModificarAlergias
Alias:	
Actores:	Usuario del Sistema, Cliente
Usuario del Sistema	Respuesta del Sistema
línea 4	Si la alergia no existe lo informa y permite intentarlo de nuevo.
línea 6	Si algún dato obligatorio esta vacío lo indica.
línea 8	Si aparece algún error en la modificación lo informa y cancela la operación.

Flujo Básico	
Nombre del Caso de Uso:	Manejo de Alergias - EliminarAlergias
Alias:	
Actores:	Usuario del Sistema, Cliente
Usuario del Sistema	Respuesta del Sistema
1. El usuario selecciona la opción eliminar alergia.	
	2. El sistema captura la selección.
	3. Muestra la pantalla para eliminar datos.
4. Ingresa el código de la alergia que desea eliminar.	
	5. Busca la alergia.
6. Clic en el botón borrar.	
	7. elimina los datos en el sistema.
Flujo alternos	
Nombre del Caso de Uso:	Manejo de Alergias - Eliminaralergiaes
Alias:	
Actores:	Usuario del Sistema, Cliente
Usuario del Sistema	Respuesta del Sistema
línea 4	Si la alergia no existe lo informa y permite intentarlo de nuevo. El sistema valida que la alergia no este en algún record medico. De ser el caso debe eliminarla antes de los record médicos. El sistema Informa.
línea 6	Si aparece algún error en la eliminación lo informa y cancela la operación.

Flujo Básico	
Nombre del Caso de Uso:	Manejo de Enfermedades Por Record – IngresarEnfermedadesRecord
Alias:	
Actores:	Usuario del Sistema, Cliente
Usuario del Sistema	Respuesta del Sistema
1. El usuario selecciona la opción ingresar enfermedad al record.	
	2. El sistema captura la selección.
	3. Muestra la pantalla de captura de datos.
4. Digita los datos los datos de la enfermedad.	
	5. Valida datos de la nueva enfermedad asociada.
6. Clic en el botón grabar	
	7. Graba datos en el sistema.
Flujo alternos	
Nombre del Caso de Uso:	Manejo de Enfermedades Por Record – IngresarEnfermedadesRecord
Alias:	
Actores:	Usuario del Sistema, Cliente
Usuario del Sistema	Respuesta del Sistema
línea 4	Si algún dato obligatorio esta vacío lo indica.
Línea 6	Si aparece algún error en la grabación lo informa y cancela la operación.

Flujo Básico	
Nombre del Caso de Uso:	Manejo de Enfermedades Por Record - ModificarEnfermedadesRecord
Alias:	
Actores:	Usuario del Sistema, Cliente
Usuario del Sistema	Respuesta del Sistema
1. El usuario selecciona la opción modificar la enfermedad asociada.	
	2. El sistema captura la selección.
	3. Muestra la pantalla para modificar datos.
4. Modifica los datos.	
	5. Valida nuevos datos.
6. Clic en el botón grabar.	
	7. Graba datos en el sistema.
Flujo alternos	
Nombre del Caso de Uso:	Manejo de Enfermedades Por Record - ModificarEnfermedadesRecord
Alias:	
Actores:	Usuario del Sistema, Cliente
Usuario del Sistema	Respuesta del Sistema
línea 4	Si algún dato obligatorio esta vacío lo indica.
línea 6	Si aparece algún error en la modificación lo informa y cancela la operación.

Flujo Básico	
Nombre del Caso de Uso:	Manejo de Enfermedades Por Record - EliminarEnfermedadesRecord
Alias:	
Actores:	Usuario del Sistema, Cliente
Usuario del Sistema	Respuesta del Sistema
1. El usuario selecciona la opción eliminar enfermedad asociada.	
	2. El sistema captura la selección.
	3. Muestra la pantalla para eliminar datos.
4. Clic en el botón borrar.	
	5. elimina los datos en el sistema.
Flujo alternos	
Nombre del Caso de Uso:	Manejo de Enfermedades Por Record - EliminarEnfermedadesRecord
Alias:	
Actores:	Usuario del Sistema, Cliente
Usuario del Sistema	Respuesta del Sistema
línea 4	Si aparece algún error en la eliminación lo informa y cancela la operación.

Flujo Básico	
Nombre del Caso de Uso:	Manejo de Medicamentos Por Record – IngresarMedicamentosRecord
Alias:	
Actores:	Usuario del Sistema, Cliente
Usuario del Sistema	Respuesta del Sistema
1. El usuario selecciona la opción ingresar Medicamento al record.	
	2. El sistema captura la selección.
	3. Muestra la pantalla de captura de datos.
4. Digita los datos los datos del Medicamento.	
	5. Valida datos del nuevo Medicamento asociada.
6. Clic en el botón grabar	
	7. Graba datos en el sistema.
Flujo alternos	
Nombre del Caso de Uso:	Manejo de Medicamentos Por Record – IngresarMedicamentosRecord
Alias:	
Actores:	Usuario del Sistema, Cliente
Usuario del Sistema	Respuesta del Sistema
línea 4	Si algún dato obligatorio esta vacío lo indica.
Línea 6	Si aparece algún error en la grabación lo informa y cancela la operación.

Flujo Básico	
Nombre del Caso de Uso:	Manejo de Medicamentos Por Record - ModificarMedicamentosRecord
Alias:	
Actores:	Usuario del Sistema, Cliente
Usuario del Sistema	Respuesta del Sistema
1. El usuario selecciona la opción modificar Medicamento asociada.	
	2. El sistema captura la selección.
	3. Muestra la pantalla para modificar datos.
4. Modifica los datos.	
	5. Valida nuevos datos.
6. Clic en el botón grabar.	
	7. Graba datos en el sistema.
Flujo alternos	
Nombre del Caso de Uso:	Manejo de Medicamentos Por Record - ModificarMedicamentosRecord
Alias:	
Actores:	Usuario del Sistema, Cliente
Usuario del Sistema	Respuesta del Sistema
línea 4	Si algún dato obligatorio esta vacío lo indica.
línea 6	Si aparece algún error en la modificación lo informa y cancela la operación.

Flujo Básico	
Nombre del Caso de Uso:	Manejo de Medicamentos Por Record - EliminarMedicamentosRecord
Alias:	
Actores:	Usuario del Sistema, Cliente
Usuario del Sistema	Respuesta del Sistema
1. El usuario selecciona la opción eliminar Medicamento asociada.	
	2. El sistema captura la selección.
	3. Muestra la pantalla para eliminar datos.
4. Clic en el botón borrar.	
	5. elimina los datos en el sistema.
Flujo alternos	
Nombre del Caso de Uso:	Manejo de Medicamentos Por Record - EliminarMedicamentosRecord
Alias:	
Actores:	Usuario del Sistema, Cliente
Usuario del Sistema	Respuesta del Sistema
línea 4	Si aparece algún error en la eliminación lo informa y cancela la operación.

Flujo Básico	
Nombre del Caso de Uso:	Manejo de Alergias Por Record - IngresarAlergiasRecord
Alias:	
Actores:	Usuario del Sistema, Cliente
Usuario del Sistema	Respuesta del Sistema
1. El usuario selecciona la opción ingresar Alergia al record.	
	2. El sistema captura la selección.
	3. Muestra la pantalla de captura de datos.
4. Digita los datos los datos de la Alergia.	
	5. Valida datos de la nueva Alergia asociada.
6. Clic en el botón grabar	
	7. Graba datos en el sistema.
Flujo alternos	
Nombre del Caso de Uso:	Manejo de Alergias Por Record - IngresarAlergiasRecord
Alias:	
Actores:	Usuario del Sistema, Cliente
Usuario del Sistema	Respuesta del Sistema
línea 4	Si algún dato obligatorio esta vacío lo indica.
línea 6	Si aparece algún error en la grabación lo informa y cancela la operación.

Flujo Básico	
Nombre del Caso de Uso:	Manejo de Alergias Por Record – ModificarAlergiasRecord
Alias:	
Actores:	Usuario del Sistema, Cliente
Usuario del Sistema	Respuesta del Sistema
1. El usuario selecciona la opción modificar Alergia asociada.	
	2. El sistema captura la selección.
	3. Muestra la pantalla para modificar datos.
4. Modifica los datos.	
	5. Valida nuevos datos.
6. Clic en el botón grabar.	
	7. Graba datos en el sistema.
Flujo alternos	
Nombre del Caso de Uso:	Manejo de Alergias Por Record – ModificarAlergiasRecord
Alias:	
Actores:	Usuario del Sistema, Cliente
Usuario del Sistema	Respuesta del Sistema
línea 4	Si algún dato obligatorio esta vacío lo indica.
Línea 6	Si aparece algún error en la modificación lo informa y cancela la operación.

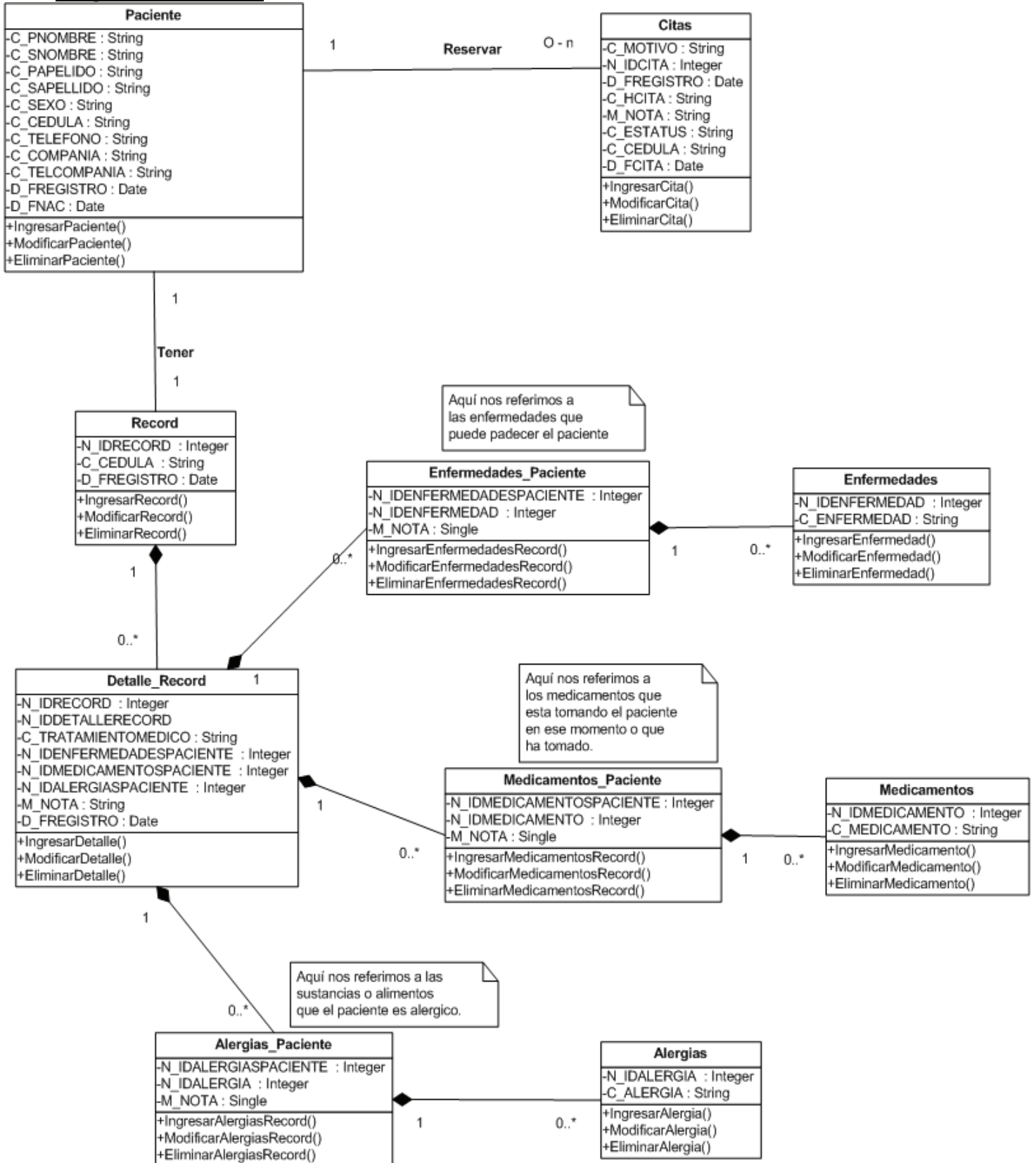
Flujo Básico	
Nombre del Caso de Uso:	Manejo de Alergias Por Record - EliminarAlergiasRecord
Alias:	
Actores:	Usuario del Sistema, Cliente
Usuario del Sistema	Respuesta del Sistema
1. El usuario selecciona la opción eliminar Alergia asociada.	
	2. El sistema captura la selección.
	3. Muestra la pantalla para eliminar datos.
4. Clic en el botón borrar.	
	5. elimina los datos en el sistema.
Flujo alternos	
Nombre del Caso de Uso:	Manejo de Alergias Por Record - EliminarAlergiasRecord
Alias:	
Actores:	Usuario del Sistema, Cliente
Usuario del Sistema	Respuesta del Sistema
línea 4	Si aparece algún error en la eliminación lo informa y cancela la operación.

Formato de eventos	
Nombre del Caso de Uso:	Imprimir – Record de pacientes
Alias:	
Actores:	Usuario del Sistema
Usuario del Sistema	Respuesta del Sistema
1. El usuario selecciona la opción Informe Record de pacientes	
	2. El sistema captura la selección.
	3. Muestra la pantalla para capturar el código del paciente.
4. Ingresa código de paciente	
	5. Captura código del paciente
6. Clic en el botón imprimir.	
	7. muestra en pantalla el record del paciente con opción a imprimir.
Formato de eventos alternos	
Nombre del Caso de Uso:	Imprimir – Record de pacientes
Alias:	
Actores:	Usuario del Sistema, Cliente
Usuario del Sistema	Respuesta del Sistema
línea 4	Si no existen datos lo Informa.
Línea 6	Si aparece algún error en la impresión lo informa y cancela la operación.

Flujo Básico	
Nombre del Caso de Uso:	Imprimir - Citas por fecha
Alias:	
Actores:	Usuario del Sistema
Usuario del Sistema	Respuesta del Sistema
1. El usuario selecciona la opción Informe Citas por fecha.	
	2. El sistema captura la selección.
	3. Muestra la pantalla para capturar la fecha inicial y la fecha final para el informe.
4. Ingresar fecha inicial y fecha final	
	5. Captura fecha inicial y fecha final
6. Clic en el botón imprimir.	
	7. muestra en pantalla las citas en el intervalo de fechas indicado con opción a imprimir.
Flujo alternos	
Nombre del Caso de Uso:	Imprimir - Citas por fecha
Alias:	
Actores:	Usuario del Sistema, Cliente
Usuario del Sistema	Respuesta del Sistema
línea 4	Si no existen datos lo Informa.
línea 6	Si aparece algún error en la impresión lo informa y cancela la operación.

Flujo Básico	
Nombre del Caso de Uso:	Imprimir - Citas por paciente por fecha
Alias:	
Actores:	Usuario del Sistema
Usuario del Sistema	Respuesta del Sistema
1. El usuario selecciona la opción Informe Citas por fecha.	
	2. El sistema captura la selección.
	3. Muestra la pantalla para capturar la fecha inicial, la fecha final y el código del paciente para el informe.
4. Ingresar fecha inicial, fecha final y código del paciente.	
	5. Captura fecha inicial, fecha final y el código del paciente
6. Clic en el botón imprimir.	
	7. muestra en pantalla las citas en el intervalo de fechas indicado para el paciente seleccionado con opción a imprimir.
Flujo alternos	
Nombre del Caso de Uso:	Imprimir - Citas por paciente por fecha
Alias:	
Actores:	Usuario del Sistema, Cliente
Usuario del Sistema	Respuesta del Sistema
línea 4	Si no existen datos lo Informa.
línea 6	Si aparece algún error en la impresión lo informa y cancela la operación.

Diagrama de Clases



Diagramas de Interacción de los Sub Casos de Uso

Diagrama de Secuencia

Caso de Uso IngresarPaciente

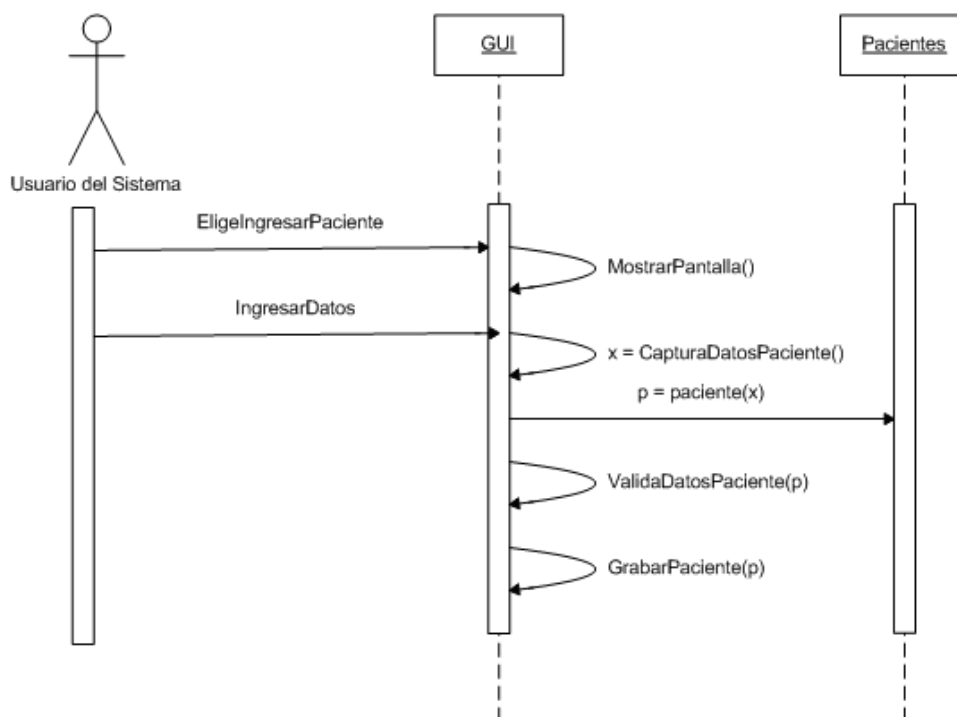


Diagrama de Colaboración

Caso de Uso IngresarPaciente

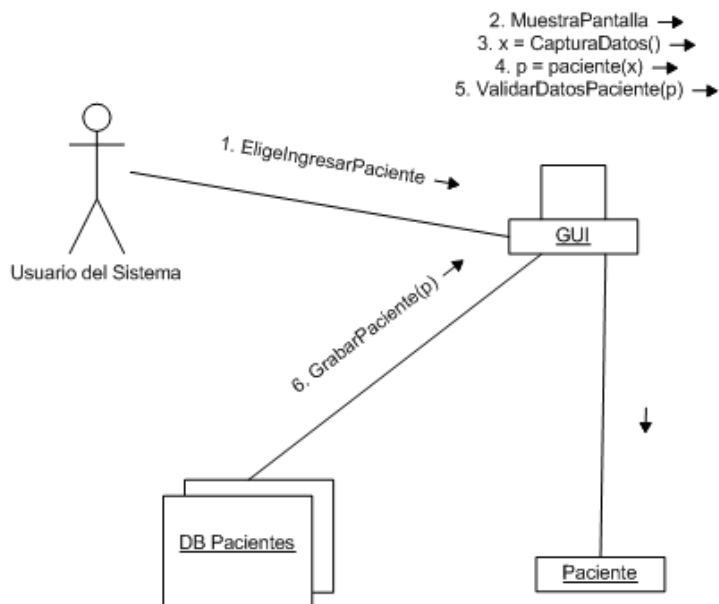


Diagrama de Secuencia

Caso de Uso ModificarPaciente

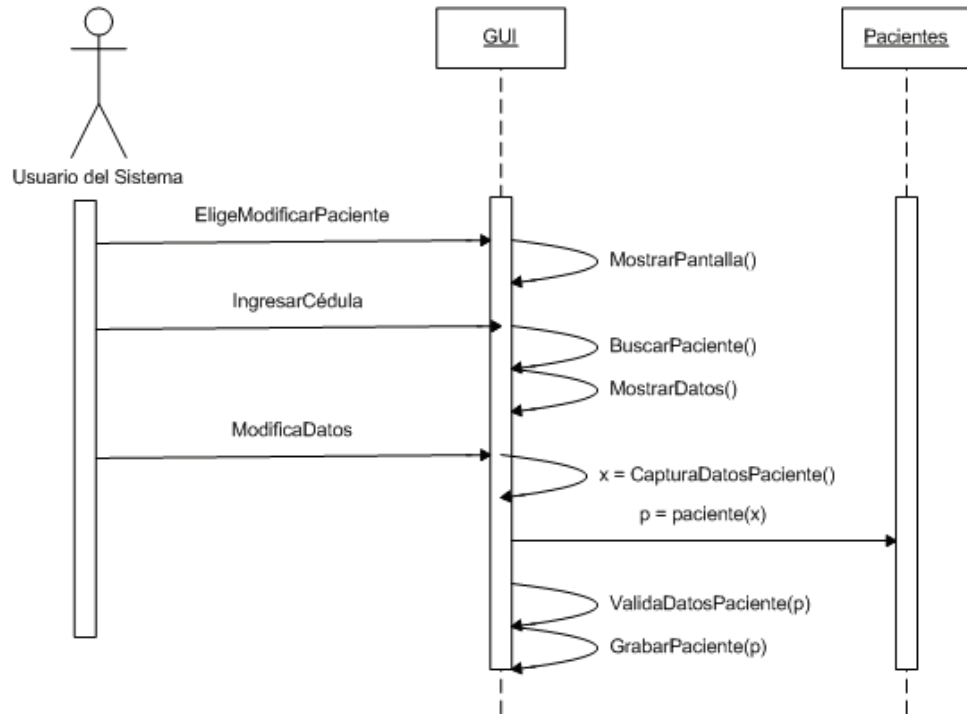


Diagrama de Colaboración

Caso de Uso ModificarPaciente

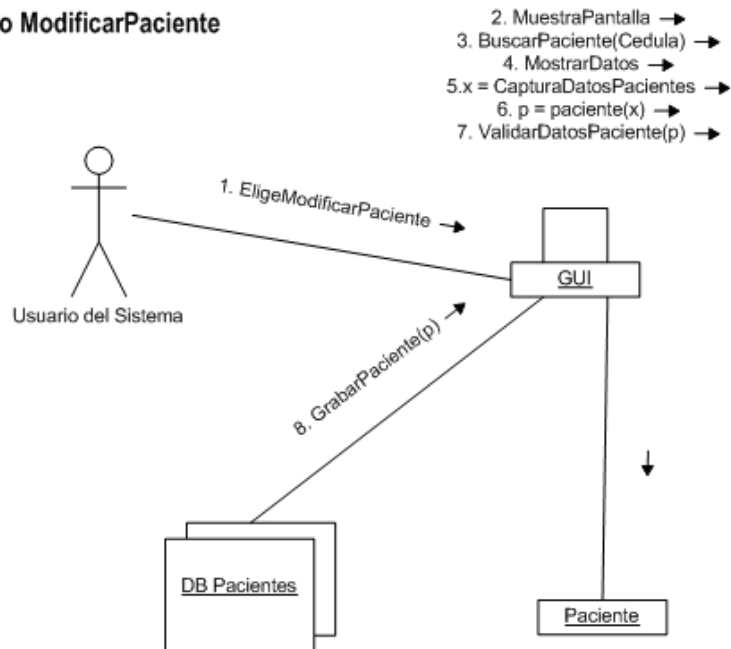
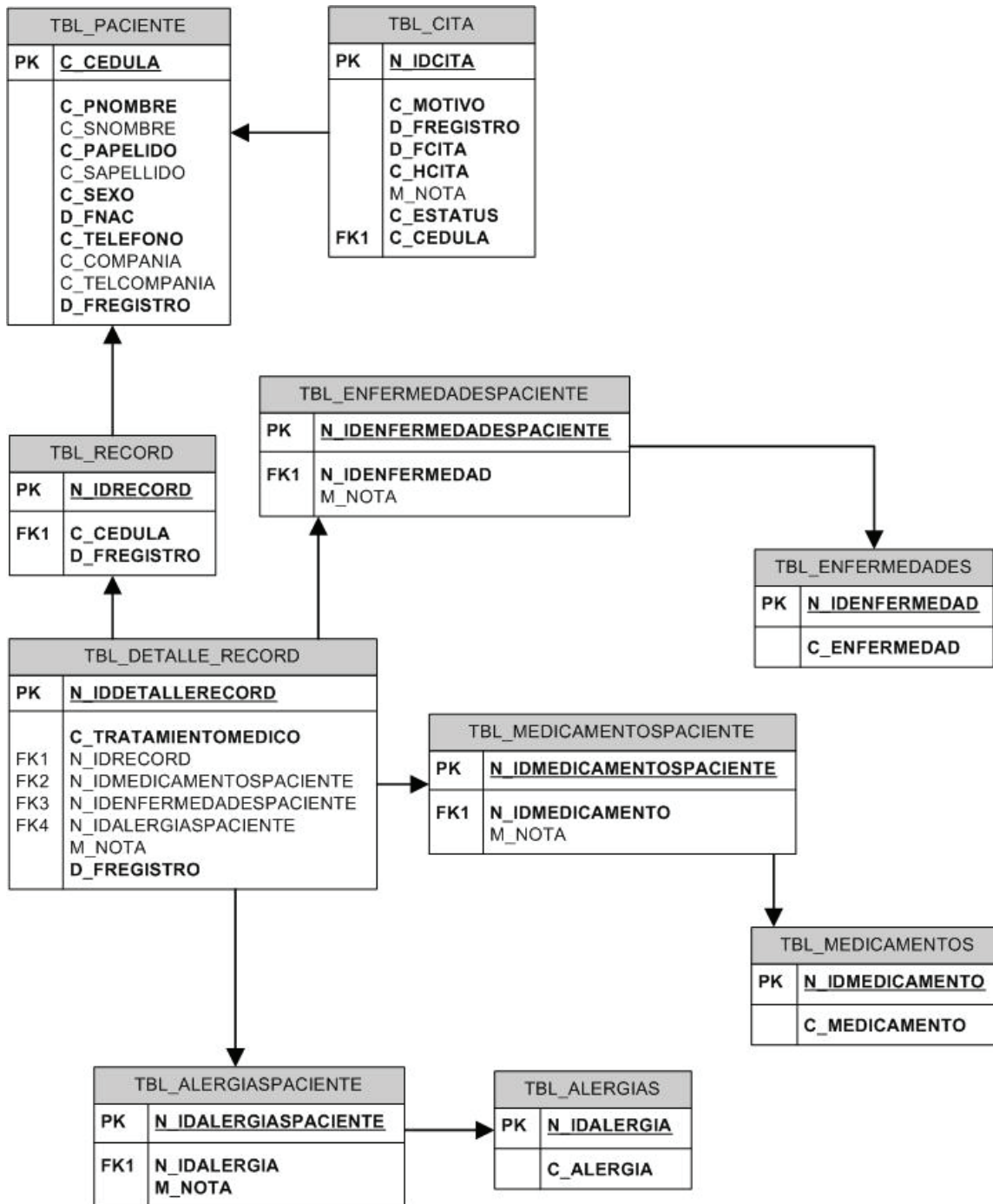


Diagrama de Base de Datos



Diccionario de datos

TBL_PACIENTE: En esta tabla se guardan los registros de los datos personales de los pacientes.

CAMPO	TIPO	LONGITUD	REQUERIDO	DESCRIPCION
C_PNOMBRE	CARACTER	20	SI	PRIMER NOMBRE DEL PACIENTE
C_SNOMBRE	CARACTER	20	NO	SEGUNDO NOMBRE DEL PACIENTE
C_PAPellido	CARACTER	20	SI	PRIMER APELLIDO DEL PACIENTE
C_SAPellido	CARACTER	20	NO	SEGUNDO APELLIDO DEL PACIENTE
C_SEXO	CARACTER	1	SI	SEXO DEL PACIENTE
D_FNAC	FECHA	10	SI	FECHA DE NACIMIENTO DEL PACIENTE
C_CEDULA(PK)	CARACTER	11	SI	CEDULA DEL PACIENTE
C_TELEFONO	CARACTER	10	SI	TELEFONO DE CONTACTO DEL PACIENTE
C_COMPANIA	CARACTER	40	NO	COMPAÑÍA DONDE LABORA EL PACIENTE
C_TELCOMPANIA	CARACTER	10	NO	TELEFONO DE LA COMPAÑÍA DONDE LABORA EL PACIENTE
D_FREGISTRO	FECHA	10	SI	FECHA EN LA QUE SE REGISTRARON LOS DATOS DEL PACIENTE

TBL_CITA: En esta tabla se guardan los registros de los datos de las citas abiertas.

CAMPO	TIPO	LONGITUD	REQUERIDO	DESCRIPCION
C_MOTIVO	CARÁCTER	80	SI	MOTIVO DE LA CITA
N_IDCITA(PK)	NUMERO	10	SI	CLAVE AUTONUMERICA QUE GENERA LOS NUMEROS DE LAS CITAS
D_FREGISTRO	FECHA	10	SI	FECHA EN LA QUE REGISTRO LA CITA EN EL SISTEMA
D_FCITA	FECHA	10	SI	FECHA PARA LA CUAL SE ACORDO LA CITA
C_HCITA	CARÁCTER	5	SI	HORA PARA LA CUAL SE ACORDO LA CITA
M_NOTA	MEMO	16	NO	NOTA SOBRE LA CITA
C_ESTATUS	CARACTER	7	SI	VALORES POSIBLE: ABIERTA O CERRADA
C_CEDULA	CARACTER	11	SI	CEDULA DEL PACIENTE

TBL_RECORD: En esta tabla se guardan los registros de los Records Medico de los pacientes.

CAMPO	TIPO	LONGITUD	REQUERIDO	DESCRIPCION
N_IDRECORD (PK)	NUMERO	10	SI	CODIGO DEL FORMULARIO
C_CEDULA (FK)	CARACTER	11	SI	CEDULA DEL PACIENTE
D_FREGISTRO	FECHA	10	SI	FECHA EN LA QUE SE REGISTRARA EL REGISTRO

TBL_DETALLE_RECORD: Esta es la tabla detalle de "Records Medico".				
CAMPO	TIPO	LONGITUD	REQUERIDO	DESCRIPCION
N_IDRECORD (FK)	NUMERO	10	SI	CODIGO DEL FORMULARIO
N_IDDETALLERECORD (PK)	NUMERO	10	SI	CAMPO CLAVE, SECUENCIAL, GENERADO POR EL SISTEMA
D_FREGISTRO	FECHA	10	SI	FECHA EN LA QUE SE REGISTRARA EL REGISTRO
C_TRATAMIENTOMEDICO	CARACTER	2	SI	SI EL PACIENTE ESTA EN TRATAMIENTO O NO, VALORES "SI" O "NO"
N_IDENFERMEDADESPACIENTE (FK)	NUMERO	10	NO	CLAVE PARA LA TABLA DE ENFERMEDADES QUE SUFRE EL PACIENTE
N_IDMEDICAMENTOSPACIENTE (FK)	NUMERO	10	NO	CLAVE PARA LA TABLA DE MEDICAMENTOS QUE ESTA TOMANDO EL PACIENTE
N_IDALERGIASPACIENTE (FK)	NUMERO	10	NO	CLAVE PARA LA TABLA DE ALERGIAS QUE SUFRE EL PACIENTE
M_NOTA	MEMO	16	SI	NOTAS

TBL_ENFERMEDADESPACIENTE: En esta tabla se registran las enfermedades que tiene un paciente.				
CAMPO	TIPO	LONGITUD	REQUERIDO	DESCRIPCION
N_IDENFERMEDADESPACIENTE (PK)	NUMERO	10	SI	CAMPO CLAVE, SECUENCIAL, GENERADO POR EL SISTEMA
N_IDENFERMEDAD (FK)	NUMERO	10	SI	CLAVE DE LA TABLA DE ENFERMEDADES
M_NOTA	MEMO	16	NO	NOTA

TBL_ENFERMEDADES: En esta tabla se registran las enfermedades.				
CAMPO	TIPO	LONGITUD	REQUERIDO	DESCRIPCION
N_IDENFERMEDAD (PK)	NUMERO	10	SI	CAMPO CLAVE, SECUENCIAL, GENERADO POR EL SISTEMA
C_ENFERMEDAD	CARACTER	100	SI	NOMBRE DE LA ENFERMEDAD

TBL_MEDICAMENTOSPACIENTE: En esta tabla se registran los medicamentos que esta tomando el paciente.				
CAMPO	TIPO	LONGITUD	REQUERIDO	DESCRIPCION
N_IDMEDICAMENTOSPACIENTE (PK)	NUMERO	10	SI	CAMPO CLAVE, SECUENCIAL, GENERADO POR EL SISTEMA
N_IDMEDICAMENTO (FK)	NUMERO	10	SI	CLAVE DE LA TABLA DE MEDICAMENTOS
M_NOTA	MEMO	16	NO	NOTA

TBL_MEDICAMENTOS: En esta tabla se registran los medicamentos.				
CAMPO	TIPO	LONGITUD	REQUERIDO	DESCRIPCION
N_IDMEDICAMENTO (PK)	NUMERO	10	SI	CAMPO CLAVE, SECUENCIAL, GENERADO POR EL SISTEMA
C_MEDICAMENTO	CARACTER	100	SI	NOMBRE DEL MEDICAMENTO

TBL_ALERGIASPACIENTE: En esta tabla se registran las alergias padece el paciente.				
CAMPO	TIPO	LONGITUD	REQUERIDO	DESCRIPCION
N_IDALERGIASPACIENTE (PK)	NUMERO	10	SI	CAMPO CLAVE, SECUENCIAL, GENERADO POR EL SISTEMA
N_IDALERGIA (FK)	NUMERO	10	SI	CLAVE DE LA TABLA DE ALERGIA
M_NOTA	MEMO	16	NO	NOTA

TBL_ALERGIAS: En esta tabla se registran las alergias.				
CAMPO	TIPO	LONGITUD	REQUERIDO	DESCRIPCION
N_IDALERGIA (PK)	NUMERO	10	SI	CAMPO CLAVE, SECUENCIAL, GENERADO POR EL SISTEMA
C_ALERGIA	CARACTER	100	SI	NOMBRE DE LA ALERGIA

Diseño de pantalla

Mantenimiento de Medicamentos


Acciones

Medicamentos

Codigo: 31

Medicamento: ASPIRINA

[Ayuda](#)



Mantenimiento Enfermedades

Acciones

Enfermedades

Código: 8

Enfermedad: PAPERAS

Mantenimiento de Alergias

Acciones

Alergias

Código: 31

Alergias: PENICILINA

Diseño de reportes

Mantenimiento de Paciente

Acciones

Agregar Editar Borrar Aceptar Cancelar Buscar Imprimir [Visualizando] 1/1:0 Cerrar

Pacientes

Cédula: 00100014489 Fecha de Registro: 01/01/2005

Primer Nombre: alex

Segundo Nombre: rafael ma

Primer Apellido: terrero

Segundo Apellido: matos

Fecha Nac: 01/01/2005

Sexo: M

Teléfono: 8095977962

Compañía: cam

Teléfono Compañía: 8095977962