



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS
CENTRO DE INFORMÁTICA INDUSTRIAL

Trabajo de Diploma para optar por el Título de Ingeniero en
Ciencias Informáticas.

Título: MÓDULO DE GESTIÓN DE DATOS PARA
HDETA

Autor: Adrian Fumero Valdés
Tutora: Ing. Kirenia Rojas Escobar

La Habana, julio de 2011
“Año 53 de la Revolución”

Agradecimientos

Quisiera agradecer a toda mi familia, en especial a mi madre Fifi y mi hermano Aza que han sido los principales ejemplos a seguir en mi vida.

A mi abu Zena, que siempre estuvo presente en toda mi vida de estudiante y fue mi segunda mamá.

A mi novia Eve por estar siempre empujándome y peleándome para hacer las cosas.

A mis compañeros, amigos y a todos aquellos que han tenido confianza en mí y que de una forma u otra ayudaron a hacer posible que este humilde servidor se haya convertido en ingeniero.

Adrian Fumero Valdés

Dedicatoria

Dedico este trabajo como representación de todos estos años de estudios:

A mi madre y mi hermano, con tanto amor y dedicación, los cuales representan la fuerza que me levanta cada día y la pasión con que vivo la vida.

A las memorias de mi padre y de mi abuelo, que aunque no estén hoy presente significan mucho para mí.

A mis abuelos por su preocupación en mi educación y formación.

Adrian Fumero Valdés

Declaración de autoría

Declaro que soy el único autor de este trabajo y autorizo a la Universidad de las Ciencias Informáticas (UCI) derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo el presente a los ___ días del mes de ___ del año ____.

Adrian Fumero Valdés
Autor

Ing. Kirenia Rojas Escobar
Tutora

Resumen

El uso de métodos para la gestión de la información se hace imprescindible cuando se debe gestionar grandes cantidades de datos, como es el caso de los datos personales y médicos de pacientes. El sistema HDETA (**H**erramienta de **D**etención **E**valuación y **T**ratamiento de la **A**mbliopía) desarrollado en la Facultad 5 de la UCI contiene diferentes módulos (Stereopsis, Transparencia y otros) para evaluar el estado y progreso de pacientes que padecen de esta patología. Dichos módulos generan una cierta cantidad de información que es necesario gestionar, con el fin de llevar estos datos organizados y que resulten de fácil acceso. En la tesis se presenta una base de datos diseñada en Microsoft SQL Server con 9 tablas que recogen la información aportada por los módulos antes mencionados y un módulo de gestión de esa información que permite dar seguimiento del estado y progreso de los pacientes según el tratamiento diseñado por los especialistas. Como salida, esta aplicación brinda la posibilidad de generar reportes con información relevante como ayuda a la toma de decisiones y la emisión de diagnósticos de los pacientes tratados en este sistema, por parte de los especialistas.

Palabras Clave: *Gestión de datos, Realidad Virtual*

Índice

| | |
|---|-----------|
| Introducción | 1 |
| Capítulo 1: Fundamentación teórica | 4 |
| 1.1 Introducción | 4 |
| 1.2 Tipos de base de datos | 4 |
| 1.2.1 Según la variabilidad de los datos | 4 |
| 1.2.2 Según el contenido | 5 |
| 1.3. Sistemas de gestión de base de datos | 6 |
| 1.4. Técnicas y métodos para el análisis de datos | 12 |
| 1.4.1. Generación de Reportes | 13 |
| 1.5. Antecedentes de la gestión de información en aplicaciones para el tratamiento de enfermedades visuales | 13 |
| Capítulo 2: Descripción y Análisis de la Solución | 15 |
| 2.1. Introducción | 15 |
| 2.2. Descripción de la solución propuesta | 15 |
| 2.3. Principales funcionalidades del sistema | 15 |
| 2.4. Herramientas y lenguajes de desarrollo utilizados | 17 |
| 2.4.1. Herramientas | 17 |
| 2.4.2. Lenguajes | 19 |
| 2.5. Modelo de Dominio | 20 |
| 2.6. Descripción de los requisitos funcionales y no funcionales. | 21 |
| 2.7. Descripción del sistema propuesto | 22 |

| | |
|---|-----------|
| 2.7.1. Descripción de los actores del sistema | 22 |
| 2.7.2. Diagrama de casos de usos del sistema | 23 |
| 2.7.3. Descripción de los casos de usos del sistema | 23 |
| 2.8. Descripción de la Arquitectura y Fundamentación | 39 |
| Capítulo 3: Diseño e Implementación..... | 40 |
| 3.1. Introducción | 40 |
| 3.2. Patrón de Arquitectura Utilizado..... | 40 |
| 3.3. Patrones de Diseño Utilizados..... | 41 |
| 3.4. Diagrama de Clases del Sistema | 42 |
| 3.4.1. Descripción de las clases más significativas..... | 45 |
| 3.5. Descripción de la implementación por Casos de Uso..... | 48 |
| 3.6 Diagrama de Componentes | 54 |
| 3.7. Diagrama de Despliegue..... | 55 |
| 3.7. Diseño de la Base de Datos..... | 55 |
| 3.7.1. Diagrama de las clases persistentes | 55 |
| 3.7.2. Diagrama entidad-relación..... | 56 |
| Capítulo 4: Análisis de los Resultados | 58 |
| 4.1. Introducción | 58 |
| 4.2. Validación teórica del software..... | 58 |
| 4.2.1. Normalización..... | 58 |
| 4.2.2. Análisis de redundancias..... | 59 |
| 4.3. Análisis de la seguridad de la base de datos | 59 |
| 4.4. Validación práctica del software | 60 |
| Conclusiones Generales | 67 |

| | |
|--|-----------|
| Recomendaciones | 67 |
| Referencias bibliográficas y bibliografías..... | 69 |
| Glosario de Términos | 71 |

Introducción

La RV se podría definir como un sistema informático que genera en tiempo real representaciones de la realidad, que de hecho no son más que ilusiones, ya que se trata de una realidad perceptiva sin ningún soporte físico, basado en el empleo de ordenadores y otros dispositivos, permitiéndole al usuario tener la sensación de estar presente en ella, es decir incursionar creativamente, hasta donde el límite de su imaginación se lo permita. Todo se basa en el engaño que se pueda dar a sus sentidos haciendo uso de ciertos dispositivos. Hoy, llegan estos engaños a varios sentidos como la vista, el tacto y audición. (RV, 2011) Esta representación virtual de la realidad, cuenta con desarrollos ya consolidados en la defensa, mediante entrenamientos virtuales aeronáuticos; en la educación, donde la capacidad de representar escenarios distantes o elementos abstractos puede permitir la mejor comprensión de algunos hechos o fenómenos para los estudiantes. Una evolución algo inesperada se ha dado en la difusión del entretenimiento y el cine, para representar actores virtuales y la simulación de escenografías tridimensionales interactivas, lo que ha constituido un negocio creciente en el sector.

Uno de los campos del conocimiento donde se han aprovechado bastante las ventajas de la RV ha sido en la medicina, ventajas que se han traducido en mejoras terapéuticas, mayor certeza al realizar un diagnóstico, comodidades para el paciente, peligrosas cirugías y hasta la representación ciberespacial de traumas de guerra. (RVM, 2011)

Muchos de los trabajos realizados en este ámbito han de necesitar de una cierta cantidad de información, ya que hay que ir llevando los datos personales y médicos de disimiles pacientes. La utilización de herramientas para ir gestionando estos datos es de gran necesidad, ya que como se conoce, permiten manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante para la organización.

El manejo de información requiere desarrollar un conjunto de habilidades que permiten definir la información necesaria, obtenerla y aprovecharla. Con el surgimiento de las computadoras, se desarrollaron herramientas informáticas que permiten mayor rapidez, reducir el esfuerzo, además representar y comunicar la información.

En la actualidad, la tendencia de soporte de la información es en formato digital. El aumento de los volúmenes de información y el acortamiento de los tiempos de procesamiento, hacen indispensable el desarrollo de software para los procesos en las organizaciones. La gestión de información es

caracterizada por la reducción de etapas y recursos, brindando la oportunidad de disminuir el tiempo de algunos procesos para invertirlo en la realización de otros.

A través de la observación y con la puesta en práctica de sistemas gestores de información se traslucen varias ventajas tales como: hacer mucho más efectivo el manejo de la información para la toma de decisiones, contar con archivos de información de gran volumen en muy poco espacio facilitando la disponibilidad para posteriores análisis, se reduce el personal que tiene que manejar la información, es decir, se gestiona mejor y con menos personas.

En el Centro de Informática Industrial de la Facultad 5 de la UCI existe un proyecto de investigación llamado HDSVE, dedicado al desarrollo de sistemas de RV, basados en tecnologías de visión estereoscópica. Este proyecto desarrolla actualmente un software denominado HDETA, el cual se especializa en la detección, evaluación y tratamiento de la ambliopía mediante los métodos de RV. El mismo presenta módulos específicos, los cuales representan un problema en cuestión de un paciente tratado por el sistema.

Es necesario que dicho sistema permita realizar un manejo eficiente y organizado de la información relativa a los pacientes, su tratamiento y evolución, de manera que permita a los especialistas realizar un seguimiento de la evolución de cada uno de los casos en el tiempo, apoyando la toma de decisiones y la emisión de diagnósticos, ya que actualmente, llevar esto sin incluir la informatización, se hace muy engorroso.

De aquí se plantea entonces como **problema científico**: ¿Cómo automatizar el manejo de la información de los pacientes tratados con HDETA?

Se define como **objeto de estudio** el proceso de gestión de información, enmarcando el **campo de acción** en los procesos de seguimiento, control y evaluación del manejo de la información de los pacientes tratados con HDETA.

Para dar respuesta al problema planteado se propone como **objetivo general**, desarrollar un módulo que permita la gestión de la información relativa a los pacientes tratados con el sistema HDETA, de manera que ayude en la toma de decisiones y emisión de diagnósticos por los especialistas.

Para dar cumplimiento al principal objetivo de esta investigación se han planteado una serie de tareas a cumplir, las cuales son:

- Analizar los sistemas de gestión de base de datos más comunes.
- Analizar técnicas y métodos para el manejo de datos que permita al especialista la toma de decisiones.
- Desarrollar el diseño lógico y físico de la base de datos.
- Construir un prototipo de aplicación integrable a la BD que facilite el uso y manejo de la información.
- Realizar un flujo de trabajo que muestre el proceso de la integración de las herramientas de gestión de datos con las implementaciones dadas.
- Analizar de los resultados.

Métodos de investigación:

Los **métodos teóricos** posibilitaron descubrir, analizar y sistematizar los resultados obtenidos, para llegar a conclusiones confiables que permitan resolver el problema. En tal sentido se usaron:

- **Analítico - Sintético:** Se realizó un estudio excesivo de los diferentes tipos de gestores de base de datos, así como el uso de éstos en el manejo organizado de la información mediante una interfaz. Analizando la documentación de lo anteriormente expuesto y sintetizando los rasgos más relevantes de acuerdo a nuestro desarrollo e investigación.
- **Análisis histórico - lógico:** Se analizó además la historia de las herramientas utilizadas así como sus principales características, adquiriendo mayor conocimiento sobre su trayectoria desde sus inicios hasta la actualidad.

Ideas a Defender:

Con este módulo de gestión de datos, se otorgará a HDETA una interactividad organizada con los datos de los pacientes, permitiendo llevar un seguimiento y evolución de estos, para posteriormente llegar a emisiones de diagnósticos y toma de decisiones por parte de los especialistas.

Estructura del documento

El presente trabajo está estructurado por cuatro capítulos, los cuales están conformados por la siguiente información:

Capítulo 1, tiene como propósito dar a la investigación un sistema coordinado y coherente de conceptos y proposiciones que permitan abordar el problema. Además se analiza las tendencias de las tecnologías y sistemas relacionados al tema, así como las técnicas, herramientas y tecnologías útiles para el desarrollo que se propone.

Capítulo 2, muestra la solución propuesta para el problema planteado. Presentándose el Modelo de dominio, los requerimientos y la descripción del sistema propuesto.

Capítulo 3, representa la construcción de la solución propuesta donde se hace énfasis en la fase de análisis y diseño e implementación.

Capítulo 4, se mostrará un efectivo análisis de los resultados obtenidos, mediante una serie de pruebas que evaluarán al software dando muestra de que logró dar respuesta al problema científico planteado.

Capítulo 1: Fundamentación teórica

1.1 Introducción

La gestión, propiamente dicha, es un proceso que se lleva a cabo en busca de mejorar la eficacia de las actividades que se desarrollan en torno a un objeto de estudio, de que se puede concluir que: *“Gestionar es emplear adecuadamente los recursos disponibles para lograr el objetivo que se haya trazado, es alcanzar los propósitos a través de la acción coordinada de personas.”* (Ruiz., 2006). La organización, ejecución, control y evaluación de la información, constituyen las fases de la gestión. Con la puesta en marcha de cada una de ellas se puede lograr una gestión efectiva. En la actualidad la gestión de datos es muy importante ya que a través de esto se mantiene la información bien organizada, además de poder acceder fácilmente a ella.

En este capítulo se muestra una serie de conceptos y características estrechamente relacionadas sobre el objeto de estudio y campo de acción, así como el análisis de distintas.

1.2 Tipos de base de datos

Se define una base de datos como una serie de datos organizados y relacionados entre sí, los cuales son recolectados y explotados por los sistemas de información de una empresa o negocio en particular. Éstas proporcionan la infraestructura requerida para los sistemas de apoyo a la toma de decisiones y para los sistemas de información estratégicos. (BD, 2009)

Las BD pueden clasificarse de varias maneras, de acuerdo al contexto que se esté manejando, la utilidad de las mismas o las necesidades que satisfagan. Teniendo en cuenta esto se pueden agrupar en:

1.2.1 Según la variabilidad de los datos

Bases de datos estáticas:

Estas son bases de datos de solo lectura, utilizadas primordialmente para almacenar datos históricos que posteriormente se pueden utilizar para estudiar el comportamiento de un conjunto de datos a través del tiempo, realizar proyecciones y tomar decisiones.

- Objetos (entidades, relaciones, registros, etc.).
- Asociaciones entre objetos (interrelaciones, etc.).
- Propiedades de los objetos o asociaciones (atributos, campos, elementos de datos, etc.).
- Dominios (conjuntos nominados de valores sobre los que se definen las propiedades).

Bases de datos dinámicas:

Éstas son bases de datos donde la información almacenada se modifica con el tiempo, permitiendo operaciones como actualización, borrado y adición de datos, además de las operaciones fundamentales de consulta. Un ejemplo de esto puede ser la base de datos utilizada en un sistema de información de un supermercado, una farmacia, un video-club o una empresa.

1.2.2 Según el contenido

Bases de datos bibliográficas:

Solo contienen un representante de la fuente primaria, que permite localizarla. Un registro típico de una BD bibliográfica contiene información sobre el autor, fecha de publicación, editorial, título, edición, de una determinada publicación. Puede contener un resumen o extracto de la publicación original, pero nunca el texto completo, porque si no se estaría en presencia de una BD a texto completo (o de fuentes primarias). Como su nombre lo indica, el contenido son cifras.

Base de datos de texto completo:

Almacenan las fuentes primarias, como por ejemplo, todo el contenido de todas las ediciones de una colección de revistas científicas.

Bases de datos o bibliotecas de información química o biológica:

Son bases de datos que almacenan diferentes tipos de información proveniente de la química, las ciencias de la vida o médicas. Se pueden considerar en varios subtipos:

- • Las que almacenan secuencias de nucleótidos o proteínas.
- • Las bases de datos de rutas metabólicas.
- • Bases de datos de estructura, comprende los registros de datos experimentales sobre estructuras de tres dimensiones (3D) de biomolecular.
- • Bases de datos bibliográficas (biológicas, químicas, médicas y de otros campos). ([TBD, 2009](#))

1.3. Sistemas de gestión de base de datos

Un SGBD es un programa de ordenador que facilita una serie de herramientas para manejar bases de datos y obtener resultados (información) de ellas. Además de almacenar la información, se le pueden hacer preguntas sobre esos datos, obtener listados impresos, generar pequeños programas de mantenimiento de la BD, o ser utilizado como servidor de datos para programas más complejos realizados en cualquier lenguaje de programación.

Además, ofrece otras herramientas propias de la gestión de BD como sistemas de permisos para autorización de accesos, volcados de seguridad, transferencia de ficheros, recuperación de información dañada, etc.

Los sistemas de gestión de bases de datos son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan.

De forma general, los SGBD son los encargados de manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante para una organización.

Seguidamente se muestran una serie de ventajas y desventajas que presentan estas:

Ventajas

Proveen facilidades para la manipulación de grandes volúmenes de datos. Entre estas:

- • Manejando las políticas de respaldo adecuadas, garantizan que los cambios de la base de datos serán siempre consistentes.
- • Organizan los datos con un impacto mínimo en el código de los programas.
- • Bajan drásticamente los tiempos de desarrollo y aumentan la calidad del sistema si son bien explotados por los desarrolladores.

Usualmente, proveen interfaces y lenguajes de consulta que simplifican la recuperación de los datos.

Desventajas

1. Típicamente, es necesario disponer de una o más personas que administren la base de datos, en la misma forma en que suele ser necesario en instalaciones de cierto porte, la disposición de una o más personas que administren los sistemas operativos. Esto puede llegar a incrementar los costos de operación en una empresa. Sin embargo hay que balancear este aspecto con la calidad y confiabilidad del sistema que se obtiene.
2. Si se tienen muy pocos datos que son usados por un único usuario por vez y no hay que realizar consultas complejas sobre los datos, entonces es posible que sea mejor usar una planilla de cálculo.
3. Complejidad: los software muy complejos y las personas que vayan a usarlo deben tener conocimiento de las funcionalidades del mismo para poder aprovecharlo al máximo.
4. Tamaño: la complejidad y la gran cantidad de funciones que tiene, hacen que sea un software de gran tamaño, que requiere de gran cantidad de memoria para poder ejecutarse.
5. Coste del hardware adicional: los requisitos de hardware para usar un SGBD por lo general son relativamente altos, por lo que estos equipos pueden llegar a costar gran cantidad de dinero.

1.3.1. Principales gestores de base de datos

Oracle, Microsoft SQL Server, MYSQL y PostgreSQL, son destacados gestores de bases de datos que se utilizan en la actualidad. ([Hernández, 2010](#))

Oracle

Se considera a Oracle como uno de los sistemas de bases de datos más completos que existe, (ORA, 2011) entre las **ventajas** que presenta se pueden citar las siguientes:

- Soporte de transacciones.
- Gran estabilidad y seguridad.
- Escalabilidad.
- Multiplataforma.

A pesar de esto presenta algunos **inconvenientes**.

- Su mayor defecto es su enorme precio, que es de varios miles de euros (según versiones y licencias).
- Otro aspecto a criticar es la seguridad de la plataforma, y las políticas de suministro de parches de seguridad, que incrementan el nivel de exposición de los usuarios. En los parches de actualización provistos durante el 2005 fueron corregidas 22 vulnerabilidades públicamente conocidas, algunas de ellas con una antigüedad de más de dos años.
- Aunque ha tenido un amplio dominio en el mercado de servidores sufre la competencia del Microsoft SQL Server de Microsoft y con licencia libre como PostgreSQL y MySQL.

MySQL

Es un SGBD cliente/servidor, multihilo y multiusuario. Se compone de un servidor SQL, varios programas clientes y bibliotecas, herramientas administrativas, y una gran variedad de Interfaces de Programación de Aplicaciones (APIs). Se puede obtener también como una biblioteca multihilo que se puede enlazar dentro de otras aplicaciones para obtener un producto más pequeño, rápido, y fácil de manejar. Es el servidor de BD relacionales más popular.

MySQL es desarrollado como software libre en un esquema de licenciamiento dual.

Cualquier empresa o persona interesada puede descargar el software de MySQL de Internet y usarlo sin pagar por ello. Además, cualquiera que lo necesite puede estudiar el código fuente y cambiarlo de

acuerdo a sus necesidades. Usa la licencia GPL (Licencia Pública General GNU), para definir qué es lo que se puede y no se puede hacer con el software para diferentes situaciones. Sin embargo, si no se está de acuerdo con la licencia GPL o tiene la necesidad de incorporar código de MySQL en una aplicación comercial es posible comprar una versión con una licencia comercial. (Muñoz, 2009)

Ventajas:

- Diseñado en vistas a la velocidad.
- Consume muy pocos recursos de CPU y memoria. Muy buen rendimiento.
- Tamaño del registro sin límite.
- Buena integración con PHP.
- Utilidades de administración (phpMyAdmin).
- Buen control de acceso usuarios-tablas-permisos.
- Trabaja bajo diferentes plataformas: AIX 4x 5x, Amiga, BSDI, Digital Unix 4x, FreeBSD 2x 3x 4x, HP-UX 10.20 11x, Linux 2x, Mac OS, NetBSD, Novell NetWare 6.0, OpenBSD 2.5, OS/2, SCO OpenServer, SCO UnixWare 7.1.x, SGI Irix 6.x, Solaris 2.5, SunOS 4.x, Tru64 Unix y Windows 9x, Me, NT, 2000, XP, 2003.
- Desarrollo de APIs para C, C++, Eiffel, Java, Perl, PHP, Python, Ruby, y Tcl.

Inconvenientes:

- No soporta subconsultas.
- No soporta transacciones.
- No soporta vistas.

Postgre

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional basado en el proyecto, es una derivación libre (OpenSource) del proyecto POSTGRES y utiliza el lenguaje SQL92/SQL99. Es un servidor de base de datos relacional, tiene soporte total para transacciones, disparadores, vistas, procedimientos almacenados, y almacenamiento de objetos de gran tamaño, incluye características de la orientación a

objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, reglas e integridad transaccional.

A pesar de esto, PostgreSQL no es un sistema de gestión de bases de datos puramente orientados a objetos, pero tiene otras características que hacen prestigioso este potente gestor, se puede ejemplificar las siguientes:

- Soporta distintos tipos de datos: además del soporte para los tipos base, también soporta datos de tipo fecha, monetarios, elementos gráficos, datos sobre redes (MAC, IP...), cadenas de bits, etc. También permite la creación de tipos propios.
- Incorpora una estructura de datos array.
- Incorpora funciones de diversa índole: manejo de fechas, geométricas, orientadas a operaciones con redes, etc.
- Permite la declaración de funciones propias, así como la definición de disparadores.
- Soporta el uso de índices, reglas y vistas.
- Incluye herencia entre tablas (aunque no entre objetos, ya que no existen), por lo que a este gestor de bases de datos se le incluye entre los gestores objeto-relacionales.
- Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos.

Actualmente la utilización del gestor de base de datos Postgre en el mundo, continúa siendo una de las de mayor demanda por el gran alcance que tiene y las innumerables ventajas que proporciona, entre ellas la que opera en una arquitectura cliente/servidor, y trabaja en múltiples plataformas, pues existen versiones para Windows y Linux. Es rápido, confiable y posee un mayor rendimiento, mejores utilidades de administración y mejor control de acceso a los usuarios. (Pos, 2011)

Microsoft SQL Server.

SQL Server es el SGBD ideal para un amplio espectro de clientes corporativos y productores independientes de software, inmersa en la creación de aplicaciones empresariales.

Las necesidades y requisitos del cliente han dado lugar a innovaciones significativas en el producto SQL Server versión, entre las que se incluyen la facilidad de uso, escalabilidad, fiabilidad, y almacenamiento de

datos. SQL Server es la oferta completa de BD y análisis, tanto por la capacidad para consultar la BD mediante un explorador, como por la compatibilidad con el XML, SQL Server es la BD totalmente habilitada para Web.

Es un gestor de BD relacionales desarrollado por Microsoft, de ahí su gran compatibilidad con IDE Visual Studio. Para el desarrollo de aplicaciones más complejas (tres o más capas), Microsoft SQL Server incluye interfaces de acceso para la mayoría de las plataformas de desarrollo, incluyendo .NET, además de ser uno de los mejores SGBD basada en el lenguaje SQL, capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea. Las necesidades y requerimientos de los clientes han llevado a la creación de innovaciones de productos significativos para facilitar la utilización, escalabilidad, confiabilidad y almacenamiento de datos. (SQL, 2011)

Ventajas:

- Soporta la configuración automática y la auto-optimización.
- Administración multiservidor para un gran número de servidores.
- Gran variedad de opciones de duplicación de cualquier BD.
- Acceso universal a los datos (Universal Data Access).
- Fácil de usar.
- Escalabilidad. Permite realizar un escalamiento hasta 32 CPU y 64 gigabytes (GB) de RAM, siendo capaz de manejar al máximo multiprocesamiento simétrico aprovechando de esta forma el hardware.
- Potencia: Microsoft SQL Server es la mejor BD para Windows NT Server.
- Posee los mejores registros de los benchmarks independientes, tanto en transacciones totales como en coste por transacción.
- Gestión: Con una completa interfaz gráfica que reduce la complejidad innecesaria de las tareas de administración y gestión de la BD.
- Múltiples instancias y tolerancia a fallos.
- Soporte para XML (cláusula FOR XML).

Inconvenientes:

- Licencias con costos altos.

➤ Plataformas Windows.

1.4. Técnicas y métodos para el análisis de datos

Existen diferentes técnicas para organizar, describir y analizar los datos recogidos a partir de diferentes instrumentos. Para llevar a cabo el análisis, se definen dos tipos de técnicas, la cuantitativa y la cualitativa, que se basan en la forma estadística y en los aspectos descriptivos e interpretativos, respectivamente.

Análisis de datos desde el punto de vista cualitativo.

1. Las técnicas de categorización.

Son técnicas de reducción y organización de datos cualitativos. Los datos se revisan rigurosamente y se reducen a unidades llamadas categorías. El investigador cualitativo no categoriza para contar hechos, por el contrario, categoriza para analizar y comparar los diversos significados producidos en cualquier categoría.

2. Las técnicas de análisis de contenido.

Es una técnica de procesamiento e investigación para la descripción objetiva, sistemática y cualitativa de los archivos de documento textual. La misma permite reducir y sistematizar cualquier tipo de información cualitativa acumulada (documentos escritos, filmes, grabaciones, etc.) en datos, respuestas o valores correspondientes a variables que investigan en función de un problema.

Es una técnica de codificación, donde se reducen grandes datos textuales a categorías que se puedan representar numérica o gráficamente.

Análisis de datos desde el punto de vista cuantitativo

Las técnicas de análisis cuantitativo son aquellas que se basan en las estadísticas, en las finanzas. Estas sirven para describir, graficar, analizar, comparar, relacionar y resumir los datos obtenidos con los instrumentos cuantitativos.

1.4.1. Generación de Reportes

Un reporte es un informe o una noticia. Este tipo de documento (que puede ser impreso, digital, audiovisual, etc.) pretende transmitir una información, aunque puede tener diversos objetivos. Existen reportes divulgativos, persuasivos y de otros tipos.

El reporte puede ser la conclusión de una investigación previa o adoptar una estructura de problema-solución en base a una serie de preguntas. En estos, el texto suele ir acompañado por gráficos, diagramas, tablas de contenido y notas al pie de página.

En el ámbito de la informática, los reportes son informes que organizan y exhiben la información contenida en una base de datos. Su función es aplicar un formato determinado a los datos para mostrarlos por medio de un diseño atractivo y que sea fácil de interpretar por los usuarios.

Un prototipo de reporte estadístico se muestra posteriormente en el Anexo No. 1.

1.5. Antecedentes de la gestión de información en aplicaciones para el tratamiento de enfermedades visuales.

Durante el desarrollo de esta aplicación se ha estado investigando acerca de otros posibles software de este tipo a nivel nacional e incluso internacional con el objetivo de encontrar mejoras para su desarrollo e incluso entrar en el mercado, lo que propiciará a la evolución del mismo.

Un ejemplo de esto fue el llamado Programa EVO. desarrollado en el período (1999- 2001) por parte de la Organización Nacional de Ciegos Españoles (ONCE), con el objetivo de aprovechar las nuevas tecnologías en la mejora del ámbito visual. Este logra gestionar los ejercicios de entrenamiento visual, además de permitir llevar una traza de las secuencias de entrenamiento y sus resultados.

EVO ha sido probado en los Centros de Recursos Educativos de la O.N.C.E. “Antonio

Vicente Mosquete”, “Joan Amades” y “Luis Braille”, obteniéndose resultados de mejora sobre los usuarios en su funcionamiento visual. ([Pro, 2001](#))

Este software alimenta de ideas y mejoras al desarrollo del sistema, consiguiendo así que este cumpla con los requerimientos.

En este capítulo se recopilan datos que permiten respaldar la labor a realizar a través de un profundo análisis del proceso de gestión de información, además de haber puntualizado los conceptos necesarios referentes al campo de acción de manera detallada para así entender la temática y por último y no menos importante estar actualizado en lo referente de trabajos o software similares desarrollados con anterioridad con el propósito de llegar a desarrollar una aplicación adaptada a sus requisitos.

Capítulo 2: Descripción y Análisis de la Solución

2.1. Introducción

En este capítulo se exponen los procesos más importantes por los cuales atravesará la aplicación durante su desarrollo, así como la construcción de la BD. Se describen las herramientas utilizadas, los requisitos funcionales y no funcionales que debe cumplir el sistema, además se muestran los diagramas de los casos de usos persistentes, así como las descripciones de cada uno de ellos, luego una fundamentación de la descripción de la arquitectura propuesta para dar solución al problema existente.

El principal objetivo es lograr la automatización del proceso de gestión de la información de HDETA con un sistema informático. Tomando como apoyo una metodología, se realiza el modelado del software que sirve para visualizar lo que posteriormente se va a construir, empleando lenguajes de programación de alto nivel sobre plataformas y ambientes de desarrollo, amigables y flexibles. De aquí se puede coincidir con Paz Jiménez cuando plantea que automatizar es: “Compatibilizar objetivos de eficiencia y flexibilidad en los sistemas productivos mediante la integración de ordenadores y máquinas”. ([Sainz, 2002](#))

2.2. Descripción de la solución propuesta

Esta investigación tiene como salida un software que se traduce en un módulo de gestión de datos de los pacientes tratados con el sistema HDETA. El software se conforma por una interfaz conectada a una base de datos donde se almacenan los datos personales y médicos de los pacientes tratados. Presenta un conjunto de funcionalidades que permiten gestionar los datos de los pacientes, así como poder llevar un seguimiento y evolución de estos, teniendo como objetivo ayudar en la toma de decisiones y emisión de diagnósticos por parte de los especialistas.

2.3. Principales funcionalidades del sistema

Insertar Paciente:

Esta funcionalidad permite al usuario registrar un nuevo paciente en la base de datos, teniendo de esta forma sus datos para futuros análisis. Esta operación se basa en la muestra de un formulario donde se encuentran varios campos vacíos con el objetivo de introducir los datos del paciente.

Modificar Paciente:

Con esta función se puede modificar cualquier dato de un paciente anteriormente seleccionado por el usuario. Se da inicio al comenzar el usuario con una selección del paciente a modificar en un formulario donde se encuentra todos los pacientes ya registrados, una vez hecho esto se prosigue el pulsar en el botón Modificar, el cual mostrará un formulario con los datos anteriormente introducidos con el fin de cambiarlos a su gusto, siempre cumpliendo con los requisitos de cada tipo de dato

Eliminar Paciente:

Elimina un paciente elegido por el usuario de la base de datos. Todo comienza en el momento en que el usuario selecciona de la lista de los pacientes ya registrados uno que desee eliminar, luego de seleccionar toma la opción del botón Eliminar, el cual eliminará por completo todo dato de este paciente de la base de datos.

Configurar Actividades:

Después de que el usuario haya seleccionado un carnet de identidad, seleccionará en un orden una serie de actividades y ejercicios a realizar. Después de esto se exportará en un fichero los datos personales del paciente seleccionado y en otro la configuración de las actividades.

Cargar Datos de Actividades de una Paciente:

Se importará de un fichero XML la configuración de una serie de datos de actividades y ejercicios realizados por un paciente, dichos datos se insertarán en la base de datos, ya que posteriormente se tomarán para realizar los reportes estadísticos.

Generar Reporte Estadístico:

El usuario debe seleccionar el carnet de identidad del paciente al cual se le va a realizar el reporte, además de otros atributos específicos de las actividades y ejercicios realizados por este, con el objetivo de poder representar éstos en forma gráfica.

Consultando el Anexo No. 2 se puede ver que se muestran prototipos de interfaces de las funcionalidades antes mencionadas.

Gestionar Usuario:

Esta funcionalidad permite tener una mayor seguridad en la gestión de datos de la base de datos, manteniendo estos fuera de cualquier ataque o corrupción. Para ellos se tendrán usuarios con respectivos permisos y restricciones.

2.4. Herramientas y lenguajes de desarrollo utilizados

Para la realización de este sistema se realizó una investigación acerca de las diferentes herramientas y lenguajes de desarrollo para según sus características individuales tomar aquellas que brinden de manera eficiente y con un mejor rendimiento la posibilidad de desarrollar dicho sistema.

2.4.1. Herramientas

Embarcadero ERStudio 7.5

Como herramienta de modelado de datos se seleccionó a Embarcadero ERStudio la cual brinda el poder de descubrir, documentar, y re-utilizar los activos de datos, con un soporte completo sobre las base de datos lo cual hace que los arquitectos de las mismas puedan fácilmente realizar ingeniería inversa, analizar y optimizar estas base de datos.

Microsoft Visual Studio 2005

Primeramente decir que se toma esta herramienta debido a que esta tesis representa un módulo más de un software ya desarrollado anteriormente el cual presentaba como requerimientos este IDE. Pero es necesario no dejar de resaltar su gran facilidad de trabajo con su ágil interfaz, su rápido compilador, su gran integración a muchas herramientas, y otras características que muestra esta herramienta como una vía fácil de desarrollo.

Microsoft SQL 2005

Se utilizó Microsoft SQL como sistema de gestión de datos por su fácil integración y compatibilidad con el IDE Visual Studio, además de su legible entorno gráfico y su gran escalabilidad, estabilidad y seguridad en el manejo de datos.

Visual Paradigm

Visual Paradigm es una herramienta UML de tipo profesional, la cual soporta el ciclo completo de desarrollo de software: análisis y diseño orientado a objetos, construcción, pruebas y despliegue. Este software de modelado UML constituye una gran ayuda en la rápida construcción de aplicaciones de calidad a un menor costo. Permite dibujar todos los tipos de diagramas de clases, código inverso, generador de código desde diagramas y generar documentación. Esta herramienta CASE (Computer Aided Software Engineering) como comúnmente se conoce está provista de un gran número de tutoriales de modelado UML, demostraciones interactivas de UML, así como proyectos de este tipo. Por ello se seleccionó esta herramienta.

Generador de reportes

Como generador de reportes estadísticos se tomó el componente DEVExpressReport del pluginDEVExpress. Dicho plugin, brinda la incorporación de nuevos componentes al

IDE, cuyo muchos de estos componentes muestran una serie de datos preestablecidos mediante gráficas en forma de tablas, líneas, barras y otras.

Se centrará el uso de los reportes estadísticos ya que es una forma de emplear las técnicas de análisis de datos, en este caso las cuantitativas, con el objetivo de lograr mayor efectividad y certeza a la hora de la toma de decisiones y la emisión de diagnósticos, ya que les sería más fácil al especialista poder ver la evolución y tratamiento de los pacientes tratados por el sistema. (Horna, 2011)

La generación de reportes estadísticos actualmente está muy valorada, al tanto que son muchos los software y herramientas que hacen uso de estos, ya que proporcionan valiosas modalidades de representar datos que posteriormente se convertirán en información relevante para una institución.

Se utilizó las gráficas de líneas ya que suelen utilizarse para comparar valores a lo largo del tiempo, haciendo que su empleo resulte atractivo y fácil de interpretar por los usuarios, aunque esto se puede hacer mediante las gráficas de barras pero suele ser un poco dificultoso a la hora de comparar un conjunto de muestras, en este caso datos de pacientes.

Seguidamente se mostrará una tabla imagen donde se podrá ver las gráficas más utilizadas, así como también su uso.

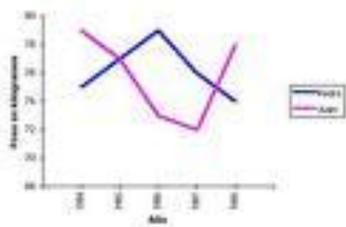
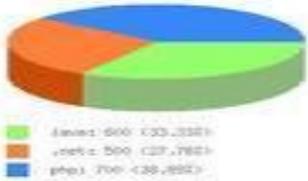
| | | |
|--------------------------------------|---|--|
| <p>Gráficas de líneas</p> | <p>Estos gráficos suelen utilizarse para comparar valores a lo largo del tiempo, en el que se pueden incluir varias muestras.</p> |  |
| <p>Gráficas de barras</p> | <p>Las gráficas de barras se utilizan generalmente cuando se pretende resaltar la representación de porcentajes de datos que componen un total, aunque también pueden hacer lo mismo que las de líneas, pero se complica al analizarse varias muestras.</p> |  |
| <p>Gráficas de pasteles o tortas</p> | <p>Expresa de manera gráfica la distribución proporcional de los eventos o datos en estudio.</p> |  |

Tabla 1 Comparación entre los gráficos más usados

2.4.2. Lenguajes

Son muchos y muy variados los lenguajes que brindan las nuevas tecnologías que permiten el desarrollo de este tipo de aplicación. Cada uno de ellos, con sus particularidades, ventajas y desventajas, se vislumbra en la amplia lista de recursos y herramientas que están al alcance de los desarrolladores de software.

C#

Para el desarrollo de la aplicación se utilizó como lenguaje C# ya que este producto como se mencionó anteriormente está integrado a otro módulo ya desarrollado bajo este lenguaje, por lo que se tomó este para mantener la línea y compatibilidad en el desarrollo.

Pero no dejar a un lado su sencillez en su uso, su modernidad, su buena orientación a objetos y su gran eficiencia, además de otras características que lo destacan sobre los demás lenguajes.

XML

Se utilizó este lenguaje como fichero de almacenamiento estático de una serie de datos donde se configuran las actividades y ejercicios de un paciente dado, además de exportar la información de un paciente, sobre los cuales se decidirá más tarde sobre el futuro análisis por parte de los especialistas.

UML

El Lenguaje de Modelado Unificado (UML por sus siglas en inglés), es un lenguaje para la especificación, visualización, construcción y documentación de los artefactos de un proceso de sistema intensivo.

El UML está compuesto por diversos elementos gráficos que se combinan para conformar diagramas. Debido a que el UML es un lenguaje, cuenta con reglas para combinar tales elementos.

Es importante recalcar que UML no es una guía para realizar el análisis y diseño orientado a objetos, es decir, no es un proceso. UML es un lenguaje que permite la modelación de sistemas con tecnología orientada a objetos.

En las versiones previas del UML, se hacía un fuerte hincapié en que UML no era un lenguaje de programación. Un modelo creado mediante UML no podía ejecutarse. En el UML 2.0, esta asunción cambió de manera drástica y se modificó el lenguaje, de manera tal que permitiera capturar mucho más comportamiento (behavior). De esta forma, se permitió la creación de herramientas que soporten la automatización y generación de código ejecutable, a partir de modelos UML.

2.5. Modelo de Dominio

Seguidamente se muestra un diagrama, el cual ayuda a comprender los conceptos que utilizan los usuarios, con los que trabajan y con los que deberá trabajar la aplicación. Es decir, un diagrama con los objetos que existen (reales) relacionados con el proyecto que se va a acometer y las relaciones que hay entre ellos.

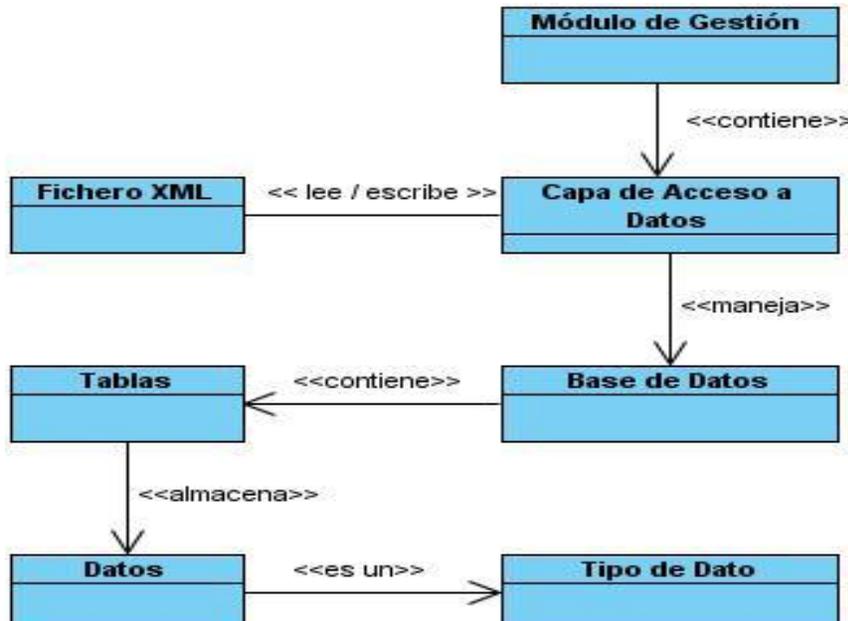


Ilustración 1 Modelo de dominio del sistema.

2.6. Descripción de los requisitos funcionales y no funcionales.

Seguidamente se muestran descriptivamente los requisitos del sistema, los cuales representan características requeridas de este, del proceso de desarrollo, del servicio prestado o de cualquier otro aspecto del desarrollo que expresa una capacidad de acción del mismo, una funcionalidad; generalmente expresada en una declaración en forma verbal o también una características que señale una restricción del sistema en desarrollo.

| Requisitos funcionales | |
|---|------------------------------------|
| RF1 Adicionar datos de paciente. | RF10 Seleccionar usuario. |
| RF2 Modificar datos de paciente. | RF11 Eliminar usuario de la BD. |
| RF3 Eliminar paciente de la BD. | RF12 Buscar usuario. |
| RF4 Seleccionar paciente. | RF13 Autenticar usuario |
| RF5 Buscar Paciente | RF14 Generar reportes estadísticos |
| RF6 Listar Pacientes | RF15 Configurar actividades |
| RF7 Cargar Datos de Actividades de una Paciente | RF16 Exportar XML de Paciente |

| | |
|---------------------------------|-----------------------------------|
| RF8 Insertar datos de usuario. | RF17 Exportar XML de Actividades |
| RF9 Modificar datos de usuario. | RF18 Exportar Reporte Estadístico |

Requisitos no funcionales

El software debe contar con una impresora, con el objetivo de poder exportar los reportes estadísticos generados por la aplicación, además de una base de datos en un servidor Microsoft SQL Server, el cual soportará grandes volúmenes de datos y velocidad de procesamiento. Debe tener plataformas Windows y tener al menos 512 de RAM para que el sistema gestor de base de datos no tenga problemas de rendimiento y disponibilidad.

2.7. Descripción del sistema propuesto

Aquí se muestra los actores en el sistema, así como la descripción de los casos de usos que representan los requerimientos funcionales con el objetivo de obtener un mejor entendimiento del sistema que se propone.

2.7.1. Descripción de los actores del sistema

Seguidamente se muestra los actores que interactúan con el sistema, los cuales se pueden centrar en uno solo, “Especialista”, el cual será el encargado de gestionar los pacientes, configurar las actividades-ejercicios que realizarán posteriormente, entre otras funcionalidades.

| Actores | Justificación |
|----------------|---|
| Especialista | Este actor es al que está dirigido la aplicación, será quien hará uso de ella, así como de todas sus funcionalidades como Gestionar Pacientes, Configurar actividades-ejercicios a pacientes, Generar reportes estadísticos, etc. |

Tabla 2 Actores que interactúan con el Sistema

2.7.2. Diagrama de casos de usos del sistema

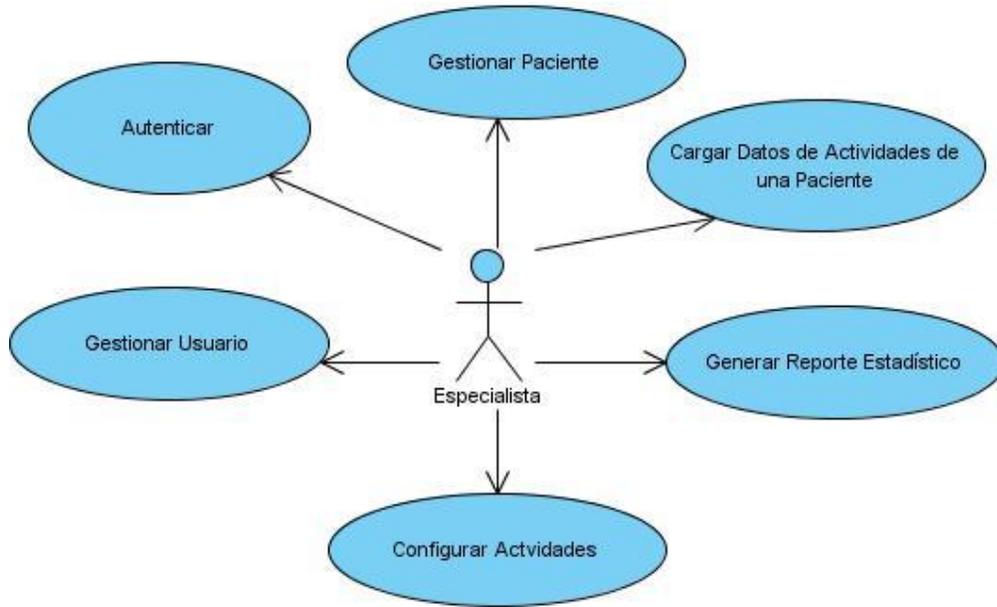


Ilustración 2 Diagrama de casos de usos del sistema

2.7.3. Descripción de los casos de usos del sistema

En la descripción de los casos de estudio se puede percibir la interacción del actor con el sistema, así como conocer las especificidades propias de cada caso de uso o funcionalidad.

Seguidamente se muestran dichas descripciones.

CUS: Gestionar paciente

| | |
|-----------------------|---|
| Caso de Uso: | Gestionar Paciente |
| Actores: | Especialista |
| Resumen: | Aquí el especialista tendrá la posibilidad de poder insertar, modificar o eliminar algún paciente, se termina el caso de uso al completarse alguna sección. |
| Precondiciones | Para realizar esta funcionalidad el especialista debe haberse autenticado primero. |

CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN

| | |
|--|--|
| Referencias | RF1, RF2, RF3, RF4, RF5, RF6 |
| Flujo Normal de Eventos | |
| Acción del Actor | Respuesta del Sistema |
| 1. Seleccionar la pestaña Gestionar Paciente. | 1.1 El sistema muestra de operaciones a realizar. |
| 2. El usuario selecciona la opción a realizar. | <p>2.1 Si el actor elige la opción Insertar, ir a Sección “Insertar Paciente”.</p> <p>2.2 Si el actor elige la opción Modificar, ir a Sección “Modificar Paciente”.</p> <p>2.3 Si el actor elige la opción Eliminar, ir a Sección “Eliminar Paciente”.</p> |
| Poscondiciones | Se logra insertar, modificar o eliminar un paciente. |
| Sección: “Insertar Paciente” | |
| Flujo Normal de Eventos | |
| Acción del Actor | Respuesta del Sistema |
| | <p>1. Muestra un formulario dando la posibilidad de introducir los datos del Paciente:</p> <p style="padding-left: 40px;">CI, Nombre, Apellidos, Edad, Distancia Int, Agudeza Visual OD y OI, Stereopsis</p> <p>Seleccionar: Ojo Vago y las opciones de Aceptar y Cancelar (Ver Alternativa 1: “Cancelar operación.”)</p> |

CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN

| | |
|--|---|
| 2. Introduce los datos del Paciente: | 2.1 Valida los datos. Si hay datos incorrectos, ver Alternativa2 : “Existen campos incorrectos.” Si hay campos vacíos, ver Alternativa 3 “Existen campos vacíos”. Si existe el paciente, ver Alternativa4 “Ya existe el paciente”. |
| | 3. Se inserta el Paciente a la base de datos. |
| | 4. El caso de uso termina. |
| Flujos Alternos: “Cancelar operación.” | |
| Acción del Actor | Respuesta del Sistema |
| 1. Selecciona la opción de Cancelar. | 1. Regresa a la vista anterior. |
| | 2. El caso de uso termina. |
| Flujos Alternos: “Existen campos incorrectos” | |
| Acción del Actor | Respuesta del Sistema |
| | 1. Muestra un indicador sobre los campos incorrectos con la información que se debe introducir en ellos. |
| Flujos Alternos: “Existen campos vacíos.” | |
| Acción del Actor | Respuesta del Sistema |
| | 1. Muestra un indicador al lado de los campos vacíos. |
| Flujos Alternos: “Ya existe el paciente” | |
| Acción del Actor | Respuesta del Sistema |

CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN

| | 1. Muestra un mensaje de error “Ya existe el paciente”. |
|--|--|
| Sección: Modificar Paciente | |
| Flujo Normal de Eventos | |
| Acción del Actor | Respuesta del Sistema |
| 1. El especialista selecciona un paciente ya registrado. | 1. 1.1 El sistema guarda la selección del paciente. |
| 2. El especialista accede a la opción Editar. | 2.1 Muestra un formulario dando la posibilidad de modificar los datos actuales del Paciente seleccionado: <div style="text-align: center;">CI, Nombre, Apellidos, Edad, Distancia Int, Agudeza Visual OD y OI, Stereopsis</div> Seleccionar: Ojo Vago y las opciones de Aceptar y Cancelar(Ver Alternativa 1: “Cancelar operación.”) |
| 3. Modifica los datos del Paciente: | 3. Valida los datos. Si hay datos incorrectos, ver Alternativa 2: “Existen campos incorrectos.” Si hay campos vacíos, ver Alternativa 3 “Existen campos vacíos”. Si existe el paciente, ver Alternativa 4 “Ya existe el paciente”. |
| | 5. Se modifica el paciente de la base de datos. |
| | 6. El caso de uso termina. |
| Flujos Alternos: “Cancelar operación.” | |

CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN

| Acción del Actor | Respuesta del Sistema |
|--|--|
| 1. Selecciona la opción de Cancelar. | 1. Regresa a la vista anterior. |
| | 2. El caso de uso termina. |
| Flujos Alternos: "Existen campos incorrectos" | |
| Acción del Actor | Respuesta del Sistema |
| | 1. Muestra un indicador sobre los campos incorrectos con la información que se debe introducir en ellos. |
| Flujos Alternos: "Existen campos vacíos." | |
| Acción del Actor | Respuesta del Sistema |
| | 1. Muestra un indicador al lado de los campos vacíos. |
| Flujos Alternos: "Ya existe el paciente" | |
| Acción del Actor | Respuesta del Sistema |
| | 1. Muestra un mensaje de error "Ya existe el paciente". |
| Sección: Eliminar Paciente | |
| Flujo Normal de Eventos | |
| Acción del Actor | Respuesta del Sistema |
| 1. El especialista selecciona un paciente ya registrado y accede a la opción Eliminar. | 1.1. Valida los datos. Si hay campos vacíos, ver Alternativa 2: "Selección Invalida". |
| | 2. Elimina el Paciente. |

CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN

| | |
|--|--|
| | 3. El caso de uso termina. |
| Flujos Alternos: “Selección Invalida” | |
| Acción del Actor | Respuesta del Sistema |
| | 1. Muestra el mensaje de error “Debe seleccionar un valor valido”. |

CUS: Configurar actividades

| | |
|--|---|
| Caso de Uso: | Configurar actividades |
| Actores: | Especialista |
| Resumen: | El sistema guarda en un fichero XML la configuración del paciente seleccionado, así como una serie de actividades y ejercicios. |
| Precondiciones | Debe haber ejecutado al menos una vez el caso de uso “Insertar Paciente”. |
| Referencias | RF15 |
| Flujo Normal de Eventos | |
| Acción del Actor | Respuesta del Sistema |
| 1. Seleccionar la pestaña Configurar Actividades | 1.1 El sistema muestra un formulario donde se encuentran los CI de todos los pacientes ya registrados. Además da la opción de Cancelar, ver Alternativa 1 “Cancelar Operación” |
| 2. Selecciona el CI de un Paciente. | 2.1 Guarda la selección del paciente. 2.2 El sistema le muestra los diferentes tipos de |

CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN

| | |
|--|--|
| | actividades a realizar. |
| 3. Selecciona la opción Configurar Paciente. | 3.1 Valida si seleccionó el paciente. Si hay campos vacíos, ver Alternativa 2 : “Selección Invalida”. |
| 4. Selecciona el nombre del fichero y la dirección donde exportará el fichero XML. | 4.1 Exporta un fichero XML en una dirección seleccionada por el actor, los datos personales del paciente: CI, Nombre, Apellidos, Edad, Distancia Int, Agudeza Visual OD y OI, Stereopsis, Ojo Vago. |
| 5. Selecciona las actividades a realizar. | 5. 1 Adiciona las actividades a la lista de actividades a configurar. |
| 6. Selecciona la opción Configurar Actividades. | 6.1 El sistema acepta las actividades seleccionadas y pide donde exportar esta configuración de actividades, lo hace mediante un fichero XML. |
| 7. Selecciona el nombre del fichero y la dirección donde exportará el fichero XML. | 7.1 Exporta un fichero XML en una dirección seleccionada por el actor la configuración de las actividades a realizar. |
| | 8. Se terminó el caso de uso. |
| Flujos Alternos: “Cancelar operación.” | |
| Acción del Actor | Respuesta del Sistema |
| 1. Selecciona la opción de Cancelar. | 1. Regresa a la vista anterior. |
| | 2. El caso de uso termina. |
| Flujos Alternos: “Selección Invalida | |
| Acción del Actor | Respuesta del Sistema |

CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN

| | |
|-----------------------|--|
| | 1. No habilita las operaciones de Configurar Paciente, ni Configurar Actividades. |
| Poscondiciones | Se exportó en un fichero de formato XML la configuración de una serie de actividades a realizar a un paciente, y en otro los datos personales de dicho paciente. |

CUS: Generar reporte estadístico

| | |
|-----------------------|---|
| Caso de Uso: | Generar Reporte Estadístico |
| Actores: | Especialista |
| Resumen: | El caso de uso inicia cuando el actor accede a la opción Generar Reporte, el sistema genera el reporte. El actor puede exportar el reporte, el caso de uso termina. |
| Precondiciones | Debe haberse insertado al menos un paciente y habersele cargado sus datos médicos. |
| Referencias | RF14, RF18 |

Flujo Normal de Eventos

| Acción del Actor | Respuesta del Sistema |
|---|---|
| 1. El caso de uso inicia cuando el actor accede a la pestaña Generar Reporte. | 1.1 Se muestra un formulario donde se debe seleccionar los siguientes atributos: CI del Paciente, Actividad, No. Ejercicio, Atributo |
| 2. Selecciona los datos que considera como criterios para generar el reporte. | 2.1 Valida los datos, si quedó alguno vacío, ver Alternativa 1 "Selección Inválida" |

CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN

| | |
|--|--|
| 3. Selecciona la opción de Generar Reporte. | 3.1 Busca los datos que cumplen con los criterios seleccionados, consultando las entidades y genera el reporte (una gráfica). Si desea exportar el reporte, ver Alternativa 2 "Exportar Reporte". Si selecciona la opción Limpiar gráfica, ver Alternativa 3 "Limpiar Gráfica". |
| | 4. El caso de uso termina |
| Flujos Alternos: "Selección Inválida" | |
| Acción del Actor | Respuesta del Sistema |
| | 1. Muestra el mensaje de error "Debe seleccionar correctamente los campos". |
| Flujos Alternos: "Exportar Reporte" | |
| Acción del Actor | Respuesta del Sistema |
| | 1. Muestra una ventana que puede exportar el reporte en forma de imagen o pdf., dando la opción de imprimir. |
| | 2. El caso de uso termina. |
| Flujos Alternos: "Limpiar Gráfica" | |
| Acción del Actor | Respuesta del Sistema |
| | 1. Limpia la gráfica generada, además de todos los campos anteriormente seleccionados. |
| | 2. El caso de uso termina. |

CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN

| | |
|-----------------------|--|
| Poscondiciones | Muestra el Reporte y se exporta en caso de haber seleccionado la opción. |
|-----------------------|--|

CUS: Cargar Datos de Actividades de un Paciente:

| | |
|-----------------------|--|
| Caso de Uso: | Cargar Datos de Actividades de un Paciente |
| Actores: | RF9 |
| Resumen: | El sistema carga de un fichero XML los datos médicos de las actividades y ejercicios de un paciente en específico. |
| Precondiciones | Debe haberse ejecutado al menos una vez el caso de uso “Insertar Paciente“ y habersele ejecutado a este paciente el caso de uso “Configurar Actividad“ al menos una vez. |
| Referencias | RF7 |

Flujo Normal de Eventos

| Acción del Actor | Respuesta del Sistema |
|---|---|
| 1. Selecciona la opción “Cargar Datos de XML. ” | 1.1 El sistema le pide la ubicación del fichero XML y da la posibilidad de aceptar y cancelar. |
| 2. Selecciona la ubicación. | <p>2.1 Si selecciona la opción Cancelar ver Alternativa 1 “Cancelar operación”.</p> <p>2.1 Si selecciona la opción Aceptar, Valida los datos. Si el paciente al que se le cargó los datos no está en la base de datos, ver Alternativa 2 “El paciente no se encuentra registrado”.</p> <p>2.3 El sistema carga los datos del fichero y los introduce en la base de datos.</p> |
| | 1. El caso de uso termina. |

CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN

| Flujos Alternos: "Cancelar operación." | |
|---|---|
| Acción del Actor | Respuesta del Sistema |
| 1. Selecciona la opción de Cancelar. | 2. Regresa a la vista anterior. |
| | 3. El caso de uso termina. |
| Flujos Alternos: "El paciente no se encuentra registrado" | |
| Acción del Actor | Respuesta del Sistema |
| | 1. Muestra el mensaje de error "El paciente no se encuentra registrado". |
| | 2. El caso de uso termina. |
| Poscondiciones | El sistema importa de una dirección selecciona por el usuario un fichero XML con la configuración de los datos médicos de paciente en específico. |

CUS: Autenticar

| | |
|-----------------------|---|
| Caso de Uso: | Autenticar |
| Actores: | Especialista |
| Resumen: | El caso de uso inicia cuando el actor desea entrar al sistema con las credenciales que les fueron asignadas, el sistema brinda la posibilidad de introducir los parámetros, el actor introduce los datos, el sistema le permite la entrada, el caso de uso termina. |
| Precondiciones | |
| Referencias | RF13 |

Flujo Normal de Eventos

CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN

| Acción del Actor | Respuesta del Sistema |
|--|---|
| 1. El caso de uso inicia cuando el actor desea entrar al sistema. | 1. Brinda la posibilidad de introducir los parámetros elementales: <ul style="list-style-type: none"> • Usuario • Contraseña |
| 2. El usuario introduce los parámetros obligatorios: <ul style="list-style-type: none"> • Usuario • Contraseña | 2.1. Si hay datos vacíos, ver Alternativa 1 : “Usuario o clave incorrecta”. Si se han introducido datos incorrectos, ver Alternativa 2 “Usuario o clave incorrecta” |
| | 5. El caso de uso termina. |
| Flujos Alternos: “Usuario o clave incorrecta” | |
| Acción del Actor | Respuesta del Sistema |
| | 1. Muestra el mensaje de error “Usuario o Contraseña incorrecto”. |
| Poscondiciones | El usuario accede a la aplicación. |

CUS: Gestionar usuario

| | |
|---------------------|--|
| Caso de Uso: | Gestionar Usuario |
| Actores: | Especialista |
| Resumen: | Aquí el especialista tendrá la posibilidad de poder insertar, modificar o eliminar algún usuario, se termina el caso de uso al completarse alguna sección. |

| | |
|--|--|
| Precondiciones | Para realizar esta funcionalidad el especialista debe haberse autenticado primero. |
| Referencias | RF8, RF9, RF10, RF11, RF12 |
| Flujo Normal de Eventos | |
| Acción del Actor | Respuesta del Sistema |
| 1. Seleccionar la pestaña Gestionar Usuario. | 1.1 El sistema muestra de operaciones a realizar. |
| 2. El usuario selecciona la opción a realizar. | 2.1 Si el actor elige la opción Insertar, ir a Sección "Insertar Usuario". 2.2 Si el actor elige la opción Modificar, ir a Sección "Modificar Usuario". 2.3 Si el actor elige la opción Eliminar, ir a Sección "Eliminar Usuario". |
| Poscondiciones | Se logra insertar, modificar o eliminar un usuario. |
| Sección: Insertar Usuario | |
| Flujo Normal de Eventos | |
| Acción del Actor | Respuesta del Sistema |
| | 1. Muestra un formulario dando la posibilidad de introducir los datos del Usuario: Usuario, Contraseña, Repetir Contraseña |

CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN

| | |
|--|--|
| | y las opciones de Aceptar y Cancelar(Ver Alternativa 1 : “Cancelar operación.”) |
| 2. Introduce los datos del Usuario: | 2.1. Valida los datos. Si las contraseñas no coinciden, ver Alternativa 2 :”Contraseñas no coinciden”. Si hay campos vacíos, ver Alternativa 3 “Existen campos vacíos”. Si existe el usuario, ver Alternativa 4 “Ya existe el usuario”. |
| | 3. Se inserta el Usuario a la base de datos. |
| | 4. El caso de uso termina. |
| Flujos Alternos: “Cancelar operación.” | |
| Acción del Actor | Respuesta del Sistema |
| 1. Selecciona la opción de Cancelar. | 1. Regresa a la vista anterior. |
| | 2. El caso de uso termina. |
| Flujos Alternos: “Contraseñas no coinciden” | |
| Acción del Actor | Respuesta del Sistema |
| | 1. Muestra el mensaje “La contraseña no se confirmó correctamente”. |
| Flujos Alternos: “Existen campos vacíos.” | |
| Acción del Actor | Respuesta del Sistema |
| | 1. Muestra un indicador al lado de los campos vacíos. |

| Flujos Alternos: "Ya existe el usuario" | |
|---|---|
| Acción del Actor | Respuesta del Sistema |
| | 1. Muestra un mensaje de error "Ya existe el usuario". |
| Sección: "Modificar Usuario" | |
| Flujo Normal de Eventos | |
| Acción del Actor | Respuesta del Sistema |
| | <p>1. Muestra un formulario dando la posibilidad de modificar los datos actuales del Usuario:</p> <p style="padding-left: 40px;">Usuario, Contraseña, Repetir Contraseña</p> <p style="padding-left: 40px;">Seleccionar: Usuario a modificar</p> <p>y las opciones de Aceptar y Cancelar(Ver Alternativa 1: "Cancelar operación.")</p> |
| 2. Modifica los datos del Usuario: | 2.1 Valida los datos. Si hay datos incorrectos, ver Alternativa 2 : "Existen campos incorrectos." Si hay campos vacíos, ver Alternativa 3 "Existen campos vacíos". Si existe el usuario, ver Alternativa 4 "Ya existe el usuario". |
| | 3. Se modifica el usuario. |
| | 4. El caso de uso termina. |
| Flujos Alternos: "Cancelar operación." | |

CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN

| Acción del Actor | Respuesta del Sistema |
|--|---|
| 1. Selecciona la opción de Cancelar. | 1. Regresa a la vista anterior. |
| | 2. El caso de uso termina. |
| Flujos Alternos: "Existen campos incorrectos" | |
| Acción del Actor | Respuesta del Sistema |
| | 1. Muestra un indicador sobre los campos incorrectos con la información que se debe introducir en ellos. |
| Flujos Alternos: "Existen campos vacíos." | |
| Acción del Actor | Respuesta del Sistema |
| | 1. Muestra un indicador al lado de los campos vacíos. |
| Flujos Alternos: "Ya existe el usuario" | |
| Acción del Actor | Respuesta del Sistema |
| | 1. Muestra un mensaje de error "Ya existe el usuario". |
| Sección: "Eliminar Usuario" | |
| Flujo Normal de Eventos | |
| Acción del Actor | Respuesta del Sistema |
| | 1. El sistema muestra un formulario mostrando los usuarios ya registrados dando la opción de Eliminar o Cancelar. |

| | |
|---|--|
| 2. El especialista selecciona un usuario ya registrado y accede a la opción Eliminar. | 2.1 Valida los datos. Si hay campos vacíos, ver Alternativa 1: “Selección Invalida” . |
| | 3. Elimina el Usuario |
| | 4. El caso de uso termina. |
| Flujos Alternos: “Selección Invalida” | |
| Acción del Actor | Respuesta del Sistema |
| | 1. Muestra el mensaje de error “Debe seleccionar un valor valido”. |

2.8. Descripción de la Arquitectura y Fundamentación

La aplicación estará bajo el funcionamiento del patrón de arquitectura MVC, el cual tendrá organizado las clases, librerías y todos aquellos archivos que interfieren en la aplicación.

Esta aplicación cuenta con clases estrechamente relacionadas. Dichas clases están conectadas a una base de datos con el objetivo de manejar desde una interfaz de fácil navegación funcionalidades que darán solución a los requisitos del software.

La BD presenta 9 tablas relacionadas, las cuales contienen de manera organizada los datos más importantes de los pacientes así como de las actividades y ejercicios que realizarán para un posterior análisis por especialistas, además de los usuarios que interactuarán con el módulo.

En este capítulo se ha brindado una clara descripción de los métodos y herramientas utilizadas, además de los requisitos funcionales y no funcionales del sistema, así como la definición detallada de los casos de usos. Con lo antes mencionado se tomó claridad a la concepción del sistema a construir y se sentaron bases de la fase de análisis. Los resultados serán punto de partida en la próxima fase.

Capítulo 3: Diseño e Implementación

3.1. Introducción

En este capítulo se muestra detalladamente la estructura de la aplicación, teniendo como punto de partida la fase de análisis. Definiendo el modelado del diagrama de clases, el diagrama de secuencia de los casos de usos o funcionalidades, así como la descripción de cada una de las clases con sus respectivas definiciones de los atributos, métodos con que consta con el objetivo de implementar las respuestas a los casos de usos que presenta dicha aplicación.

3.2. Patrón de Arquitectura Utilizado

Para el desarrollo del sistema fue necesaria la utilización del patrón de arquitectura “por capas”, ya que en este se define como organizar el modelo de diseño a través de capas que pueden estar físicamente distribuidas. Además que simplifica la comprensión y la organización del desarrollo del sistema.

Para esto se utilizaron las siguientes capas:

Presentación: Esta capa reúne todos los aspectos del software que tienen que ver con las interfaces y la interacción con los diferentes tipos de usuarios. Además de que es responsable de presentar información e interactuar con capas inferiores.

Lógica del Negocio: Implementan la lógica de negocio de la aplicación. Es responsable de implementar operaciones solicitadas por los clientes a través de la capa de presentación.

Acceso a Datos: Esta es la capa que propicia la conexión a la Base de Datos del sistema, la cual se realiza a través de la clase Conexión. La mayoría de las aplicaciones y servicios necesitan obtener acceso al repositorio de datos en un momento determinado del proceso de negocios. Por lo tanto, es razonable abstraer la lógica necesaria para obtener acceso a los datos en una capa, ya que de este modo se centraliza la funcionalidad de acceso a datos y se facilita la configuración y el mantenimiento de la misma.



Ilustración 3 Patrón de Arquitectura por Capas

3.3. Patrones de Diseño Utilizados

Para el desarrollo de la esta aplicación se propone la utilización de los Bajo Acoplamiento y Creador. A través de estos patrones el sistema será más fácil de mantener, modificar y adicionar nuevos módulos o clases que brinden nuevas funcionalidades.

- Creador: Define quien debería ser el responsable de la creación de una nueva instancia de alguna clase. Guía la asignación de responsabilidades relacionadas con la creación de objetos. El propósito fundamental de este patrón es encontrar un creador que se debe conectar con el objeto producido en cualquier evento.
- Bajo Acoplamiento: Este patrón plantea la idea de tener las clases lo menos posibles atadas entre sí. De tal forma que en caso de producirse una modificación en alguna de ellas, la repercusión en el resto de las clases sea lo menor posible, potenciando así la reutilización y disminuyendo la dependencia entre las clases. El acoplamiento es una medida de la fuerza con que una clase está relacionada a otras clases, una clase con bajo o débil acoplamiento no depende de muchas otras.
- Fabricación pura: el cual se utiliza en la creación de clases que se encarguen de desarrollar funcionalidades específicas, para de esta forma aumentar la cohesión. Ej. (ClassConnection)

3.4. Diagrama de Clases del Sistema

Diagrama estático en el cual se muestra la estructura de un sistema detallando las clases que serán utilizadas dentro de éste y las relaciones que existen entre ellas. Permite visualizar las relaciones entre las clases que involucran el sistema, las cuales pueden ser asociativas, de herencia, de uso y de convencimiento. Un diagrama de clases está compuesto por los siguientes elementos: Clase: atributos, métodos y visibilidad. Relaciones: Herencia, Composición, Agregación, Asociación y Uso.

CUS: Gestionar Paciente

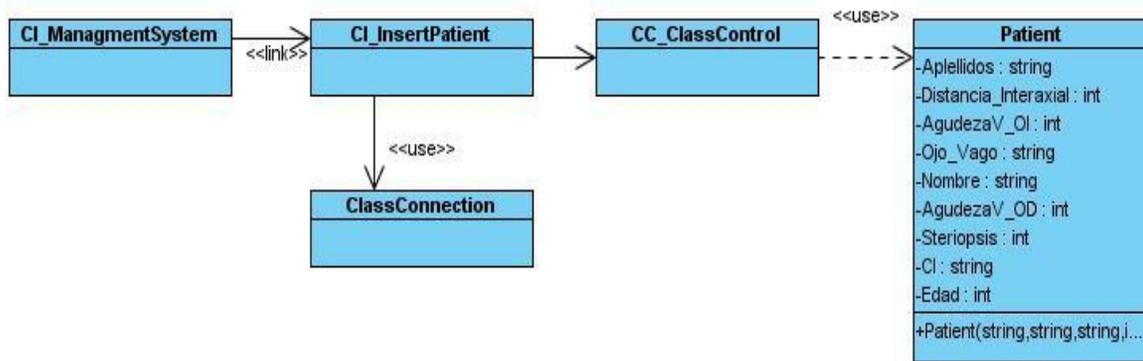


Ilustración 4 Escenario. Insertar Paciente

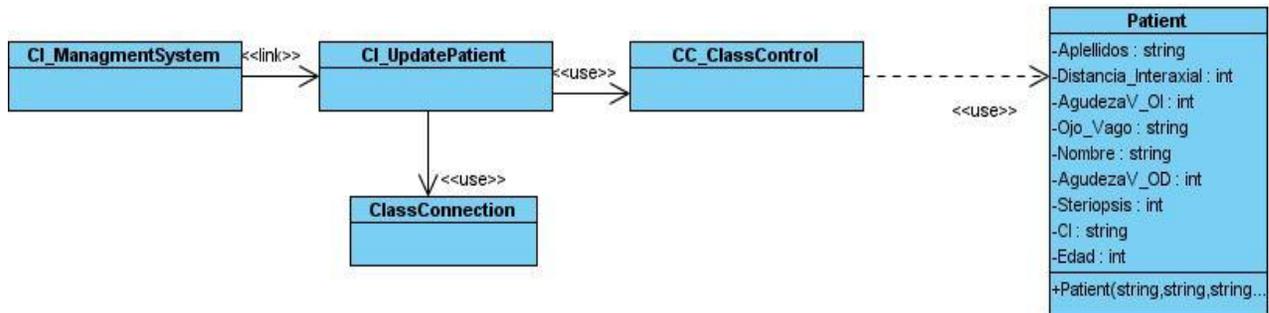


Ilustración 5 Escenario. Modificar Paciente

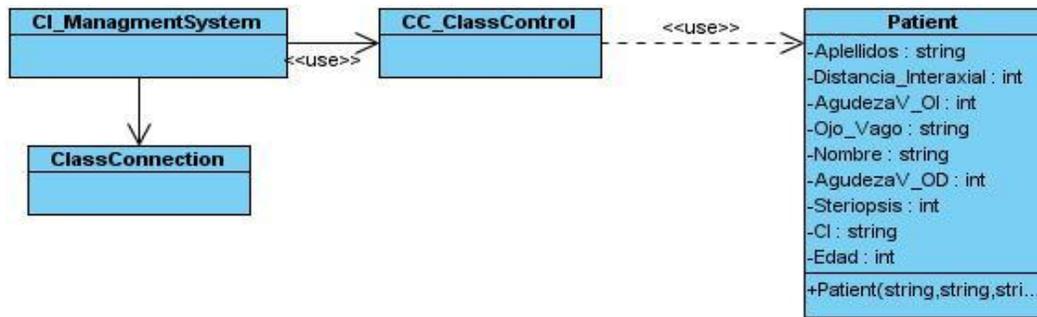
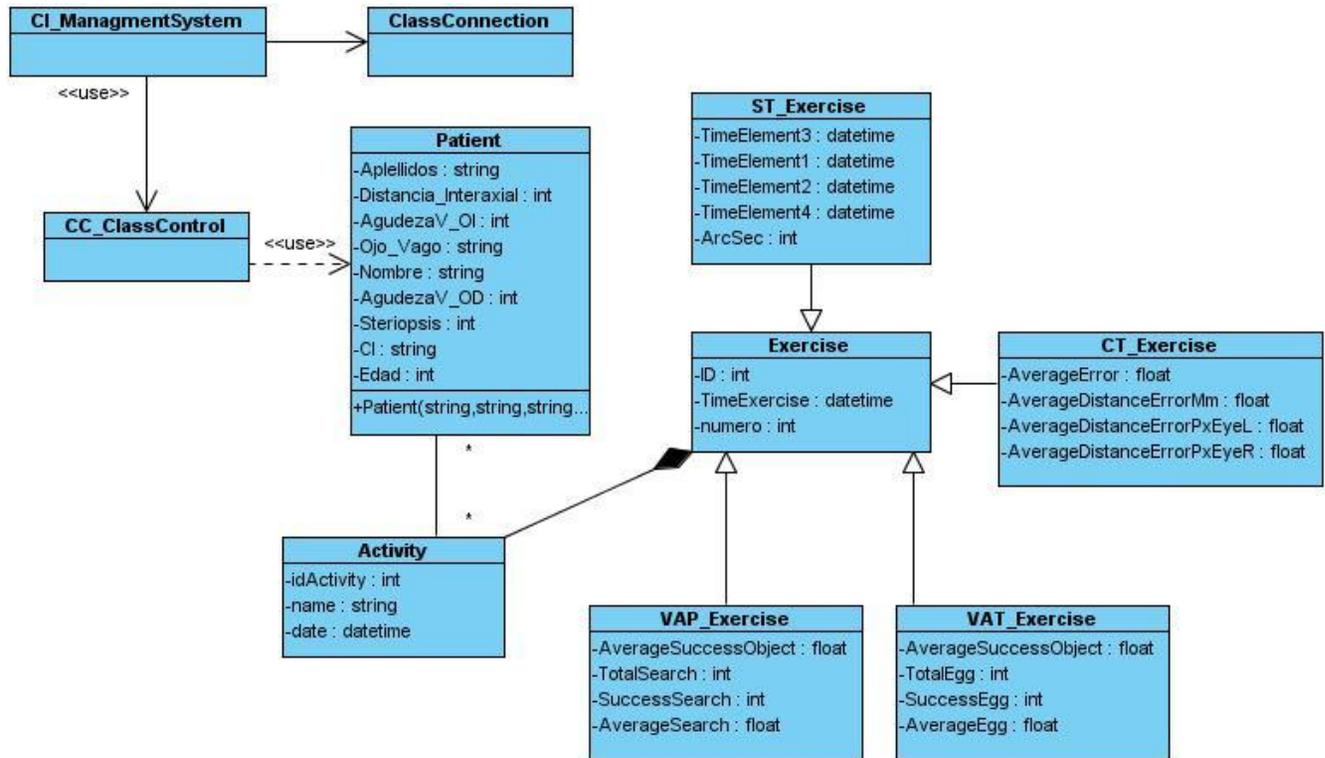


Ilustración 6 Escenario. Eliminar Paciente

Se quiere resaltar que el siguiente DCD detalla los casos de usos:

- Configurar Actividades
- Gestionar Reportes Estadísticos
- Cargar Datos de Actividades de una Paciente



CUS: Gestionar Usuario

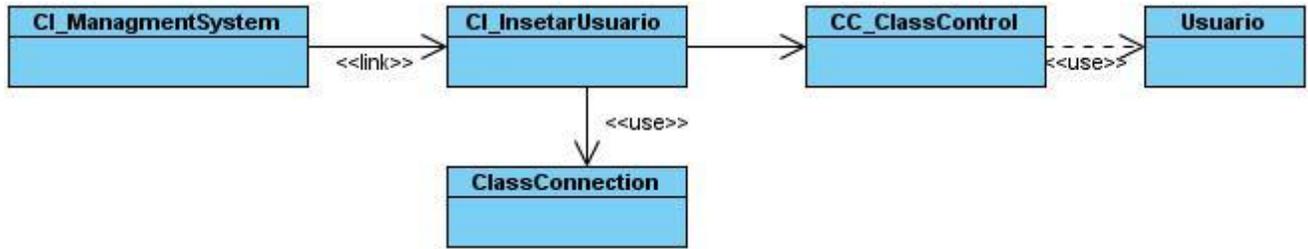


Ilustración 7 Escenario. Insertar Usuario

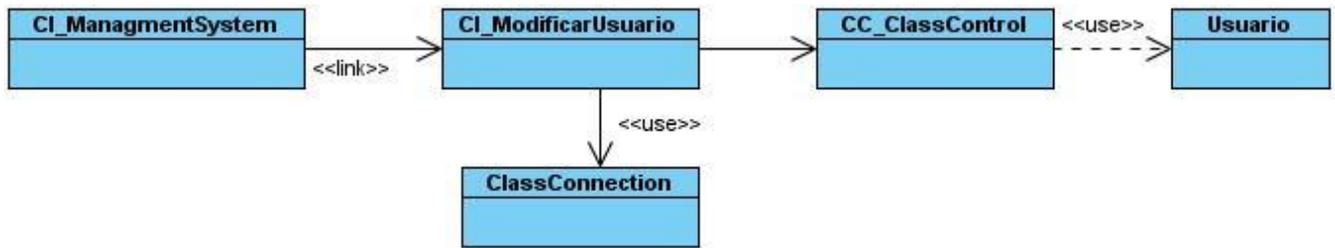


Ilustración 8 Escenario. Modificar Usuario

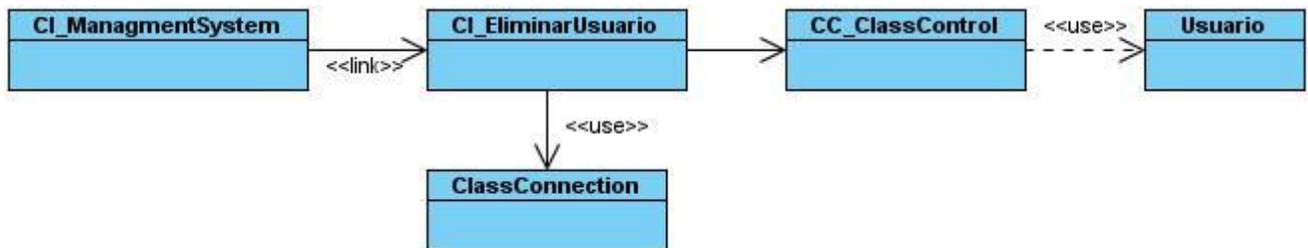


Ilustración 9 Escenario. Eliminar Usuario

CUS: Aumentar

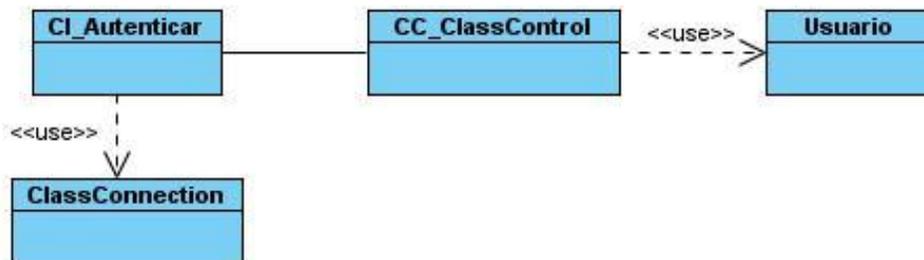


Ilustración 10 CUS. Aumentar

3.4.1. Descripción de las clases más significativas

Seguidamente se muestran las descripciones de las principales clases en el sistema

| | |
|-----------------------------------|---|
| Nombre: CI_ManagmentSystem | |
| Tipo de Clase | Interfaz |
| Atributos | |
| Nombre | Tipo |
| activityTableAdapter | ActivityTableAdapter |
| direccion | string |
| actividad | int |
| IdActividad | int |
| Act | struct |
| Actividades | ArrayList |
| classControl | ClassControl |
| Métodos | |
| Nombre | Descripción |
| voidCI_ManagmentSystem() | Inicializa la interfaz y los atributos auxiliares. |
| void InsertPatient_Click() | Muestra la interfaz CI_InsertPatient |
| voidEliminarPatient_Click() | Elimina de la BD un paciente seleccionado por el actor. |
| voidConfPatient_Click() | Exporta en un fichero XML los datos personales de un paciente seleccionado por el actor. |
| voidConfActividades_Click() | Exporta en un fichero XML la configuración de un grupo de actividades y ejercicios de un paciente seleccionado. |
| voidCargarDatosXML() | Carga de un fichero XML una configuración de datos médicos de actividades realizadas por un paciente registrado en el sistema. |
| voidGenerarReporte() | Se formará de forma gráfica un conjunto de datos preestablecidos, una vez que el actor seleccionó los campos de búsqueda. Mediante esto se puede analizar el seguimiento y evolución de uno o varios pacientes. |
| voidExportarReporte() | Da la posibilidad de exportar en forma pdf o imagen el reporte |

| | |
|--------------------------|---|
| | generado. |
| boolValidarCI(string ci) | Devuelve true si este CI está registrado. |

Tabla 3 Clase ManagementSystem

| | |
|--|---|
| Nombre: CC_ClassControl | |
| Tipo de Clase | Controladora |
| Atributos | |
| Nombre | Tipo |
| connections | ClassConnection |
| listPatient | ArrayList<Patient> |
| listUsuario | ArrayList<Usuario> |
| Métodos | |
| Nombre | Descripción |
| voidCC_ClassControl (ClassConnection) | Inicializa la clase y atributos para la gestión de los pacientes. |
| boolAddPatient(string CI, string nombre, string apellid, intedad, int DI, string ojoVago, int AVOD, int AVOI, intSterep) | Añade un paciente a la base de datos y devuelve verdadero o falso si se puede |
| void UpdatePatient (string ciActual, string ciNew, string nomb, string apell, intDI,intedad, string ojoVago, int AVOD, int AVOI, intStereop) | Modifica los datos actuales de un paciente seleccionado por el actor. |
| voidDeletePatient(string ci) | Elimina de la base de datos un paciente dado. |
| void ConfAct(XmlDomument doc, AcivityTableAdapter) | Configura un grupo de actividades y ejercicios a un paciente seleccionado. |
| voidDeleteActivity(stringact) | Elimina una actividad dada a un paciente dado. |
| DataReaderGenerarReporte(string, string, string, string) | Devuelve una respuesta a la búsqueda realizada según los atributos. |
| boolValidadAutent(string user, string pass) | Devuelve verdadero o falso si esta correcta la autenticación. |
| boolInsertUser(Usuario u) | Inserta si se puede, un usuario en la base de datos. |

CAPÍTULO 3: DISEÑO E IMPLEMENTACIÓN

| | |
|---|--|
| boolUpdateUser(string userAct, string userNew, passNew) | Modifica sí se puede, un usuario seleccionado por el actor. |
| voidDeleteUser(stringuser) | Elimina un usuario seleccionado. |
| ArraylistListActiv_Patient(string ci) | Devuelve la lista de actividades de un paciente dado. |
| Void Actualizar() | Actualiza las listas de paciente y usuarios. |
| DatareaderListarDatos(string CI, string act,intnum,stringatrib) | Devuelve una lista de datos referente a la búsqueda con dichos parámetros. |

Tabla 4 Clase ClassControl

| | |
|--|---|
| Nombre: CE_Patient | |
| Tipo de Clase | Entidad |
| Atributos | |
| Nombre | Tipo |
| AgudezaV_OD | string |
| AgudezaV_OI | string |
| Nombre | string |
| Apellidos | string |
| Edad | int |
| Dist_int | int |
| CI | string |
| Stereopsis | int |
| OjoVago | string |
| listActiv | ArrayList |
| Métodos | |
| Nombre | Descripción |
| void CC_ClassControl (string,string, string, string, int, int, string, int string) | Inicializa la clase y atributos de un paciente. |

Tabla 5 Clase Patient

| | |
|--------------------------------|----------|
| Nombre: ClassConnection | |
| Tipo de Clase | Auxiliar |
| Atributos | |

| Nombre | Tipo |
|---|--|
| string_connection | string |
| database | SqlConnection |
| Métodos | |
| Nombre | Descripción |
| void ClassConnection(string _string_connection) | Inicializa la clase y atributos para la conexión a la base de datos. |
| void OpenConnection() | Abre la conexión con el servidor. |
| void CloseConnection() | Cierra la conexión con el servidor. |

Tabla 6 ClassConnection

3.5. Descripción de la implementación por Casos de Uso

Aquí se muestran las descripciones de los principales casos de usos del sistema, mostrando detalladamente cómo se manifestó la implementación de cada uno de ellos. Estos diagramas muestran la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada caso de uso, mostrando de forma secuencial los envíos de mensajes entre objetos. El diagrama puede así mismo mostrar los flujos de datos intercambiados durante el envío de mensajes.

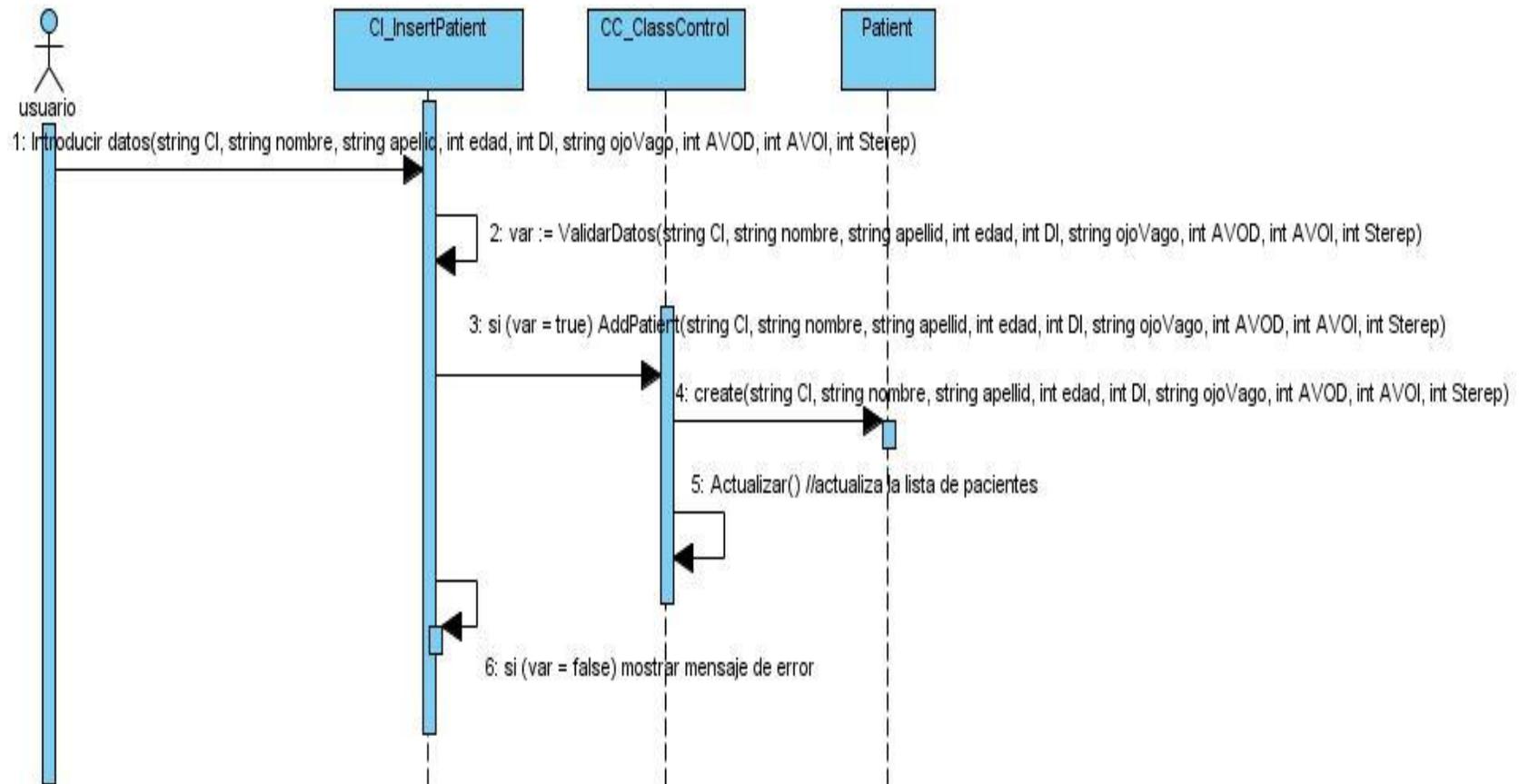


Ilustración 11 Escenario Insertar Paciente

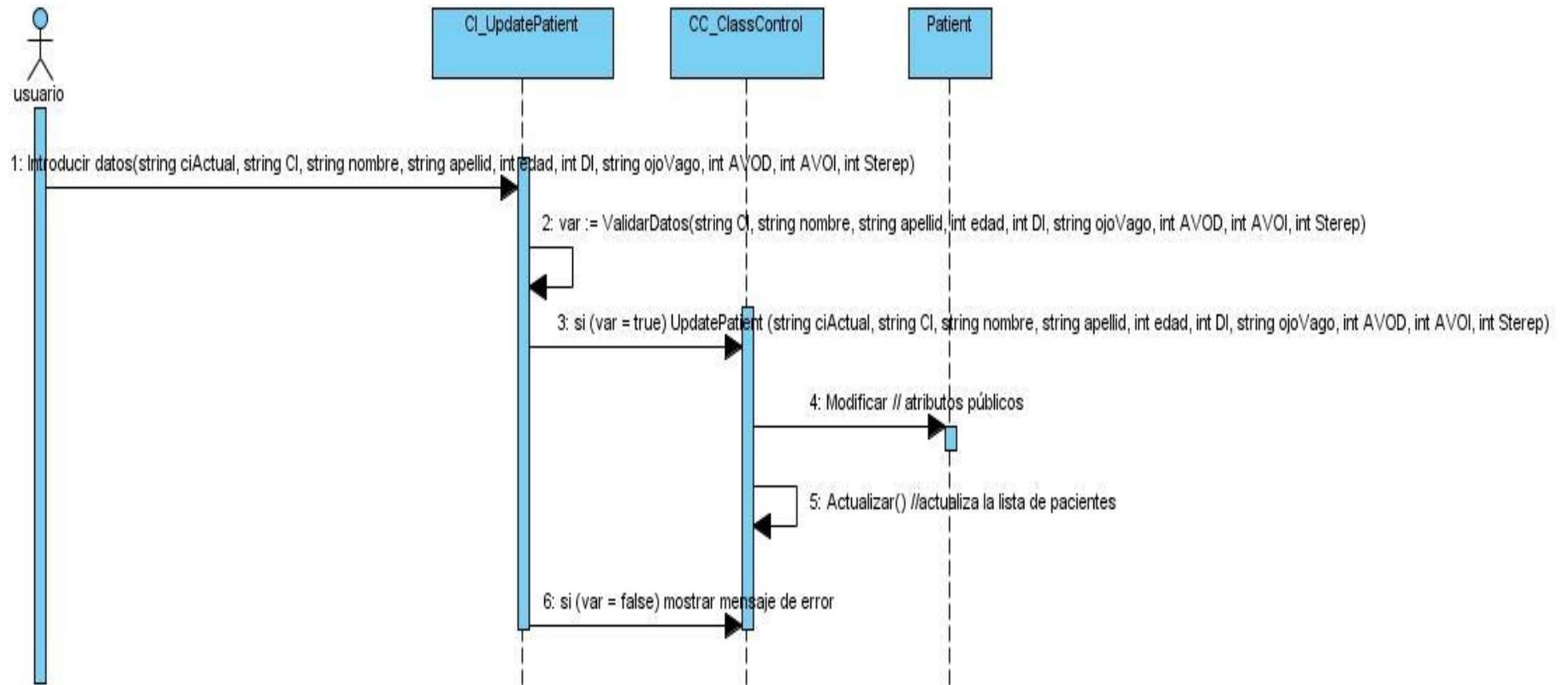


Ilustración 12 Escenario Modificar Paciente

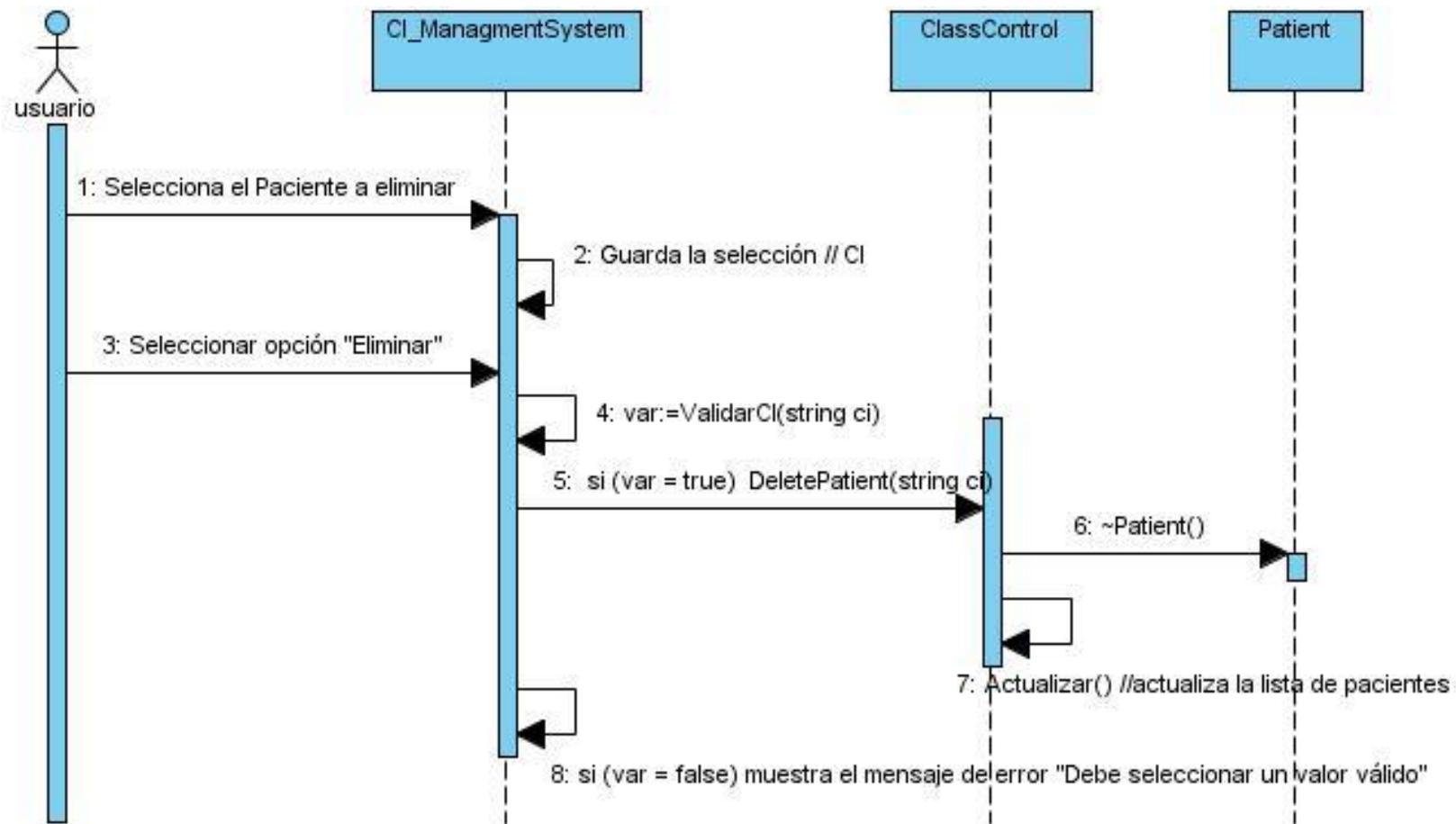


Ilustración 13 Escenario Eliminar Paciente

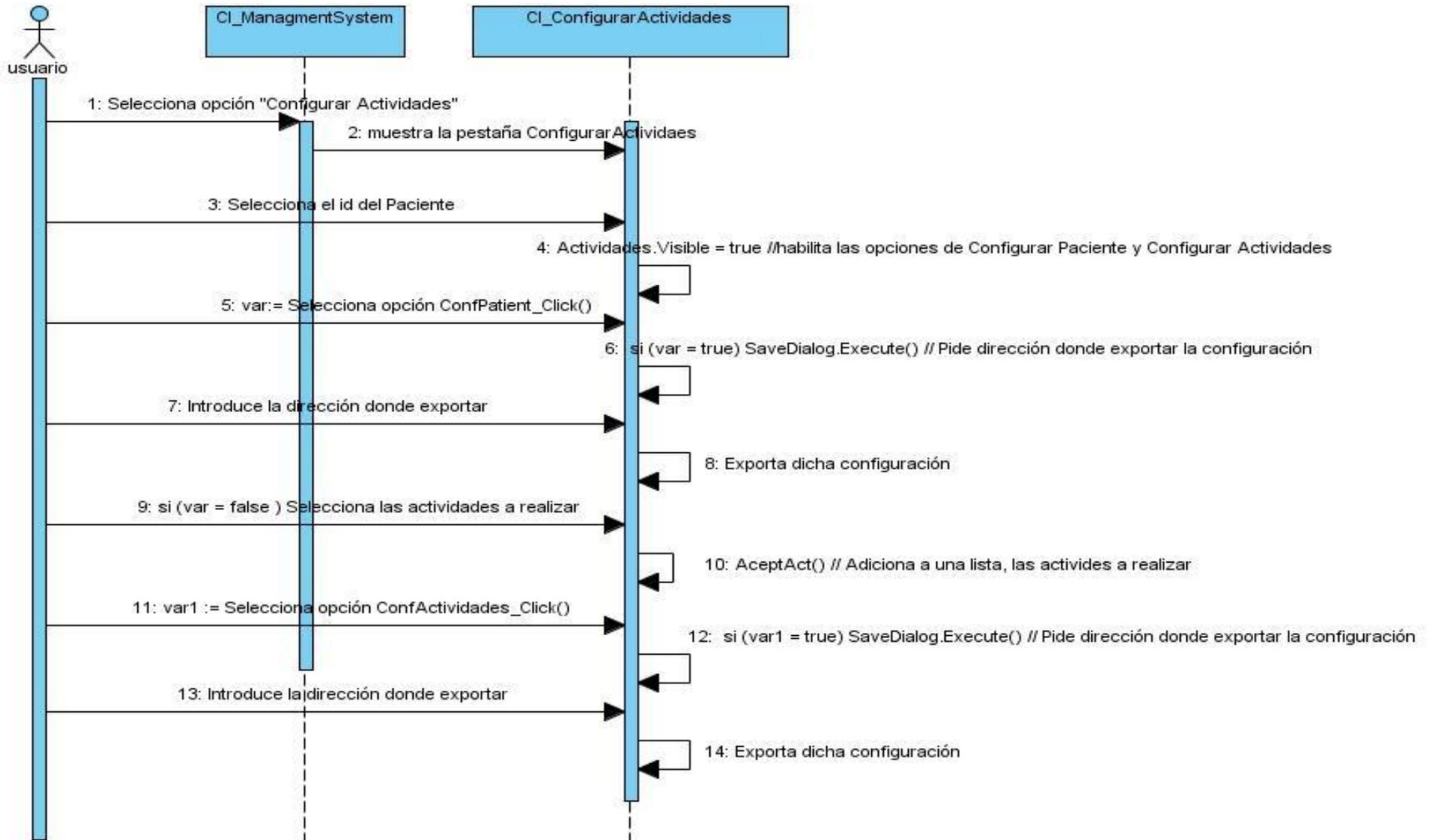


Ilustración 14 CU: Configurar Actividades

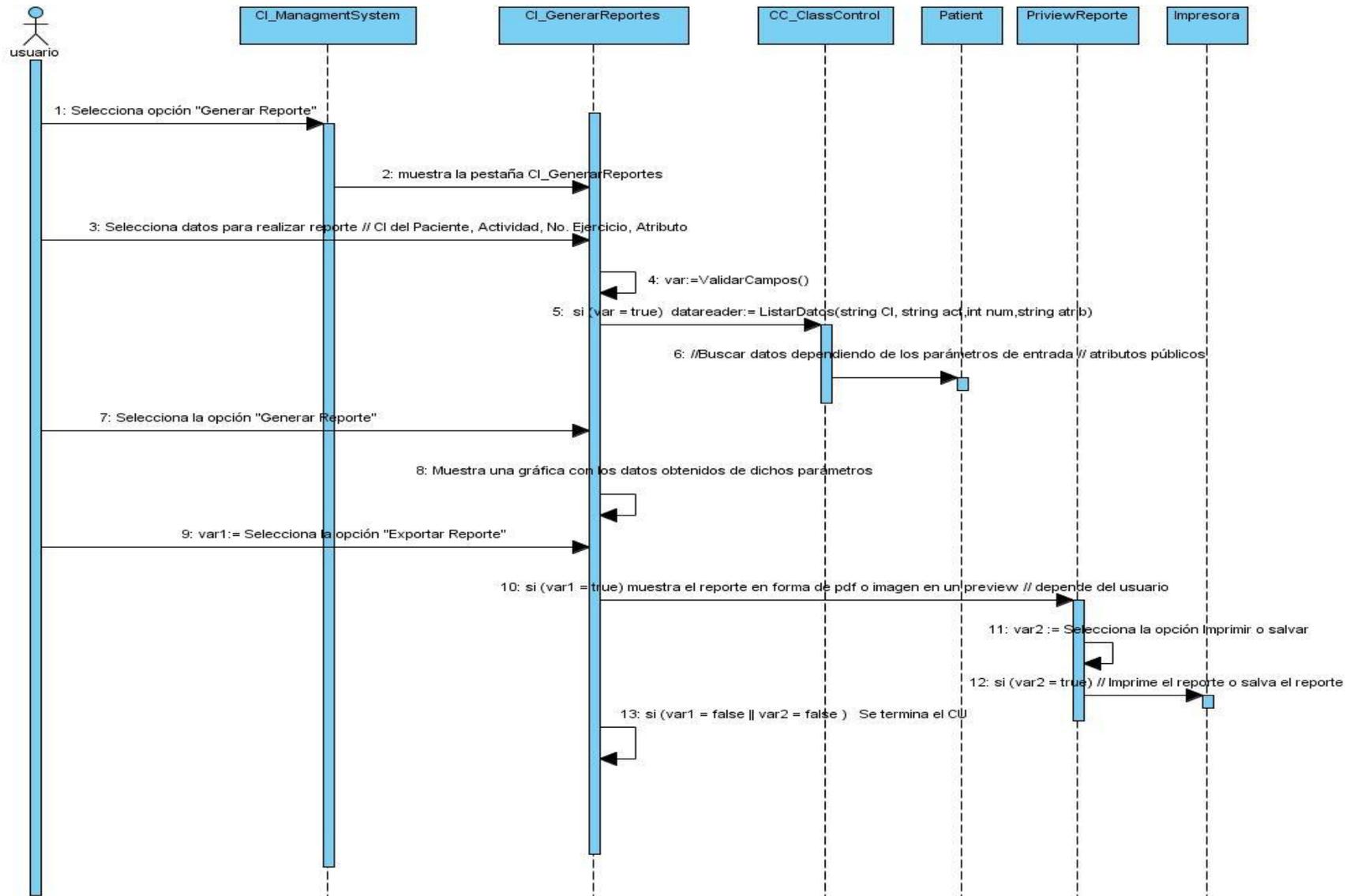


Ilustración 15 CU: Generar Reporte Estadístico

3.7. Diagrama de Despliegue

La aplicación consta con un servidor de BD, que será donde se encontrarán almacenados los datos personales y médicos de los pacientes tratados en el sistema. Cuenta además con PCs clientes conectadas a este servidor, las cuales estarán haciendo uso del software.

Por último, una impresora, mediante la cual se imprimirán los reportes estadísticos, en caso de desearlo el usuario.

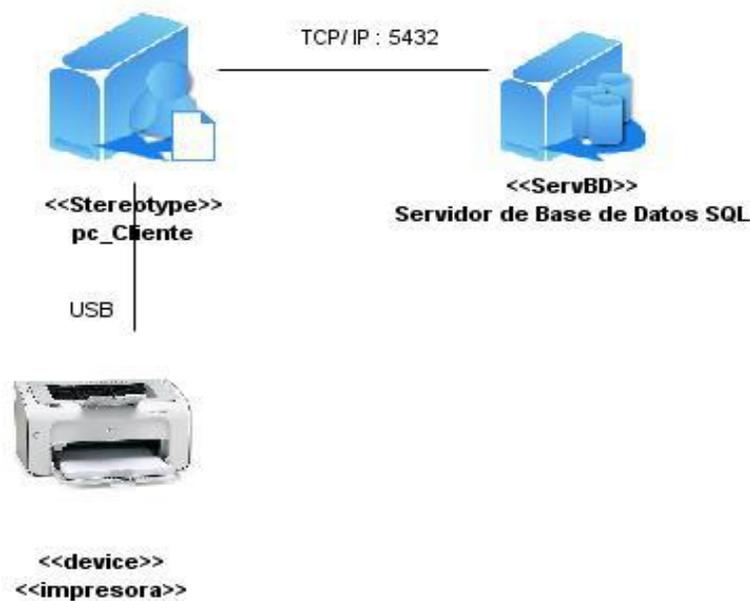


Ilustración 17 Diagrama de despliegue.

3.7. Diseño de la Base de Datos

3.7.1. Diagrama de las clases persistentes

En el siguiente diagrama se muestran las distintas clases de la base de datos, detallando sus respectivos atributos, además de las estrechas relaciones que existen entre ellas, con el objetivo de comprender la forma en que se almacenarán los datos, tanto de los pacientes tratados por el sistema, como de las actividades y ejercicios realizados por estos.

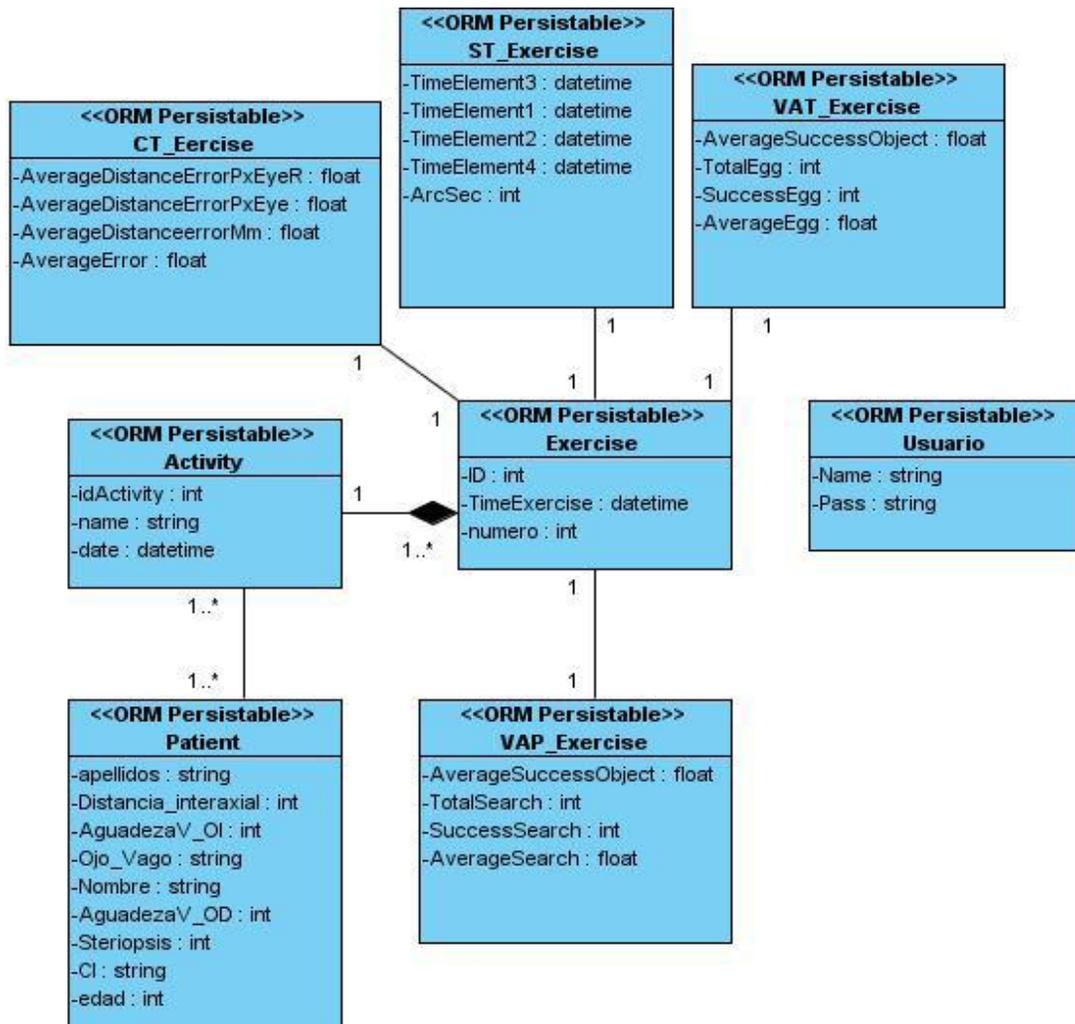


Ilustración 18 Diagrama de las clases persistentes

3.7.2. Diagrama entidad-relación

Se define diagrama entidad-relación o modelo de datos como la herramienta intelectual que permite estructurar los datos de forma que se capte la semántica de los mismos. Éste ofrecerá un conjunto de conceptos y reglas que permitirán representar con mayor o menor fidelidad, un conjunto de datos interrelacionados y operaciones sobre los mismos, a los que afectan unas restricciones que se han de cumplir en todo momento.

En el siguiente diagrama se muestra cómo va a quedar estructurada las tablas de la base de datos, así como la descripción de sus atributos y sus relaciones.

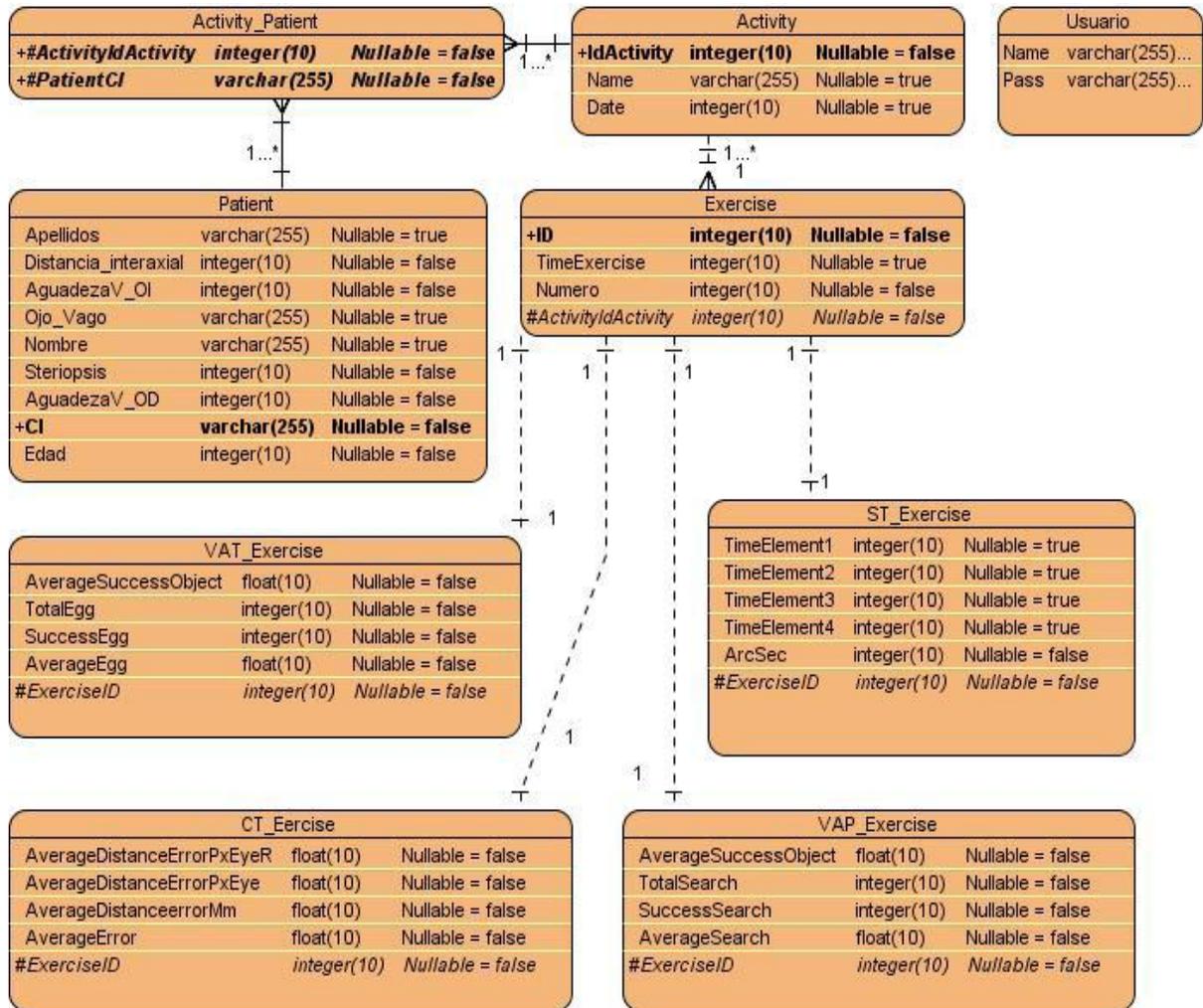


Ilustración 19 Diagrama entidad-relación

En este capítulo se definieron artefactos importantes de la construcción del sistema.

Se construyó el diagrama de despliegue en donde se detalló la relación hardware y software del sistema final. Se mostró de forma detallada como está estructurado el diseño e implementación que presenta el sistema en cuestión.

Capítulo 4: Análisis de los Resultados

4.1. Introducción

En este capítulo se brinda información sobre la validación teórica realizada al diseño descrito en el Capítulo 3, como la normalización y el análisis de la redundancia de la información. Se toman aspectos como la seguridad y control de acceso a la BD.

Se muestran además una serie de pruebas de software las cuales permitieron validar las funcionalidades del mismo, como la conexión a la base de datos, la gestión del paciente, la configuración de actividades, entre otras.

4.2. Validación teórica del software

4.2.1. Normalización

Para los esquemas de relación propuestos para la BD se tuvo en cuenta la normalización, la cual se aplicó al diagrama Modelo-Entidad-Relación, con el objetivo de analizar la relación de la BD y las dependencias funcionales entre sus atributos, para evitar las anomalías en las actualización, inserción, y eliminación, asegurando la consistencia de los datos, con vista a garantizar un adecuado diseño de la BD. (Rojas, 2008)

Se puede afirmar que los diseños de los módulos de la BD propuestos se encuentran normalizados hasta Tercera Forma Normal. Se puede plantear que los esquemas de relación para estos módulos están en 1ra Forma Normal puesto que se puede asegurar que cada tupla contiene exactamente un valor para cada atributo de las tablas de la BD, es decir, no existen campos multievaluados. También se cumple que los esquemas de relación están en 2da Forma Normal, porque primeramente se encuentran en 1ra Forma Normal, y además todos los atributos que no son claves en las tablas, dependen totalmente de la clave primaria. Finalmente se puede plantear que los esquemas de relación se encuentran en 3ra Forma Normal, porque primeramente se encuentran en 2da Forma Normal, y además que no existen dependencias transitivas entre llaves candidatas y atributos no primos. Solo se llegará hasta la 3ra Forma

Normal, debido a que un esquema relacional grande como el que se está desarrollando es muy difícil llevarlo a las otras formas normales y durante el proceso se puede crear características no deseadas.

4.2.2. Análisis de redundancias

Se dice que hay redundancia de datos cuando la misma información es almacenada varias veces en la misma BD. Esto es siempre algo que se debe evitar, la redundancia dificulta la tarea de modificación de datos, y es el motivo más frecuente de inconsistencia de datos. Además requiere un mayor espacio de almacenamiento, que influye en mayor coste y mayor tiempo de acceso a los datos. Después de llevar la BD a 3FN quedó libre de redundancias aunque en caso que se quisiera agrupar en relaciones más grandes la BD se podría utilizar la técnica de Síntesis Relacional que precisamente consiste en agrupar las relaciones en otras más grandes a conveniencia del proceso de negocio.

4.3. Análisis de la seguridad de la base de datos

La seguridad es un punto esencial en las BD para evitar ataques, impedir cualquier acceso no autorizado, con la intención de modificar, usar y/o difundir información almacenada en las BD. El SGBD a utilizar para diseñar las BD incluye formas de restringir el acceso al sistema. Esta función se denomina control de acceso y se pone en práctica creando cuentas de usuarios y contraseñas para que el SGBD controle el proceso de entrada al sistema.

El administrador de la BD (ABD) juega un papel importante en la seguridad de la BD porque es quien otorga privilegios a los usuarios, los clasifica, así como a los datos.

Las órdenes de los ABD incluyen las siguientes acciones:

1. Creación de cuentas.
2. Concesión de privilegios.
3. Revocación de privilegios.

La creación de cuentas sirve para controlar el acceso al SGBD en general, la concesión y revocación de privilegios para controlar autorizaciones discrecionales. Estas autorizaciones constituyen un importante mecanismo de seguridad.

4.4. Validación práctica del software

Para la validación práctica del software es necesario realizar un grupo de pruebas que permitirán determinar si existen fallas en el sistema, no importa que ésta esté funcionando, nunca se puede afirmar que carecen de fallas hasta que se llegue a esta fase y se hagan casos de pruebas en sus funcionalidades.

El método que se propone para diseñar los casos de pruebas es una variante de particiones equivalentes, propuesto por RUP, el cual se basa en hacer por cada caso de uso un caso de prueba.

CU: Gestionar Paciente

CAPÍTULO 4: ANÁLISIS DE LOS RESULTADOS

Escenario: Insertar Paciente

| Escenario | Descripción | CI | Nombre | Apellidos | Edad | DI | OV | AVOD | AVOI | St | Respuesta del sistema |
|------------------------------------|--|-------------|--------|---------------|------|----|---------|------|------|-----|--|
| EC Insertar Paciente | Debe insertarse correctamente. | V | V | V | V | V | V | V | V | V | El sistema inserta el paciente correctamente en la base de datos y vuelve al formulario principal. |
| | | 87101917486 | Adrian | Fumero Valdés | 23 | 64 | Derecho | 5 | 7 | 230 | |
| EC Seleccionar la opción Cancelar. | Mediante este escenario se cancela la operación Insertar Paciente. | NA | NA | NA | NA | NA | NA | NA | NA | NA | El sistema limpia los campos del formulario y vuelve al formulario principal |
| EC Existen datos incorrectos | Al insertar un paciente se introducen datos incorrectos. | I | V | V | V | V | V | V | V | V | El sistema muestra un indicador sobre los campos incorrectos con la información que se debe introducir en ellos. |
| | | V | I | V | V | V | V | V | V | V | |
| | | V | V | I | V | V | V | V | V | V | |
| | | V | V | V | I | V | V | V | V | V | |
| | | V | V | V | V | I | V | V | V | V | |
| | | V | V | V | V | V | V | I | V | V | |
| | | V | V | V | V | V | V | V | I | V | |
| | | V | V | V | V | V | V | V | V | I | |
| | | V | V | V | V | V | V | V | V | I | |
| EC Existen campos | Al insertar un paciente se dejan | NA | V | V | V | V | V | V | V | V | El sistema muestra un indicador sobre los campos |
| | | V | NA | V | V | V | V | V | V | V | |
| | | V | V | NA | V | V | V | V | V | V | |

CAPÍTULO 4: ANÁLISIS DE LOS RESULTADOS

| | | | | | | | | | | | |
|--------------------------|--|---|---|---|----|----|----|----|----|----|---|
| vacíos | campos vacíos. | V | V | V | NA | V | V | V | V | V | incorrectos con la información que se debe introducir en ellos. |
| | | V | V | V | V | NA | V | V | V | V | |
| | | V | V | V | V | V | NA | V | V | V | |
| | | V | V | V | V | V | V | NA | V | V | |
| | | V | V | V | V | V | V | V | NA | V | |
| | | V | V | V | V | V | V | V | V | NA | |
| EC Ya existe el paciente | Se inserta un paciente con un CI ya existente. | I | V | V | V | V | V | V | V | V | El sistema muestra un mensaje “Ya existe el paciente” |

Escenario: Modificar Paciente

| Escenario | Descripción | CI | Nombre | Apellidos | Edad | DI | OV | AVOD | AVOI | St | Respuesta del sistema |
|------------------------------------|---|-------------|--------|---------------|------|----|---------|------|------|-----|---|
| EC Modificar Paciente | Debe modificarse correctamente. | V | V | V | V | V | V | V | V | V | El sistema modifica el paciente correctamente en la base de datos y vuelve al formulario principal. |
| | | 87101917485 | Adrian | Fumero Valdés | 23 | 64 | Derecho | 5 | 7 | 230 | |
| EC Seleccionar la opción Cancelar. | Mediante este escenario se cancela la operación modificar paciente. | NA | NA | NA | NA | NA | NA | NA | NA | NA | El sistema limpia los campos del formulario y vuelve al formulario principal |
| EC Existen | Al insertar un paciente se | I | V | V | V | V | V | V | V | V | El sistema muestra un indicador sobre los |
| | | V | I | V | V | V | V | V | V | V | |

CAPÍTULO 4: ANÁLISIS DE LOS RESULTADOS

| | | | | | | | | | | | |
|--------------------------|--|----|----|----|----|----|----|----|----|----|--|
| datos incorrectos | introducen datos incorrectos. | V | V | I | V | V | V | V | V | V | campos incorrectos con la información que se debe introducir en ellos. |
| | | V | V | V | I | V | V | V | V | V | |
| | | V | V | V | V | I | V | V | V | V | |
| | | V | V | V | V | V | I | V | V | V | |
| | | V | V | V | V | V | V | I | V | V | |
| | | V | V | V | V | V | V | V | I | V | |
| | | V | V | V | V | V | V | V | V | I | |
| EC Existen campos vacíos | Al modificar un paciente se dejan campos vacíos. | NA | V | V | V | V | V | V | V | V | El sistema muestra un indicador sobre los campos incorrectos con la información que se debe introducir en ellos. |
| | | V | NA | V | V | V | V | V | V | V | |
| | | V | V | NA | V | V | V | V | V | V | |
| | | V | V | V | NA | V | V | V | V | V | |
| | | V | V | V | V | NA | V | V | V | V | |
| | | V | V | V | V | V | NA | V | V | V | |
| | | V | V | V | V | V | V | NA | V | V | |
| | | V | V | V | V | V | V | V | NA | V | |
| | | V | V | V | V | V | V | V | V | NA | |
| EC Ya existe el paciente | Se modifica un paciente con un CI ya existente. | I | V | V | V | V | V | V | V | V | El sistema muestra un mensaje "Ya existe el paciente" |

Escenario: Eliminar Paciente

| Escenario | Descripción | CI | Respuesta del sistema |
|-------------------------|---|-------------|--|
| EC Eliminar paciente | Debe permitir eliminar correctamente un paciente seleccionado por el usuario. | V | El sistema elimina el paciente seleccionado y lo redirecciona a la página principal. |
| | | 87101917486 | |
| EC "Selección Inválida" | No se selecciona un CI de un paciente. | I | El sistema muestra un mensaje de error "Debe seleccionar un valor valido". |

CU: Generar Reporte Estadístico

| Escenario | Descripción | CI | Act | Atributo | No. Ej | Respuesta del sistema |
|--------------------------------|---|-------------|------------|--------------|--------|--|
| EC Generar reporte estadístico | Debe permitir generarse el reporte correctamente. | V | V | V | V | El sistema autentica el usuario y lo redirecciona a la página principal. |
| | | 87101917486 | Stereopsis | TimeElement1 | 1 | |
| EC Selección Inválida | Se introducen datos incorrectamente en la operación de Generar Reporte Estadístico. | I | V | V | V | Muestra el mensaje de error "Debe seleccionar correctamente los campos". |
| | | V | I | V | V | |
| | | V | V | I | V | |
| | | V | V | V | I | |

CAPÍTULO 4: ANÁLISIS DE LOS RESULTADOS

| | | | | | | |
|---------------------------|---|----|----|----|----|--|
| EC Exportar Reporte | Se habilita la opción Exportar Reporte al haberse | V | V | V | V | Una vez generado el reporte, se exporta el reporte en forma de pdf o imagen. |
| EC Limpiar Gráfica | Debe permitir limpiar la gráfica actual. | NA | NA | NA | NA | El sistema limpia la gráfica actual que esté generada, sino no hay ninguna no pasa nada. |

CU: Configurar Actividades

| Escenario | Descripción | CI | Actividades | Respuesta del sistema |
|-------------------------------------|---|-------------|-------------|---|
| EC Configurar Paciente | Debe Exportar en un fichero XML los datos personales de un paciente seleccionado. | V | NA | El sistema exporta en una dirección seleccionada los datos personales del paciente con el CI seleccionado en un fichero XML. |
| | | 87101917486 | | |
| EC Selecciona la opción Cancelar | Se introducen datos incorrectamente en la operación de Generar Reporte Estadístico. | I | V | El sistema limpia los campos del formulario. |
| | | V | I | |
| EC Configurar Actividades | Se habilita en el momento que se selecciona un CI. | V | V | El sistema exporta en una dirección seleccionada la configuración de las actividades del paciente con el CI seleccionado en un fichero XML. |
| | | 87101917486 | | |

CAPÍTULO 4: ANÁLISIS DE LOS RESULTADOS

| | | | | | |
|----------------|-----------|---|---|----|--|
| EC Inválida | Selección | No se selecciona correctamente el CI. | I | NA | El sistema no habilita las operaciones de Configurar Paciente, ni Configurar Actividades. |
|----------------|-----------|---|---|----|--|

Las celdas de la tabla contienen V, I, o NA. V indica válido, I indica inválido, y NA que no es necesario proporcionar un valor del dato en este caso, ya que es irrelevante.

Se diseñaron los casos de prueba, faltaría probar contra la aplicación y verificar si se cumplieron los requisitos establecidos en los casos de usos. El nivel de las pruebas a realizar en este caso podría ser de liberación o de aceptación por parte del cliente. Las pruebas realizadas al sistema alcanzaron resultados satisfactorios en más del 95% de los casos. Se garantizó la robustez y a la vez flexibilidad del sistema, además de la seguridad, su fácil uso y la gran utilidad en el desarrollo del proceso en general.

En este capítulo se realizaron pruebas que permitieron verificar y revelar la calidad del producto o software, con el objetivo de identificar posibles fallos de implementación, calidad o usabilidad. Además, se realizaron pruebas teóricas a la base de datos con el fin de realizar un diseño eficiente de ésta.

Conclusiones Generales

Se desarrolló un módulo que permite gestionar pacientes, la configuración de sus actividades y ejercicios, que posibilita además la generación de reportes estadísticos como ayuda a la toma de decisiones y la emisión de diagnósticos, así como dar seguimiento a la evolución de los pacientes. El módulo tiene implementada la gestión de usuario con el objetivo de tener seguridad en la información. Además, se diseñó una base de datos que contendrá los datos personales y médicos de los pacientes tratados por el sistema HDETA.

Con lo anterior planteado se puede afirmar que este módulo resuelve las limitaciones presentadas en el problema existente, además se pone a disposición de uso en el tratamiento a la población, lo que permitirá una mejora de vida, lográndose una vez más llevar la informática al sector de la salud.

Recomendaciones

- ✓ Es importante el uso de esta herramienta como ayuda a los especialistas en su toma de decisiones y le emisión de diagnósticos, en vistas a mejorar los resultados en los tratamientos de los pacientes tratados por HDETA.
- ✓ Continuar con el desarrollo de la aplicación, así como de la base de datos, con el objetivo de llevar este sistema más allá de los problemas de visión estereoscópica.
- ✓ Continuar con la investigación referente a la vía de gestionar los datos con el objetivo de minimizar el uso de código y rendimiento en el servidor.
- ✓ Desarrollar otros tipos de reportes estadísticos de manera que a los especialistas les sea más fácil seguir la evolución de los pacientes y emitir tratamientos sobre estos.

Referencias bibliográficas y bibliografías

- [BD, 2009] (2009). Base de datos. <http://www.monografias.com/trabajos14/basededatos/basededatos.shtml#>
- [TBD, 2009] (2009). Tipos de base de datos. <http://www.monografias.com/trabajos55/basede-datos/base-de-datos.shtml>
- [Int, 2010] (2010). Integridad. <http://www.monografias.com/trabajos19/administracionbase-datos/administracion-base-datos.shtml>
- [Rep, 2011] (2011). Definición de reporte. <http://definicion.de/reporte/>
- [Tra, 2011] (2011). Definición de trazabilidad. <http://www.gestiontrazabilidad.com/normativa.php>
- [RV, 2011] (2011). La realidad virtual. <http://www.fib.upc.edu/retroinformatica/avui/realitatvirtual.html>
- [RVM, 2011] (2011). La realidad virtual en la medicina. <http://www.saludymedicinas.com.mx/articulos/2385/realidad-virtual-aplicada-enmedicina/4>
- [Pos, 2011] (2011). Postgresql. <http://www.ecured.cu/index.php/PostgreSQL>
- [ORA, 2011] (2011). SGBD Oracle. <http://www.monografias.com/trabajos81/sistemasdistribuidos-ii-oracle/sistemas-distribuidos-ii-oracle.shtml>
- [SQL, 2011] (2011). Sql server <http://www.sqlmax.com/caracter.asp>
- [Hernández, 2010] Hernández, L. (2010). Sistema de gestión de base de datos. <http://www.miportal.edu.sv/portal/comunidad/blogs/lucitahernandezl/archive/2010/06/20/sistemade-gestion-de-base-de-datos.aspx>.
- [Horna, 2011] Horna, A. V. (2011). Análisis de datos. <http://es.scribd.com/doc/7799033/Cap5-Analisis-de-Datos>
- [Muñoz, 2009] Muñoz, A. (2009). Mysql ¿qué es? <http://www.miempresaenlinea.com/blog/post/MySQL-Que-es.aspx>
- [Ruiz., 2006] Ruiz., G. (2006). Gestión institucional: Conceptos introductorios. <http://www.educ.ar/educar/site/educar/Gestiose-uri=urn:kbee:ff9221c0-13a9-11dc-b8c4-0013d43e5fae>

- [Sainz, 2002] Sainz, J. (2002). La automatización en la empresa actual.
www.eui.upm.es/wdoe/automatizacion.ppt
- [Rojas, 2008] Rojas, A. R. (2008). Base de datos de gestor de contenidos de ajedrez. Master'sthesis, Universidad de las Ciencias Informáticas
- [Pro, 2001] (2001). Manual de usuario, programa evo.
- **Dapena, MSc. Martha D. Delgado Dapena.** Definición del modelo del negocio y del dominio utilizando. s.l. : Centro de Estudios de Ingeniería de Sistemas., 2009.
- Definición de modelo de datos. <http://definicion.de/modelo-de-datos/>.
- **Lisset, Domínguez López, Rotceh y Bolmey Romero, Yurisney del Carmen Avila de la Cruz.** Trabajo de Diploma para optar por el Título de Ingeniero Informático. Sistema de Gestión de Información en el Proceso de Planificación de Materiales Gastables de Uso Médico del Ministerio de Salud Pública (MINSAP). La Habana : s.n., Mayo del 2008.
- **Sanchez, María A. Mendoza.** Metodologías de desarrollo de Software.
http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html
- **García Molina J., Moreira, Ana; Rossi, Gustavo; 2004.** *UML: el lenguaje estándar para el modelado de software.* Ediciones SA.2004.
- **Ezequiel Rozic, Sergio; 2004.** *Bases de Datos y su aplicación con SQL.*
<http://bibliodoc.uci.cu/pdf/reg03438.pdf>
- **Angel Sicilia, Miguel; 2008.** *Funciones de los Sistemas Gestores de Bases de Dato.*
<http://cnx.org/content/m17543/latest/>
- **Álvarez, Sara; 2007.** *Sistemas gestores de bases de datos.*
<http://www.desarrolloweb.com/articulos/sistemas-gestores-bases-datos.html>
- **Pressman, R.S.; 2005.** *Ingeniería del Software, Un enfoque práctico.* 5a. ed. La Habana, Mc Graw Hill.
- **González Cornejo, J. E.; 2001.** *Arquitectura en Capas. Un camino hacia los procesos distribuidos.*
http://www.docirs.cl/arquitectura_tres_capas.htm
- **Patrón MVC; 2008** <http://www.proactiva-calidad.com/java/patrones/mvc.htm>.
- **Larman, C.; 1999.** *UML y Patrones: Introducción al análisis y programación orientada a objetos.*
<http://bibliodoc.uci.cu/pdf/reg00062.pdf>

Glosario de Términos

HDSVE Herramienta de Desarrollo para Sistemas de Visión Estereoscópica

HDETA Herramienta para la Detección Evaluación y Tratamiento de la Ambliopía

UCI Universidad de la Ciencias Informáticas

RV Realidad Virtual

SGBD Sistema Gestor de Base de Datos

BD Base de Datos

IDE Entorno de Desarrollo Integrado, Integrated Development Environment

UML Lenguaje Unificado de Modelado

XML Lenguaje de Marcado Extensible, Extensible Markup Language

benchmarks Técnica utilizada para medir el rendimiento de un sistema