



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS
FACULTAD 5

“Módulo Registro de Evaluación para el Simulador Quirúrgico”

Trabajo para optar por el Título de Ingeniería en
Ciencias Informáticas

Autores: Aylin Arango Bujardón.

Yohandri Quintana Echevarria.

Tutores: Ing. Juan Carlos Quevedo.

Ing. Osmanys Valdez Puga.

Ciudad de La Habana

Junio, 2008

*“El progreso es la habilidad del hombre para hacer
complejo lo sencillo.”*

Thor Hierdahl

Declaración de Autoría

Declaramos que somos los únicos autores de este trabajo, y autorizamos al Proyecto Simulador Quirúrgico (Kheipros) de la Facultad 5 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmamos la presente a los ____ días del mes de _____ del año _____.

Autores:

Aylin Arango Bujardón

Yohandri Quintana Echevarria

Tutores:

Ing. Juan Carlos Quevedo

Ing. Osmanys Valdez Puga

Datos de Contacto

Nombre y Apellidos: Ing. Juan Carlos Quevedo Lusson.

Graduado de la Universidad de las Ciencias Informáticas en la primera graduación, julio 2007.

Instructor recién graduado.

4 años de experiencia en el desarrollo de software.

Jefe de Colectivo de Bases de Datos y Gestión de Software de la Facultad 5.

Líder de proyecto de Supervisión Energética.

E-mail: jquevedo@uci.cu

Nombre y Apellidos: Ing. Osmanys Valdes Puga.

Graduado de la Universidad de las Ciencias Informáticas

Profesor Adiestrado, 2 años de experiencia.

E-mail: ovaldes@uci.cu

Dedicatoria

Aylin

A mis padres en especial a mi mamá por su apoyo, amor y dedicación!!!!

A mi hermano por ser muy especial en vida

A mi novio (Yohandri) por quererme, apoyarme y estar siempre a mi lado.

A una persona muy especial que ya no está, pero que en todo momento estará y será parte de mi vida, esto es para ti abuelita (Frangelina).

A toda mi familia en general

Yohandri

A mis padres con todo el amor de mi corazón!!!!

A mi hermano por ser alguien tan importante para mí

A mi novia (Aylin) por estar a mi lado en todo momento y ayudarme a lograr mi anhelo.

A mi familia

Agradecimientos

Agradecemos a la Revolución Cubana y en especial a nuestro Comandante en jefe por habernos dado la posibilidad de formarnos como personas de bien, y de ver nuestro sueño hecho realidad.

*A nuestro tutor Juan Carlos por haber tenido tanta paciencia con nosotros, por ser tan especial y habernos demostrado que todavía existe la bondad y la perseverancia en las personas. **MUCHAS GRACIAS!!!***

Al ingeniero Yoan Parra por su ayuda incondicional.

A todos los profesores que han sido importantes en nuestras vidas (Juan Manuel, Souchay, Barzaga, Yenieris y Vero).

A todas aquellas personas que de una forma u otra contribuyeron al desarrollo de este trabajo.

Yohandri

*Ante todo a mi familia, en especial a mis padres (Haydee y Jaime) por creer en mí, por apoyarme en todo momento, por brindarme todo el amor del mundo, **LOS QUIERO MUCHO!!!***

A mi hermano (Jaimito) por ser una persona muy especial en mi vida.

A mi novia Aylin gracias por existir.

A la memoria de mi abuela (Fela) que me enseñó a ser una persona correcta.

A mis abuelos del alma (Aurelia, Nene y en especial a Berto).

A Yaneira por hacer tan feliz a mí hermano.

A mis tíos (en especial a Odila).

A mis primos (Leonardito, Orestico, Indira, Melvis, Milenis).

A la familia de mi novia, que siempre estuvo ahí para los momentos difíciles (Frangelina, Andresito, Andrés, Meco, Paco, Annia, Maydolis, Liena)

A todos los socios del apto (Ricar, Leo, Yera, Yausell, Lester, Rainer, Candima, Niche, Yaself, Pupi, Flaco, Marlon, Chacal).

A todos mis amigos de la vocacional que siempre hemos mantenido el contacto (Daimara, Erick, Humberto, Pedro, Armando, Yuliet, chino).

A mi profesor Antero por ser una persona que siempre decía que se podía en las peores circunstancias.

Aylin

A mi mamá por haberme apoyado, por ser tan importante en mi vida, por ser la luz que me ilumina y por dejarme sin palabras para expresarle lo que significa para mí Te quiero con la vida!!!

A mi papá por darme su apoyo y confiar en mí Te quiero mucho!!!

A mi hermano por ser tan importante y especial, por siempre brindarme su amor

A mi novio Yohandri por ser el sol que ilumina mi sendero en la vida Demostrándome en los momentos difíciles con su ternura y su pasión que todo se vuelve más fácil

A mi tío Meco por saberme guiar en la vida y quererlo casi como a mi papá

A mi tía Carmita por ser tan buena, ser mi amiga y quererla casi como a mi mamita

A mi primo Rolo por ser más que mi primo, ser mi hermano Cuídate mi tesoro.

A toda mi familia (Yulio, Paco, Annia, Yisell, Nati, Tío pepe, Marta, Liena, Maydolis)

A toda la familia de mi novio en especial a mis suegros y mi cuñado por ser tan buenos y haberme ayudado y apoyado tanto.

A Zory y Lisi por ser tan buenas y especiales conmigo.

A todos mis amigos y más que amigos, hermanos Cordero, Linito y el grande, por estar siempre en los momentos más importantes de mi vida

A todo mi grupo por haberme dado los años más felices de toda mi vida (Loiret, Leo, Liena, Yaself, Yurian, Lester, Susej, Yausell, Janet, Yera, Nichesito, Yole, Pupi, el flaco, Saily, Rainer, Lili, Yani lis, Yaines, Alex).

A Loiret por ser mi amiga, hermana, consejera, por siempre brindarme su comprensión y haberme demostrado que la amistad sincera todavía existe, por ser tan especial....

A todas las personas que han sido y que son importantes en mi vida (Masi, Jorge, Yohandra, Henry, Deisy, Pablo, Reinier, Blanquilla, Mamita, Pepe, Paquita, Yaneira, Roberto, Guille).

Resumen

Este trabajo tiene como objetivo desarrollar un Módulo de registro de evaluación para el simulador quirúrgico que permita brindar un reporte detallado de los resultados obtenidos por el médico cirujano en la realización de un examen. Todo esto para mantener organizada la información de cada médico con su respectivo resultado en el desempeño de un examen y luego poder dar un reporte del estado actual del cirujano.

Para la realización de este trabajo se emplea como herramienta el Gestor de Base de Datos Firebird, para el análisis, implementación y documentación de sistemas orientados a objetos se tiene como proceso de desarrollo de software, el Proceso Unificado Racional (RUP) que junto, con el Lenguaje Unificado de Modelado (UML), constituyen la metodología a utilizar.

Se obtuvo un módulo de registro de evaluación que contiene datos personales de los médicos a evaluar, además de brindar la posibilidad de llevar un control específico de todos los movimientos que se realizan durante el examen, como también almacenar los valores de las variables de cada examen realizado, permitiendo gestionar la información de los resultados alcanzados por cada usuario del sistema. Se logró que la manipulación de la información sea de una manera segura y confiable.

Palabras clave: registro de evaluación, Firebird, simulador quirúrgico, gestionar información.

Índice

DEDICATORIA	I
AGRADECIMIENTOS	II
RESUMEN	IV
INTRODUCCIÓN	1
CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA	5
INTRODUCCIÓN	5
1.1 GESTIÓN DE LA INFORMACIÓN	5
1.1.1 Información y Conocimiento.....	6
1.1.2 Información.....	7
1.1.3 Conocimiento.....	7
1.1.4 La Información y el conocimiento en la salud.....	8
1.2 REGISTRO	9
1.2.1 Autenticidad.....	11
1.2.2 Fiabilidad.....	11
1.2.3 Integridad.....	12
1.2.4 Usabilidad.....	13
1.3 ESTUDIO DE SITUACIONES	14
1.4 TENDENCIAS ACTUALES	15
1.4.1 De un Sistema de Gestión de Bases de Datos se espera:.....	21
1.5 BASE DE DATOS	22
1.5.1 Funcionalidad de una Base de Datos.....	22
1.6 GESTOR DE BASE DE DATOS. CONCEPTO.	24
1.6.1 Ventajas de los Gestores de Base de Datos:.....	24
1.6.2 Desventajas de los Gestores de Base de Datos:.....	25
1.7 ANÁLISIS DE LOS GESTORES DE BASE DE DATOS	25
1.7.1 Oracle.....	25
1.7.2 SQL Server.....	26
1.7.3 MySQL.....	27
1.7.4 PostgreSQL.....	28
1.7.5 Firebird.....	30
1.8 LIBRERÍA IBPP	32
1.9 LENGUAJE C++	32
1.9.1 Ventajas del C++.....	33
1.10 PROCESO DE DESARROLLO DE SOFTWARE	33
1.11 HERRAMIENTA A UTILIZAR	34
CAPÍTULO 2 CARACTERÍSTICAS DEL SISTEMA	36
INTRODUCCIÓN	36
2.1 OBJETO DE AUTOMATIZACIÓN	36
2.2 PROPUESTA DEL SISTEMA	36
2.3 MODELO DEL DOMINIO	36
2.4 ESPECIFICACIÓN DE LOS REQUISITOS DEL SOFTWARE	37
2.4.1 Requerimientos funcionales.....	37
2.4.2 Requerimientos no funcionales.....	39
2.5 MODELO DE CASOS DE USO DEL SISTEMA	39
2.5.1 DEFINICIÓN Y JUSTIFICACIÓN DE LOS ACTORES DEL SISTEMA A AUTOMATIZAR.....	39
2.5.2 Diagrama de casos de uso del sistema a automatizar.....	40
2.5.3 Descripción de los casos de uso del sistema.....	40
2.6 ANÁLISIS Y DISEÑO	41
2.6.1 Análisis	41

2.6.1.1 Diagrama de Clase de Análisis.....	41
2.7 DISEÑO	43
2.7.1 Diagrama de Clases del Diseño	43
Arquitectura de tres Capas	43
2.7.2 Capa de Presentación.	44
2.7.3 Capa de Negocio	45
2.7.4 Capa de Acceso a Datos	48
2.7.5 Diagrama de Secuencia	49
2.7.6 Diagrama de clases Persistentes	59
2.7.7 Modelo de Datos.....	59
2.7.8 Descripción de las tablas del modelo de datos.	60
CAPÍTULO 3 IMPLEMENTACIÓN Y PRUEBA DEL SISTEMA.	61
INTRODUCCIÓN.....	61
3.1 IMPLEMENTACIÓN	61
3.2 DIAGRAMAS DE COMPONENTES	61
3.2.1 Diagrama de Paquetes del modelo de Implementación.....	62
3.2.2 Diagrama de componentes Capa Acceso a datos (Vista ObjetoBD).....	62
3.2.3 Diagrama de componentes Capa Acceso a datos (Vista Control de Acceso Firebird).....	63
3.2.4 Diagramas de componentes Capa de Presentación.....	63
3.2.4.1 Diagrama de componentes Administrador.....	63
3.2.4.2 Diagrama de componentes Autenticar.	63
3.2.4.3 Diagrama de componentes Examen.....	64
3.2.4.4 Diagrama de componentes Profesor.	64
3.2.4.5 Diagramas de componentes Usuario.....	64
3.2.5 Diagramas de componentes Capa de Negocio.....	64
3.2.5.1 Diagrama de componentes Administrador.....	64
3.2.5.2 Diagrama de componentes Autenticar.	65
3.2.5.3 Diagrama de componentes Examen.....	65
3.2.5.4 Diagrama de componentes Profesor.	65
3.2.5.5 Diagramas de componentes Usuario.....	66
3.3 DIAGRAMA DE DESPLIEGUE	66
3.4 MODELO DE PRUEBA	67
3.4.1 Objetivos.....	67
3.4.2 Alcance	67
3.4.3 Prueba de software.....	68
3.4.3.1 Caso de Prueba Registrar Nuevo Usuario.....	68
3.4.3.2 Caso de Prueba Autenticar Usuario.....	69
3.4.3.3.0 Caso de Prueba Gestionar usuarios del sistema (Sección “Asignar Rol”).....	69
3.4.3.3.1 Caso de Prueba Gestionar usuarios del sistema (Sección “Modificar Usuario”).....	70
3.4.3.3.2 Caso de Prueba Gestionar usuarios del sistema (Sección “Eliminar Usuario”).....	71
3.4.3.3.3 Caso de Prueba Gestionar usuarios del sistema (Sección “Quitar Rol”).....	71
3.4.3.4 Caso de Prueba Registrar Inicio Examen.	72
3.4.3.5 Caso de Prueba Registrar Evento.....	73
3.4.3.6 Caso de Prueba Registrar Fin Examen.....	73
3.4.3.7 Caso de Prueba Mostrar Estadísticas de Médicos Cirujanos.	73
3.4.3.8 Caso de Prueba Consultar Reporte.	73
3.4.4 Prueba de Caja Negra.....	74
3.4.5 Descripción de la prueba de caja negra realizada.....	74
3.4.5.1 Caso de Prueba Registrar Nuevo Usuario.....	74
3.4.5.2 Caso de Prueba Autenticar Usuario.....	75
3.4.5.3 Caso de Prueba Gestionar usuarios del sistema (Sección “Asignar Rol”).....	76
3.4.5.4 Caso de Prueba Gestionar usuarios del sistema (Sección “Modificar Usuario”).....	77
3.4.5.5 Caso de Prueba Gestionar usuarios del sistema (Sección “Eliminar”).....	77
3.4.5.6 Caso de Prueba Gestionar usuarios del sistema (Sección “Quitar Rol”).....	78
3.4.5.7 Caso de Prueba Registrar Inicio Examen.	79
3.4.5.8 Caso de Prueba Registrar Evento.....	79

3.4.5.9 Caso de Prueba Registrar Fin Examen.....	80
3.4.5.10 Caso de Prueba Consultar Reporte.....	80
3.4.5.11 Caso de Prueba Mostrar Estadísticas de Médicos Cirujanos.....	81
CONCLUSIONES GENERALES	82
RECOMENDACIONES.....	83
REFERENCIAS BIBLIOGRÁFICAS	84
BIBLIOGRAFÍA CONSULTADA	87
APÉNDICES	90
GLOSARIO DE ABREVIATURAS	90
GLOSARIO DE TÉRMINOS.....	91
ANEXO.....	93
Anexo 1 Descripciones Textuales.....	93
Anexo 2 Tablas de Clases del Diseño.....	100
Anexo 3 Descripción de las tablas del modelo de datos.....	103

Introducción

El ejercicio de la medicina es una prueba exigente a la que cada día tienen que enfrentarse médicos de todo el mundo. En cuestión de minutos, a veces segundos, el profesional tiene que obtener la información procesarla y tomar decisiones que conllevan responsabilidades médicas, sociales y económicas. Esta situación es extremadamente importante en el área de Urgencias y en determinadas patologías donde los pacientes aportan poca información y ésta puede ser crítica. Hasta hace unos años este proceso de aprendizaje se realizaba en el día a día sobre pacientes reales con los riesgos que ello conllevaba. Por lo que existió la necesidad de crearse simuladores para el apoyo en la medicina, que no son más que modelos virtuales de distintas partes del cuerpo dotados también de herramientas virtuales con las que los cirujanos pueden entrenarse en distintas técnicas quirúrgicas. [1]

La simulación quirúrgica está destinada a la educación y entrenamiento de los médicos, para que puedan practicar los conocimientos adquiridos durante sus estudios, en casos reales y con procesos de trabajos similares a los realizados en la vida real.

Durante la realización del caso clínico, el simulador va guardando todos los actos médicos realizados, y a través de algoritmos propietarios, presenta un informe detallado no solo de lo que se ha hecho y su conveniencia, sino que comunica al médico si le ha faltado alguna prueba o actuación por realizar. Para esto es necesario la utilización de una base de datos que genere potentes sistemas de archivo permitiendo almacenar información de manera organizada y posteriormente recuperarla y modificarla. Además de brindar medidas de seguridad que permiten mantener la integridad de los datos, trayendo esto un gran aumento en la demanda de estos sistemas. Los cuales son un cimiento fundamental en el desarrollo de estrategias de comercialización, de tal forma que pueden proporcionar la respuesta adecuada a las inquietudes de comercialización, y así evitar gastos económicos, de tiempo y de otros recursos innecesarios.

Cuba está muy vinculada al desarrollo de la informática y las telecomunicaciones, viéndose avances de gran envergadura como: la realización de juegos virtuales, simuladores, todo esto con vista al desarrollo y bienestar del pueblo. En el campo de la medicina se vio la necesidad de crear las perspectivas para el desarrollo de un simulador que le sirva de práctica, y de especialización a los médicos cirujanos, por lo que La Universidad de las Ciencias Informáticas (en lo adelante UCI) se da a la tarea de llevar a cabo un Simulador Quirúrgico que se encuentra enmarcado en el desarrollo de su

primera versión, la cual presenta problemas con el almacenamiento de información, ya que cuando se realizan los exámenes no se puede obtener un resultado, debido a que los parámetros que deben medir los mismos no se encuentran almacenados, impidiendo presentar un informe con los datos de los exámenes realizados, siendo esta la situación problemática.

En función de lo planteado anteriormente el **problema científico** es: ¿Cómo garantizar el registro y almacenamiento de los datos, para el simulador quirúrgico que se desarrolla en la UCI con vista al entrenamiento de los médicos cirujanos, de una forma exitosa?

Por tanto, se define que el **objeto de estudio** es el proceso de gestión de la información para los simuladores.

De ahí que el **campo de acción** que abarca este trabajo es el proceso de desarrollo de Base de Datos en el simulador quirúrgico para el entrenamiento de los médicos cirujanos.

Con la finalidad de dar solución al problema en cuestión el **objetivo general** de este trabajo es desarrollar un Módulo de registro de evaluación para el simulador quirúrgico que permita brindar un reporte detallado de los resultados obtenidos por el cirujano.

Para darle cumplimiento al objetivo propuesto se plantea la siguiente idea a defender:

Con el desarrollo del sistema de registro de evaluación, se obtendrá un reporte detallado, que informe los resultados, que, con el uso del simulador quirúrgico tienen los médicos cirujanos.

Para alcanzar el objetivo propuesto y darle solución a dicha problemática se trazan las siguientes **tareas de investigación**:

1. Investigar y estudiar el proceso de gestión de la información.
2. Analizar soluciones similares de los simuladores que se han desarrollado en el mundo para el apoyo en la realización de la aplicación.
3. Estudiar posibles tecnologías a utilizar, para un mejor entendimiento y utilización de las mismas.
4. Aplicar la metodología de análisis y diseño del sistema.
5. Implementar el módulo para solucionar el problema planteado.

Para el desarrollo de este trabajo se utilizan como métodos científicos de la investigación los que a continuación se hacen referencia.

Se emplea como métodos teóricos **Analítico – sintético** ya que permite buscar la esencia de los fenómenos, sus características particulares y las que los distinguen, para así darse cuenta de cuál sería la mejor solución para el problema. Además de permitir analizar teorías, documentos, y otros escritos que contengan elementos importantes relacionados con el objeto de estudio. También el **Análisis histórico lógico** con el fin de investigar sobre las aplicaciones informáticas de este tipo desarrolladas en Cuba y el resto del mundo. Además de analizar el avance que ha alcanzado, así como las ventajas y desventajas de este sistema.

Como métodos empíricos se utiliza la **entrevista** a especialista de la medicina para recoger toda la información posible acerca de las funcionalidades que debe tener el sistema y de cómo desea que esté estructurada la información. También se utilizan métodos estadísticos en el análisis de los costos que generará el desarrollo de la aplicación.

En esta investigación se pretende desarrollar una base de datos que le permita al simulador quirúrgico almacenar los datos de los médicos cirujanos, así como las estadísticas obtenidas durante la realización del ejercicio. Todo esto para mantener organizada la información de cada médico con su respectivo desempeño en la realización de los exámenes y luego poder dar un reporte del estado actual del cirujano. Para la realización de este trabajo se emplea como herramienta el Gestor de Base de Datos Firebird y como lenguaje de programación C++ estándar, para el análisis, implementación y documentación de sistemas orientados a objetos se tiene como proceso de desarrollo de software, el Proceso Unificado Racional (RUP) que junto, con el Lenguaje Unificado de Modelado (UML), constituyen la metodología a utilizar.

El contenido de este trabajo se encuentra estructurado en tres capítulos de la siguiente manera:

En el Capítulo 1, Fundamentación Teórica que aborda el estudio del estado del arte de las base de datos en la actualidad, además de las tendencias, tecnologías y técnicas que existen actualmente en el mundo y que han sido usadas para la solución del problema en cuestión.

En el Capítulo 2, Características del Sistema, se especifica el objeto de automatización, la propuesta del sistema, se describe el modelo del dominio, además de exponer todos los requisitos tanto funcionales como no funcionales, la representación del diagrama del sistema y la descripción textual de los casos de usos conformados. En este capítulo también se definen los distintos artefactos del flujo de trabajo de análisis y diseño.

En el capítulo 3, Implementación y Prueba del Sistema, se describen y especifican las funcionalidades de los nodos de cómputos con que cuenta el sistema. En los diagramas del modelo de implementación se representan las relaciones entre los distintos componentes y nodos. Además en este capítulo se explican y describen los objetivos y alcance del flujo de trabajo de Prueba y se especifica el tipo de prueba a utilizar.

Capítulo 1 Fundamentación Teórica.

Introducción

En las últimas décadas el gran desarrollo tecnológico de la informática ha potenciado el almacenamiento de grandes volúmenes de datos y ha permitido el desarrollo de los sistemas para la gestión y procesamiento de los mismos. Por lo que se puede decir que las bases de datos ocupa un lugar importante a tal punto que esta puede verse como un proceso relativamente independiente dentro del diseño del sistema y compuesto por una serie de etapas. Es así como surgen los sistemas de información, que en su definición más general, se refieren al conjunto de todos los componentes necesarios que se interrelacionan, con el objetivo de tramitar y facilitar la información sobre el tema de interés para su consumo en cualquier medio, momento y lugar.

En este capítulo se van a describir fundamentos teóricos de los procesos a automatizar de acuerdo a las tendencias modernas. Se analizará conceptos asociados al dominio del problema, tendencias actuales alrededor del problema a resolver, una breve descripción de la tecnología usada así como el lenguaje de programación y sistema gestor de base de datos; esto va a favorecer la comprensión sobre el tema de trabajo. Para este análisis se necesita información que ha sido obtenida, consultada y recopilada a través de la revisión bibliográfica, entrevistas y otras fuentes de información.

1.1 Gestión de la información

La gestión de la información ofrece al usuario la capacidad de gestionar los datos. Incluye funcionalidades como: jerarquización de la información, arquitectura de bajo acoplamiento de la aplicación, formatos de almacenamiento.

Según Woodman la gestión de información: Comprende las actividades relacionadas con la obtención de la información adecuada, a un precio adecuado, en el tiempo y lugar adecuado, para tomar la decisión adecuada.

Teniendo en cuenta el contexto de este trabajo la gestión de información no es más que el proceso de determinar la información que se precisa, recogerla, analizarla, registrarla, recuperarla, utilizarla y divulgarla. Todo esto para facilitar el trabajo con grandes volúmenes de información.

La gestión de información es una disciplina que se ocupa de utilizar los recursos básicos (económicos, físicos, humanos, materiales) para manejar información dentro y para la sociedad a la que sirve. Por otra parte, dentro de una organización se pueden identificar diferentes tipos de información según las funciones que se realizan, ésta se debe manejar y utilizar de forma adecuada y sistemática para obtener los mejores beneficios.

Una sociedad de la información es cualquier conglomerado humano, en cuyas acciones de supervivencia y desarrollo predomine un uso, distribución, almacenamiento y creación intensa de recursos de información y conocimientos mediatizados por las nuevas tecnologías de la información y la comunicación. Las tres características que la distinguen son:

- √ La información se utiliza como recurso económico para elevar la eficacia y eficiencia en las organizaciones, con el apoyo de mejoras de la calidad de los productos y servicios que oferta.
- √ Las personas hacen un mejor e intenso uso de la información en su papel de consumidores y también como ciudadanos para cultivar sus derechos civiles y sus responsabilidades.
- √ La información conforma un sector dentro de la economía, el cual va en aumento a un ritmo superior que el del propio sector económico, ya que se ocupará de satisfacer las demandas informacionales de las personas. [3]

1.1.1 Información y Conocimiento

La sociedad de la información y el conocimiento (en lo adelante SIC) alcanza su máxima expresión de desarrollo en los inicios del siglo XXI, donde la producción, el manejo y el uso de la información y el conocimiento se convierten en pilares fundamentales del desarrollo informacional del hombre. Ésta se caracteriza por ser un espacio en el que la adquisición, el procesamiento, la organización, el almacenamiento, la recuperación, la utilización, el seguimiento, la distribución y la venta de información conforman actividades prioritarias para la economía de los países que las fomentan, en virtud de su alto índice de generación de valor agregado.

En los estudios hechos a la SIC aparecen siempre de forma reiterada los conceptos de datos, información y conocimiento. Por eso se precisa establecer una clara diferenciación de cada uno.

Los términos datos, información y conocimiento se usan casi siempre como sinónimos en los distintos medios, lo que puede repercutir desfavorablemente ante determinada situación prolémica. He aquí uno de los dilemas actuales de la ciencia de la información: la definición y relación consecutiva que se produce entre las tres expresiones.

Los datos constituyen la materia prima fundamental que interviene en el proceso de construcción de información, pero para esto deben estar bien estructurados, ya que no tienen un significado por sí mismos. Los datos son hechos discretos sobre acontecimientos.

1.1.2 Información

Por otra parte, se entiende por información, todas las ideas, hechos y trabajos imaginativos de la mente que se han comunicado, registrado, publicado y propagado formal o informalmente en cualquier forma. La información se considera científica si el carácter de las exposiciones correspondientes admite la comprobación sin la participación del autor.

Para que la información se convierta en conocimiento es necesario que se le clasifique, procese y analice para que, por medio de la reflexión del producto resultante de la primera, nazca el segundo. De tal modo, la información puede ser organizada y analizada sistemáticamente con vista a producir conocimiento, que no es más que el cúmulo de entendimiento obtenido acerca de los objetos del mundo real y de las ideas.

1.1.3 Conocimiento

Por conocimiento se entiende toda información, que ha sufrido determinados procesos mentales: interiorización, análisis, fijación, aplicación, etcétera, que permiten al sujeto tener un dominio de los sucesos o hechos que ocurren en la sociedad y solucionar determinado problema; es identificar, estructurar y, sobre todo, utilizar la información para obtener un resultado. En otras palabras, requiere aplicar la intuición y la sabiduría, propios de la persona, a la información.

El conocimiento está categorizado como activo intangible, medidor del capital intelectual de una organización. Este último sería el conocimiento propio, interno de la organización, que no se puede ver ni medir a simple vista, es de lo que se nutre la organización, su valor.[\[4\]](#)

1.1.4 La Información y el conocimiento en la salud.

En el contexto de la atención médica, los datos son los hechos primarios y las observaciones adquiridas a medida que se ofrecen los servicios, como pueden ser, por ejemplo, el valor numérico de una medida de la presión sanguínea o la historia familiar acerca de un padre fallecido por una enfermedad determinada. Estos datos se transforman en información cuando indican una evaluación o una acción, como pudiera ser un diagnóstico de hipertensión arterial, el cálculo de posibilidades en un paciente con la misma afección sufrida con anterioridad por sus padres o el análisis de la cantidad de glóbulos rojos en la sangre de alguno de ellos para determinar si tiene anemia. De ahí que la información en el ámbito médico esté concebida como la base sobre la cual los médicos fundamentan sus decisiones, comparando los datos y la información específica de una persona con la base científica reconocida generalmente como verdad sobre la salud humana y la enfermedad.

A partir de la problemática que genera la aparición de las TIC en relación con la socialización del conocimiento, es vital que los datos estén debidamente indizados para evitar que se tergiverse la información en el proceso de toma de decisiones y se generen conocimientos errados. En el sector de la salud se pueden dar esas posibles confusiones en la base del conocimiento científico acumulado, ya sea éste tácito o explícito.

El impacto del desarrollo científico y tecnológico, producto de la revolución de la información en el marco de la SIC, se manifiesta en:

La velocidad de la generación de los conocimientos.

El desarrollo de las TIC.

La eliminación de las fronteras del conocimiento.

Los cambios operados en la práctica médica, que se expresan por medio de los siguientes logros:

- √ Mejoramiento significativo de la salud de la población mundial.
- √ Introducción de medicamentos de alta eficacia para el tratamiento de enfermedades.

√ Desarrollo de instituciones de salud de nuevo tipo.

√ Difusión del uso de las TIC como soporte de los modelos sanitarios.

1.2 Registro

El registro es el complemento de la gestión de la información ya que evita la pérdida de datos y permite que se realicen procedimientos con la información que en él se encuentra.

Según el Diccionario de la real academia cita textual registro es: "Acción y efecto de registrar. Conjunto de datos relacionados entre sí, que constituyen una unidad de información en una base de datos."

Según la Revista digital líder en Informática cita textual registro: "Es una pequeña unidad de almacenamiento destinada a contener cierto tipo de datos. Puede estar en la propia memoria central o en unidades de memoria de acceso rápido."

El principal objetivo de los registros, es proporcionar información útil sobre la prevalencia, evolución, resultados y necesidades. Los registros deben cumplir una serie de características, tales como la validez, la exactitud, la confianza y la calidad.

La utilidad de los registros depende básicamente de la calidad de los datos recogidos. Si los datos son de pobre calidad los resultados obtenidos con ellos pueden ser inciertos. Además, si la calidad de los datos no ha sido verificada, los resultados pueden ser de poca credibilidad, especialmente si se trata de datos que son inesperados. Se considera que un registro es de calidad, si proporcionan información válida tanto para una persona como para otra.

El registro es el conjunto de datos relacionados entre sí ya que toda la información que en él se encuentran están en estrecha relación una con otra mediante la acción del usuario con el sistema. Es una pequeña unidad de almacenamiento destinada a contener cierto tipo de datos, teniendo en cuenta que se trata como un área pequeña de almacenamiento de alta velocidad donde se guardan datos referentes a una persona y su desempeño en la realización de un examen.

El actual desarrollo rápido de sistemas de información tecnológica facilita el uso de sistemas electrónicos para recoger datos y mantener las bases de datos. Entre las ventajas del uso de sistemas electrónicos tenemos el obtener registros más competentes y eficaces (los datos pueden ser entrados continuamente sin limitaciones ni de tiempo, ni de lugar, el aumentar la calidad de los mismos por

permitir su inmediata computarización y chequeo de su consistencia y el permitir una más fácil diseminación de la información.

La calidad de los datos reportados debe garantizarse y validarse. Existen una serie de pasos a realizar para que el desarrollo de un registro sea de calidad. Estos son:

- √ Definir explícitamente los usos que se van a hacer de los datos obtenidos.
- √ Desarrollar el registro, de manera que sea comprensivo en términos de utilidad pero sin limitar su tamaño (decidir qué datos recoger, en qué tipo de personas, en que proceso, que resultados, todo ello previamente estandarizado).
- √ Desarrollo del mismo con la finalidad de reducir los errores de codificación y transcripción.
- √ Minimizar los errores de entrada de datos, si es necesario utilizando un procedimiento doble.

Los registros sostenibles se definen como aquellos objetos electrónicos y sus metadatos que los definen como registros, que han de ser guardados de forma continua por la organización propietaria o creadora de los mismos hasta que los registros puedan ser destruidos o, en su caso, trasladados a un archivo especializado para su archivo permanente.

Los elementos clave pueden resumirse como:

- √ La estructura de un registro, debe permanecer intacto su formato y las relaciones entre los elementos que componen el registro.
- √ El contexto en el que el registro fue creado, recibido y utilizado debe quedar aparente en el registro (incluyendo el proceso del cual la transacción forma parte, la fecha y hora de la transacción y los participantes en la transacción).
- √ Deben estar presentes los enlaces entre documentos que existen de forma separada pero que, combinados, forman un registro.

Las características esenciales de un registro son:

- √ Autenticidad

√ Fiabilidad

√ Integridad

√ Usabilidad

1.2.1 Autenticidad

Un registro auténtico es el que demostrablemente:

√ Ha sido creado o enviado por la persona que dice haberlo creado o enviado.

√ Ha sido creado o enviado en el momento que dice haber sido creado o enviado.

Para asegurar la autenticidad de los registros, las organizaciones deben implementar y documentar políticas y procedimientos que controlen la creación, la recepción, la transmisión, el mantenimiento y la disposición de los registros con el fin de asegurar que los creadores de registros estén autorizados e identificados, y que los registros queden protegidos contra adiciones, eliminaciones, modificaciones, usos y ocultaciones no autorizados.

Autenticidad puede existir únicamente si están presentes elementos suficientes de las otras tres características, puesto que la autenticidad en un ambiente electrónico sólo puede establecerse cuando las demás características también están presentes. Como tal, la autenticidad es un valor implícito, derivado o supuesto por la presencia de los elementos explícitos que caracterizan las otras tres características.

1.2.2 Fiabilidad

Un registro fiable es uno cuyo contenido puede ser considerado como una representación completa y exacta de las transacciones, actividades o hechos a que se refiere y en el que se puede confiar en el transcurso de posteriores transacciones o actividades. Los registros deben crearse en el momento de la transacción o incidente al que se refieren, o poco tiempo después, por personas que tienen conocimientos directos de los hechos o por instrumentos utilizados en forma rutinaria dentro de la actividad normal para realizar la transacción.

Característica de fiabilidad que: Para cualquier sistema desplegado el gestionar registros debe ser capaz de funcionar de modo continuo y regular, y debe estar basado en procedimientos responsables.

Un sistema de registros debe:

- √ Capturar de forma rutinaria todos los registros que estén dentro del alcance de las actividades que cubre.
- √ Organizar los registros de una manera que refleje los procesos de negocios del creador de los registros.
- √ Proteger los registros contra modificación o disposición no autorizadas.
- √ Funcionar de forma rutinaria como fuente principal de información sobre las acciones documentadas en los registros.
- √ Proporcionar un acceso fácil a todos los registros pertinentes y a sus metadatos asociados.

Puede argumentarse que la fiabilidad es un aspecto o función de los requisitos operativos inmediatos del negocio. Con el fin de implementar políticas y completar transacciones, cada negocio necesita registros fiables posicionados en un contexto lógico. Si la fiabilidad no forma parte de los procesos operativos mediante la adopción de una funcionalidad de gestión de registros en el momento de la creación y captura del registro, es poco probable que pueda postularse posteriormente con confianza.

1.2.3 Integridad

La integridad de un registro se refiere a su cualidad de completo e inalterado. Es necesario que un registro sea protegido contra la modificación no autorizada. Las políticas y procedimientos de gestión de registros deben especificar las adiciones o anotaciones que pueden ser realizadas en un registro después de su creación, las circunstancias bajo las cuales dichas adiciones o modificaciones pueden ser autorizadas, y las personas autorizadas para realizarlas. Cualquier anotación, adición o eliminación autorizada a un registro debe quedar indicada de forma explícita y ser trazable.

Los controles que definen la integridad incluyen:

- √ Establecer privilegios de acceso sobre la creación.
- √ Modificación.

- √ Anotación.
- √ Reubicación.
- √ Destrucción de registros.
- √ Implementar procedimientos para impedir, descubrir y corregir la pérdida o la corrupción de los registros.
- √ Implementar medidas que garanticen la continuada identidad e integridad de los registros pese al deterioro de los soportes y los cambios tecnológicos.
- √ Si existen múltiples copias de los registros, identificar formalmente el registro prevaleciente.
- √ Identificar y mantener de forma clara, junto a los registros, toda la documentación necesaria para comprender su contexto legal, administrativo y técnico.

La característica de la integridad puede en sí ser dividida en cuatro sub-elementos:

- √ Trazabilidad.
- √ Periodos de retención.
- √ Reglas, estándares y normas aplicables.
- √ Riesgo.

La cuestión de integridad de los registros está estrechamente relacionada con la planificación eficaz de continuidad del negocio, porque para clarificar el coste de mantener la integridad de un registro es necesario evaluar el riesgo para la organización si el registro ha sido retenido en forma incompleta o con limitadas posibilidades de auditoría.

1.2.4 Usabilidad

Un registro utilizable puede ser localizado, recuperado, presentado e interpretado. Debe ser posible presentarlo posteriormente como directamente relacionado con la actividad o transacción comercial que lo produjo. Los vínculos contextuales de los registros deben llevar la información necesaria para comprender las transacciones que los crearon y utilizaron. Debe ser posible identificar un registro dentro de un contexto más amplio de actividades y funciones de negocio. Deben mantenerse las

relaciones entre registros que documentan una secuencia de actividades.

La usabilidad no debe en ningún momento comprometer la integridad del registro.

La usabilidad consta de al menos cuatro elementos clave:

- √ El o los formato/s en que la organización puede desear visualizar o publicar esta información.
- √ La posibilidad de producir nuevas versiones en otros formatos como instancias adicionales del registro, manteniendo los enlaces al registro original.
- √ Los permisos de acceso que permiten el acceso al registro o a instancias editadas del registro (por ejemplo, si es necesario publicar o divulgar un subconjunto limitado de la información, pero reteniendo algunos detalles como nombres o direcciones).
- √ La posibilidad que tiene el usuario de saber dónde se obtuvo esta información, y dónde puede ser localizada todavía, y recuperada si se establece la necesidad de autenticación.

1.3 Estudio de situaciones

La informática ha tenido un gran auge en estos últimos años, y sobre todo sus avances en el campo de la realidad virtual a dado un paso de adelanto con el desarrollo de los simuladores en especial aquellos que se encuentran enmarcados en el área de la salud, más específicamente los relacionados con la cirugía.

Algunos simuladores disponen de salas de control las cuales contienen sistemas de grabación audiovisual que permiten reproducir posteriormente los escenarios propuestos, siendo esto muy incomodo para llevar el registro del desempeño de un médico, ya que debe existir un especialista para cada estudiante a evaluar, en caso de que sea el mismo especialista sería muy agotador y podría causar deficiencia en el momento de registrar los resultados debido al cansancio de tener que estar varias horas dedicado completamente a este trabajo tan riguroso. Además de que si va a realizar el registro de las trazas de la evaluación por el video, tendría que ver completa la grabación, para poder llegar a conclusiones (más información en el acápite 1.2).

Mientras otros tienen que generar potentes registros de archivos para el almacenamiento de datos. Estos son los más utilizados ya que dan más rapidez y eficiencia al sistema, puesto que si el

especialista quiere ver el reporte puede seleccionar la categoría por la cual desea buscar (más información en el acápite 1.2).

En Cuba el proceso de registrar el desempeño de un cirujano durante la ejecución de un examen se realiza mediante una hoja que tiene el especialista el cual se encuentra observando este proceso y va registrando los parámetros que se miden en el examen. Cuando concluye el examen los resultados alcanzados por el médico son archivados. Este proceso de registro no es óptimo ya que siempre tiene que existir un especialista para que un médico pueda evaluarse. Además de que se demora en darle la evaluación al cirujano. Esto también trae dificultad a la hora de buscar un examen archivado, debido al volumen de datos guardados, existiendo la posibilidad de que se pueda extraviar algún documento.

1.4 Tendencias Actuales

En el mundo de la medicina, los simuladores de paciente a escala real se introdujeron en los años 90 en el área de la anestesiología. Pero la mayoría de las especialidades sanitarias utilizan la técnica de la simulación con diferentes tipos de simuladores para mejorar la seguridad de los pacientes.

La simulación clínica es una herramienta que aporta a los profesionales un nuevo modelo capaz de superar los hábitos del tradicional aprendizaje pasivo en su trabajo diario; facilita una formación y un entrenamiento de excelencia en habilidades técnicas y multidisciplinarias; estimula a los médicos en la necesidad de innovar, facilita el trabajo en equipos clínicos de alto rendimiento y permite determinar el grado de destreza del cirujano, siendo esto diferente en cada simulador desarrollado. Existen algunos simuladores que incluyen contadores de tiempo, que miden el tiempo que el cirujano tarda en identificar los blancos propuestos, e incluso proporcionan una puntuación (como si de un videojuego se tratase) determinada cada vez que el cirujano realiza correctamente una maniobra. Así mismo, cualquier error convierte la imagen en borrosa, simulando el alcance de un órgano y la consiguiente pérdida de sangre que emborrona la lente. Esto hace bajar los “puntos” conseguidos por el cirujano.

Un aspecto esencial del nuevo proceso de formación, basado en el simulador, será su capacidad de valorar el grado de destreza del cirujano tras completar determinados procedimientos quirúrgicos, maniobras o tareas.

Para valorar en el simulador el grado de destreza del cirujano hay que combinar las maniobras de exposición quirúrgica, disección, corte, aprehensión y sutura dentro del mismo procedimiento laparoscópico. Estos parámetros serán tomados como variables del desarrollo del simulador, de forma

que progresivamente el cirujano debe perfeccionar cada uno de los aspectos quirúrgicos hasta completar las maniobras.

La simulación clínica es una técnica, no una tecnología, que pretende reemplazar la experiencia con pacientes reales por una práctica que evoque los aspectos fundamentales de la realidad de una manera completamente interactiva. En el sector de la simulación, se ofrecen ilimitadas posibilidades para el avance de las técnicas docentes sanitarias y cuyo crecimiento es imparable y representa un potencial de evolución para este nuevo milenio, ejemplo de ello, son los simuladores que se mencionan a continuación teniéndose en cuenta el proceso de evaluación en cada uno.

√ El primer centro de España que utilizó la tecnología de los simuladores de paciente a escala real para el entrenamiento, fue el de Santander. Teniendo en las salas de simulación instalados dos potentes simuladores de paciente a escala real el más popular es Celedonio, que permiten recrear cualquier escenario clínico de una manera segura, y por el que ya han pasado casi 6.000 alumnos. Estas salas de control del simulador disponen de sistemas de grabación audiovisual que permiten reproducir posteriormente los escenarios propuestos, y en las aulas de análisis se visualizan los casos clínicos mediante sistemas de control y visionado en pantallas planas, que permiten un posterior análisis con el grupo de alumnos.

√ El Tester Psicomotriz Avanzado de Dundee (en lo adelante TPAD) es un aparato controlado por medio de una computadora, desarrollado por el Hospital de Ninewells, de la Universidad de Dundee en México, para la evaluación objetiva del comportamiento de la tarea endoscópica.

El hardware del TPAD consiste de un mecanismo de soporte cardánico dual que acepta instrumentos endoscópicos estándares y una cámara para la manipulación bilateral, y un espacio definido de trabajo en tres dimensiones. Sobre el isocentro del aparato, se coloca una caja de prueba. Esta caja comprende cinco diferentes objetivos de tareas, basados en las principales acciones implicadas en la manipulación endoscópica. El TPAD está enlazado a una computadora estándar para la información de las tareas y el almacenaje de los datos. Algunos de los parámetros que se tienen en cuenta para la evaluación del practicante son: tiempo de ejecución, la culminación exitosa, el tiempo total de error de placa y el tiempo total de error de examen probatorio.

√ El Xitact LS500 es desarrollado por Xitact SA, Morges, Suiza, y es un simulador de colecistectomía laparoscópica de realidad virtual.

La plataforma de laparoscopia Xitact LS500 es una plataforma de capacitación modular de realidad virtual, que comúnmente trabaja con disecciones de peritoneo y la tarea de diseccionar y cortar de la colecistectomía laparoscópica.

Los ingenieros de software y hardware, trabajando junto con un consejo asesor de cirujanos laparoscópicos de gran fama y experiencia, desarrollaron el simulador. Éste es un híbrido, que combina un objeto físico que contiene el hardware mecánico, la imagen abierta, por ejemplo el abdomen virtual, y lo conecta con una computadora que proporciona la escenografía de la realidad virtual y la realimentación háptica. La computadora asigna un informe confeccionado y almacena conjuntos de datos de los participantes.

√ El Simulador de Cirugía Laparoscópica LapMentor del Centro de Alta Tecnología Henry Mayer, Hospital Clínico Universidad de Chile le proporciona al cirujano una sensación táctil y de manipulación de tejidos similar a la realidad. Esto se logra mediante dispositivos de realimentación de fuerza, denominados dispositivos hápticos. Además, este sistema tiene la capacidad de evaluar la destreza del cirujano frente a determinadas técnicas (corte, disección, prehensión) lo que posibilita objetivar destrezas. Aprender estas técnicas, requiere desarrollar nuevas habilidades y destrezas para manejar ciertas características propias como son la falta de percepción táctil directa, la mayor coordinación de movimientos complejos y la pérdida de percepción de profundidad.

Este simulador es un equipo de alta fidelidad que presenta el descrito sistema háptico, confiriéndole resistencia a los tejidos al ser manipulados. Además, posee un sistema de retroalimentación, otorgándole al cirujano una evaluación de su desempeño mediante la medición del tiempo que requirió, evaluando la precisión de los ejercicios, la eficiencia de los movimientos, y el cumplimiento de parámetros de seguridad ejercidos.

La empresa fabricante (Symbionix) denominó a los aparatos como "mentors", nombre acertado para estos verdaderos "mentores electrónicos", que hasta entregan un informe estadístico después de cada procedimiento. En él quedarán registrados los avances y también las fallas del alumno, quien podrá

esclarecer cuánto tiempo demoró en hacer la inspección de cada segmento o si encontró las patologías que debía encontrar.

√ ProMIS es uno de los simuladores existentes en el mercado. Permite a los cirujanos entrenarse en modelos virtuales y físicos en una misma unidad. Utiliza instrumentos y materiales quirúrgicos reales. Dispone de aparatos de lectura gráfica y vídeo. El simulador ofrece un modelo físico en el que pueden entrenarse en la colocación de trócares y el uso de instrumental. Los instrumentos pueden ser intercambiados, retirados y reinsertados. Las señales de audio contribuyen a que la experiencia sea más realista. Los cirujanos también pueden entrenar en equipo. Cada módulo de aprendizaje de ProMIS se divide en distintas tareas y niveles, lo que proporciona una trayectoria clara en el aprendizaje del usuario. ProMIS analiza, entre otras cosas, el tiempo empleado, las longitudes de las trayectorias o la suavidad de los movimientos del cirujano: en qué grado sus movimientos son fluidos o erráticos.

√ LapSim System es otro de los simuladores existentes en el mercado. El entrenamiento de las habilidades básicas incluye la navegación con cámara, la navegación con instrumental, la coordinación, grapado, disección, colocación de clips, sutura y medición de precisión y velocidad. En todos los ejercicios el cirujano debe identificar el objeto propuesto y dirigir hacia él, el instrumental, realizando la tarea requerida con la mayor precisión y en el menor tiempo posible.

√ El simulador Wesst-OT de la Pontificia Universidad Javeriana de Colombia requiere diversas habilidades; por ejemplo, la habilidad de orientación es aquella que permite al cirujano navegar, desplazarse e identificar toda la anatomía involucrada; por otra parte, la coordinación, de acuerdo con el doctor Carlos Gamboa, especialista en otorrinolaringología, es una habilidad muy importante que permite la navegación e interacción de los instrumentos quirúrgicos con la anatomía del seno para nasal simultáneamente. Esta debe darse al manipular tanto el instrumento quirúrgico como el endoscopio.

Según el diseño de este simulador se analizarán las clases relacionadas con la habilidad de coordinación y las relacionadas con el entorno de personalización de sesiones para ver cómo interactúan las clases en el proceso de evaluación.

Las clases relacionadas con la habilidad de coordinación.

La clase Habilidad es fundamental; esta permite definir los componentes y la información específica de cada una de las habilidades quirúrgicas que se deben tener en el entorno para realizar la cirugía; esta clase posibilita definir habilidades como la orientación y la coordinación, entre otras. La clase Habilidad es uno de los principales cambios que se pueden observar respecto al diseño original de West-OT y fue creada para posibilitar la generalización del entorno de práctica. Adicionalmente existe un Entorno asociado a cada habilidad; esta clase crea el entorno gráfico y despliega en él los elementos anatómicos y no anatómicos necesarios para la habilidad quirúrgica previamente seleccionada por el usuario. El panel de instrumentos quirúrgicos se visualiza y se usa principalmente para la habilidad de coordinación y elementos como símbolos de preguntas se cargan en el entorno en lugares específicos dependiendo de la habilidad a practicar; la dependencia de Entorno con Habilidad permite crear una instancia de los objetos necesarios para el desarrollo de la habilidad elegida únicamente. También, la clase Ruta maneja puntos de referencia para determinar si el camino que está siguiendo el usuario es el adecuado; esta maneja puntos de tolerancia para evaluar el camino seguido y observa si el usuario está haciendo las fases de la cirugía de senos para nasales en el orden correcto.

Las clases relacionadas con el entorno de personalización de sesiones.

La clase Personalizar Sesión brinda la posibilidad al administrador de modificar el entorno que será visualizado por un usuario practicante, al servir de puente entre la clase que captura los parámetros de la personalización y la clase que realiza las operaciones sobre la base de datos. La clase Anomalía sirve para realizar operaciones de creación, eliminación, actualización y consulta de anomalías sobre la base de datos del simulador. La clase Parámetros Personalización obtiene los parámetros de la anomalía asociados con la posición, la escala, el color y el texto.

√ El proyecto SINERGIA propone integrar un simulador quirúrgico laparoscópico en el sistema de formación en cirugía de mínima invasión, intentando perfeccionar el aprendizaje convencional en estas técnicas quirúrgicas. Dicho entorno virtual será el complemento idóneo del periodo de aprendizaje en cirugía laparoscópica, con capacidad para gestionar una base de datos con modelos tridimensionales de estructuras orgánicas que se emplearán para el aprendizaje y las prácticas en el entrenamiento en cirugía laparoscópica.

√ Beatriz Robles, directora de la carrera de Enfermería de la Universidad Andrés Bello Viña del Mar en Chile explica que con los simuladores que cuenta la universidad los alumnos pueden realizar procesos de aprendizaje como son evaluación y valoración de pacientes.

El laboratorio de simulación cuenta con programas de software de última generación, que recrean signos y síntomas de enfermedades que permiten dar cuidados específicos, crear registro del número de oportunidades que el alumno ensayó, número de errores efectuados, tiempo de demora en reaccionar; al final el docente sabe cuáles son las dificultades en el aprendizaje del estudiante.

De esta forma se puede ver como en la actualidad existe un gran número de empresas, sitios web y simuladores que necesitan almacenar y administrar un gran volumen de información. Para ello emplean gestores de base de datos que son sistemas que permiten la realización y administración de las bases de datos. En el mundo existen muchos gestores que ofrecen innumerables ventajas para administrar unas bases de datos, para ello se analizará el comportamiento de los gestores de base de datos desde sus inicios hasta la actualidad.

“El diseño de bases de datos es el proceso por el que se determina la organización de una base de datos, incluidos su estructura, contenido y las aplicaciones que se han de desarrollar. Durante mucho tiempo, el diseño de bases de datos fue considerado una tarea para expertos: más un arte que una ciencia. Sin embargo, se ha progresado mucho en el diseño de bases de datos y éste se considera ahora una disciplina estable, con métodos y técnicas propios. El diseño de bases de datos (en lo adelante BD) ha pasado a constituir parte de la formación general de los informáticos, en el mismo nivel que la capacidad de construir algoritmos usando un lenguaje de programación convencional. [2]

A finales de la década de 1960, cuando las bases de datos entraron por primera vez en el mercado del software, los diseñadores de bases de datos actuaban como artesanos, con herramientas muy primitivas: diagramas de bloques y estructuras de registros eran los formatos comunes para las especificaciones, y el diseño de bases de datos se confundía frecuentemente con la implantación de las bases de datos. Esta situación ahora ha cambiado: los métodos y modelos de diseño de bases de datos han evolucionado paralelamente con el progreso de la tecnología en los sistemas de bases de datos. Se ha entrado en la era de los sistemas relacionales de bases de datos, que ofrecen poderosos

lenguajes de consulta, herramientas para el desarrollo de aplicaciones e interfaces amables con los usuarios.

En la actualidad los Sistemas de Gestión de Base de Datos (en lo adelante SGBD) relacionales están en plena transformación para adaptarse a tres tecnologías de éxito reciente, fuertemente relacionadas: la multimedia, la de orientación a objetos (en lo adelante OO) para su uso en la simulación, Internet y la web.

Los primeros SGBD en los años sesenta todavía no se les denominaba así, estaban orientados a facilitar la utilización de grandes conjuntos de datos en los que las interrelaciones eran complejas. La aparición de los SGBD relacionales en los años ochenta supone un avance importante para facilitar la programación de aplicaciones con BD y para conseguir que los programas sean independientes de los aspectos físicos de las BD.

Los tipos de datos que se pueden definir en los SGBD relacionales de los años ochenta y noventa son muy limitados. La incorporación de tecnologías multimedia –imagen y sonido – en los Sistemas Informáticos (en lo adelante SI) hace necesario que los SGBD relacionales acepten atributos de estos tipos. Sin embargo, algunas aplicaciones no tienen suficiente con la incorporación de tipos especializados en multimedia. Necesitan tipos complejos que el desarrollador pueda definir a medida de la aplicación. En definitiva, se necesitan tipos abstractos de datos (en lo adelante TAD).

Los SGBD más recientes ya incorporaban esta posibilidad, y abren un amplio mercado de TAD predefinidos o librerías de clases. Esto lleva a la orientación a objetos. Al final de los años ochenta tuvo un gran éxito, en el desarrollo de software básico, en las aplicaciones de ingeniería industrial y en la construcción de interfaces gráficas con los usuarios, ha hecho que durante la década de los noventa se extendiese en prácticamente todos los campos de la informática.

En los Sistemas Informáticos se inicia también la adopción, de la programación OO. La utilización de lenguajes como C++ o Java requiere que los SGBD relacionales se adapten a ellos con interfaces adecuadas.

1.4.1 De un Sistema de Gestión de Bases de Datos se espera:

√ Permita a los usuarios crear otras bases de datos y especificar su esquema por medio de algún lenguaje de definición.

- √ Ofrezca a los usuarios la capacidad de consultar los datos y modificarlos, usando para ello un lenguaje de consulta y manipulación.
- √ Brinde soporte al almacenamiento de cantidades voluminosas de datos durante un largo período, protegiéndolo contra accidentes o utilización no autorizada.
- √ Controlar el acceso concurrente.

1.5 Base de Datos

Base de datos de un Sistema Informático es la representación integrada de los conjuntos de entidades instancia correspondiente a las diferentes entidades tipo del SI y de sus interrelaciones. Esta representación informática (o conjunto estructurado de datos) debe poder ser utilizada de forma compartida por muchos usuarios de distintos tipos.

En otras palabras, una base de datos es un conjunto estructurado de datos que representa entidades y sus interrelaciones. La representación será única e integrada, a pesar de que debe permitir utilidades varias y simultáneas.

Una base de datos es un conjunto de datos almacenados entre los que existen relaciones lógicas y ha sido diseñada para satisfacer los requerimientos de información de una empresa u organización. En una base de datos, además de los datos, también se almacena su descripción. [5]

1.5.1 Funcionalidad de una Base de Datos

Las bases de datos, son utilizadas en sistemas que requieren una interacción fluida con la aplicación; estas se encargan muchas veces de administrar, editar, y dar de alta. Usualmente la base de datos, está ligada a la programación directa del site, causando que una edición en ella cause una modificación directa en lo que ve el usuario.

El conjunto de componentes o herramientas conceptuales que un SGBD proporciona para modelar recibe el nombre de modelo de BD. Los cuatro modelos de BD más utilizados en los Sistemas Informáticos son: el modelo relacional, el modelo jerárquico, el modelo en red y el modelo relacional con objetos.

El Primer modelo que apareció a principio de los años sesenta, fue el modelo jerárquico. Sus estructuras son registros interrelacionados en forma de árboles. El SGBD clásico de este modelo es el IMS/DL1 de IBM.

A principios de los setenta surgieron SGBD basados en un modelo en red. Como en el modelo jerárquico, hay registros e interrelaciones, pero un registro ya no está limitado a ser “hijo” de un solo registro tipo.

Durante los años ochenta apareció una gran cantidad de SGBD basados en el modelo relacional propuesto en 1969 por E.F. Codd, de IBM, y prácticamente todos utilizaban como lenguaje nativo el SQL. El modelo relacional se basa en el concepto matemático de relación, que aquí se puede considerar de momento equivalente al término tabla (formada por filas y columnas). La mayor parte de los SI que actualmente están en funcionamiento utilizan SGBD relacionales, pero algunos siguen utilizando los jerárquicos o en red (especialmente en SI antiguos muy grandes).

Así como en los modelos pre relacionales (jerárquico y en red), las estructuras de datos constan de dos elementos básicos (los registros y las interrelaciones), en el modelo relacional constan de un solo elemento: la tabla, formada por filas y columnas. Las interrelaciones se deben modelizar utilizando las tablas.

Otra diferencia importante entre los modelos pre relacionales y el modelo relacional es que el modelo relacional se limita al nivel lógico (no hace absolutamente ninguna consideración sobre las representaciones físicas). Es decir, da una independencia física de datos total. Esto es así, si se habla del modelo teórico, pero los SGBD del mercado proporcionan una independencia limitada.

Estos últimos años se está extendiendo el modelo de BD relacional con objetos. Se trata de ampliar el modelo relacional, añadiéndole la posibilidad de que los tipos de datos sean tipos abstractos de datos, TAD. Esto acerca los sistemas relacionales al paradigma de la OO.

Hay modelos de datos que no son utilizados por los SGBD del mercado: sólo se usan durante el proceso de análisis y diseño, pero no en las realizaciones.

Los más conocidos de estos tipos de modelos son los modelos semánticos y los funcionales. Éstos proporcionan herramientas muy potentes para describir las estructuras de la información del mundo real, la semántica y las interrelaciones, pero normalmente no disponen de operaciones para tratarlas.

Se limitan a ser herramientas de descripción lógica. Son muy utilizados en la etapa del diseño de BD y en herramientas CASE. El más extendido de estos modelos es el conocido como modelo ER (entity-relationship).

Actualmente, la práctica más extendida en el mundo profesional de los desarrolladores de SI es la utilización del modelo ER durante el análisis y las primeras etapas del diseño de los datos, y la utilización del modelo relacional para acabar el diseño y construir la BD con un SGBD.

1.6 Gestor de Base de datos. Concepto

Un gestor de base de datos o sistema de gestión de base de datos es un software que permite introducir, organizar y recuperar la información de las bases de datos; en definitiva, administrarlas. Existen distintos tipos de gestores de bases de datos: relacional, jerárquico, red... El modelo relacional es el utilizado por casi todos los gestores de bases de datos para PC's. El modelo relacional es un software que almacena los datos en forma de tablas. [\[6\]](#)

Se puede decir que el propósito general de los sistemas de gestión de base de datos es el de manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información, evitando además la redundancia de los datos.

1.6.1 Ventajas de los Gestores de Base de Datos:

- √ Seguridad y confidencialidad integral
- √ Independencia de los datos respecto a los tratamientos y viceversa.
- √ Coherencia de los resultados.
- √ Mejor disponibilidad de los datos para el conjunto de los usuarios.
- √ Mayor valor informativo (puesto que se captan relaciones propias del mundo real entre los datos, tiene más información que la suma individual de los datos por separado).
- √ Mejor y más normalizada documentación de la información, la cual está integrada con los datos.
- √ No hay duplicidad de información, ya que comprueba la información en el momento de introducir la misma.
- √ Gran velocidad en muy poco tiempo.
- √ Mayor eficiencia en la recogida, validación e introducción de los datos en el sistema.

√ Reducción del espacio de almacenamiento.

1.6.2 Desventajas de los Gestores de Base de Datos:

√ Instalación y la actualización del hardware y software son muy elevados.

√ Tiene que trabajar en ella personal especializado.

√ Un mal diseño de esta puede originar problemas en el futuro.

√ Falta de rentabilidad a corto plazo.

√ Generan campos vacíos en exceso.

√ Desfase entre teoría y práctica. Esto no tiene que ocurrir necesariamente, se refiere a un desfase entre la concepción teórica, de lo que es la base de datos, que pueden los directivos con respecto a lo que ofrece la tecnología realmente. [7]

Actualmente existe una gran competencia entre los gestores, ¿Cual es el mejor gestor? , ¿Cuál debo usar?, son preguntas que normalmente surgen cuando se presenta un problema y para su solución se necesita del desarrollo de una base de datos. Las más conocidas en la actualidad son el Oracle, SQLServer, MySQL, PostgreSQL y Firebird.

1.7 Análisis de los Gestores de Base de Datos

Para seleccionar la herramienta a utilizar se debe de realizar un análisis de las características principales de los distintos gestores de base de datos más utilizados en la actualidad, para este análisis y selección vamos a tener en cuenta los siguientes gestores: Oracle, SQL Server, MySQL, PostgreSQL y Firebird.

1.7.1 Oracle

Oracle es un sistema de gestión de base de datos relacional. Se considera como uno de los sistemas de bases de datos más completos, destacando su:

√ Soporte de transacciones, que no es más que una interacción con una estructura de datos que, aún siendo compleja y estar compuesta por varios procesos que se han de aplicar uno después del otro, se pretende que sea equivalente a una interacción. Es decir, que se realice de una sola vez.

√ Estabilidad, lo que significa que su nivel de fallo disminuye, en dependencia de la estabilidad que se requiera.

√ Escalabilidad, es decir, su posibilidad de estar preparado para hacerse más grande sin perder calidad en los servicios ofrecidos, siendo capaz de cambiar su tamaño o configuración para adaptarse a las circunstancias cambiantes.

√ Es multiplataforma.

Su mayor defecto es su enorme precio, que es de varios miles de euros (según versiones y licencias).

Oracle resulta muy difícil de instalar. No dispone de un fichero rpm de instalación. Por lo que se debe bajar un instalador particular del website de Oracle.

Aunque su dominio en el mercado de servidores empresariales ha sido casi total hasta hace poco, recientemente sufre la competencia del Microsoft SQL Server de Microsoft y de la oferta de otros RDBMS con licencia libre como PostgreSQL, MySQL.

1.7.2 SQL Server

Es un sistema de gestión de base de datos relacionales, capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea. Este es un lenguaje de programación interactivo y estandarizado, declarativo de acceso. Agrupa características del álgebra y el cálculo relacional, permitiendo la realización de consultas para recuperar de forma sencilla la información de una base de datos. Teniendo unas ventajas que a continuación se pueden describir.

√ De transacciones.

√ Escalabilidad, estabilidad y seguridad.

√ Soporta procedimientos almacenados.

√ Incluye también un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.

√ Permite trabajar en modo cliente-servidor donde la información y datos se alojan en el servidor y las terminales o clientes de la red sólo acceden a la información.

√ Además permite administrar información de otros servidores de datos.

Microsoft SQL Server, no es multiplataforma, ya que sólo está disponible en Sistemas Operativos de Microsoft (Windows) con nuestros productos Plan Empresarial, Servidor Virtual, Servidor Virtual Ecommerce y Servidor Virtual ISP o como Hosting Base de Datos SQL Server, la licencia es de alto costo.

1.7.3 MySQL

MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario que gestiona bases de datos relacionales poniendo las tablas en ficheros diferenciados, con más de seis millones de instalaciones, es uno de los gestores de base de datos más populares desarrollados bajo la filosofía de código abierto. (Open Source).

MySQL es muy utilizado en aplicaciones web como MediaWiki o Drupal, en plataformas (Linux/Windows-Apache-MySQL-PHP/Perl/Python), y por herramientas de seguimiento de errores como Bugzilla. [\[8\]](#)

MySQL tiene como principales características:

√ Facilidad de configuración e instalación.

√ Su gran velocidad y su precio reducido. Es uno de los servidores de bases de datos más rápido y el de menor precio.

√ MySQL es muy utilizado en aplicaciones PHP o Perl en servidores Linux. En general, si no necesita características como transacciones, procedimientos almacenados, triggers o sentencias SQL complejas, MySQL cumplirá la misma función que otras bases de datos más potentes, pero de forma más rápida y con un coste menor.

√ Para aplicaciones Windows, MySQL es una alternativa económica a SQL Server, pues su coste es menor y además se puede aprovechar todo el espacio para datos, mientras que SQL Server necesita una parte del espacio para el log de transacciones. [9]

Aspectos a considerar para el uso de MySQL.

Un aspecto a tener en cuenta de la versión 5 es que la misma hace mucho hincapié en InnoDB, (su principal motor de almacenamiento transaccional para el código abierto) el cual fue comprado hace unos años por Oracle, por lo que MySQL ha estado desarrollando una alternativa llamada Falcon, que se liberará con MySQL 6.0.

Sun Microsystems tras la compra el 16 de enero de 2008 de MySQL por 1.000 millones de dólares, el futuro de este gestor es muy incierto aunque la empresa se comprometió en no interferir en el modelo de desarrollo. Según criterios de expertos algunos usuarios se han mostrado escépticos tras esta adquisición, por el impacto que pudiera tener esta en el futuro de MySQL. [10]

MySQL es gratis incluso para su uso comercial mientras trabaje como servidor de web, pero si se desea trabajar con otras aplicaciones será entonces necesario obtener una licencia.

1.7.4 PostgreSQL

El Sistema Gestor de Bases de Datos Relacionales Orientadas a Objetos. PostgreSQL es el gestor de bases de datos de código abierto más avanzado hoy en día, ofreciendo control de concurrencia multi-versión, soportando casi toda la sintaxis SQL (incluyendo subconsultas, transacciones, y tipos y funciones definidas por el usuario). PostgreSQL se diseñó como una base de datos orientada a objetos. Esto significa, que las tablas no son tablas, sino objetos, y las tuplas son instancias de ese objeto. Mediante un sistema denominado MVCC (Acceso concurrente multiversión) PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos.

Esto evita que la transacción vea datos inconsistentes que pueden ser causados por la actualización de otra transacción concurrente en la misma fila de datos, proporcionando aislamiento transaccional para cada sesión de la base de datos. PostgreSQL es capaz entonces de manejar los registros sin necesidad de que los usuarios tengan que esperar a que los registros estén disponibles.

Los bloqueos de tabla han sido sustituidos por el control de concurrencia multiversión, el cual permite a los accesos de sólo lectura continuar leyendo datos consistentes durante la actualización de registros, y permite copias de seguridad en caliente desde `pg_dump` mientras la base de datos permanece disponible para consultas.

Es apropiada para aplicaciones más complejas, en las cuáles se utilizan las normas ANSI-SQL, vistas (views), métodos (stored procedures) y disparadores (triggers). [\[11\]](#)

Características del PostgreSQL:

√ PostgreSQL es el servidor de bases de datos de código abierto más potente que existe y es por tanto la alternativa a MySQL cuando se necesitan características avanzadas como transacciones, procedimientos almacenados, triggers, vistas, etc.

√ Es el servidor de bases de datos más utilizado por los programadores de servlets de Java y, en general, por todos aquellos que realizan aplicaciones cliente servidor complejas o críticas en el mundo Linux/Unix.

√ Para aplicaciones Windows, PostgreSQL es una alternativa económica a SQL Server, pues su coste es menor y tiene similares prestaciones. Esta diferencia económica es especialmente sustancial si se necesita un Servidor Dedicado de bases de datos.

√ Buena escalabilidad. Debido a su diseño, conforme se aumenta el número de procesadores aumenta el rendimiento.

√ Soporta un subconjunto del estándar SQL mayor que el de MySQL y tiene propiedades de orientación a objetos.

√ Incorpora una estructura de datos array.

√ Permite la declaración de funciones propias, así como la definición de disparadores.

√ Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos.

[\[12\]](#)

En principio, tiene un sistema de bases de datos de mayor nivel a MySQL, a la altura de Oracle, Sybase.

Limitaciones:

√ El soporte a orientación a objetos es una simple extensión que ofrece prestaciones como la herencia, no un soporte completo.

√ La mayor limitación de PostgreSQL viene dada por su velocidad: es el sistema de bases de datos más lento, sobrecarga el sistema bastante más que MySQL.

1.7.5 Firebird

Firebird es un sistema de administración de base de datos relacional (en lo adelante RDBMS) de código abierto, basado en la versión 6 de Interbase, cuyo código fue liberado por Borland en 2000. [\[13\]](#)

El Firebird surgió a finales de la década de 1990, Borland decidió liberar el código de Interbase. Diversos integrantes de la plantilla crearon una nueva empresa denominada IBPhoenix, y junto a otros desarrolladores independientes, crearon el fork ahora conocido como Firebird. Más tarde, Borland decidiría volver a privatizar Interbase y comercializar sus licencias. Sin embargo, Firebird sigue siendo un proyecto de código abierto bajo una licencia similar a la del Mozilla Public License (en lo adelante MPL).

La arquitectura del Firebird está compuesta por dos partes: el servidor, y el cliente que se comunica con él, donde el servidor es la parte que hace el trabajo: almacena los datos, los procesa, ejecuta los procedimientos almacenados y aplica las validaciones, y el cliente es el encargado de la comunicación entre las aplicaciones y el servidor. Maneja los protocolos, envía y recibe datos, envía comandos.

Características del Firebird:

√ Es multiplataforma, y actualmente puede ejecutarse en los sistemas operativos: Linux, HP-UX, FreeBSD, Mac OS, Solaris y Microsoft Windows.

√ Ejecutable pequeño, con requerimientos de hardware bajos.

√ Arquitectura Cliente/Servidor sobre protocolo TCP/IP y otros.

- √ Soporte de transacciones y claves foráneas.
- √ Es medianamente escalable.
- √ Buena seguridad basada en usuarios/roles.
- √ Bases de datos de sólo lectura, para aplicaciones que corran desde dispositivos sin capacidad de escritura, como CD-ROM.
- √ Existencia de controladores Java DataBase Connectivity (ODBC), Object Linking and Embedding for DataBase "Incrustación y enlace de objetos para bases de datos" (OLEDB).
- √ Controlador nativo para PHP.
- √ Requisitos de administración bajos, siendo considerada como una base de datos libre de mantenimiento, al margen de la realización de copias de seguridad y restauraciones periódicas.
- √ Pleno soporte del estándar SQL-92, tanto de sintaxis como de tipos de datos.
- √ Posibilidad de programación del servidor mediante disparadores (triggers) y procedimientos almacenados (stored procedures).
- √ Capacidad de almacenar elementos Binary Large Objects (BLOB).
- √ Posibilidad de definir procedimientos almacenados que devuelvan 'tablas virtuales'.
- √ Integridad referencial.
- √ Soporte para funciones externas.
- √ Prácticamente no necesita configuración.
- √ Backups incrementales.

√ Un lenguaje SQL amplio y potente, que contempla funciones avanzadas como son:

- Subconsultas.
- Uniones.
- Funciones de agregación.
- Ordenamiento por columnas calculadas.
- Vectores multidimensionales (matrices).

√ Disponible para arquitecturas de 64bits.

1.8 Librería IBPP

IBPP es una biblioteca de clases, libre de cualquier herramienta de desarrollo, fue confeccionada para añadir al Firebird acceso en cualquier aplicación C + +, en ella se encuentra casi la totalidad de lo que se necesita para acceder a una base de datos Firebird, y manipular los datos. IBPP también ofrece acceso a la mayoría de las administraciones de tareas: la creación de una base de datos, modificando su estructura, la realización de copias de seguridad en línea, la administración de cuentas de usuario en el servidor y así sucesivamente.

1.9 Lenguaje C++

Un lenguaje de programación es una sintaxis y un léxico particular, como cualquier otro lenguaje, se utiliza para escribir programas que se ejecutan por medio de un computador, teniendo cada lenguaje su propia sintaxis.

El C++ es un lenguaje de programación, diseñado a mediados de los años 1980, por Bjarne Stroustrup, como extensión del lenguaje de programación C, la cual es la variante más difundida y aceptada. Se puede decir que C++ es un lenguaje que abarca tres paradigmas de la programación: la programación estructurada, la programación genérica y la programación orientada a objetos.[\[14\]](#)

Esta versión combina la flexibilidad y el acceso de bajo nivel de C con las características de la programación orientada a objetos como abstracción, encapsulación y ocultación.

La POO es una nueva filosofía de programación que se basa en la utilización de objetos, sus mecanismos básicos son: objetos, mensajes, métodos y clases.

1.9.1 Ventajas del C++

- √ Lenguaje muy eficiente puesto que es posible utilizar sus características de bajo nivel para realizar implementaciones óptimas.
- √ A pesar de su bajo nivel es el lenguaje más portado en existencia, habiendo compiladores para casi todos los sistemas conocidos.

- √ Proporciona facilidades para realizar programas modulares y/o utilizar código o bibliotecas existentes.

1.10 Proceso de desarrollo de software

El Proceso Unificado Racional (Rational Unified Process en inglés, habitualmente resumido como RUP) es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. El RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización.

RUP es uno de los procesos más generales de los existentes actualmente, ya que en realidad está pensado para adaptarse a cualquier proyecto y no tan solo al software.

RUP se repite a lo largo de una serie de ciclos que constituyen la vida de un producto. Cada ciclo concluye con una generación del producto para los clientes. Cada ciclo consta de cuatro fases:

- √ Inicio: se hace un plan de fases, se identifican los principales casos de uso y se identifican los riesgos.
- √ Elaboración: se hace un plan de proyecto, se completan los casos de uso y se eliminan los riesgos.
- √ Construcción: se concentra en la elaboración de un producto totalmente operativo y eficiente y el manual de usuario.
- √ Transición: se implementa el producto en el cliente y se entrena a los usuarios. Como consecuencia de esto suelen surgir nuevos requisitos a ser analizados.

Los autores de RUP destacan que el proceso de software propuesto por RUP tiene tres características esenciales: está dirigido por los Casos de Uso, está centrado en la arquitectura, y es iterativo e incremental.

En RUP se han agrupado las actividades en grupos lógicos definiéndose 9 flujos de trabajo principales. Los 6 primeros son conocidos como flujos de ingeniería (Modelamiento del negocio, Requerimientos, Análisis y Diseño, Implementación, Prueba, Instalación) y los tres últimos como de apoyo (Administración del proyecto, Administración de configuración y cambios, Ambiente).

1.11 Herramienta a utilizar

Luego de haber realizado este análisis tan exhaustivo de las distintas características de los diferentes gestores de más repercusión en el mundo, se hará en este epígrafe un resumen comparativo para seleccionar la herramienta de trabajo a utilizar en el proceso de desarrollo de base de datos.

Teniendo en cuenta que Oracle es muy costoso, que SQL Server no es multiplataforma, y que ambos son software propietario se puede deducir que ambos no son de utilidad a este trabajo, aunque sean gestores muy potentes.

Por lo tanto nos queda más reducido este grupo, ya que solo quedaría la comparación entre el PostgreSQL, MySQL y Firebird, que son software libre, aunque no se tendrá muy en cuenta a MySQL ya que este mientras se le dé un uso particular es gratis, incluso si se le da uso comercial mientras trabaje como servidor web, pero si se utiliza con otras aplicaciones se tendrá que obtener una licencia. Además de tener actualmente un futuro incierto debido a la compra realizada por Sun Microsystems el 16 de enero de este año. .

En el campo de empleo de la velocidad se puede decir el PostgreSQL es un poco lento y esa es una de las características que más dificulta su uso. El Firebird por su parte es rápido con bajos consumo de recursos y utiliza poco requerimientos de hardware.

Como resultado de este profundo análisis sobre los gestores más populares a nivel mundial se decidió escoger el Firebird ya que lo que se persigue es diseñar y construir una base de datos potente. Este gestor además de las características analizadas anteriormente, utiliza como medio de conexión la librería IBPP la cual es una biblioteca de clases, libre de cualquier herramienta de desarrollo, que fue confeccionada para añadir al Firebird acceso en cualquier aplicación C + +, permitiendo su uso en

plataformas como Windows y Linux, facilitando que en otras versiones del proyecto que se decida migrar, la capa de acceso a datos no sufra cambios o en caso de que haya que realizarle alguno, no sean de gran envergadura. Todas estas características demuestran que es idóneo para las base de datos grandes como la que requiere un Simulador Quirúrgico.

Como metodología de desarrollo de software se va emplear RUP, ya que es uno de los procesos más generales que existe actualmente, además de estar basado en la orientación a objeto y utilizar al UML como lenguaje de modelado. También garantiza que se desarrollen todas las fases con los respectivos flujos de trabajo, pero en el caso de este trabajo se van a tener en cuenta los flujos de trabajos de ingeniería que son los más importantes para el desarrollo del mismo.

Capítulo 2 Características del Sistema.

Introducción.

En este capítulo se especifica el objeto de automatización, la propuesta del sistema, se describe el modelo del dominio, además de exponer todos los requisitos tanto funcionales como no funcionales, la representación del diagrama del sistema y la descripción textual de los casos de usos conformados. En este capítulo también se definen los distintos artefactos del flujo de trabajo de análisis y diseño.

2.1 Objeto de automatización.

Con el desarrollo de este trabajo se pretende automatizar el proceso de almacenamiento de datos, referente a la realización de un ejercicio y de igual forma se va a tener en cuenta los datos personales del médico cirujano, como son: nombre, apellidos, usuario, contraseña, dirección particular, teléfono, centro de trabajo, años de experiencia. El almacenamiento de los parámetros que van a poder medir el desempeño del usuario durante la realización de las pruebas se hace con el objetivo de que cuando el profesor desee ver el reporte de cada médico cirujano, el sistema le permita acceder, a los resultados obtenidos por el médico en el examen.

2.2 Propuesta del Sistema.

Se propone la realización de un módulo de registro de evaluación que contenga datos personales de los médicos a evaluar, además de brindar la posibilidad de llevar un control específico de todas las variables que intervienen en la realización de un examen, permitiendo gestionar la información de los resultados alcanzados por cada usuario del sistema. Se debe lograr que la manipulación de la información sea de una manera segura y confiable.

2.3 Modelo del dominio.

Para dar un enfoque a todo el proceso de actualización de la información, este trabajo se basará en el modelo de dominio, ya que este modelo permite de manera visual mostrar al cliente los principales conceptos que se manejan en el dominio del sistema de desarrollo. Esto ayuda a los clientes y desarrolladores e interesados, a utilizar un vocabulario común para poder entender el contexto en que

se trabaja en el sistema. Seguidamente se identificarán todos los conceptos asociados con el problema.

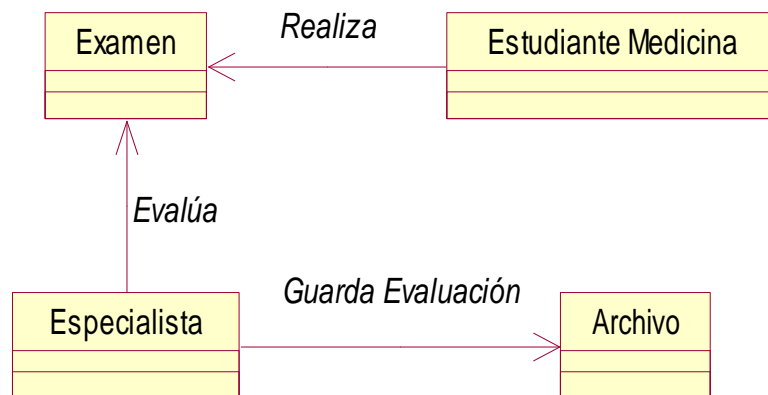
Examen: Método para saber los conocimientos adquiridos por un estudiante.

Estudiante Medicina: Persona que cursa estudios en el campo de la medicina.

Archivo: Conjunto de documentos de evaluaciones de los estudiantes que son guardados en un local.

Especialista: Es la persona que se dedica a una rama específica en este caso de la medicina y puede dar un criterio sobre el desempeño del estudiante.

2.3.1 Diagrama del Modelo del Dominio



2.4 Especificación de los requisitos del software.

Para lograr un buen funcionamiento del módulo de almacenamiento de los datos, es necesario que se cumplan una serie de requisitos funcionales y no funcionales.

2.4.1 Requerimientos funcionales.

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir. [\[15\]](#)

- R 1. El sistema debe permitir registrar un usuario.
- R 2. El sistema debe permitir la autenticación del usuario.

R 3. Gestionar usuarios.

3.1 El sistema debe permitir asignar un nuevo rol al usuario.

3.2 El sistema debe permitir modificar los datos de un usuario.

3.3 El sistema debe permitir eliminar un usuario.

Requisitos para el módulo de gestión del estudiante.

R 4. El sistema debe almacenar los posibles ejercicios de entrenamiento a utilizar por un usuario.

R 5. El sistema debe almacenar los resultados obtenidos por el estudiante en el examen realizado.

Ejercicio 1 - Requisitos funcionales ejercicio Coordinación.

R 6. Almacenar datos del ejercicio de Coordinación.

6.1 Almacenar estadísticas del tiempo de indecisión (tiempo en que están las pinzas sin moverse).

6.2 Almacenar estadísticas errores cometidos (no correspondencia de pinzas con objetos).

6.3 Almacenar estadística de aciertos (cantidad de colisiones pinza-objeto).

R 7. El sistema debe dar un reporte del desempeño del estudiante en el ejercicio de Coordinación.

Ejercicio 2 - Requisitos funcionales ejercicio Cámara.

R 8. Almacenar datos del ejercicio de Cámara.

8.1 Almacenar estadísticas errores de focalización.

8.2 Almacenar estadísticas de velocidad de movimiento de la cámara.

8.3 Almacenar estadísticas del tiempo de ausencia de movimiento de la cámara.

R 9. El sistema debe dar un reporte del desempeño del estudiante en el ejercicio de Cámara.

Ejercicio 3 - Requisitos funcionales ejercicio Selección.

R 10. Almacenar datos del ejercicio de Selección.

10.1 Almacenar estadísticas del tiempo de indecisión (tiempo en que están las pinzas sin moverse).

10.2 Almacenar estadísticas errores cometidos (no correspondencia de pinzas con objetos).

10.3 Almacenar estadísticas de transiciones correctas.

10.4 Almacenar estadísticas cantidad de objetos caídos.

R 11. El sistema debe dar un reporte del desempeño del estudiante en el ejercicio de Selección.

R 12. El sistema debe mostrar al profesor las estadísticas del entrenamiento en los tres ejercicios de los médicos cirujanos.

R 13. El sistema debe registrar y almacenar el inicio y el fin de un ejercicio.

2.4.2 Requerimientos no funcionales.

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. [16]

Rendimiento: Como toda aplicación de tiempo real, debe tener alta velocidad de procesamiento o cálculo, tiempo de respuesta y de recuperación, y disponibilidad. Las operaciones realizadas por la base de datos serán correctas y rápidas, al interactuar el usuario con el sistema, esta realizará las consultas pertinentes a las tablas involucradas de forma eficiente.

Soporte: El producto debe ser multiplataforma, que funcione tanto en Windows como en Linux, aunque en la primera versión del sistema será en Windows.

Hardware: Escasa disponibilidad de recursos. Hardware para el servidor (mínimo): Pentium PII 300Mhz de la PC. 128Mb RAM, 4.0GB disco duro. Hardware para el servidor (recomendado): Pentium PIII 600MHz, 256Mb RAM, 20 GB disco duro.

Diseño e implementación: La base de datos del sistema será implementada en el gestor de base de datos Firebird. Todo el código será en lenguaje C/C++, regido por la teoría de Programación Orientada a Objetos.

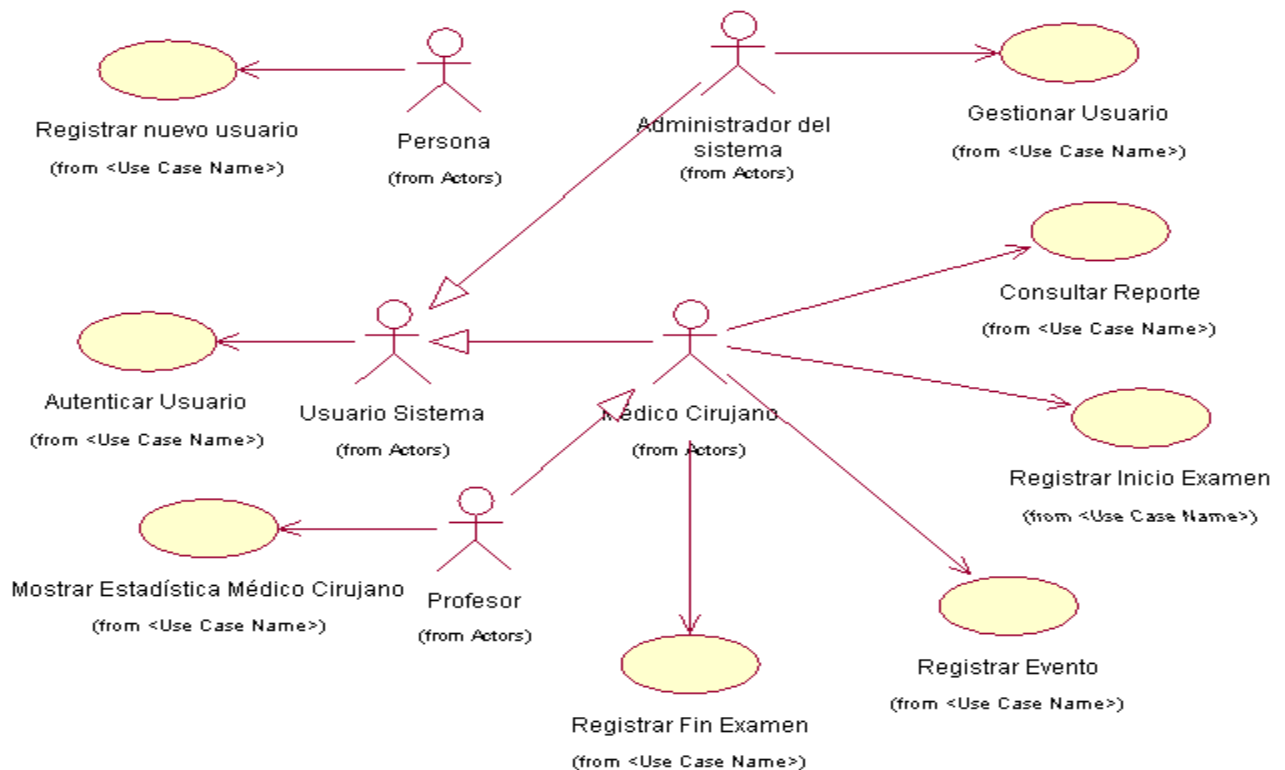
2.5 Modelo de Casos de uso del Sistema.

2.5.1 Definición y justificación de los actores del sistema a automatizar.

Actor	Descripción
Persona	Surge para desarrollar la oportunidad de que cualquier persona pueda registrarse en el sistema convirtiéndose en usuario del mismo.
Usuario del Sistema	Surge por la necesidad de un actor que generaliza a todos los demás actores de manera que todos se

	<p> puedan autentificar al entrar en el sistema.</p>
<p>Administrador del Sistema</p>	<p>Es el encargado de gestionar todos los usuarios, asignando los permisos correspondientes según los roles.</p>
<p>Médico cirujano</p>	<p>Utilizará el sistema para entrenar las habilidades necesarias a través de los ejercicios.</p>
<p>Profesor</p>	<p>Usuario que también podrá entrenarse al igual que el usuario médico cirujano, pero que además podrá conocer las evaluaciones obtenidas por estos médicos cirujanos en cada una de sus sesiones de entrenamiento.</p>

2.5.2 Diagrama de casos de uso del sistema a automatizar.



2.5.3 Descripción de los casos de uso del sistema.

[Ver Anexo 1.](#)

2.6 Análisis y Diseño.

El análisis y diseño de aplicaciones informáticas de gestión debe abordarse, con técnicas y metodologías adecuadas, acompañadas por una precisa gestión de proyectos y una eficaz gestión de la calidad. También es importante poder contar con el soporte de entornos y herramientas adecuadas, que faciliten la tarea del profesional informático y de los usuarios a la hora de desarrollar sistemas de información. [17]

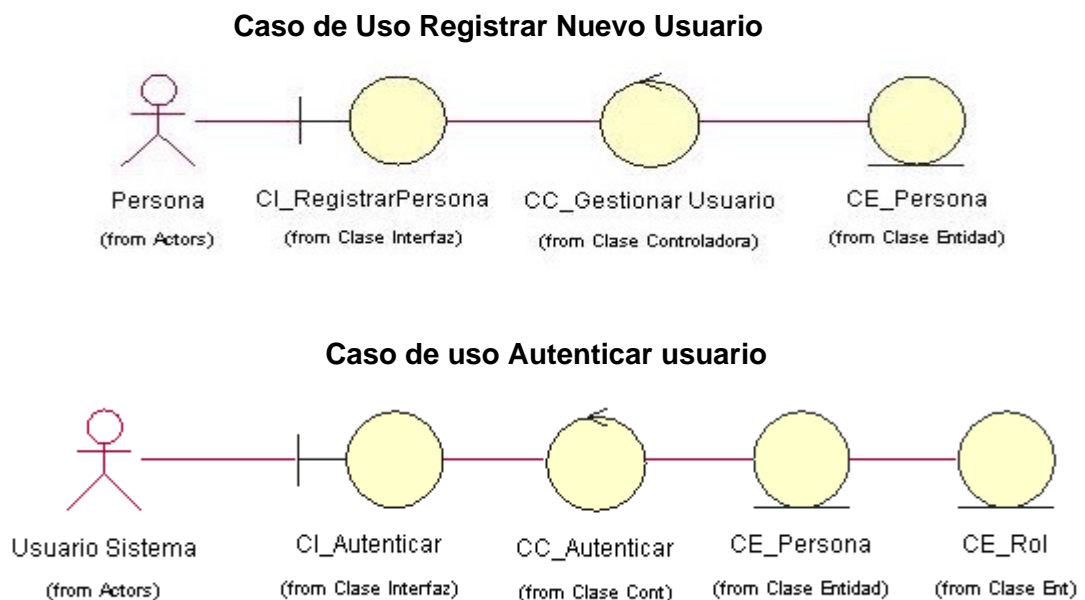
El objetivo de este flujo de trabajo es traducir los requisitos a una especificación que describe cómo implementar el sistema. [18]

2.6.1 Análisis.

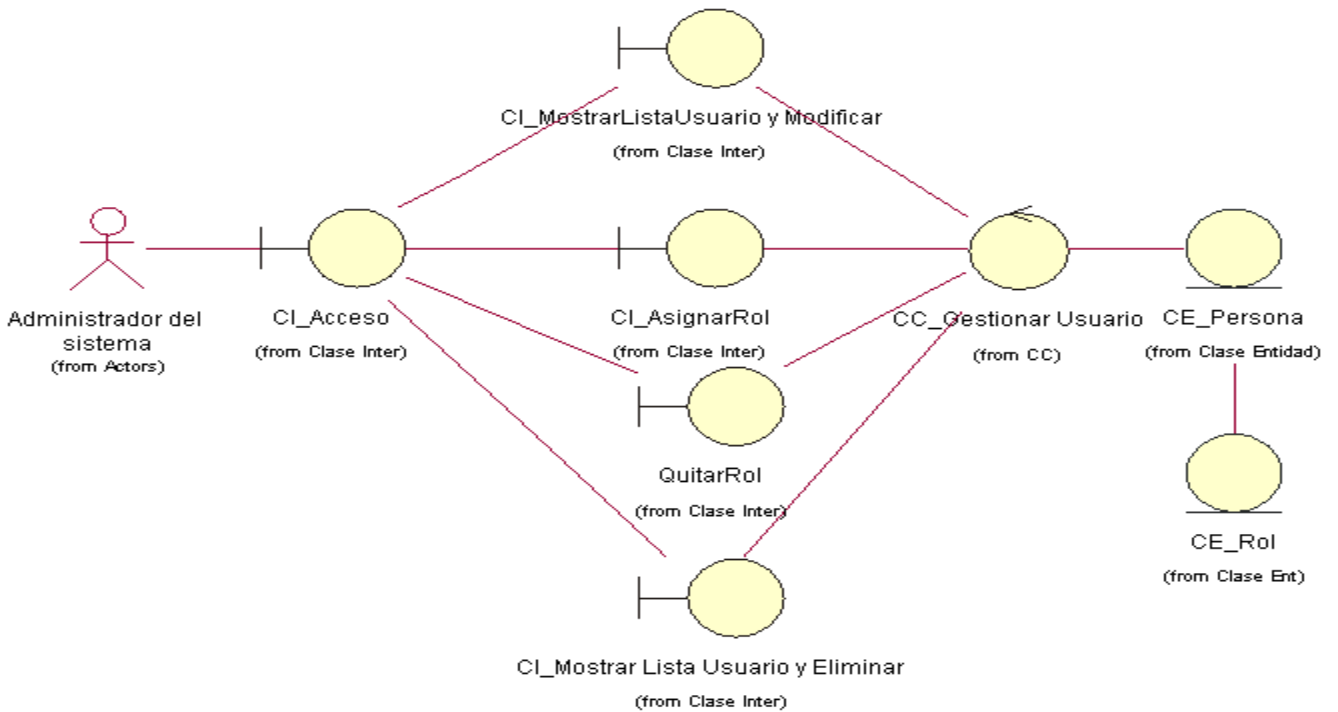
El análisis consiste en obtener una visión del sistema que se preocupa de ver qué hace, de modo que sólo se interesa por los requisitos funcionales. [19]

2.6.1.1 Diagrama de Clase de Análisis

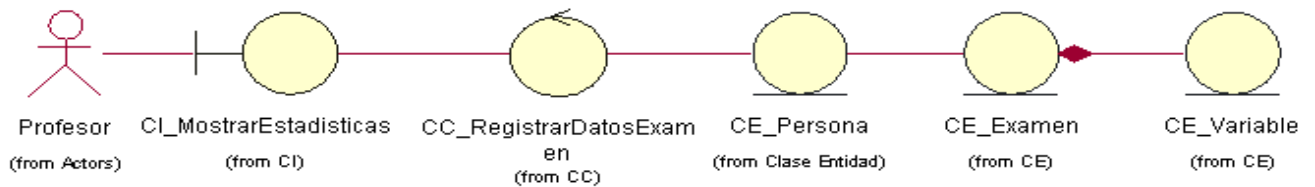
El diagrama de clase del análisis es un artefacto que permite representar conceptos de un dominio del problema, siendo un lenguaje más entendible para el cliente, ya que no incluye aspectos de implementación.



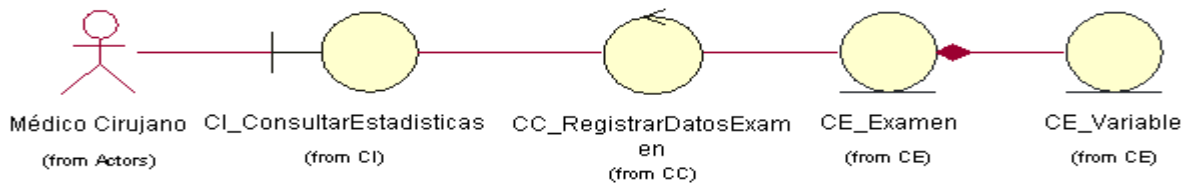
Caso de uso Gestionar usuarios del sistema.



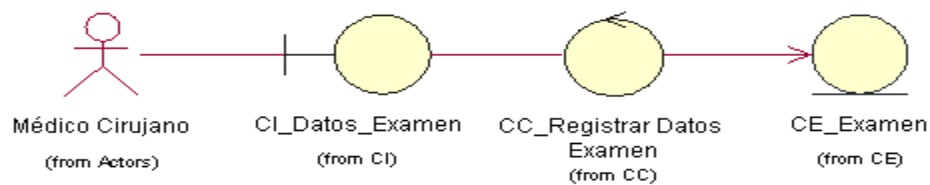
Caso de uso Mostrar Estadísticas de Médicos Cirujanos.



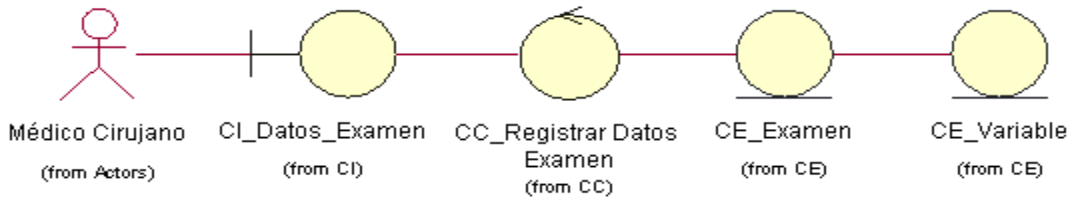
Caso de uso Consultar Reporte.



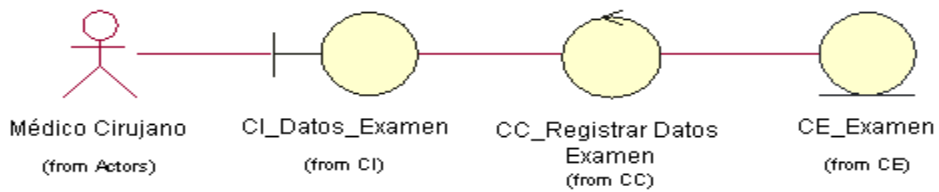
Caso de uso Registrar Inicio Examen.



Caso de uso Registrar Evento.



Caso de uso Registrar Fin Examen.



2.7 Diseño

El diseño es un refinamiento del análisis que tiene en cuenta los requisitos no funcionales, en definitiva cómo cumple el sistema sus objetivos. [20]

2.7.1 Diagrama de Clases del Diseño

Arquitectura de tres Capas

La arquitectura por capas tiene como objetivo primordial la separación de la lógica de negocios de la lógica de diseño, un ejemplo básico de esto es separar la capa de datos de la capa de presentación al usuario.

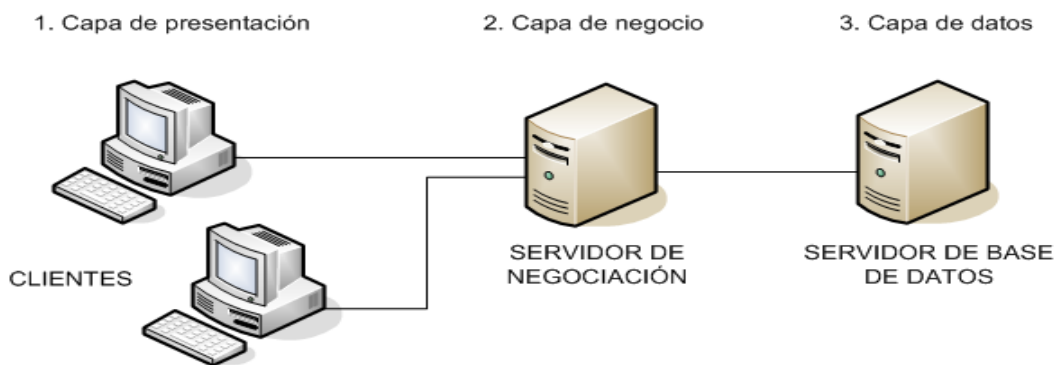


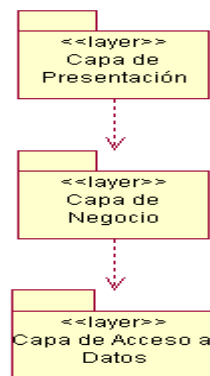
Fig.1 Arquitectura de tres capas

Básicamente la arquitectura se centra en una arquitectura de 3 partes, las cuales pueden distribuirse en una, dos y tres capas.

1.- Capa de presentación: es la que ve el usuario, le comunica y captura la información, realiza un filtrado previo para comprobar que no hay errores de formato. Esta capa se comunica únicamente con la capa de negocio. También es conocida como interfaz gráfica y debe tener la característica de ser amigable (entendible y fácil de usar) para el usuario. [21]

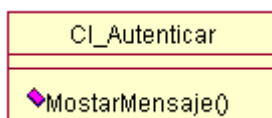
2.- Capa de negocio: es donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio (e incluso de lógica del negocio) pues es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor, almacenar o recuperar información. [22]

3.- Capa de datos: es donde residen los datos y es la encargada de acceder a ellos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio. [23]

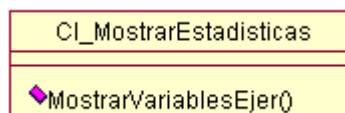


2.7.2 Capa de Presentación.

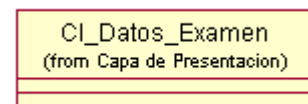
Paquete Autenticar



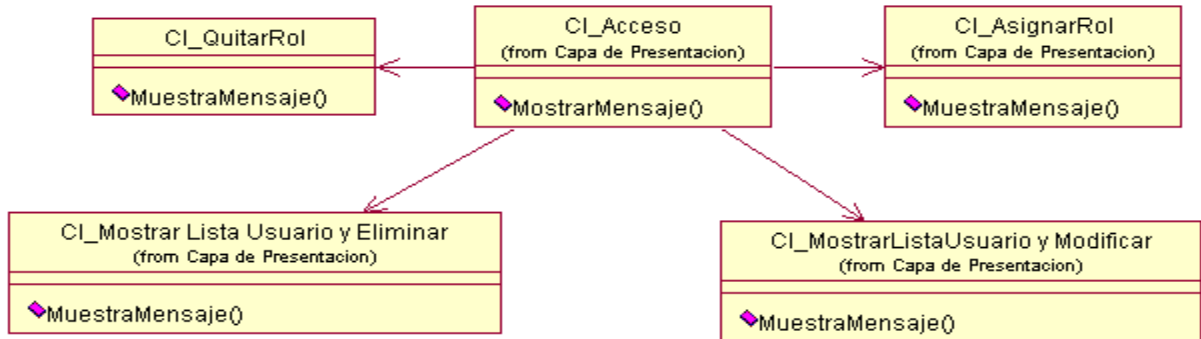
Paquete Profesor



Paquete Examen



Paquete Administrador.

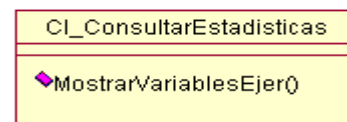


Paquete Usuario

Vista RegistrarPersona

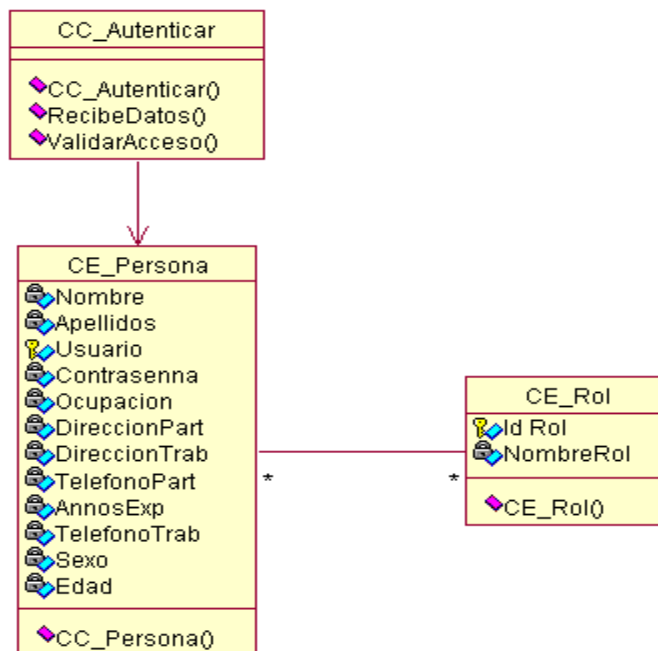


Vista ConsultarEstadisticas

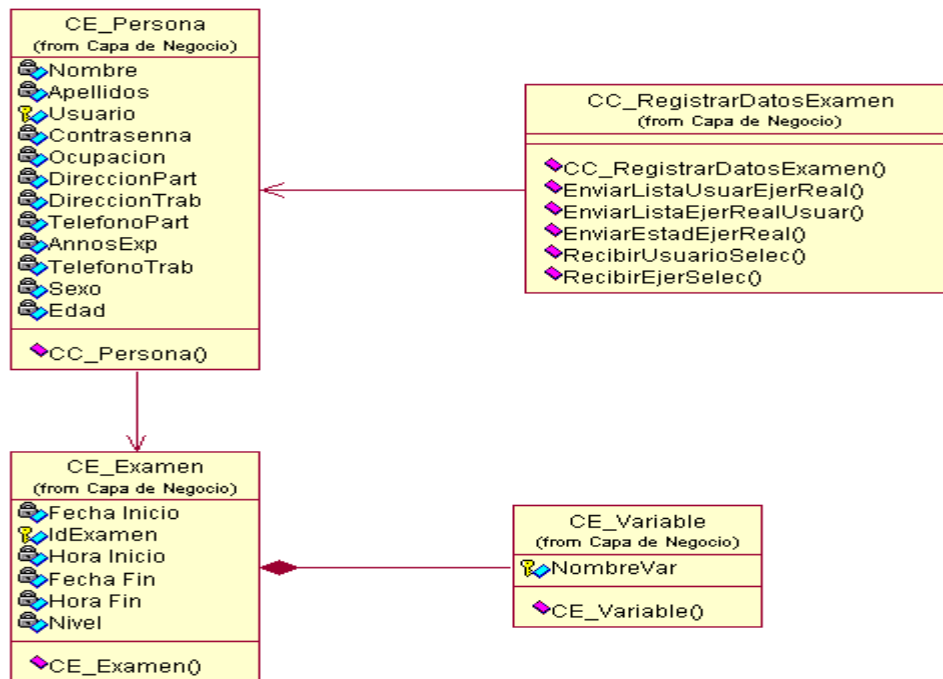


2.7.3 Capa de Negocio

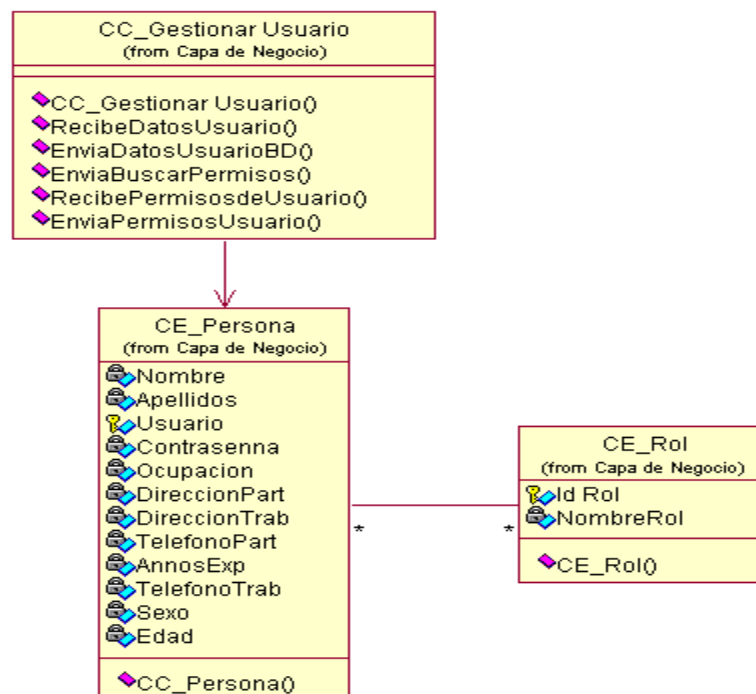
Paquete Autenticar



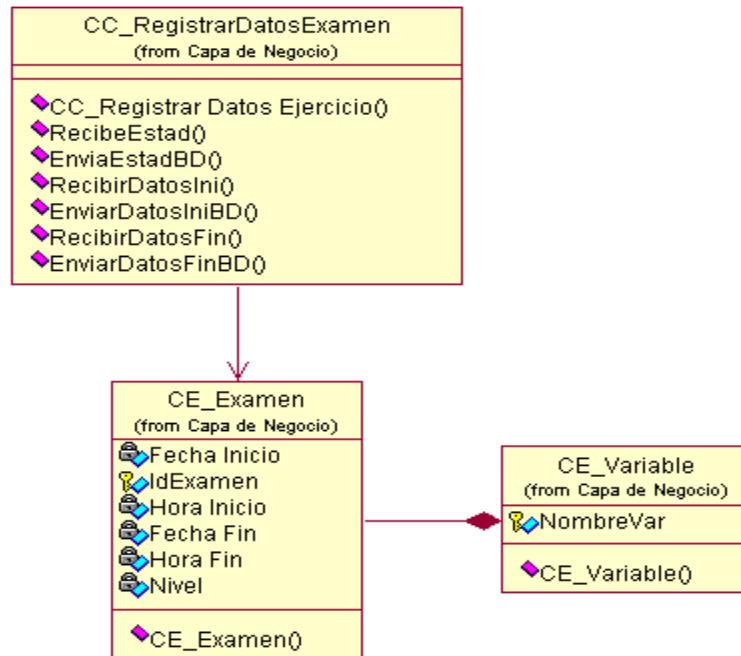
Paquete Profesor



Paquete Administrador

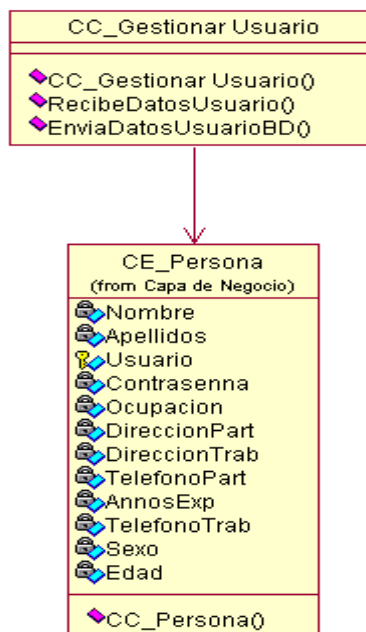


Paquete Examen

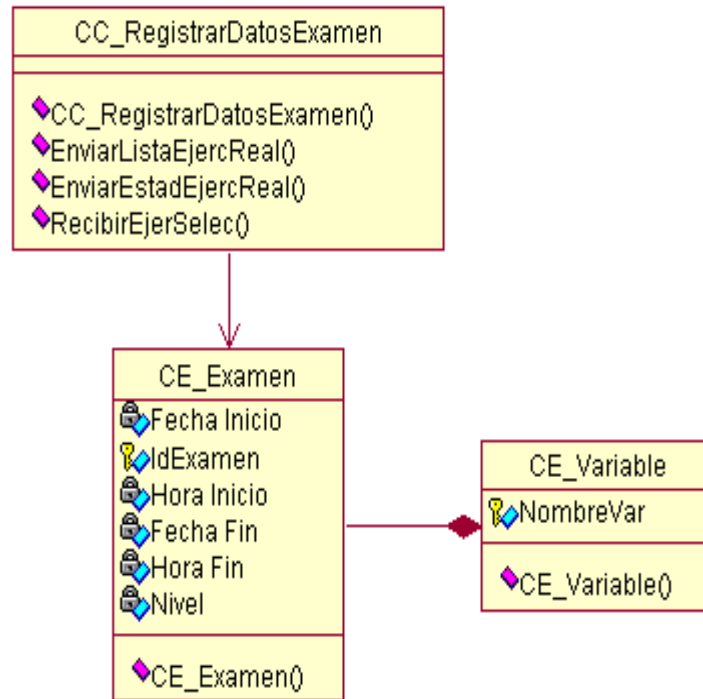


Paquete Usuario

Vista RegistrarPersona



Vista ConsultarEstadisticas



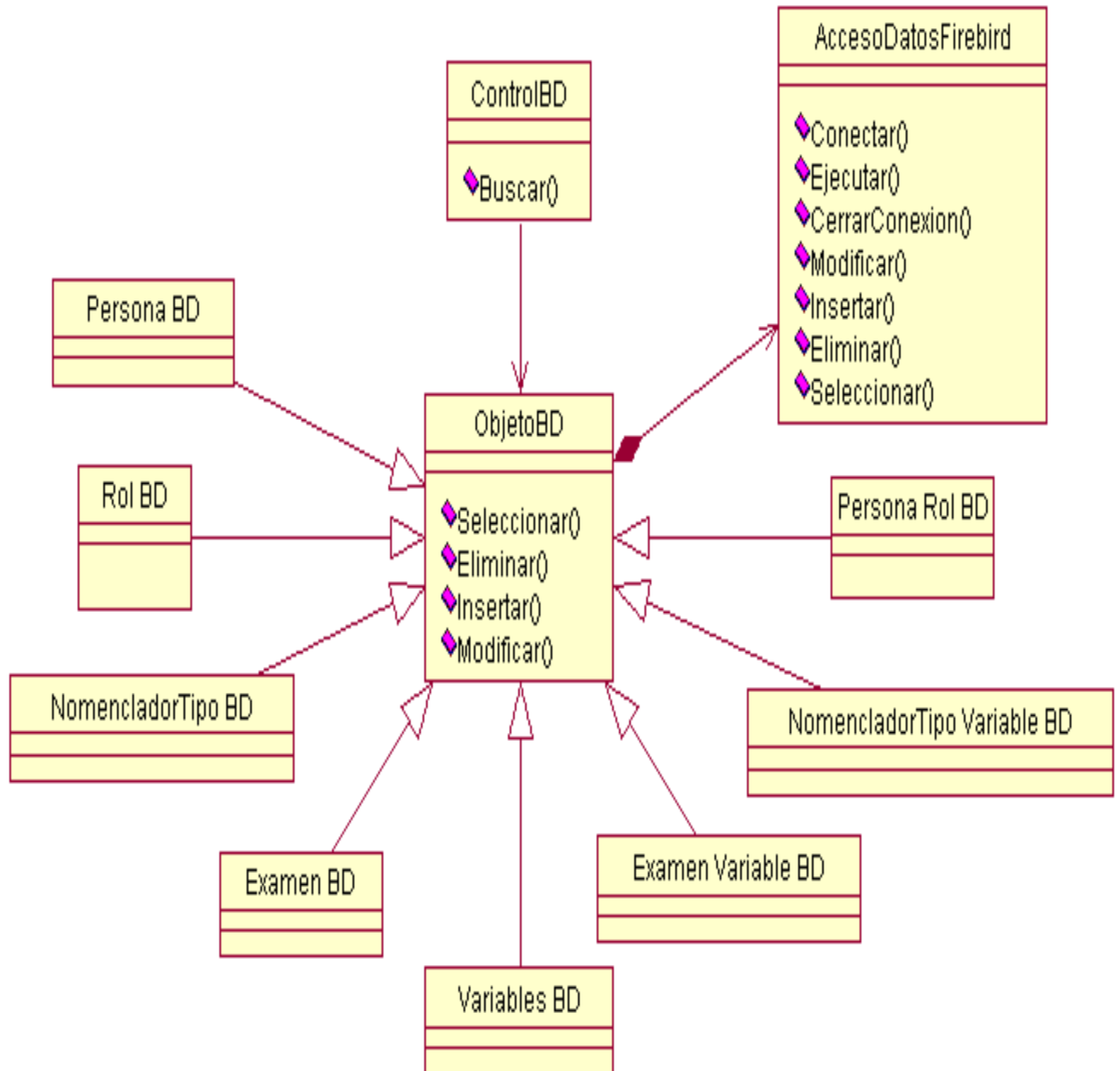
Descripción de las Clases de Diseño. [Ver Anexo2](#)

2.7.4 Capa de Acceso a Datos

Se utilizó para el desarrollo de esta capa, el patrón de diseño Row Data Gateway ya que este permite separar el acceso al SQL de la capa de dominio y situarlos en clases diferentes. Representa fielmente la estructura de una fila de la tabla en la base de datos, donde cada campo se convierte en una propiedad del objeto. Es ideal para auto-generar código. [24]

Uno de los beneficios de utilizar este acercamiento es que el resto de las clases no necesita tratar con SQL por lo que todo el código SQL es más fácil de localizar y mantener, encajando muy bien con una manera de pensar Orientado a Objeto. [25]

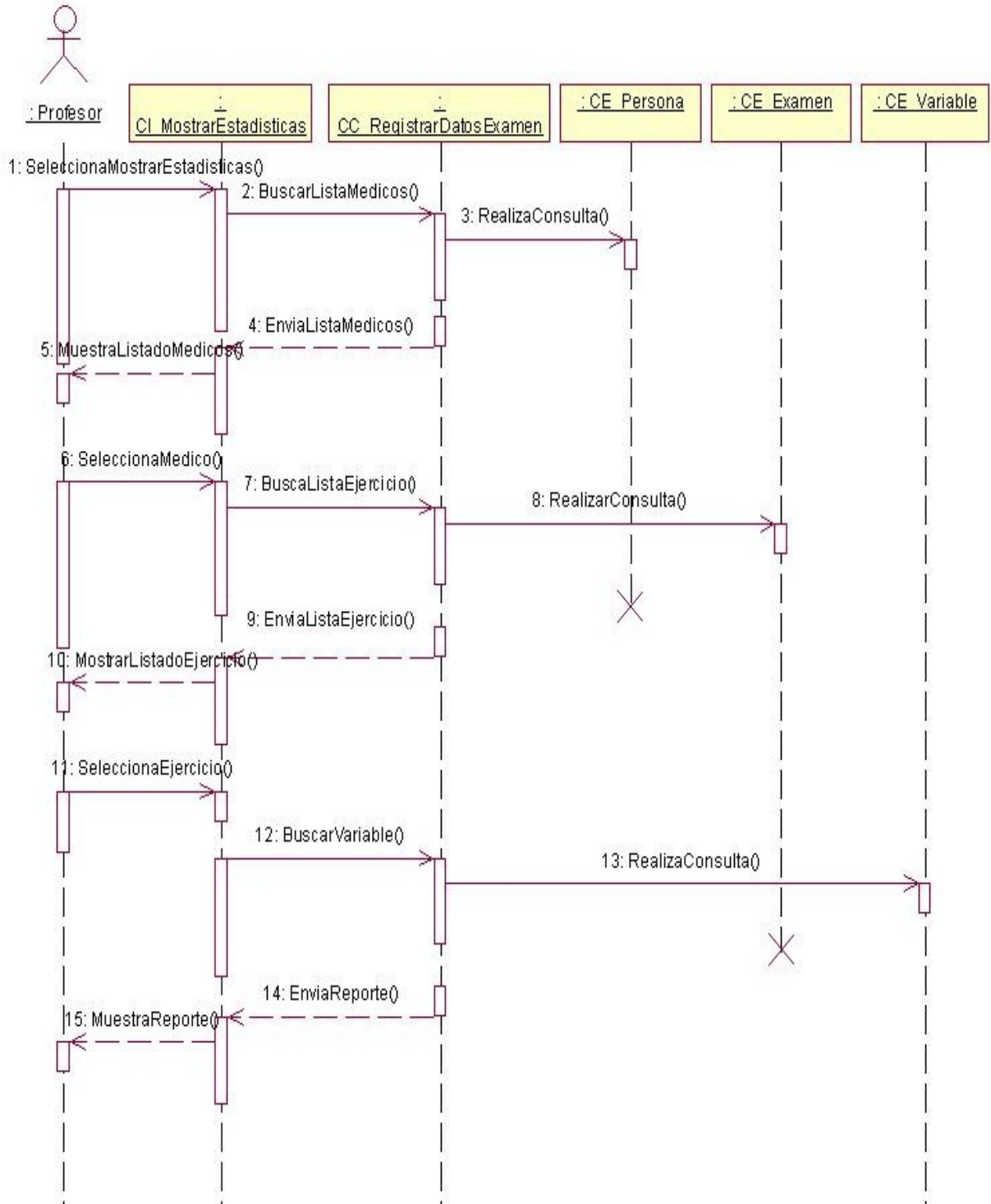
Tiene propiedades que reflejan las columnas de la tabla, y métodos de actualización en la base de datos.



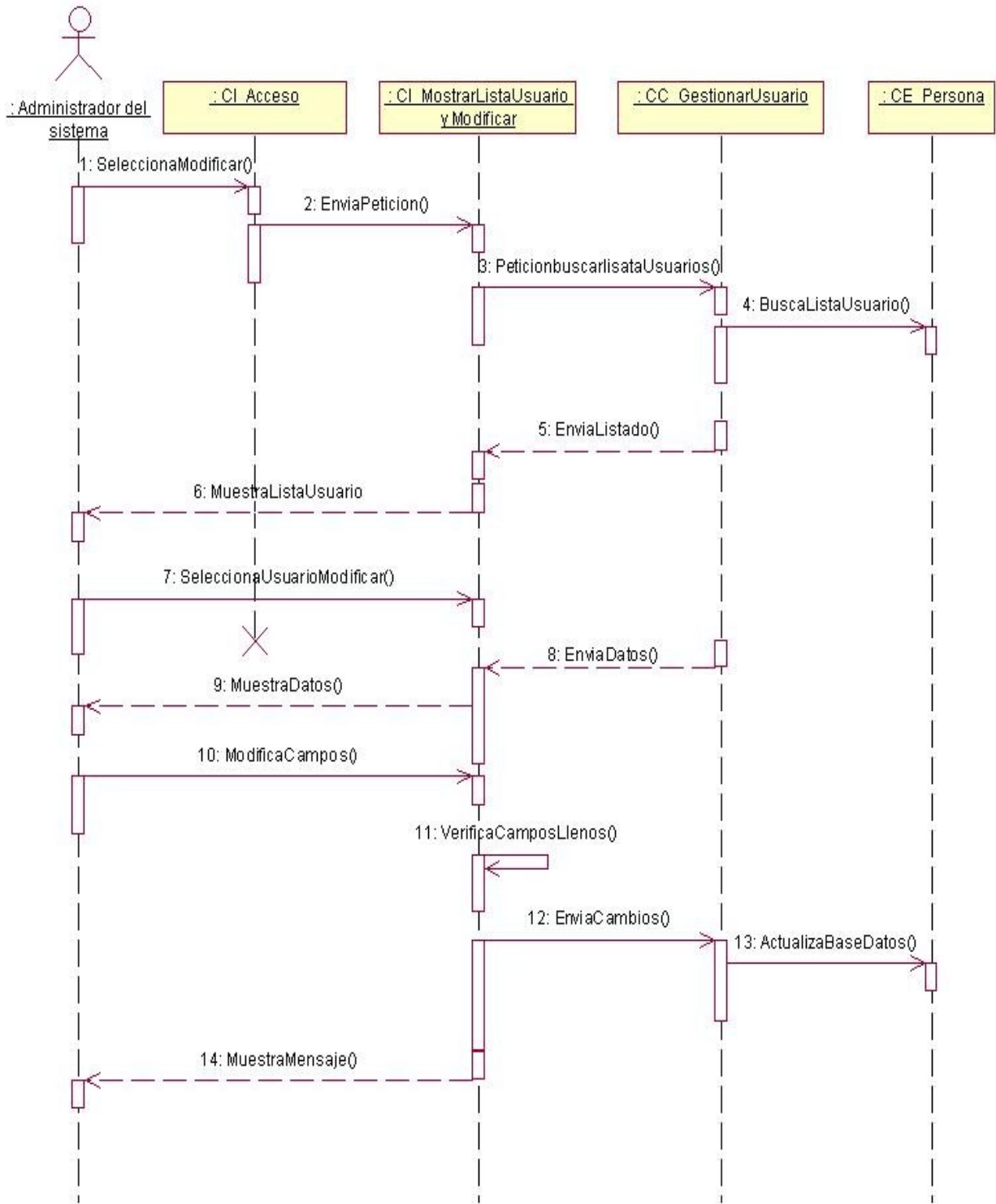
2.7.5 Diagrama de Secuencia

En el diagrama de secuencia se muestra la interacción de los objetos que componen un sistema de forma temporal. [26]

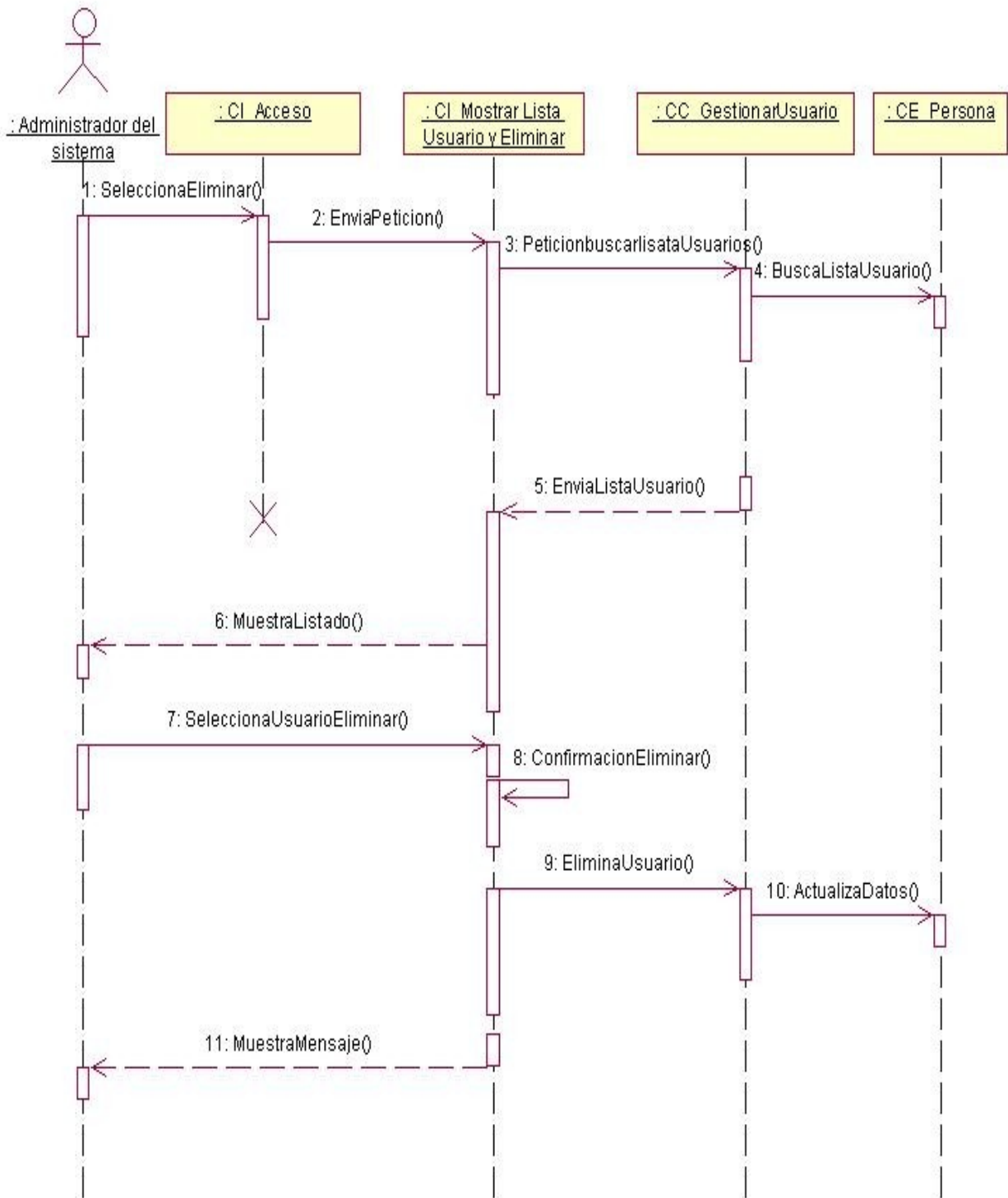
Mostrar Estadísticas Médico Cirujano



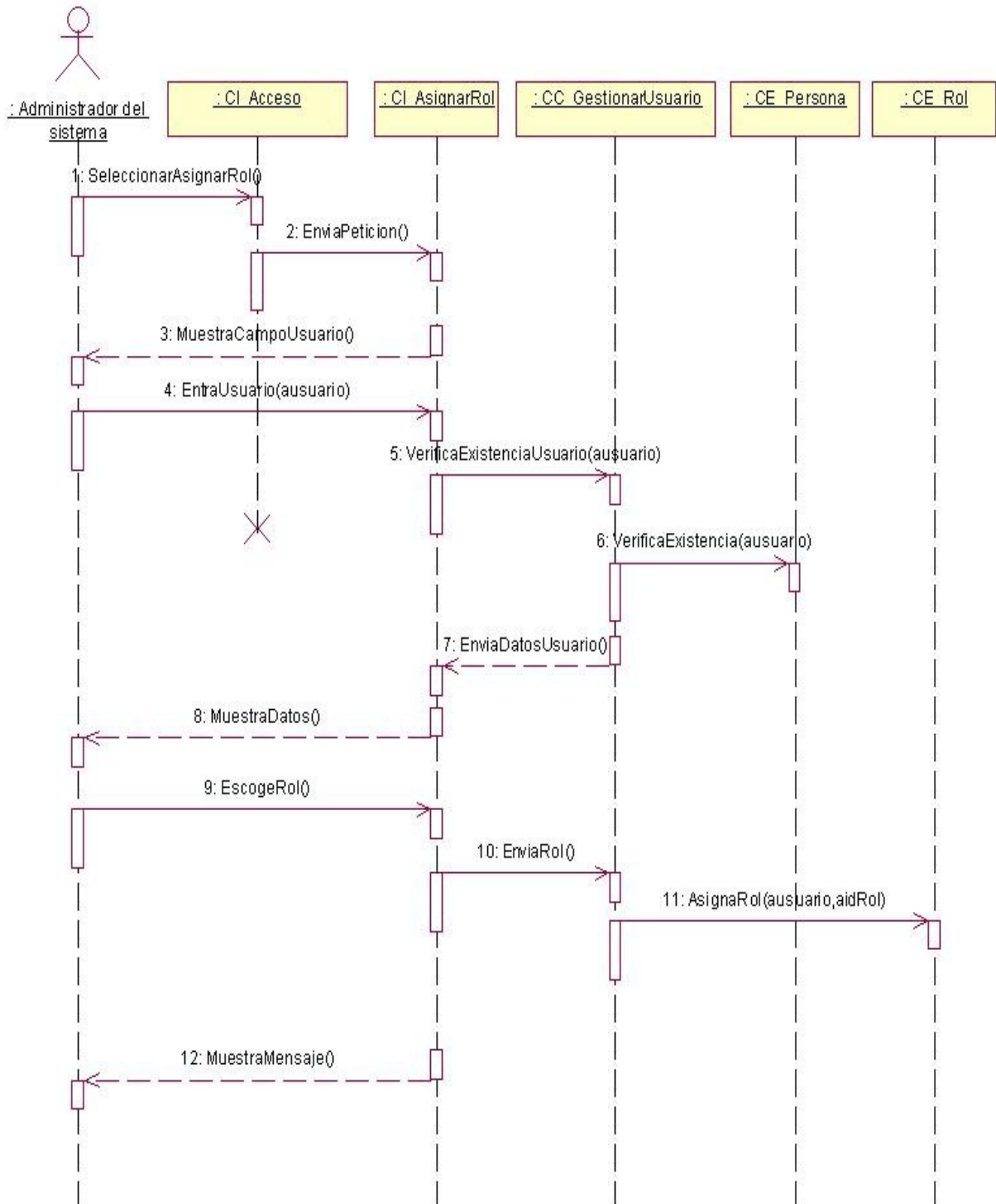
Gestionar Usuario (Modificar)



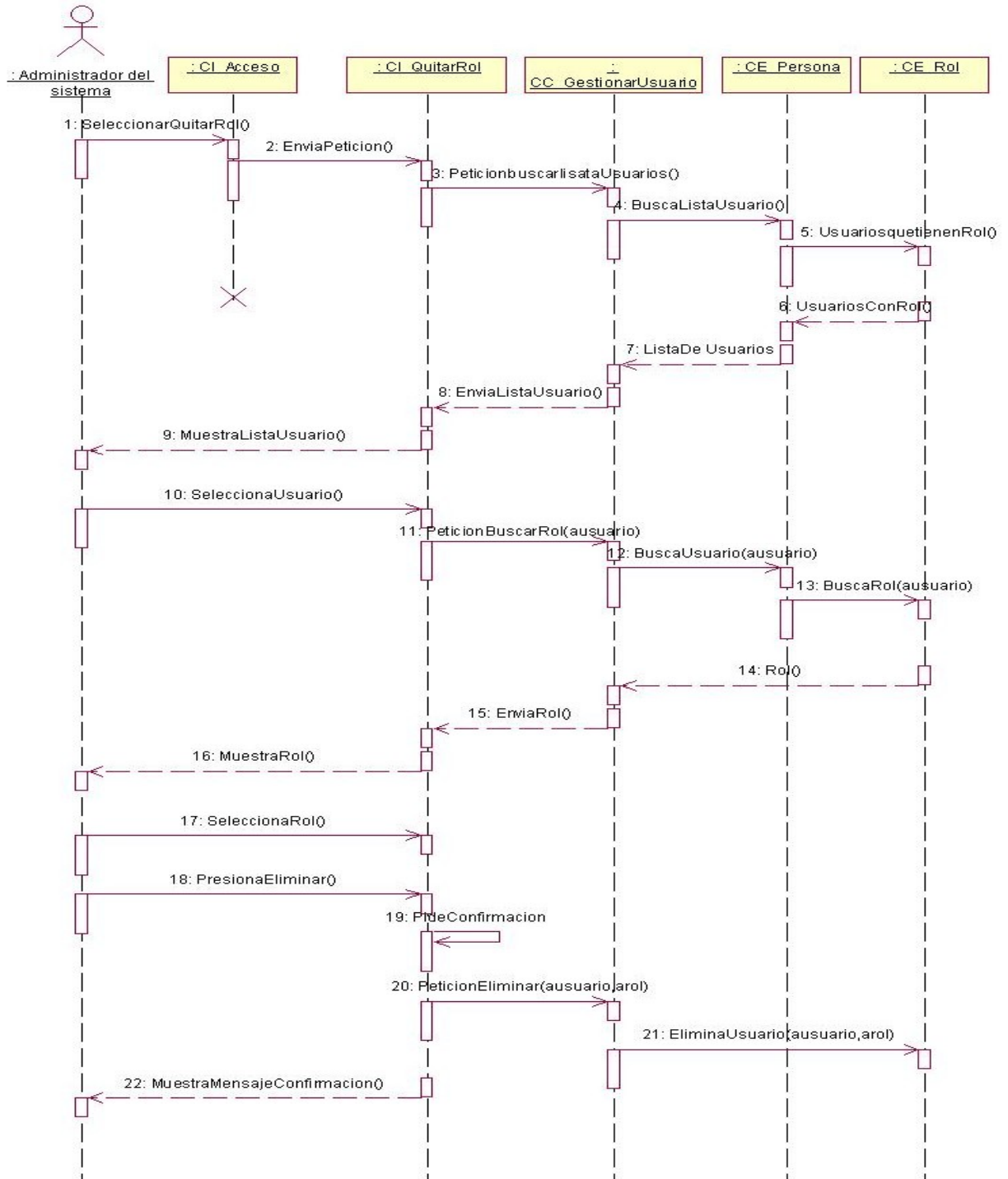
Gestionar Usuario (Eliminar)



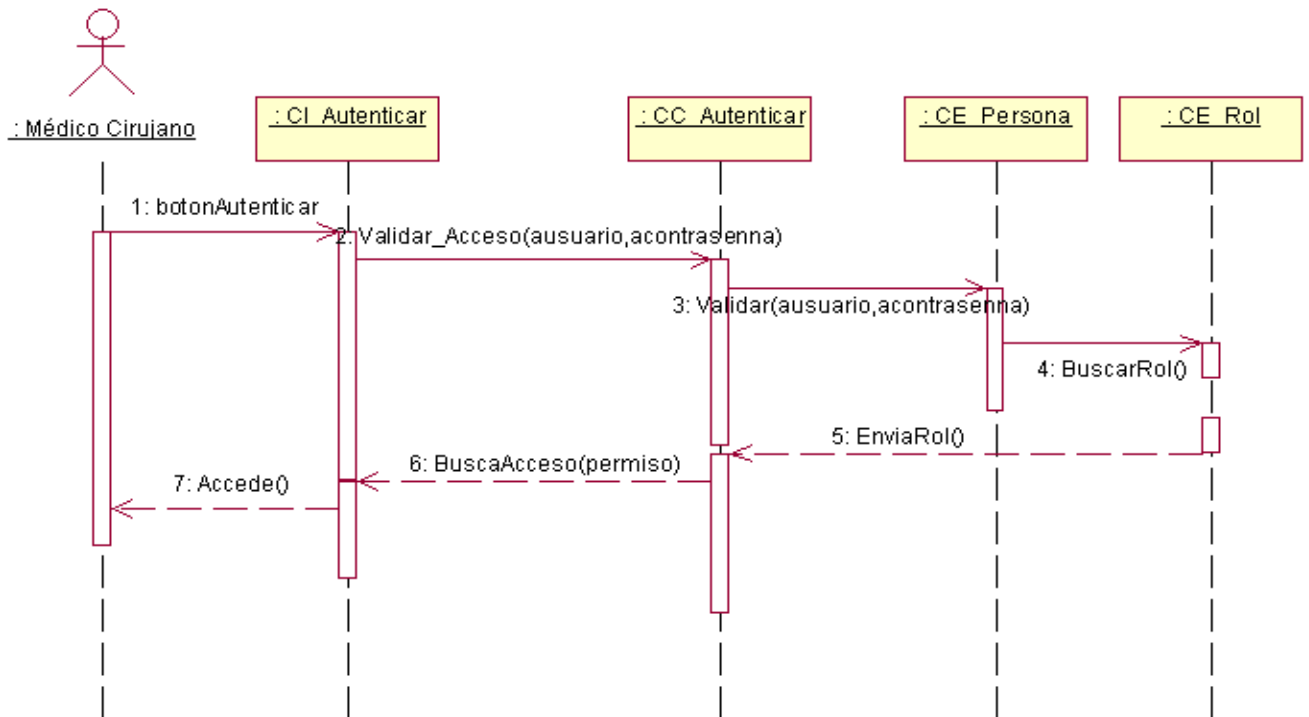
Gestionar Usuario (Asignar Rol)



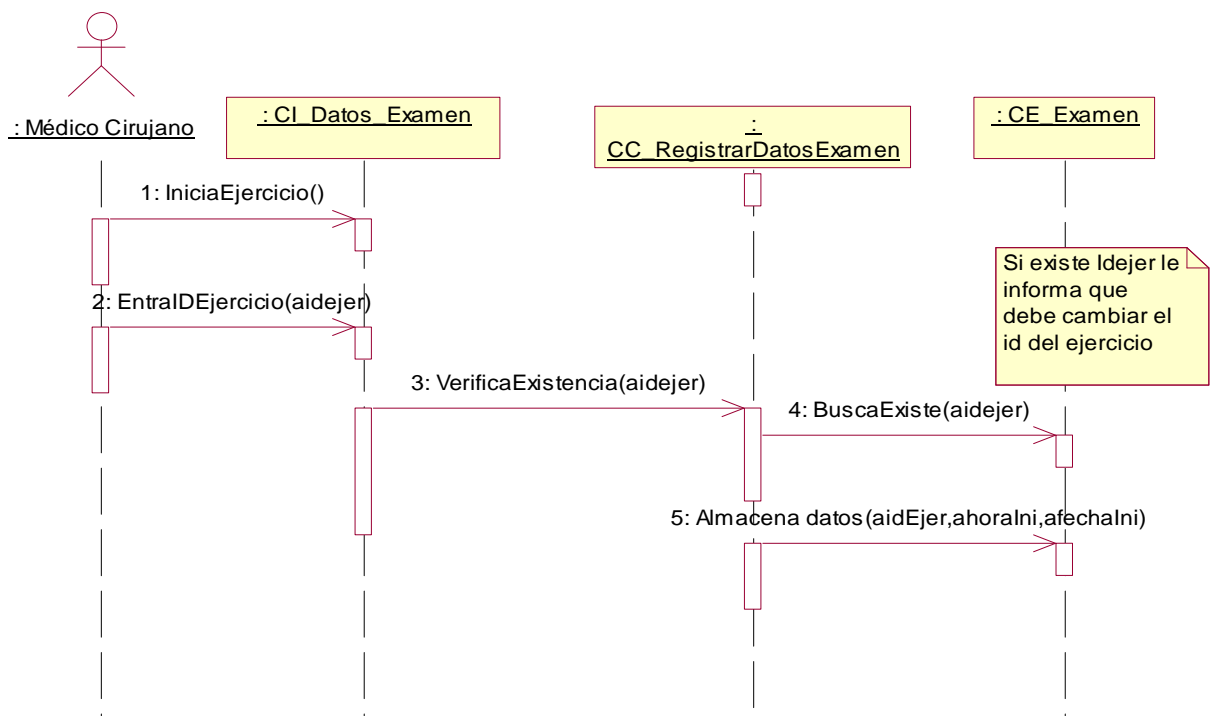
Gestionar Usuario (Quitar Rol)



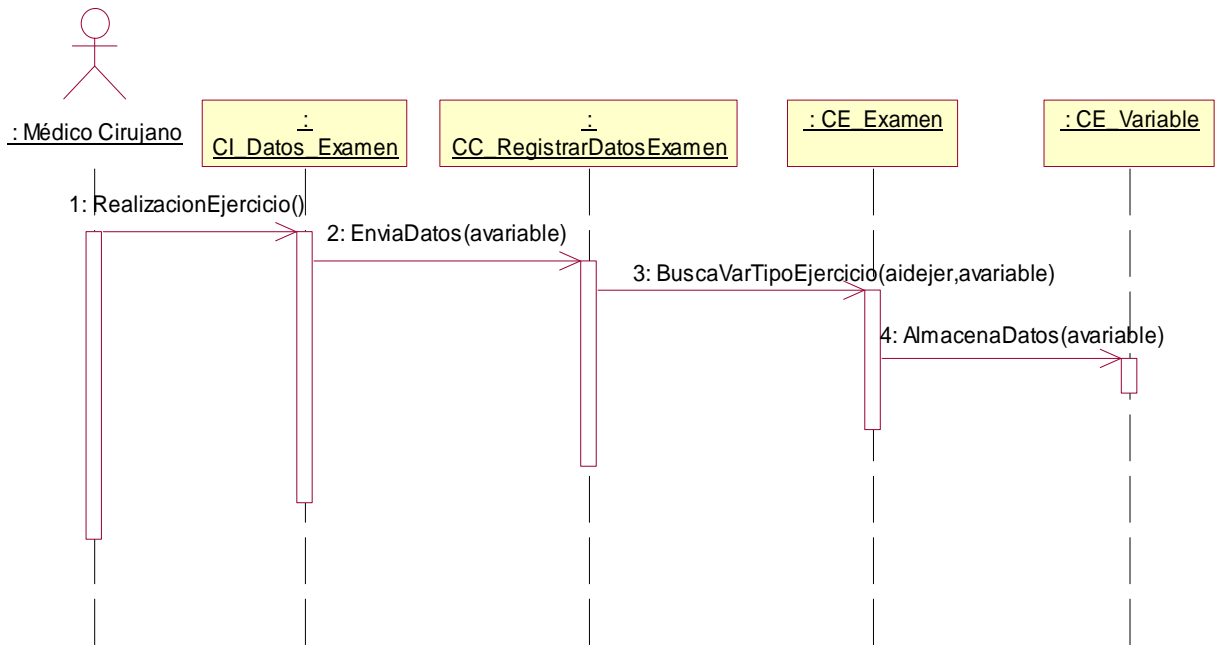
Autenticar



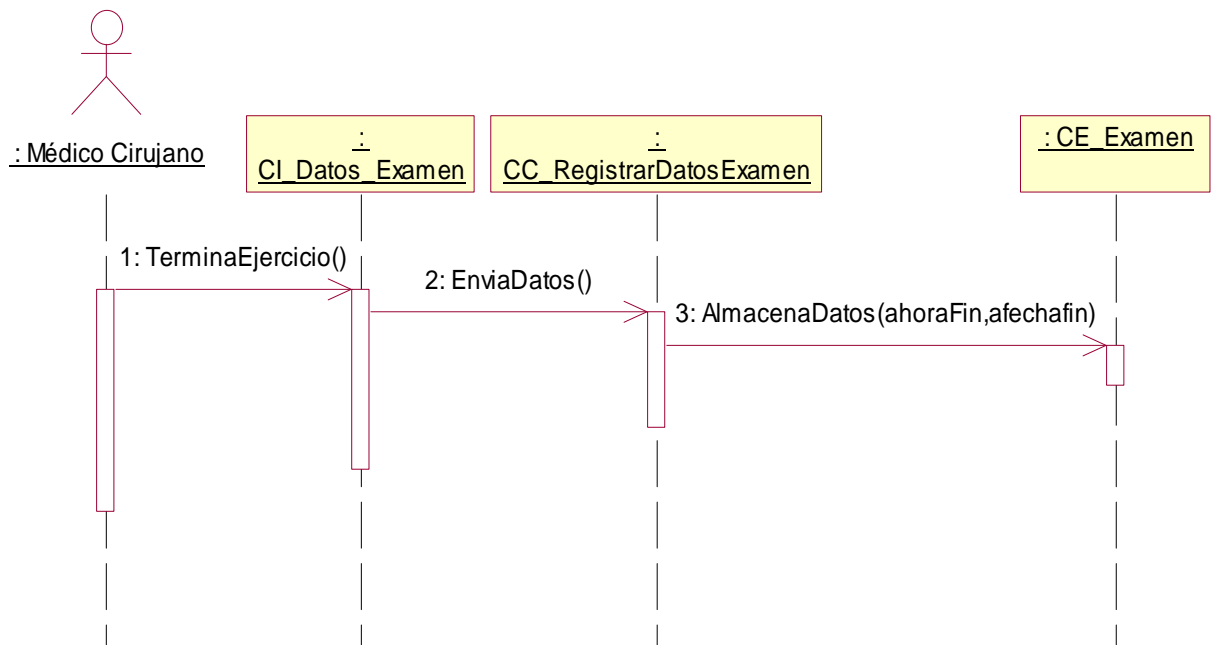
Registrar Inicio Ejercicio



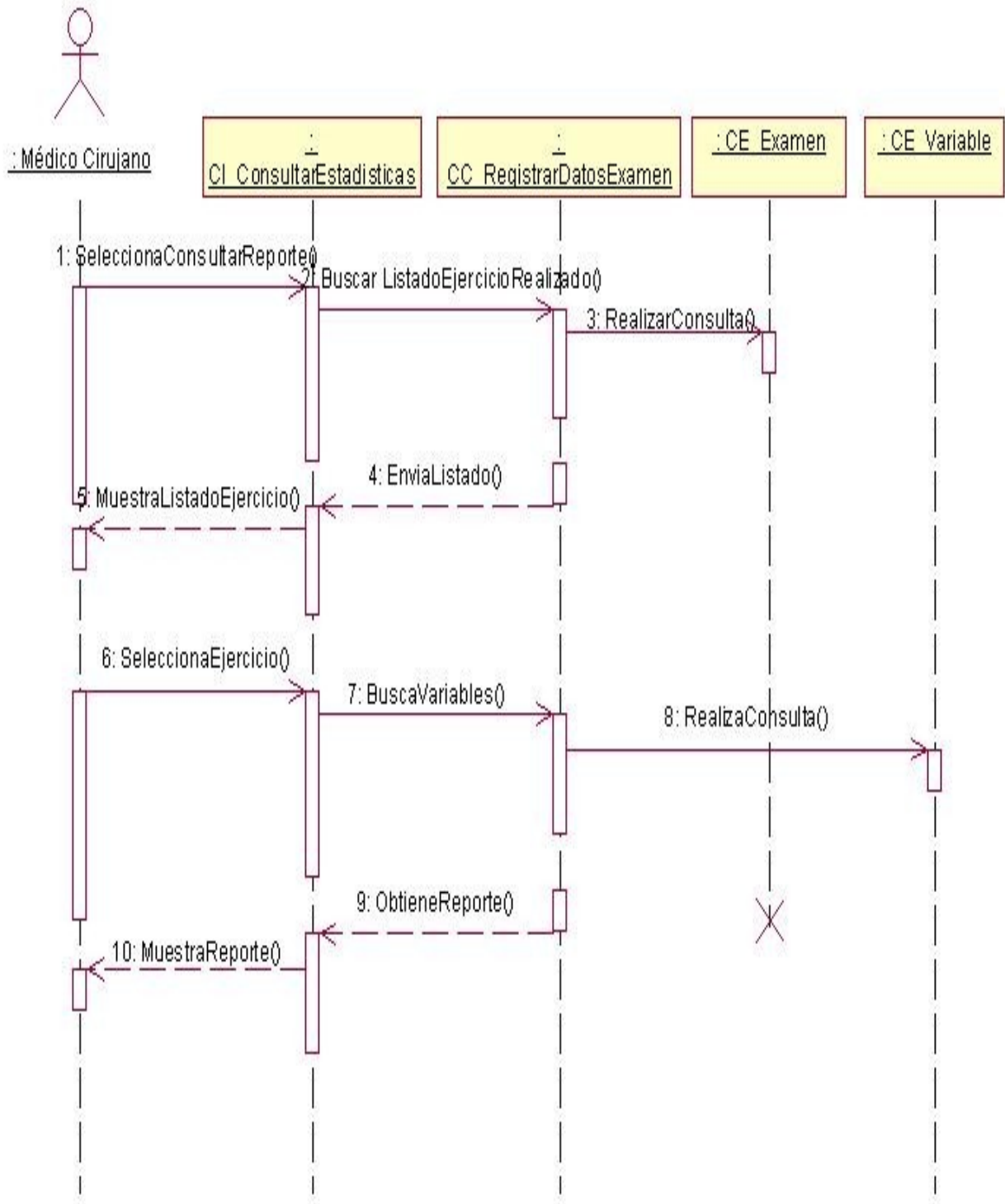
Registrar Evento



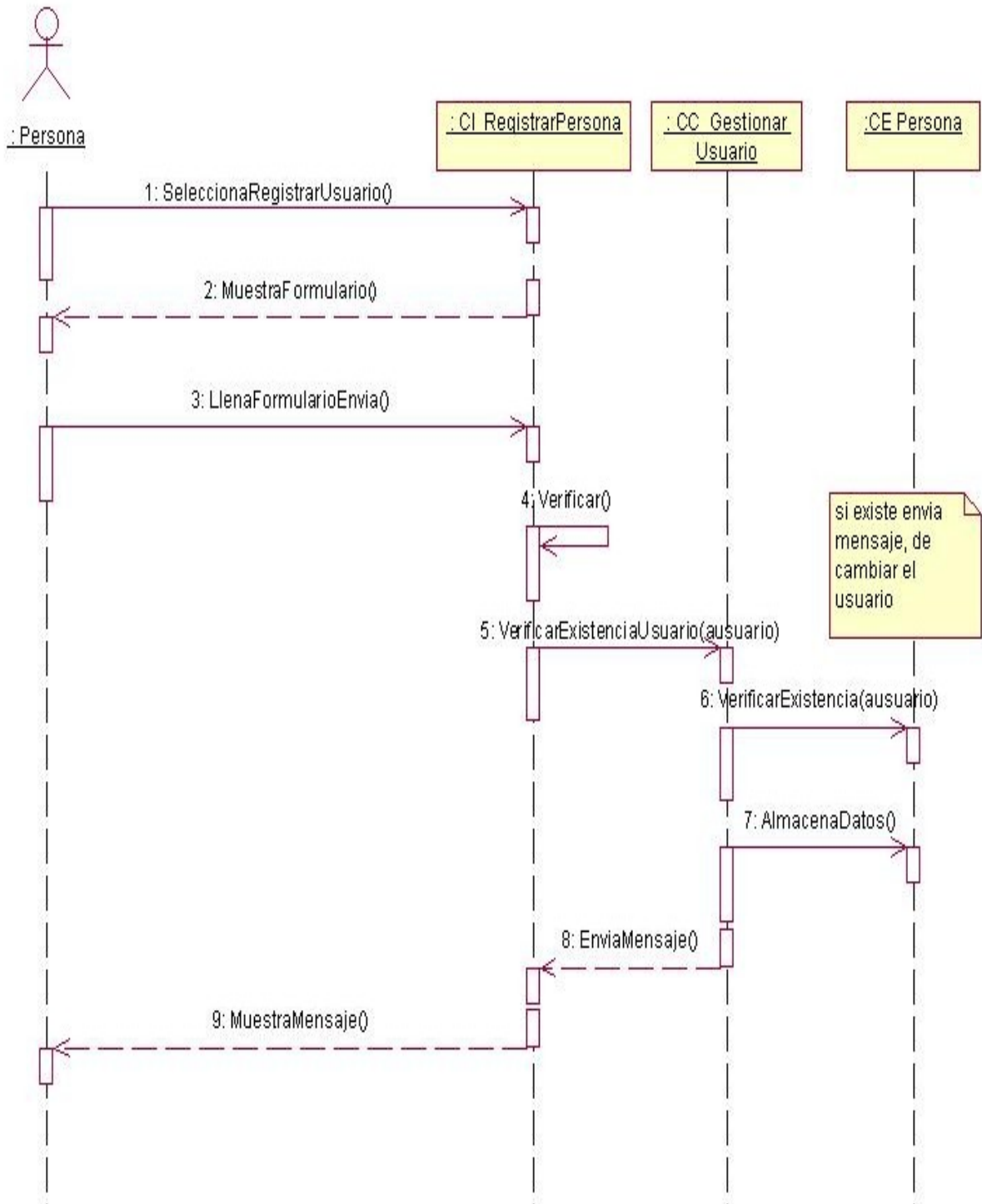
Registrar Fin Ejercicio



Consultar Reporte

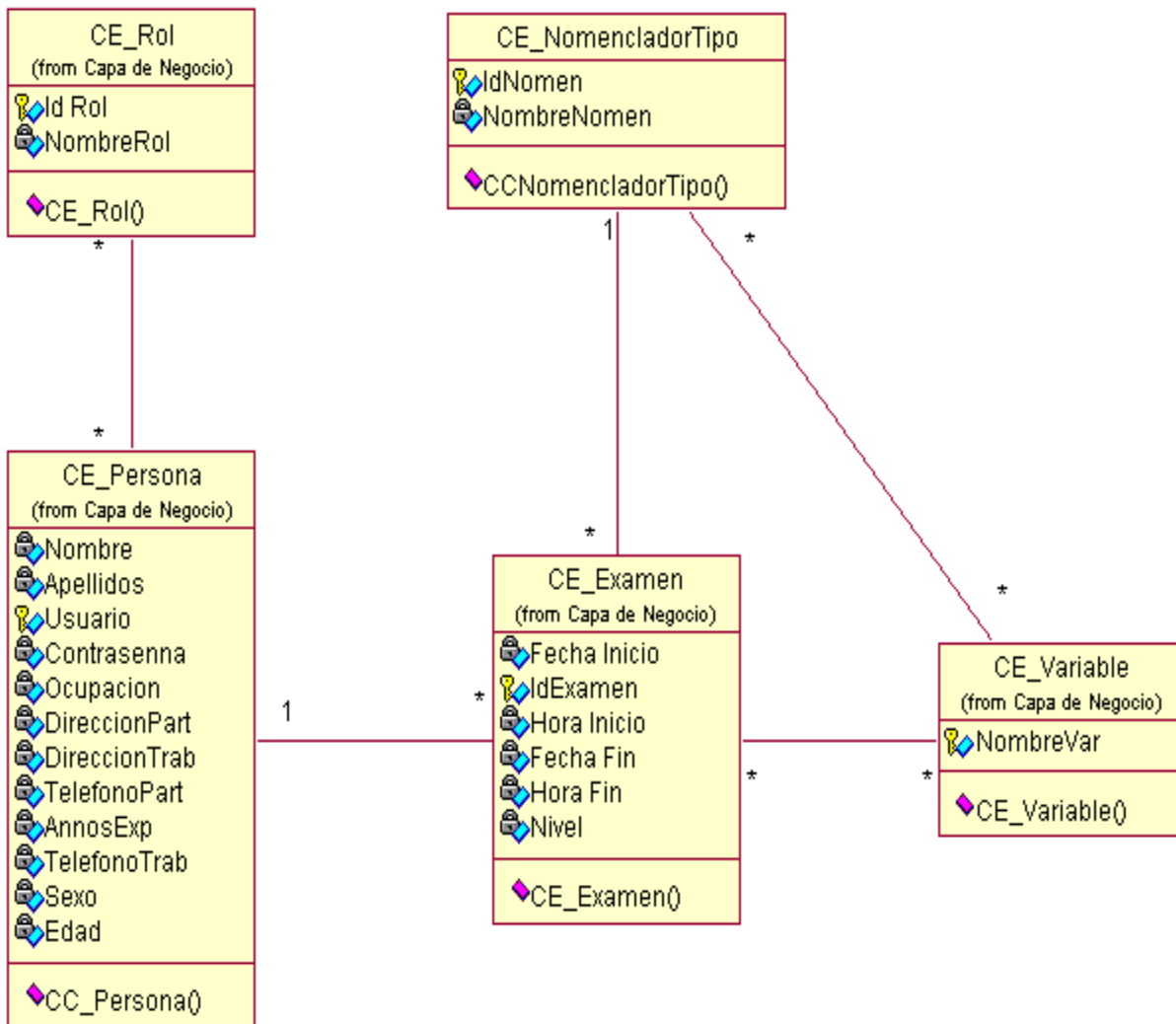


Registrar Persona



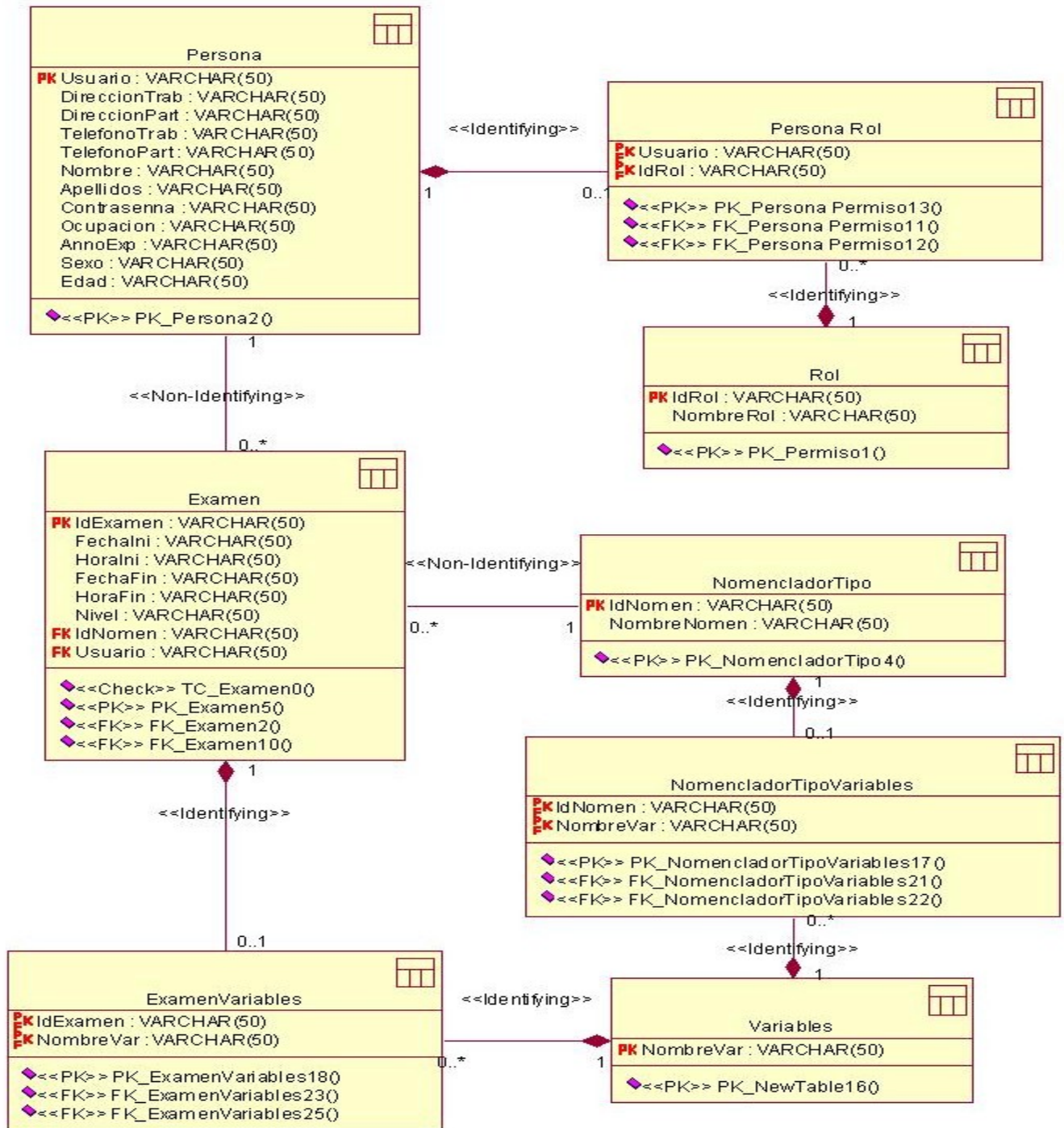
2.7.6 Diagrama de clases Persistentes

La persistencia es la capacidad de un objeto de mantener su valor en el espacio y en el tiempo. Las clases persistentes tienen como origen las clases clasificadas como entidad porque ellas modelan la información y el comportamiento asociado de algún fenómeno o concepto.



2.7.7 Modelo de Datos

Un modelo de datos es aquel que describe de una forma abstracta cómo se representan los datos, conjunto de conceptos, reglas y convenciones que permite almacenar métodos y recuperar información de esos contenedores. [27]



2.7.8 Descripción de las tablas del modelo de datos.

[Ver Anexo 3](#)

Capítulo 3 Implementación y Prueba del Sistema.

Introducción

En este capítulo se describen y especifican las funcionalidades de los nodos de cómputos con que cuenta el sistema. El flujo de trabajo de implementación describe cómo los elementos del modelo del diseño se implementan en términos de componentes y cómo estos se organizan de acuerdo a los nodos específicos en el modelo de despliegue. En los diagramas del modelo de implementación se representan las relaciones entre los distintos componentes. Además en este capítulo se explican y describen los objetivos y alcance del flujo de trabajo de Prueba y se especifica el tipo de prueba a utilizar.

3.1 Implementación

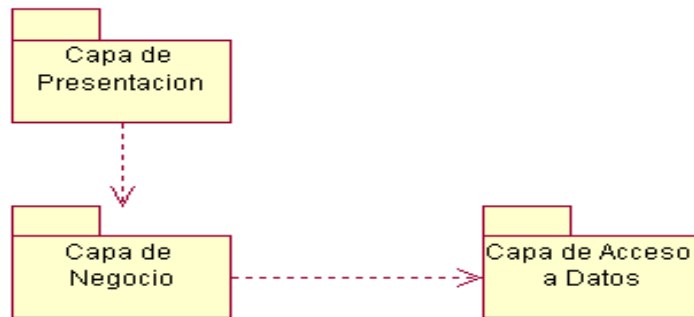
No es más que uno de los flujos de trabajo que tiene definido RUP en su metodología, para definir como se organizan las clases y objetos en componentes. Además de la ubicación de estos componentes en los nodos correspondientes.

3.2 Diagramas de Componentes

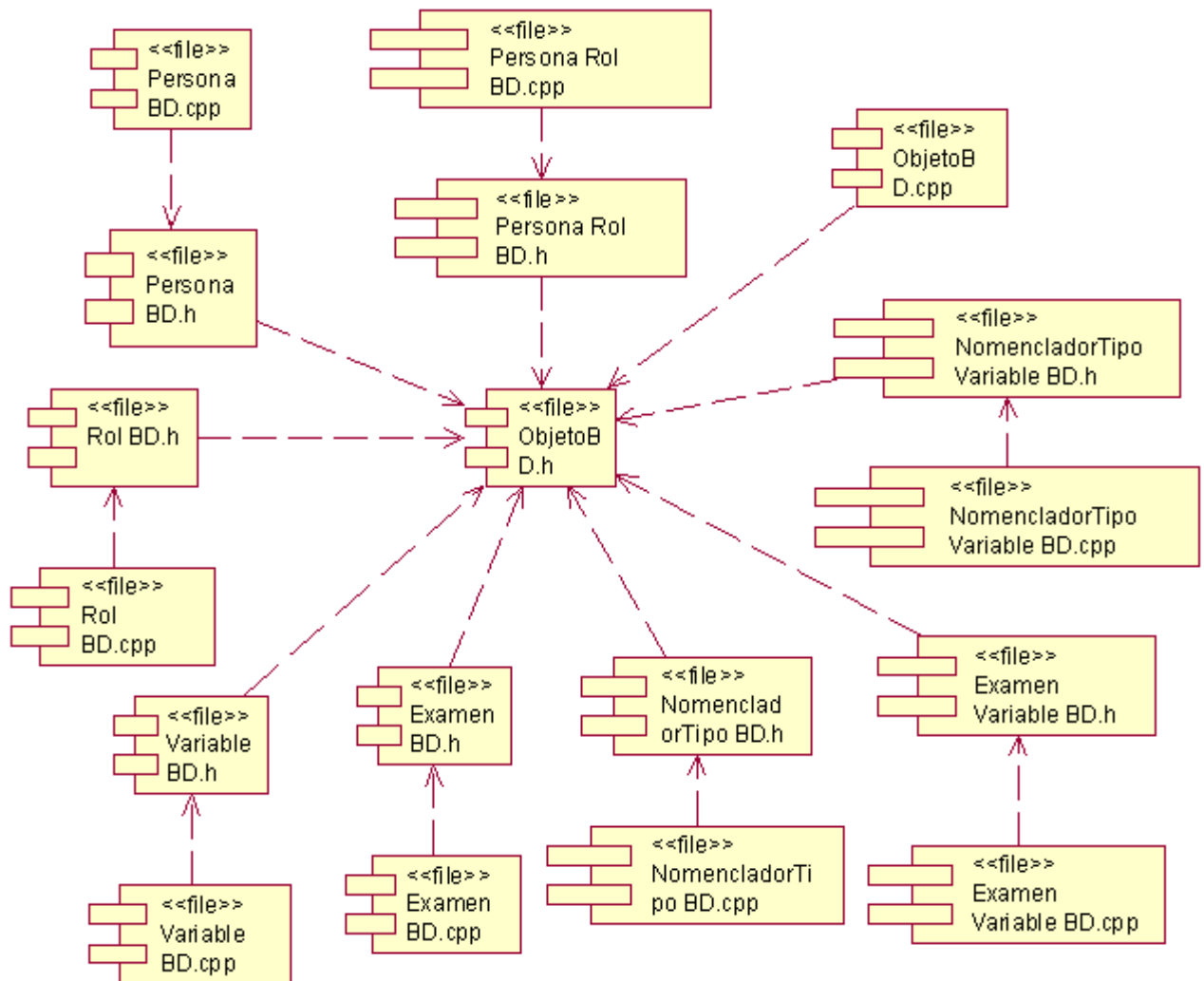
Los componentes se utilizan para modelar los elementos físicos que pueden hallarse en un nodo, tales como ejecutables, bibliotecas, tablas, archivos y documentos, un componente es una parte física y reemplazable de un sistema. [\[28\]](#)

Un diagrama de componentes muestra la organización y dependencias entre un conjunto de componentes. Son fundamentalmente diagramas de clases que se centran en los componentes de un sistema. Los diagramas de componentes no solo son importantes para visualizar, especificar y documentar sistemas basados en componentes sino también para construir sistemas ejecutables mediante ingeniería inversa o directa. [\[29\]](#)

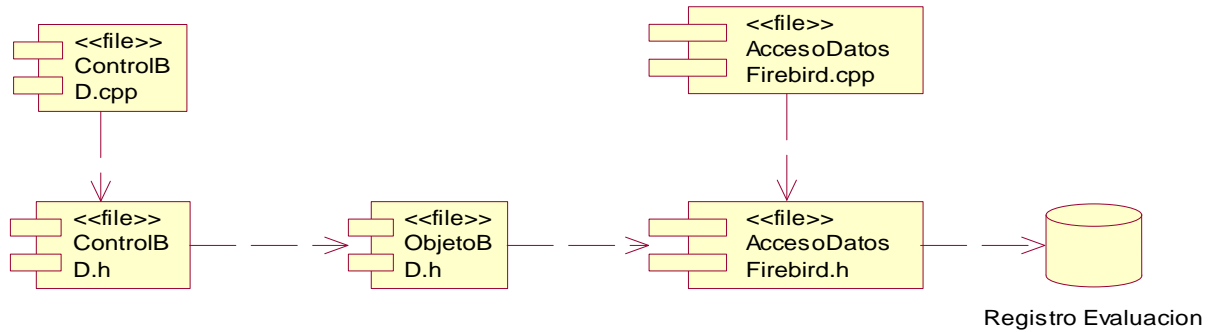
3.2.1 Diagrama de Paquetes del modelo de Implementación.



3.2.2 Diagrama de componentes Capa Acceso a datos (Vista ObjetoBD).

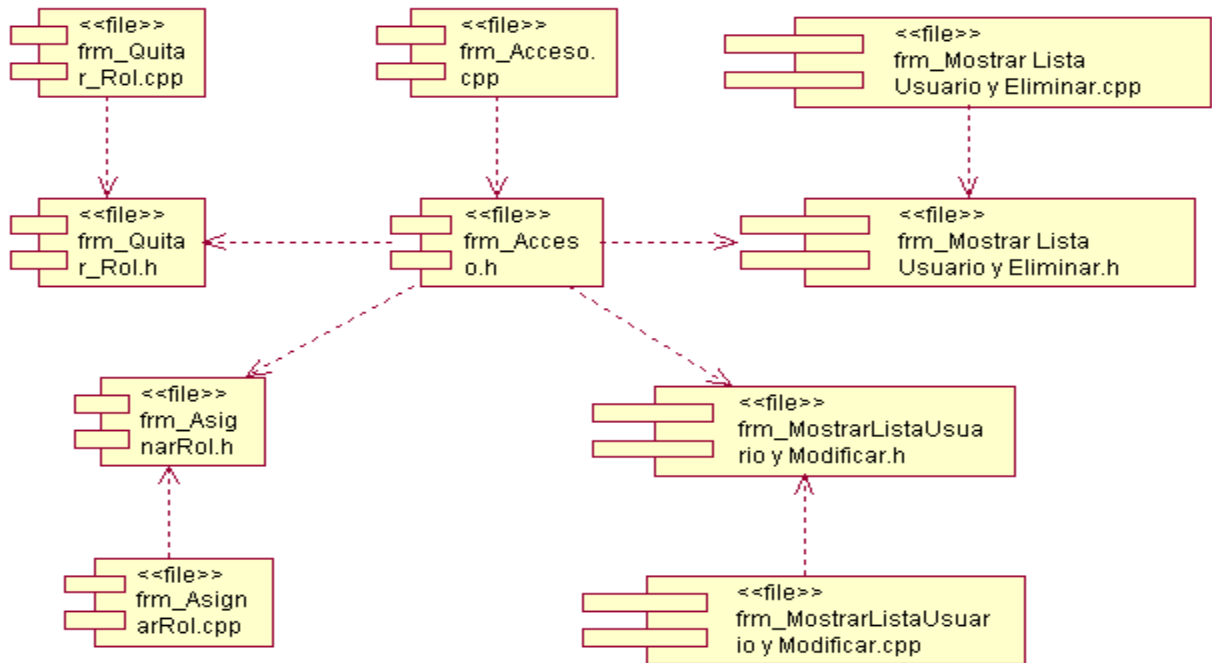


3.2.3 Diagrama de componentes Capa Acceso a datos (Vista Control de Acceso Firebird).



3.2.4 Diagramas de componentes Capa de Presentación.

3.2.4.1 Diagrama de componentes Administrador.



3.2.4.2 Diagrama de componentes Autenticar.



3.2.4.3 Diagrama de componentes Examen.



3.2.4.4 Diagrama de componentes Profesor.



3.2.4.5 Diagramas de componentes Usuario.

Vista RegistrarUsuario

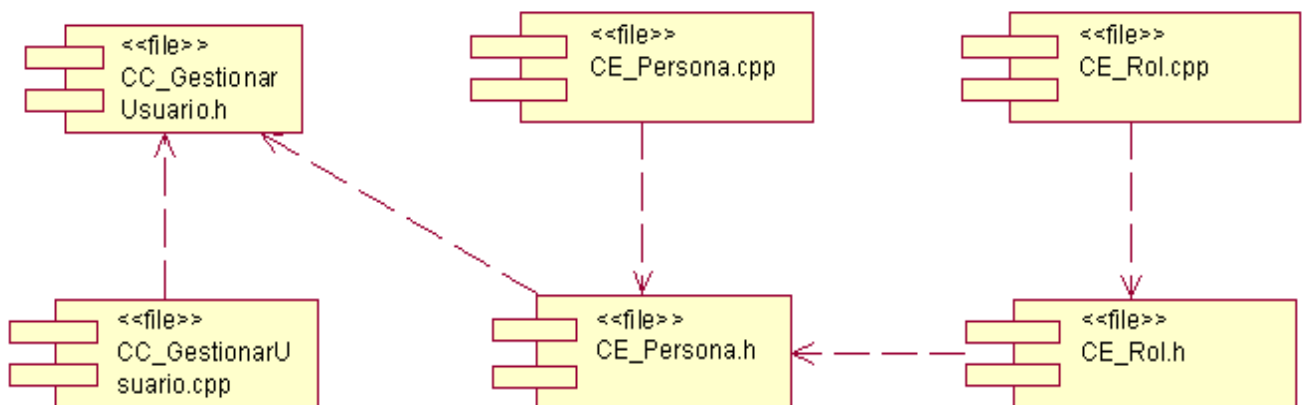


Vista ConsultarEstadisticas

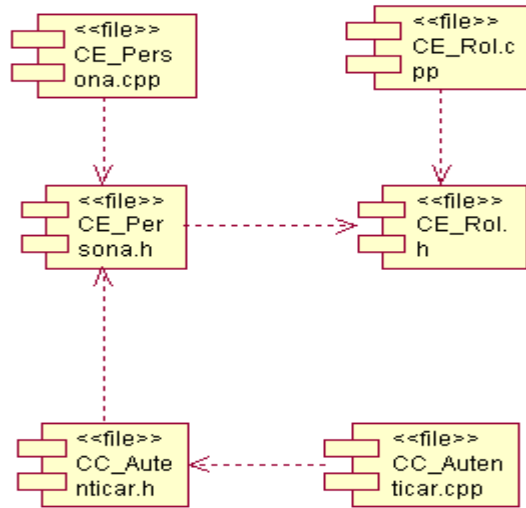


3.2.5 Diagramas de componentes Capa de Negocio.

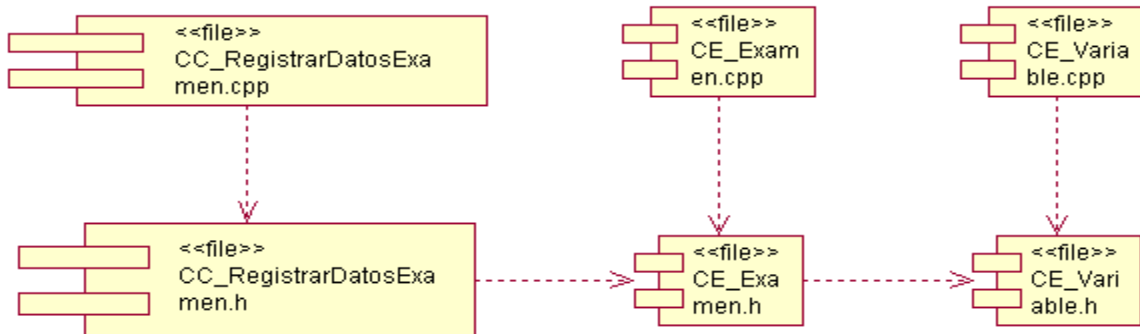
3.2.5.1 Diagrama de componentes Administrador.



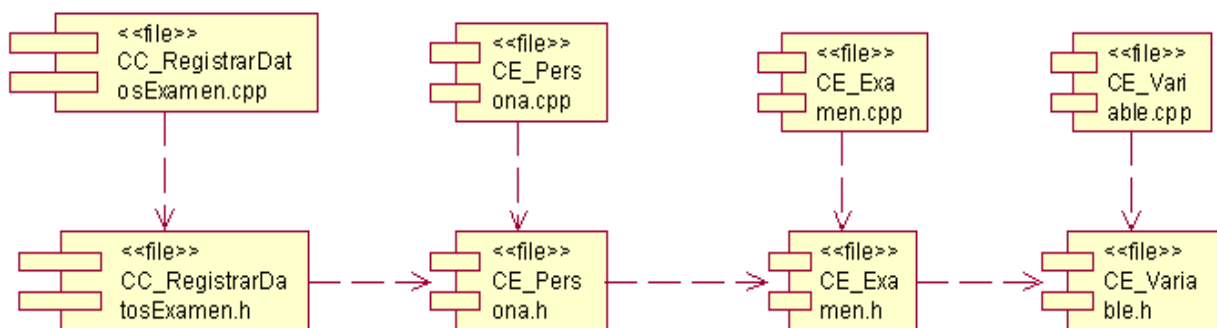
3.2.5.2 Diagrama de componentes Autenticar.



3.2.5.3 Diagrama de componentes Examen.

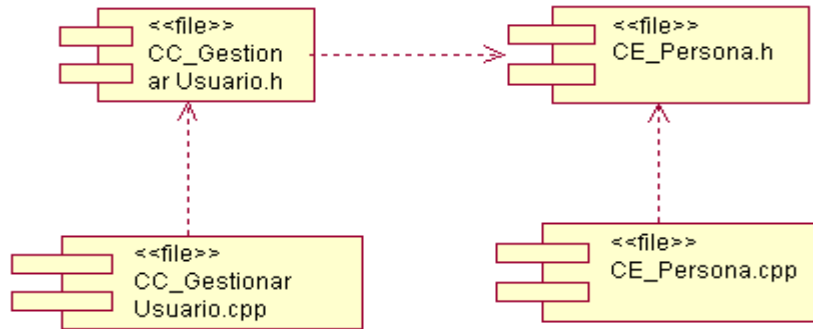


3.2.5.4 Diagrama de componentes Profesor.

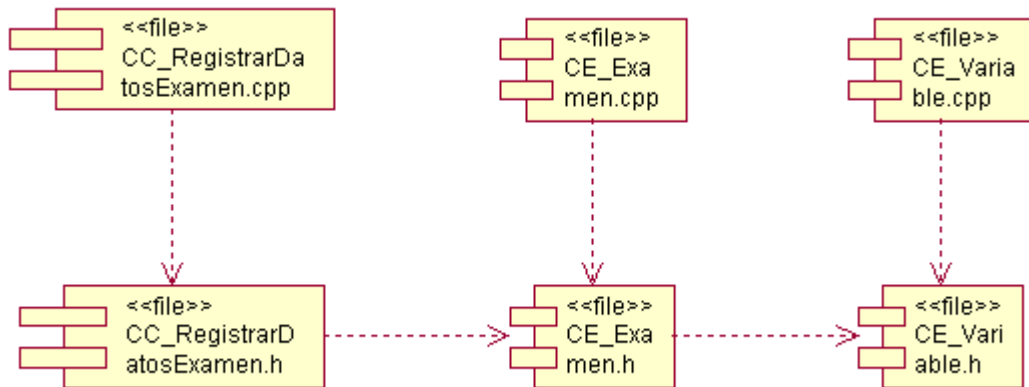


3.2.5.5 Diagramas de componentes Usuario.

Vista RegistrarUsuario



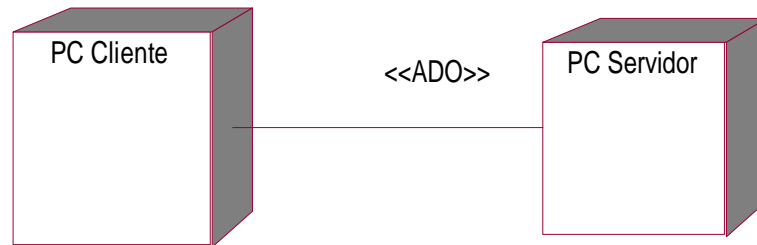
Vista ConsultarEstadísticas



3.3 Diagrama de Despliegue

El Diagrama de Despliegue es un Modelo de Objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. Es una colección de nodos y arcos, donde cada nodo representa un recurso de cómputo, normalmente un procesador o un dispositivo de hardware similar y cada arco la conectividad que existe entre dichos nodos. [30]

Muestra las relaciones entre el hardware y el software en el sistema final. Se representa como un grafo de nodos unidos por conexiones de comunicación.



3.4 Modelo de Prueba

El desarrollo del software implica una serie de actividades de producción en las que las posibilidades de que aparezca la falibilidad humana son comunes. Debido a la imposibilidad humana de trabajar y comunicarse de forma perfecta, el desarrollo del software ha de ir acompañado de una actividad que garantice la calidad. La prueba de software es un elemento crítico para la garantía de la calidad del software y representa una revisión final de las especificaciones del diseño y de la codificación.

La prueba de software es un proceso que corre en paralelo al proceso de desarrollo de software, y que se realiza por el convencimiento de que todo sistema debe ser “revisado” con el objetivo de establecer el nivel de calidad requerido. [\[31\]](#)

3.4.1 Objetivos

El objetivo de la fase de pruebas del sistema es verificar que el comportamiento externo del sistema software satisface los requisitos establecidos por los clientes y futuros usuarios del mismo. A medida que aumenta la complejidad de los sistemas software y aumenta la demanda de calidad, se hacen necesarios procesos y métodos que permitan obtener buenos conjuntos de pruebas del sistema. Este trabajo describe los modelos necesarios para generar de manera sistemática un conjunto de pruebas que permitan verificar la implementación de los requisitos funcionales de un sistema software. [\[32\]](#)

3.4.2 Alcance

Se lleva a cabo la prueba y se evalúan los resultados obtenidos frente a los resultados esperados. Si se descubren datos erróneos implica que hay un error y hay que corregirlo y empieza el proceso de depuración de errores. Se basa en las estructuras de control del diseño procedimental para generar los casos de prueba que:

√ Garanticen que se recorren por lo menos una vez todos los caminos independientes de cada módulo.

- √ Se ejecutan todas las decisiones lógicas en su parte verdadera y en su parte falsa.
- √ Se utilizan las estructuras de datos internas para garantizar su validez.
- √ Se invierte tiempo y esfuerzo en los detalles de control debido a que:
- √ Los errores suelen estar en situaciones fuera de las normales.

3.4.3 Prueba de software

Las pruebas de integración se llevan a cabo durante la construcción del sistema, involucran a un número creciente de módulos y terminan probando el sistema como conjunto.

Estas pruebas se pueden plantear desde un punto de vista estructural o funcional.

Las pruebas estructurales de integración son similares a las pruebas de caja blanca; pero trabajan a un nivel conceptual superior. En lugar de referirse a sentencias del lenguaje, se describirán las llamadas entre módulos. Se trata pues de identificar todos los posibles esquemas de llamadas y ejercitarlos para lograr una buena cobertura de segmentos o de ramas.

Las pruebas funcionales de integración son similares a las pruebas de caja negra. Aquí se tratarán de encontrar fallos en la respuesta de un módulo cuando su operación depende de los servicios prestados por otro(s) módulo(s). Según se vaya acercando al sistema total, estas pruebas se van basando más y más en la especificación de requisitos del usuario. [\[33\]](#)

3.4.3.1 Caso de Prueba Registrar Nuevo Usuario.

Flujo normal.

Entradas	Resultados	Condiciones
Seleccionar Registrar Nuevo Usuario.	Mostrar Interfaz con el formulario.	
Usuario, Dirección del trabajo, Teléfono del trabajo, Dirección particular, Teléfono particular, Nombre, Apellidos, Contraseña, Años de experiencia, Ocupación, Sexo, Edad.	Son almacenados los datos y se muestra un mensaje de confirmación.	Que ese usuario no exista, que estén llenos los campos.

Flujos Alternos.

Entradas	Resultados	Condiciones
Seleccionar Registrar Nuevo	Mostrar Interfaz con el	

Usuario.	formulario.	
Usuario.	Muestra un mensaje informando que el usuario ya existe.	Existe el Usuario.

Entradas	Resultados	Condiciones
Seleccionar Registrar Nuevo Usuario.	Mostrar Interfaz con el formulario.	
Usuario, Dirección del trabajo, Teléfono del trabajo, Dirección particular, Teléfono particular, Nombre, Apellidos, Contraseña, Años de experiencia, Ocupación, Sexo, Edad.	Muestra un mensaje informando que debe llenar todos los campos.	Campos Vacios.

3.4.3.2 Caso de Prueba Autenticar Usuario.

Flujo Normal.

Entradas	Resultados	Condiciones
Usuario y Contraseña.	Se le habilitan las funcionalidades según el rol que tenga.	Que los campos estén llenos, que se haya registrado o que coincidan estos datos con los almacenados.

Flujos Alternos.

Entradas	Resultados	Condiciones
Usuario y Contraseña.	Mostrar un mensaje informándole que debe de registrarse primero o que los datos son incorrectos.	No se ha registrado o no coinciden los datos entrados con los almacenados.

Entradas	Resultados	Condiciones
Usuario y Contraseña.	Mostrar un mensaje informándole que debe llenar todos los campos.	Campos vacios

3.4.3.3.0 Caso de Prueba Gestionar usuarios del sistema (Sección "Asignar Rol").

Flujo Normal.

Entradas	Resultados	Condiciones
Seleccionar Asignar Rol.	Va a la sección Asignar Rol y muestra un formulario.	Se tiene que haber autenticado como

		administrador.
Usuario	Mostrar datos del usuario.	Que exista el usuario
Rol	Mostrar un mensaje informando que ya ha sido asignado el rol al usuario.	Que ese usuario no tenga asignado ese permiso.

Flujos Alternos.

Entradas	Resultados	Condiciones
Seleccionar Asignar Rol.	Va a la sección Asignar Rol y muestra un formulario.	Se tiene que haber autenticado como administrador.
Usuario	Mostrar un mensaje informando que el usuario no existe, que debe de registrarlo primero.	No existe el usuario.

Entradas	Resultados	Condiciones
Seleccionar Asignar Rol.	Va a la sección Asignar Rol y muestra un formulario.	Se tiene que haber autenticado como administrador.
Usuario	Mostrar datos del usuario.	Que exista el usuario
Rol	Mostrar un mensaje informando que ese usuario ya tiene ese rol.	El usuario ya tiene asignado ese rol.

3.4.3.3.1 Caso de Prueba Gestionar usuarios del sistema (Sección “Modificar Usuario”).

Flujo Normal.

Entradas	Resultados	Condiciones
Seleccionar Modificar Usuario.	Va a la sección Modificar Usuario y muestra el listado de todos los Usuarios.	Se tiene que haber autenticado como administrador.
Usuario	Mostrar los datos del usuario.	Existe el usuario.
El dato que se vaya a modificar.	Son almacenados los datos actualizados y se muestra un mensaje informando que la acción fue realizada.	Campos llenos.

Flujos Alternos.

Entradas	Resultados	Condiciones
Seleccionar Modificar Usuario.	Va a la sección Modificar Usuario y muestra el listado de todos los Usuarios.	Se tiene que haber autenticado como administrador.
Usuario	Mostrar los datos del usuario.	Existe el usuario.
El dato que se vaya a modificar.	Muestra un mensaje informando que debe llenar todos los campos.	Campos Vacios.

3.4.3.3.2 Caso de Prueba Gestionar usuarios del sistema (Sección “Eliminar Usuario”).

Flujo Normal.

Entradas	Resultados	Condiciones
Seleccionar Eliminar Usuario.	Va a la sección Eliminar Usuario y muestra el listado de todos los Usuarios.	Se tiene que haber autenticado como administrador.
Usuario	Se elimina el usuario seleccionado y se emite un mensaje informando que la acción fue realizada.	Existe el usuario.

3.4.3.3.3 Caso de Prueba Gestionar usuarios del sistema (Sección “Quitar Rol”).

Flujo Normal.

Entradas	Resultados	Condiciones
Seleccionar “Quitar Rol”.	Va a la sección Quitar Rol y muestra el listado de todos los Usuarios.	Se tiene que haber autenticado como administrador.
Selecciona el usuario.	Muestra el rol que tiene el usuario.	Existe el usuario y que tenga un rol.
Selecciona el permiso y presiona el botón Eliminar.	Se le muestra un mensaje informándole que la operación se realizó correctamente.	Campos llenos y que el usuario que se seleccionó no sea el administrador universal.

Flujos Alternos.

Entradas	Resultados	Condiciones
Seleccionar “Quitar Rol”.	No se le muestra las opciones referentes al administrador.	No se autenticó como administrador.

Entradas	Resultados	Condiciones
Seleccionar "Quitar Rol".	Va a la sección Quitar Rol y muestra el listado de todos los Usuarios.	Se tiene que haber autenticado como administrador.
Selecciona el usuario.	Muestra el rol que tiene el usuario.	Existe el usuario y que tenga rol.
Selecciona el permiso y presiona el botón Eliminar.	Muestra un mensaje informándole que debe de llenar todos los campos.	Campos vacios.

Entradas	Resultados	Condiciones
Seleccionar "Quitar Rol".	Va a la sección Quitar Rol y muestra el listado de todos los Usuarios.	Se tiene que haber autenticado como administrador.
Selecciona el usuario.	Muestra el rol que tiene el usuario.	Existe el usuario y que tenga rol.
Selecciona el permiso y presiona el botón Eliminar.	Muestra un mensaje informándole que a ese usuario no se le puede quitar el rol.	El usuario que se seleccionó es el administrador universal.

3.4.3.4 Caso de Prueba Registrar Inicio Examen.

Flujo Normal.

Entradas	Resultados	Condiciones
Seleccionar Registrar Inicio Examen.	Mostrar interfaz y un formulario.	Se tiene que haber autenticado.
IdExamen, tipo de examen y nivel.	Coge el valor de la hora y la fecha del sistema operativo. Estos datos son almacenados.	No exista el id.

Flujos Alternos.

Entradas	Resultados	Condiciones
Seleccionar Registrar Inicio Examen.	Mostrar interfaz y un formulario.	Se tiene que haber autenticado.
IdExamen	Muestra un mensaje informando que ese IdExamen ya existe y le da la posibilidad de entrar otro id.	Existe el id.

3.4.3.5 Caso de Prueba Registrar Evento.

Flujo Normal.

Entradas	Resultados	Condiciones
Realización del examen.	Almacenar todos los datos de las variables de ese examen.	Que se realice un ejercicio.

3.4.3.6 Caso de Prueba Registrar Fin Examen.

Flujo Normal.

Entradas	Resultados	Condiciones
	Coge el valor de la hora y la fecha del sistema operativo. Estos datos son almacenados.	Terminar el ejercicio.

3.4.3.7 Caso de Prueba Mostrar Estadísticas de Médicos Cirujanos.

Flujo Normal.

Entradas	Resultados	Condiciones
Seleccionar Mostrar Estadísticas.	Mostrar lista de los médicos cirujanos que han realizado exámenes.	Se tiene que haber autenticado.
Selecciona Usuario.	Mostrar un listado de los ejercicios realizados por el usuario seleccionado.	Haber seleccionado el usuario.
Selecciona Ejercicio.	Mostrar las variables del ejercicio.	Haber seleccionado el ejercicio.

3.4.3.8 Caso de Prueba Consultar Reporte.

Flujo Normal.

Entradas	Resultados	Condiciones
Seleccionar Consultar Reporte.	Mostrar un listado de los exámenes realizados por él.	Se tiene que haber autenticado.
Selecciona examen.	Mostrar las variables del examen.	Haber seleccionado el examen.

3.4.4 Prueba de Caja Negra

Pruebas de caja negra: Pruebas que se llevan a cabo sobre la interfaz del software. El objetivo es demostrar que las funciones del software son operativas, que las entradas se aceptan de forma adecuada y se produce un resultado correcto, y que la integridad de la información externa se mantiene (no se ve el código). [34]

Esta prueba se centra principalmente en los requisitos funcionales del software, permitiendo encontrar: Funciones incorrecta o ausente.

Errores de interfaz.

Errores en estructuras de datos o en accesos a las Bases de Datos externas.

Errores de rendimiento.

Errores de inicialización y terminación.

3.4.5 Descripción de la prueba de caja negra realizada.

Este tipo de prueba se le aplicó a todos los casos de uso, para verificar el cumplimiento de los requisitos funcionales del sistema. Como método de prueba de caja negra se escogió el método de partición equivalente que divide el campo de entrada en clases de datos, para descubrir clases de errores, reduciendo la cantidad de casos de pruebas a desarrollar.

3.4.5.1 Caso de Prueba Registrar Nuevo Usuario.

Clases Válidas	Clases No Válidas	Resultados	Evaluación	Observaciones
Una persona selecciona "Registrar Nuevo Usuario".	Que no se le muestre el formulario.	Luego de seleccionar esta opción, se le muestra el formulario.	Satisfactorio	La operación se realizó correctamente.
La persona entra los datos que se le piden (Usuario, Dirección del	Ese nombre de usuario ya existe y se le muestra un mensaje	Luego de entrar los datos se le muestra un mensaje	Satisfactorio	La operación se realizó correctamente.

trabajo, Teléfono del trabajo, Dirección particular, Teléfono particular, Nombre, Apellidos, Contraseña, Años de experiencia, Ocupación, Sexo, Edad).	informándole que debe cambiar el nombre de usuario. Si hay algún campo vacío se le muestra un mensaje informándole que debe de llenar todos los campos.	informándole que el usuario fue insertado con éxito.		
---	---	--	--	--

3.4.5.2 Caso de Prueba Autenticar Usuario.

Clases Válidas	Clases No Válidas	Resultados	Evaluación	Observaciones
El usuario selecciona "Autenticar".	Que no se le muestre el formulario.	Luego de seleccionar esta opción, se le muestra el formulario.	Satisfactorio	La operación se realizó correctamente.
El usuario entra el nombre de usuario y la contraseña.	Ese nombre de usuario no existe o los datos entrados son incorrectos y se le muestra un mensaje informándole la situación. Si hay algún campo vacío se le muestra un mensaje informándole que debe de llenar todos los campos.	Luego de entrar el nombre de usuario y la contraseña, se le habilitan las funcionalidades según el rol que tenga.	Satisfactorio	La operación se realizó correctamente.

3.4.5.3 Caso de Prueba Gestionar usuarios del sistema (Sección “Asignar Rol”).

Clases Válidas	Clases No Válidas	Resultados	Evaluación	Observaciones
El usuario (administrador) selecciona “Asignar Rol”.	El usuario tiene que ser administrador, sino, no se le muestra la opción “Asignar Rol”.	Luego de seleccionar esta opción, se le muestra un formulario.	Satisfactorio	La operación se realizó correctamente.
El administrador entra el nombre del usuario que se le va asignar el rol.	El administrador entra el nombre del usuario y este no existe, y se le muestra un mensaje diciéndole que debe de registrarlo primero.	Luego de entrar el nombre del usuario se le muestran todos los datos del mismo y se le da la posibilidad al administrador de asignarle el rol.	Satisfactorio	La operación se realizó correctamente.
El administrador selecciona el rol que se la va asignar al usuario.	El administrador selecciona el rol y se le muestra un mensaje informándole que ese usuario ya tiene asignado ese rol.	Luego de seleccionar el rol se le muestra un mensaje informándole que el rol fue asignado.	Satisfactorio	La operación se realizó correctamente.

3.4.5.4 Caso de Prueba Gestionar usuarios del sistema (Sección “Modificar Usuario”).

Clases Válidas	Clases No Válidas	Resultados	Evaluación	Observaciones
El usuario (administrador) selecciona “Modificar Usuario”.	El usuario tiene que ser administrador, sino, no se le muestra la opción “Modificar Usuario”.	Luego de seleccionar esta opción, se le muestra el listado de todos los usuarios que están en la base de datos.	Satisfactorio	La operación se realizó correctamente.
El administrador escoge de la lista el usuario al que se le van hacer los cambios.	No se le muestra la lista de los usuarios.	Luego de escoger de la lista el usuario al que se le van hacer los cambios, se le muestran todos los datos del usuario.	Satisfactorio	La operación se realizó correctamente.
El administrador realiza los cambios deseados.	No se le muestran los datos del usuario para poder hacer los cambios, dejó algún campo vacío.	Luego de realizar los cambios deseados, se le muestra un mensaje confirmándole que se realizó la operación correctamente.	Satisfactorio	La operación se realizó correctamente.

3.4.5.5 Caso de Prueba Gestionar usuarios del sistema (Sección “Eliminar”).

Clases Válidas	Clases No Válidas	Resultados	Evaluación	Observaciones
El usuario	El usuario tiene que	Luego de	Satisfactorio	La operación se

(administrador) selecciona “Eliminar Usuario”.	ser administrador, sino, no se le muestra la opción “Eliminar Usuario”.	seleccionar esta opción, se le muestra el listado de todos los usuarios que están en la base de datos.		realizó correctamente.
El administrador escoge de la lista el usuario el que se va a eliminar.	No se le muestra la lista de los usuarios.	Luego de escoger de la lista el usuario a eliminar, se le muestra un mensaje informándole que el usuario fue eliminado con éxito.	Satisfactorio	La operación se realizó correctamente.

3.4.5.6 Caso de Prueba Gestionar usuarios del sistema (Sección “Quitar Rol”).

Clases Válidas	Clases No Válidas	Resultados	Evaluación	Observaciones
El usuario (administrador) selecciona “Quitar Rol”.	El usuario tiene que ser administrador, sino, no se le muestra la opción “Quitar Rol”.	Luego de seleccionar esta opción, se le muestra el listado de todos los usuarios que tienen rol.	Satisfactorio	La operación se realizó correctamente.
El administrador escoge de la lista el usuario al que se le desea quitar el rol.	No se le muestra la lista de los usuarios.	Luego de escoger de la lista el usuario al que se le van hacer los cambios, se le	Satisfactorio	La operación se realizó correctamente.

		muestran el rol que tiene el mismo.		
El administrador selecciona el rol si desea quitárselo al usuario.	No se le muestran el rol del usuario, dejó algún campo vacío.	Luego de seleccionar el rol y presionar el botón Eliminar, se le muestra un mensaje confirmándole que se realizó la operación correctamente.	Satisfactorio	La operación se realizó correctamente.

3.4.5.7 Caso de Prueba Registrar Inicio Examen.

Clases Válidas	Clases No Válidas	Resultados	Evaluación	Observaciones
El usuario selecciona "Registrar Inicio Examen".	Que no se le muestre el formulario.	Luego de seleccionar esta opción, se le muestra el formulario para registrar el examen.	Satisfactorio	La operación se realizó correctamente.
El usuario entra el identificador del examen, escoge el tipo de examen y el nivel del mismo.	Existe ya ese identificador de examen, y se le muestra un mensaje informándole que debe de entrar otro identificador.	Coge el valor de la hora y la fecha del sistema operativo. Estos datos y los entrados son almacenados.	Satisfactorio	La operación se realizó correctamente.

3.4.5.8 Caso de Prueba Registrar Evento.

Clases Válidas	Clases No Válidas	Resultados	Evaluación	Observaciones
Realización del examen.	Que no se almacenen las variables con sus valores.	Almacenar todos los datos de las variables de ese examen.	Satisfactorio	La operación se realizó correctamente.

3.4.5.9 Caso de Prueba Registrar Fin Examen.

Clases Válidas	Clases No Válidas	Resultados	Evaluación	Observaciones
Terminar el examen	Que no se almacenen los datos.	Coge el valor de la hora y la fecha del sistema operativo. Estos datos son almacenados.	Satisfactorio	La operación se realizó correctamente.

3.4.5.10 Caso de Prueba Consultar Reporte.

Clases Válidas	Clases No Válidas	Resultados	Evaluación	Observaciones
El usuario selecciona "Consultar Estadísticas" y se le muestra la lista de los exámenes realizados por ese usuario.	El usuario selecciona "Consultar Estadísticas" y no se le muestra la lista de los exámenes realizados por él.	Luego de seleccionar esta opción, el usuario observa la lista para la selección del examen.	Satisfactorio	La operación se realizó correctamente.
El usuario selecciona el examen y se le muestran las variables (con sus respectivos valores) que miden la	El usuario selecciona el examen y no se le muestran las variables.	Luego de seleccionar esta opción, el usuario observa las variables con sus respectivos valores.	Satisfactorio	La operación se realizó correctamente.

realización del examen.				
-------------------------	--	--	--	--

3.4.5.11 Caso de Prueba Mostrar Estadísticas de Médicos Cirujanos.

Clases Válidas	Clases No Válidas	Resultados	Evaluación	Observaciones
El usuario (en este caso el profesor) selecciona "Mostrar Estadísticas" y se le muestra la lista de usuarios que han realizado exámenes.	El usuario selecciona "Mostrar Estadísticas" y no se le muestra la lista de usuarios que han realizado exámenes.	Luego de seleccionar esta opción, el profesor observa la lista para la selección del usuario.	Satisfactorio	La operación se realizó correctamente.
El profesor selecciona el usuario y se le muestra la lista de los exámenes realizados por ese usuario.	El profesor selecciona el usuario y no se le muestra la lista de los exámenes realizados por ese usuario.	Luego de seleccionar esta opción, el profesor observa la lista de los exámenes realizados por el usuario, para seleccionar luego el examen a ver.	Satisfactorio	La operación se realizó correctamente.
El profesor selecciona el examen y se le muestran las variables (con sus respectivos valores) que miden la realización del examen.	El profesor selecciona el examen y no se le muestran las variables.	Luego de seleccionar esta opción, el profesor observa las variables con sus respectivos valores.	Satisfactorio	La operación se realizó correctamente.

Conclusiones Generales

Para dar cumplimiento al objetivo general de este trabajo, -en correspondencia con las exigencias del proyecto Simulador Quirúrgico-, fue necesario primeramente hacer un estudio de las técnicas, tecnologías y tendencias en cuanto al desarrollo alcanzado por los gestores de base de datos. Se analizaron funcionalidades y características básicas de cada uno de los gestores más utilizados a nivel mundial, y en dependencia de las necesidades del trabajo fue seleccionado el gestor de base de datos Firebird por ser el que más condiciones presentaba para el desarrollo de la aplicación. Se esbozó y estructuró el modelo de diseño de la base de datos, permitiendo esto la implementación de los procedimientos almacenados para el acceso a la base de datos.

A partir de estas premisas se obtuvo un módulo de registro de evaluación totalmente funcional, que cumple con los requisitos propuestos, permitiendo manipular los datos tanto personales del usuario como el desempeño del mismo en la realización de un examen, dándole esto respuesta a la situación problemática planteada. Además de facilitar la migración a otras plataformas debido a que quedó muy bien delimitada la separación entre la interfaz del usuario y la capa lógica del negocio y esta última con la capa de acceso a datos, impidiendo esto que el código no sufra cambios de gran envergadura.

Recomendaciones

Una vez concluida la investigación, se realizan las siguientes recomendaciones:

- √ Darle un seguimiento a las distintas etapas del proceso de evolución del Simulador Quirúrgico para en dependencia de los requisitos que se levanten en cada una de ellas, ajustar el diseño e implementación de la base de datos.
- √ Integrar este módulo a los demás confeccionados.
- √ Implementar un algoritmo de encriptación más eficiente.
- √ Hacer un refinamiento para próximas iteraciones del proyecto, con el objetivo de incrementar más funcionalidades a la aplicación.
- √ Hacer un diseño genérico para la utilización de un Simulador con características diferentes, donde se generen las tablas y las características de la misma en dependencia de las necesidades del proyecto.

Referencias bibliográficas

[1] Simuladores 3D: Cirugía en pacientes virtuales (15 de mayo del 2008). Disponible en:

http://www.ercilla.cl/web/index2.php?option=com_content&do_pdf=1&id=14

[2] **Batini, C., Ceri, S. y Navathe, S. (1994)**; Diseño Conceptual de bases de datos. Un enfoque de entidades-interrelaciones. Addison Wesley Iberoamericana. Disponible en

[http://books.google.com/books?id=-DP-](http://books.google.com/books?id=-DP-0puz338C&pg=PA5&dq=Dise%C3%B1o+conceptual+de+las+base+de+datos&hl=es&sig=hdlb7ZPrhU3ZiwnBF4mH50PoU)

[0puz338C&pg=PA5&dq=Dise%C3%B1o+conceptual+de+las+base+de+datos&hl=es&sig=hdlb7ZPrhU3ZiwnBF4mH50PoU](http://books.google.com/books?id=-DP-0puz338C&pg=PA5&dq=Dise%C3%B1o+conceptual+de+las+base+de+datos&hl=es&sig=hdlb7ZPrhU3ZiwnBF4mH50PoU)

[3] **Espinosa, Lic. Aloyma Casas Valdés Lic. José Antonio López.** *Aspectos teóricos en torno a la gestión del conocimiento en la medicina basada en evidencias.* España Disponible en:.

http://bvs.sld.cu/revistas/aci/vol17_2_08/aci03208.htm

[4]-Ídem al 3

[5] Sistema de base de datos, 25 de octubre del 2007]. Disponible en:

<http://www3.uji.es/~mmarques/f47/apun/node4.html>

[6] Gestores de Base de Datos, 26 de octubre del 2007]. Disponible en:

<http://wwwdi.ujaen.es/~barranco/publico/ofimatica/tema7.pdf>

[7] Proyecto final de carrera Sistema de Base de Datos cliente, 26 de octubre del 2007]. Disponible en:

<http://www.uv.es/marjoari/pfc/entregado-julio.pdf>

[8] **PEREZ, J. M** ¿Que es MYSQL?, 10 de octubre de 2007]. Disponible en:

<http://www.esepestudio.com/articulo/desarrollo-web/base-de-datos-mysql/Que-es-MySQL.htm>

[9] [¿Qué es MySQL? Historia de MySQL Características principales](#), 4 de noviembre del 2007].

Disponible en: <http://www.uaem.mx/posgrado/mcruz/cursos/miic/MySQL.pdf>

[10] *Sun Microsystems compra MySQL por mil millones de dólares.* Nueva York : s.n., 2008. Disponible

en: http://www.lostiempos.com/noticias/16-01-08/16_01_08_ultimas_eco10.php

[11] E, Quiñones, A, Introducción a PostgreSQL, 2 de noviembre de 2007]. Disponible en:

http://www.eqsoft.net/presentas/introduccion_a_postgresql.pdf

[12] **PostgreSQL** proporciona un gran número de características..., 2 de noviembre de 2007].

Disponible en: <http://www.sobl.org/traduccion/practical-postgres/node19.html>

[13] **www.sonorika.com**. [Online] [Cited: febrero 25, 2008.] <http://es.sonorika.com/firebird>.

[14] Comeau, Greg. Manual Básico de Programación en C++. Disponible en:

<http://www.bairesrobotics.com.ar/poo/manual%20programacion%20c++.pdf>

[15] **teleformacion.uci.cu**. [Online] [Cited: febrero 13, 2008.]Disponible en:

<http://teleformacion.uci.cu/mod/resource/view.php?id=8865>

[16] Ídem 15

[17] **Velthuis, Mario G. Piattini. 2003. ANÁLISIS Y DISEÑO DE APLICACIONES INFORMÁTICAS DE GESTIÓN. UNA PERSPECTIVA DE INGENIERÍA DEL SOFTWARE**. s.l. : Ra-ma, 2003.Disponible en:

<http://www.agapea.com/Analisis-y-diseno-de-Aplicaciones-Informaticas-de-Gestion-Una-perspectiva-de-Ingenieria-del-Software--n11658i.htm>

[18] **teleformacion.uci.cu**. [Online] [Cited: marzo 10, 2008.]Disponible en:

<http://teleformacion.uci.cu/mod/resource/view.php?id=10349>

[19] Ídem 18 (teleformacion.uci.cu.)

[20] Ídem 18 (teleformacion.uci.cu.)

[21] **Arquitectura en Capas DNA**, Un camino hacia los procesos distribuidos. [Online] marzo 25 , 2001. [Cited: febrero 21, 2008.]. Disponible en: http://www.docirs.cl/arquitectura_tres_capas.htm

[22] Ídem 21 (Arquitectura en Capas DNA)

[23] Ídem 21 (Arquitectura en Capas DNA)

[24] Implementación de Patrones de Diseño con .NET en el Mundo Real. Disponible en:

http://www.google.com.cu/search?hl=es&q=I+Bosque_Msdn_SesionPatrones_es-es&btnG=Buscar+con+Google&meta

[25] Mapeo de base de datos relacionales I: Patrones arquitectónicos. Disponible en:

<http://www.thinkinginblue.com/?p=49>

[26] El Lenguaje Unificado de Modelado (UML), autor: Enrique Hernández Orallo. Disponible en:

<http://www.disca.upv.es/enheror/pdf/ActaUML.PDF>

[27] **Tema 2 - Modelo de datos.**, 2005. Disponible en: <http://avellano.usal.es/~adeluis/dbd/t2.pdf>

[28] **teleformacion.uci.cu.** [Online] [Cited: marzo 3, 2008.] Disponible en:

<http://teleformacion.uci.cu/course/view.php?id=43>

[29] **Jacobson, I y Booch, G y Rumbaugh, J.** *El Proceso Unificado de Desarrollo del Software.* España : Addison Wesley. 2000.458.

[30] Ídem 29 (Jacobson)

[31] e-Quallity Corp. "Fundamentos". [Online] Javier J. Gutiérrez, 2006. [Cited: enero 12, 2007.]

Disponible en: <http://www.e-quallity.net/definiciones.php>

[32] **Javier J. Gutiérrez, María J. Escalona, Manuel Mejías y Antonia M. Reina.** 2006. Modelos de pruebas para pruebas del sistema. CIMNE, Barcelona : s.n., 2006.disponible en:

<http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-227/paper07.pdf>

[33] **Mañas, José A.** Prueba de Programas. 16 de marzo, 1994. Disponible en:

<http://www.lab.dit.upm.es/~lprg/material/apuntes/pruebas/testing.htm#s2>

[34] **Aguilar., Ing. Violena Hernández.** *Conferencia # 5 "Pruebas" Ingeniería de Software II.* Ciudad de La habana : UCI, 2007. Disponible en: [http://teleformacion.uci.cu/file.php/43/Curso_2007-](http://teleformacion.uci.cu/file.php/43/Curso_2007-2008/PPT_Pruebas/Conf_Pruebas.ppt)

[2008/PPT_Pruebas/Conf_Pruebas.ppt](http://teleformacion.uci.cu/file.php/43/Curso_2007-2008/PPT_Pruebas/Conf_Pruebas.ppt)

Bibliografía consultada

Antonio Sala, Ernesto Cullen, Marc Guillot, Juan Tendero y José Velasco. Notas de la Versión 1.5. febrero 2004. Disponible en: <http://www.firebirdsql.org/rlsnotes/es/Firebird-1.5-ReleaseNotes-Spanish.pdf>

BORRIE, HELEN. The Firebird Book: A Reference for Database Developers. s.l. : Chris Mills, 2004. Disponible en: <http://lecture.ecc.u-tokyo.ac.jp/~cueda/varilex/art/manual.pdf>

Tinoco, Antonio Ruiz. Manual de lenguaje SQL aplicado a la base de datos de Varilex. Universidad Sophia, Tokio : s.n., 2004. Disponible en: <http://lecture.ecc.u-tokyo.ac.jp/~cueda/varilex/art/manual.pdf>

Rodríguez, Juan José. Transacciones en InterBase y Firebird (0.2.6). Cuenca, España : s.n., junio 2002. Disponible en: <http://www.intitec.com/varios/transacciones-0.2.6.pdf>
Firebird 1.5 - Guía de arranque rápido. s.l. : Editores de IBPhoenix, 2003. Disponible en: http://www.firebirdsql.org/pdfmanual/Guia_arranque_rapido.pdf

Daily Echeverría Sadradin, Rudhy Ruiz Sosa, Jorge Cruz Peña. QS Versión 1.0 Software simulador de redes de transmisión de datos y de telecomunicaciones con facilidad de análisis de tráfico, fallos y comportamiento. Ciego de Ávila : s.n., 2003. Disponible en: <http://www.cujae.edu.cu/eventos/cittel/trabajos/Trabajos/Comision%202/CITTEL-101.pdf>

Molpeceres, Alberto. *Proceso de desarrollo: Rup, XP, FDD.* 2002. Disponible en: <http://www.willydev.net/descargas/articulos/general/cualxpfddrup.pdf>

Gil, Robin Alberto Castro. *Estructura básica del proceso unificado de desarrollo de software.* 2004. Disponible en: http://dspace.icesi.edu.co/dspace/bitstream/item/399/1/rcastro_estructura-bas-puds.pdf

Brufal, Paco. *Autenticación mediante PostgreSQL en Apache y ProFTPd.* 2002. Disponible en: <http://www.servitux.org/view.php/page/postgres>

Lockhart, Thomas. *El equipo de desarrollo de PostgreSQL.* 1996. Disponible en: <http://palomo.usach.cl/Docs/postgres/Postgres-Tutorial.pdf>

Lacaci, Antonio Tejada. BULMA: *Traducción del manual de PostgreSQL.* 2001. Disponible en: <http://bulma.net/pdf.phtml?nIdNoticia=273>

Orallo, Enrique Hernández. *El Lenguaje Unificado de Modelado (UML)*. 2003. Disponible en:

<http://www.disca.upv.es/enheror/pdf/ActaUML.PDF>

Rubiano, Sandra Merchán. *EL LENGUAJE UNIFICADO DE MODELADO*. s.l. : Fundación

Universitaria Manuela Beltrán, Octubre de 2005. Disponible

en:http://is.umb.edu.co/aulaumb/file.php/37/UML_EXPO.ppt

Marc Gibert Ginestà, Oscar Pérez Mora. *Bases de datos en PostgreSQL*. 2002. Disponible en:

http://ocw.uoc.edu/computer-science-technology-and-multimedia/bases-de-datos/bases-de-datos/XP06_M2109_02146.pdf

Lockhart, Thomas. *Guía del Programador de PostgreSQL*. 2001. Disponible en:

<https://forja.rediris.es/docman/view.php/312/461/Postgres-Programmer.pdf>

Rodríguez, JI. *La simulación del espacio quirúrgico. Un nuevo enfoque formativo*. s.l. : Centro de Entrenamiento Quirúrgico y transferencia tecnológica(CEQTt), 2004. Disponible en:

<http://www.uniovi.es/ceqtt/simulacion.pdf>

PROAÑO, DIEGO JAVIER BURBANO. *ANALISIS COMPARATIVO DE BASES DE DATOS DE CODIGO ABIERTO VS CODIGO CERRADO (DETERMINACION DE INDICES DE COMPARACION)*.

5 de Mayo del 2006. Disponible en: <http://www.mysql-hispano.org/articulos/num43/analisis-comparativo.pdf>

Tisselli., Eugenio. *Bases de Datos MySQL*. 2000. Disponible en:

<http://www.tecn.upf.es/~ocelma/cpom/mysql/Bases%20de%20Datos%20MySQL.ppt>

Shumer, Erick. *Introducción a las Bases de Datos*. 1999. Disponible en:

<http://www.fdi.ucm.es/cfi/access/Tema1.pdf>

Ballester, Jose Luis Domenech. *Bases de Datos*. octubre del 2007. Disponible en:

<http://www.ticat.org/treballs/twt/jldb-bd-p.pdf>

D., Gonzalo Rojas. *Algoritmos de Inserción Segura para Esquemas Conceptuales de Bases de Datos*.

agosto del 2002. Disponible en: <http://www.inf.udec.cl/revista/ediciones/edicion7/grojas.pdf>

Alejandro Fanjul, Mikel Sagues. *Arquitectura de 3 capas*. 2003. Disponible en:

<http://www.mhproject.org/media/blogs/mhpenlaces/Interno/Presentaciones/ATS-Interactiva/Mini-Presentacion%20-%20ATS-Interactiva.pps>

Astudillo, Marcello Visconti y Hernán. *Fundamentos de Ingeniería de Software*. Universidad Técnica Federico Santa María : s.n., 2000. Disponible en:

<http://www.inf.utfsm.cl/~visconti/ili236/Documentos/PautaCaso.pdf>

Guerrero, Luis A. *Análisis y Diseño Orientado a Objetos*. Chile : Departamento de Ciencias de la Computación, 2007. Disponible en:

http://www.eici.ucm.cl/Academicos/R_Villaruel/descargas/ing_sw_1/Analisis_Diseño_OO.pdf

Hugo Hernando Andrade Sosa, Marlene Lucila Guerrero Julio, Oscar Mauricio Vargas Rincón, Luis Carlos Gómez Florez. *MICROMUNDO PARA EL APRENDIZAJE DE LAS CIENCIAS DE LA NATURALEZA DE SEXTO Y SÉPTIMO GRADO*. Bucaramanga – Colombia. : Universidad Industrial de Santander, 2006. Disponible en:

http://dinamica-sistemas.mty.itesm.mx/congreso/ponencias_pdf/15.Ciencias_de_la_naturaleza.pdf

Garsaball, Xavier Martínez. *Inicio rápido a Firebird*. 08 de noviembre de 2004. Disponible en:

<http://www.clubdevelopers.com/delphi/articulos/sqbd/startfastfb/downloads/startfastfb.pdf>

Herrera, Álvaro. *Introducción a PostgreSQL*. 13 de noviembre de 2004. Disponible en:

<http://alvherre.atentus.cl/charlas/10-Introduccion.pdf>

Revista Digital Líder en Informática. [En línea] Master Magazine, 2004. [Citado el: 22 de marzo de 2008]. Disponible en: <http://www.mastermagazine.info/termino/6510.php>

Woodman L. Information management in large organizations. En: Information management from strategies to action. London: ASLIB; 1985. p. 95-114. Disponible en:

<http://eprints.rclis.org/archive/00008960/>

Requisitos genéricos para el sostenimiento de la información electrónica en el tiempo. Francia : s.n., 2006. Disponible en: http://www.scielo.org.ar/scielo.php?script=sci_arttext&pid=S1851-17402005000100008&lng=es&nrm=iso&tlng=es

Apéndices

Glosario de abreviaturas

API: Application Program Interface.

BD: Base de datos.

ER: Entidad Relación.

MVC: Patrón de Arquitectura. Modelo Vista Controlador.

MVCC: Control de Concurrencia Multi-Versión.

OO: Orientación a Objetos.

PC: Computadora personal, computadora de escritorio y microcomputadora.

RF: Requerimientos funcionales.

RNF: Requerimientos no funcionales.

RUP: Proceso Unificado de Modelado.

SGBD: Sistemas de Gestión de Base de Datos.

SGBDR: Sistema de Gestión de Base de Datos Relacional.

SQL: Structured Query Language

SI: Sistemas Informáticos.

TAD: Tipo de datos abstracto.

UCI: Universidad de las Ciencias Informáticas.

UML: El Lenguaje Unificado de Modelado.

Glosario de términos

Actor: entidad externa que inicia una actividad en el sistema (actor primario) o que interactúa en cualquier actividad (actores secundarios).

Administrador: es la persona que posee privilegios para determinadas funcionalidades del sistema.

C: es un lenguaje de programación creado en 1969 por Ken Thompson y Dennis M. Ritchie en los Laboratorios Bell como evolución del anterior lenguaje B.

C++: es un lenguaje de programación, diseñado a mediados de los años 1980, por Bjarne Stroustrup, como extensión del lenguaje de programación C.

Caso de uso: es una operación o tarea específica que se realiza tras una orden de algún agente externo, sea desde una petición de un actor o bien desde la invocación desde otro caso de uso.

Diagrama de Clases: muestra un conjunto de clases y las relaciones entre ellas. Muestra el diseño de un sistema desde un punto de vista estático.

Diagrama de Casos de Usos del Sistema: modelo gráfico que representa a todos los actores que interactúan con el sistema.

MySQL: es un sistema de gestión de base de datos relacional.

ODBC: conectividad abierta de bases de datos (Open DataBase Connectivity), que es un estándar de acceso a Bases de Datos.

Open Source: denominación para aquellas aplicaciones que tienen su código fuente liberado. Se puede obtener su código fuente, arreglarlo y modificarlo para satisfacer las necesidades de sí mismo.

Perl: Practical Extraction and Report Language. Es un lenguaje de programación desarrollado por Larry Wall inspirado en otras herramientas de UNIX como son: sed, grep, awk, c-shell.

PHP: Hypertext Preprocessor. Es un ambiente script del lado del servidor que permite crear y ejecutar aplicaciones Web dinámicas e interactivas. Con PHP se pueden combinar páginas HTML y scripts. Con el objetivo de crear aplicaciones potentes.

PL/Python: es una lengua procesal que permite que las funciones de PostgreSQL sean escritas en la lengua de Python (lenguaje de programación).

PostgreSQL: es un Sistema de Gestión de Bases de Datos Objeto-Relacionales.

Software: componente intangible de una computadora, es decir, un conjunto de programas y procedimientos necesarios para hacer posible la realización de una tarea específica.

Software libre: es el software que, una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente.

Software privativo: se refiere a cualquier programa informático en el que los usuarios tienen limitadas las posibilidades de usarlo, modificarlo o redistribuirlo (con o sin modificaciones), o cuyo código fuente no está disponible o el acceso a éste se encuentra restringido.

Tupla (fila o registro): corresponde a una fila de una tabla.

Anexo

Anexo 1. Descripciones Textuales.

2.5.3 Descripción de los casos de uso del sistema.


2.5.3.1 Caso de Uso Registrar Nuevo Usuario

Caso de Uso:	Registrar Nuevo Usuario	
Actores:	Persona (Inicia)	
Resumen:	El CU permite a una persona que quiera registrarse para convertirse en usuario del simulador, entrar todos los datos personales necesarios.	
Precondiciones:	--	
Referencias	R-1	
Prioridad	Crítico	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
<ol style="list-style-type: none"> 1. Seleccionar la opción de Registrar Nuevo Usuario 2. Llena el formulario 	<ol style="list-style-type: none"> 1.1 Muestra un formulario con los campos necesarios a almacenar de cada usuario del Simulador. 2.1 Verificar que todos los campos estén llenos. 2.2 Verifica que el usuario no exista. 2.3 Almacena los datos del nuevo usuario. 2.4 Muestra un mensaje informando que se ha registrado como nuevo usuario del Simulador. 	
Prototipo de Interfaz		
Flujos Alternos		
Acción del Actor	Respuesta del Sistema	
<ol style="list-style-type: none"> 2. Llena el formulario y deja algún campo sin llenar. 	<ol style="list-style-type: none"> 2.1 Muestra un mensaje informando que todos los campos deben estar llenos. 2.2 Muestra un mensaje informando que el usuario ya existe. 	
Prototipo de Interfaz		

Poscondiciones	Se guardaron los datos sin problemas y el usuario se registró como nuevo usuario del Simulador.
-----------------------	---

2.5.3.2 Caso de uso Autenticar usuario

Caso de Uso:	Autenticar Usuario
Actores:	Usuario del sistema(Inicia)
Resumen:	El caso de uso inicia cuando el usuario del sistema entra los datos que se le especifican para acceder al sistema, estos se verifican y finaliza dándole los permisos y habilitándole la entrada.
Precondiciones:	El usuario debe estar registrado como usuario del Simulador
Referencias	R-2
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
El usuario selecciona del menú principal la opción "Autenticarse". 2. El usuario entra usuario y contraseña.	1.1 El sistema le muestra el formulario correspondiente. 2.1 El sistema encripta la contraseña. 2.2 El sistema verifica los datos de entrada con los almacenados en la BD. 2.3 Si son correctos se le asignan el rol correspondiente.

Prototipo de Interfaz	
	
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
2. El usuario entra usuario y contraseña (incorrectos).	2.2 En caso de no ser validos los datos entrados se le muestra un mensaje informándole que debe registrarse o los datos son incorrectos.
Prototipo de Interfaz	
Poscondiciones	Se ha autenticado un usuario en el sistema donde se le habilitan las funcionalidades según sus privilegios.

2.5.3.3 Caso de uso Gestionar usuarios del sistema

Caso de Uso:	Gestionar usuarios del sistema
Actores:	Administrador del sistema(Inicia)
Resumen:	El caso de uso le va a permitir al administrador del sistema Asignar Rol, Modificar usuario, Eliminar usuario o Quitar Rol en dependencia de las necesidades este selecciona la acción a realizar y el sistema le habilita las opciones.
Precondiciones:	Administrador del sistema ya autenticado.
Referencias	R-3.1, R-3.2, R-3.3
Prioridad	Secundario
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El administrador del sistema necesita asignar o quitar un rol, modificar o eliminar los datos de un usuario.	1.1 El sistema ejecuta algunas de las siguientes acciones: Si decide asignar rol, ir a la sección "Asignar Rol" Si decide modificar los datos de un usuario, ir a la sección "Modificar usuario". Si decide eliminar un usuario, ir a la sección "Eliminar usuario". Si decide quitarle un rol a un usuario, ir a la sección "Quitar Rol".

Sección “Asignar Rol”	
Flujo Normal	
Acción del Actor	Respuesta del Sistema
<p>3. El administrador del sistema selecciona el nombre del usuario. 4. El administrador según los datos del usuario le asigna un rol.</p>	<p>2. El sistema muestra la lista de usuarios. 3.1 El sistema muestra los datos de ese usuario. 4.1 Se muestra un mensaje informándole al administrador que ya ha sido asignado el rol al usuario, finalizando así el caso de uso.</p>
Flujo Alterno	
Sección “Modificar usuario”	
Flujo Normal	
Acción del Actor	Respuesta del Sistema
<p>3. El administrador del sistema selecciona el usuario a modificar. 4. El administrador del sistema realiza las actualizaciones deseadas.</p>	<p>2. El sistema muestra la lista de usuarios. 3.1 El sistema brinda la posibilidad de modificar los datos. 4.1 Se verifica que todos los campos estén llenos. 4.2 Se actualiza la información incorporada y se emite un mensaje informando que la modificación se realizó satisfactoriamente. Finalizando así el caso de uso.</p>
Flujo Alterno	
<p>4. El administrador del sistema realiza las actualizaciones deseadas y deja algún campo vacío.</p>	<p>4.1 Se emite un mensaje informando que todos los campos deben de estar llenos.</p>
Sección “Eliminar usuario”	
Flujo Normal	
Acción del Actor	Respuesta del Sistema
<p>3. El administrador del sistema selecciona el usuario a eliminar.</p>	<p>2. El sistema muestra el listado de usuarios. 3.1 El sistema pide la confirmación de la eliminación del usuario. 3.2 El sistema elimina al usuario seleccionado y emite un mensaje informando que la eliminación se realizó satisfactoriamente. Finalizando así el caso de uso.</p>
Flujo Alterno	
Sección “Quitar Rol”	
Flujo Normal	
Acción del Actor	Respuesta del Sistema
<p>2. El administrador del sistema selecciona el usuario al que se le va a quitar el rol.</p>	<p>1.1 El sistema muestra el listado de todos los usuarios que tienen algún rol asignado. 2.1 El sistema muestra el rol que tiene ese usuario.</p>

<p>3. El administrador del sistema selecciona el rol a ser quitado.</p>	<p>3.1 El sistema verifica que todos los campos estén llenos. 3.2 El sistema verifica que el usuario seleccionado no sea el administrador universal. 3.3 El sistema pide la confirmación mostrando un mensaje (“Esta seguro que desea quitarle ese rol al usuario”). 3.4 El sistema le elimina el rol seleccionado al usuario y emite un mensaje informando que la operación se realizó satisfactoriamente. Finalizando así el caso de uso.</p>
<p>Flujo Alterno</p>	
<p> </p>	
<p> </p>	
<p>Poscondiciones</p>	<p>Se ha actualizado sin problema los cambios en la base de datos.</p>

2.5.3.4 Caso de Uso Registrar Inicio Ejercicio

<p>Caso de Uso:</p>	<p>Registrar Inicio Ejercicio</p>	
<p>Actores:</p>	<p>Médico Cirujano</p>	
<p>Resumen:</p>	<p>El caso de uso se inicia cuando el cirujano va a comenzar el ejercicio, el sistema se apropia de la hora y la fecha que tiene en ese momento el sistema operativo, esta se le asigna al ejercicio y se almacenan en la base de datos.</p>	
<p>Precondiciones:</p>	<p>El médico cirujano debe haberse autenticado previamente.</p>	
<p>Referencias</p>	<p>R-13</p>	
<p>Prioridad</p>	<p>Crítico</p>	
<p>Flujo Normal de Eventos</p>		
<p>Acción del Actor</p>	<p>Respuesta del Sistema</p>	
<p>1. El médico decide realizar un ejercicio, escoge el tipo de ejercicio, el nivel y registra el id del ejercicio.</p>	<p>1.1 El sistema verifica que el id no exista. 1.2 El sistema genera el ejercicio con el id y se apropia de la hora y la fecha del sistema operativo. 1.3 El sistema almacena los datos.</p>	
<p>Flujo Alterno</p>		
<p>El médico decide realizar un ejercicio, escoge el tipo de ejercicio, el nivel y registra el id del ejercicio (ya existe).</p>	<p>1.1 Si existe el id, el sistema envía un mensaje diciendo que entre otro id para el ejercicio.</p>	
<p> </p>		
<p>Poscondiciones</p>	<p>Se almacenan en la BD los datos registrados.</p>	

2.5.3.5 Caso de uso Registrar Evento

<p>Caso de Uso:</p>	<p>Registrar Evento</p>	
<p>Actores:</p>	<p>Médico Cirujano</p>	
<p>Resumen:</p>	<p>El caso de uso se inicia cuando el cirujano está realizando el ejercicio y el sistema almacena los datos.</p>	
<p>Precondiciones:</p>	<p>El médico cirujano debe haberse autenticado previamente.</p>	

Referencias	R-5, R-6, R-6.1, R-6.2, R-6.3, R-6.4, R-8, R-8.1, R-8.2, R-8.3, R-10, R-10.1, R-10.2, R-10.3, R-10.4
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El médico cirujano realiza el ejercicio.	1.1 El sistema almacena las estadísticas del tiempo de indecisión, de los errores cometidos, estadística de aciertos, errores de focalización, de velocidad de movimiento de la cámara, el tiempo de ausencia de movimiento de la cámara, de transiciones correctas y de cantidad de objetos caídos.
Flujo Alternativo	
Poscondiciones	Se almacenan en la BD los datos del ejercicio realizado.

2.5.3.6 Caso de Uso Registrar Fin Ejercicio

Caso de Uso:	Registrar Fin Ejercicio
Actores:	Médico Cirujano
Resumen:	El caso de uso se inicia cuando el cirujano finaliza el ejercicio y el sistema se apropia de la hora y la fecha que tiene en ese momento el sistema operativo, y se almacena en la base de datos como fecha y hora de culminación del ejercicio.
Precondiciones:	El médico cirujano debe haber comenzado el ejercicio.
Referencias	R-13
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El médico decide finalizar el ejercicio.	1.1 El sistema se apropia de la hora y la fecha del sistema operativo. 1.2 El sistema almacena los datos.
Flujo Alternativo	
Poscondiciones	Se almacenan en la BD los datos registrados.

2.5.3.7 Caso de uso Mostrar Estadísticas de Médicos Cirujanos

Caso de Uso:	Mostrar Estadísticas de médicos cirujanos
Actores:	Profesor
Resumen:	El caso de uso inicia cuando el Profesor decide consultar las estadísticas de los resultados en el entrenamiento de un médico cirujano y el sistema le muestra la lista de usuarios que han realizado exámenes y este selecciona el usuario y luego le muestra los exámenes realizados por ese usuario y este puede

	seleccionar el examen del cual quieren ver sus estadísticas y el sistema le muestra según el examen seleccionado.
Precondiciones:	El profesor debe haberse autenticado previamente.
Referencias	R-12
Prioridad	Secundario
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
El profesor selecciona la opción de mostrar estadísticas del entrenamiento de los médicos Cirujanos. El Profesor selecciona el médico cirujano al cual desea consultar. 3. El profesor selecciona el ejercicio que quiere consultar. 4. El profesor cierra la aplicación.	1.1 El sistema muestra la lista de los médicos cirujanos que han realizado su entrenamiento. 2.1 El sistema muestra un listado de los ejercicios realizados por el cirujano escogido. 3.1 El sistema le muestra las estadísticas de errores y aciertos cometidos en el entrenamiento del ejercicio.
Flujo Alternativo	
Poscondiciones	El sistema mostró las estadísticas de un médico cirujano durante su entrenamiento, y queda disponible para realizar otra operación.

2.5.3.8 Caso de uso Consultar Reporte

Caso de Uso:	Consultar Reporte
Actores:	Médico Cirujano
Resumen:	El caso de uso inicia cuando el médico cirujano decide consultar las estadísticas de los resultados en el entrenamiento de un ejercicio y el sistema le muestra los exámenes realizados por ese usuario y este puede seleccionar el examen del cual quieren ver sus estadísticas y el sistema le muestra según el examen seleccionado.
Precondiciones:	El médico cirujano debe haberse autenticado previamente.
Referencias	R-7,R-9, R-11
Prioridad	Secundario
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El médico cirujano selecciona la opción de consultar sus estadísticas. 2. El médico cirujano selecciona el ejercicio del cual quiere ver sus resultados. 3. El médico cirujano cierra la aplicación	1.1 El sistema muestra la lista de los ejercicios realizados por el cirujano. 2.1 El sistema le muestra las estadísticas de errores y aciertos cometidos en el entrenamiento del ejercicio.
Flujo Alternativo	
Poscondiciones	El sistema mostró las estadísticas de uno de los ejercicios realizados por el médico cirujano, y este queda disponible para realizar otra operación.

Anexo 2 Tablas de Clases del Diseño

Tabla 1. Descripción de la tabla CC_Autenticar.

Nombre: CC_Autenticar	
Tipo de clase: controladora	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Descripción:
Recibe Datos	Este se encarga de recibir los datos del usuario para poder hacer la validación de acceso.
Validar acceso	Esta responsabilidad se encarga de validar si los datos pasados como parámetro son correctos para garantizar la autenticación del usuario en el sistema.

Tabla 2. Descripción de la tabla CE_Persona.

Nombre: CE_Persona	
Tipo de clase: entidad	
Atributo	Tipo
Usuario	Char*
DireccionTrab	Char*
TelefonoTrab	Integer
DireccionPart	Char*
TelefonoPart	Integer
Nombre	Char*
Apellidos	Char*
Contrasenna	Char*
Ocupación	Char*
AnnoExp	Char*
Sexo	Char*
Edad	Char*

Tabla 3. Descripción de la tabla CE_Rol.

Nombre: CE_Rol	
Tipo de clase: entidad	
Atributo	Tipo
IdRol	Integer
NombreRol	Char*

Tabla 4. Descripción de la tabla CC_GestionarUsuario.

Nombre: CC_ControlDatosUsuario	
Tipo de clase: controladora	

Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Descripción:
RecibirDatos	Esta responsabilidad se encarga de recibir los datos entrados por el usuario.
EnviarDatos	Esta responsabilidad se encarga de enviar los datos de un usuario, a la base de datos para que estos sean registrados.
EnviaListaUsuario	Esta responsabilidad se encarga de enviar la lista de usuarios para que el administrador escoja a cual se le van a ser los cambios o cual va ser eliminado.
EnviaBuscarPermiso	Por la petición del usuario, esta responsabilidad se encarga de mandar un mensaje a la clase entidad, para la búsqueda de los permisos dado el usuario.
RecibirRolUsuario	Esta responsabilidad se encarga de recibir el rol que tenga un usuario.
EnviaRolUsuario	Es la encargada de enviar el rol obtenido y enviárselo a la clase interfaz.

Tabla 5. Descripción de la tabla CC_ RegistrarDatosExamen.

Nombre: CC_RegistrarDatosExamen	
Tipo de clase: controladora	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Descripción:
EnviarListaEjerReal	Esta responsabilidad se encarga de enviarle al usuario el listado de ejercicios realizados por él.
RecibirEjerSelec	Se encarga de recibir el ejercicio escogido por el usuario.
EnviarEstadEjerReal	Se usa para devolver el reporte del usuario en el ejercicio.
RecibirDatos	Esta responsabilidad se encarga de recibir los datos de inicio y fin del ejercicio, además de recibir los datos de las variables que se miden en cada ejercicio.
EnviarDatos	Este es encargado de enviar los datos de inicio y fin del ejercicio a la base de datos, además de los datos de las variables del ejercicio realizado.
EnviarListaUsuarEjerReal	Esta responsabilidad se encarga de enviarle al profesor el listado de los Usuarios que se han evaluado.
RecibirUsuarioSelec	Esta responsabilidad se encarga de recibir el usuario escogido por el profesor para ver sus estadísticas.

Tabla 6. Descripción de la tabla CE_Examen.

Nombre: CE_Examen	
Tipo de clase: entidad	
Atributo	Tipo
IdExamen	Integer

FechaIni	Char*
FechaFin	Char*
HoraIni	Char*
HoraFin	Char*
Nivel	Char*

Tabla 7. Descripción de la tabla CE_Variable.

Nombre: CE_Variable	
Tipo de clase: entidad	
Atributo	Tipo
NombreVar	Char*

Tabla 8. Descripción de la tabla CC_Gestionar Usuario.

Nombre: CC_Gestionar Usuario	
Tipo de clase: controladora	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Descripción:
RecibirDatosUsuario	Esta responsabilidad se encarga de recibir los datos de los usuarios.
EnviarDatosUsuarioBD	Este es encargado de enviar los datos de los usuarios a la base de datos.

Tabla 9. Descripción de la tabla ObjetoBD.

Nombre: ObjetoBD	
Tipo de clase: entidad	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Descripción:
Insertar	Método redefinido de AccesosDatosFirebird.
Eliminar	Método redefinido de AccesosDatosFirebird.
Modificar	Método redefinido de AccesosDatosFirebird.
Seleccionar	Método redefinido de AccesosDatosFirebird.

Tabla 10. Descripción de la tabla AccesosDatosFirebird.

Nombre: AccesosDatosFirebird	
Tipo de clase: entidad	

Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Descripción:
Conectar	Esta funcionalidad es la encargada de establecer la conexión al servidor de bases de datos
CerrarConexion	Permite cerrar la conexión con la base de datos
Ejecutar	Permite la ejecución de la base de datos.
Insertar	Permite la inserción de datos en una tabla de la base de datos.
Eliminar	Permite la eliminación de una tupla en la base de datos.
Modificar	Permite actualizar datos en la base de datos.
Seleccionar	Permite seleccionar los datos que se quieren mostrar.

Tabla 11. Descripción de la tabla CC_ControlBD.

Nombre: CC_ControlBD	
Tipo de clase: controladora	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Descripción:
Buscar	Permite realizar búsquedas en la BD.

Anexo 3 Descripción de las tablas del modelo de datos

Tabla 1. Descripción de la tabla Persona.

Nombre:	Persona	
Descripción:	En esta tabla se almacenan los datos correspondientes a los usuarios.	
Atributo	Tipo	Descripción
Usuario	Varchar(50)	Identificador de la tabla.
Dirección Trab	Varchar(50)	Dirección del centro de trabajo.
Dirección Part	Varchar(50)	Dirección particular del usuario.
Nombre	Varchar(50)	Nombre del usuario.
Apellidos	Varchar(50)	Apellidos del usuario.
TelefonoTrab	Varchar(50)	Teléfono del centro de trabajo.
Contrasenna	Varchar(50)	Contraseña con la cual el usuario puede acceder a las aplicaciones.
TelefonoPart	Varchar(50)	Teléfono particular del usuario.
Ocupación	Varchar(50)	Si es estudiante o profesor.
AnnoExp	Varchar(50)	Años de práctica del usuario.
Sexo	Varchar(50)	Sexo del usuario.
Edad	Varchar(50)	Edad del usuario.

Tabla 2. Descripción de la tabla Rol.

Nombre:	Rol	
Descripción:	En esta tabla es donde se asigna los diferentes roles.	
Atributo	Tipo	Descripción
IdRol	Varchar(50)	Identificador de la tabla.
NombreRol	Varchar(50)	Nombre de los diferentes roles.

Tabla 3. Descripción de la tabla PersonaRol.

Nombre:	Persona Rol	
Descripción:	En esta tabla se almacenan los usuarios con el id del rol que tiene.	
Atributo	Tipo	Descripción
Usuario	Varchar(50)	Identificador de la tabla.
IdRol	Varchar(50)	Identificador de la tabla.

Tabla 4. Descripción de la tabla Examen.

Nombre:	Examen	
Descripción:	En esta tabla se almacenan los datos correspondientes a los exámenes.	
Atributo	Tipo	Descripción
Usuario	Varchar(50)	Llave foránea de la tabla. Referencia a la Tabla Persona.
IdNomen	Varchar(50)	Llave foránea de la tabla. Referencia a la Tabla NomencladorTipo.
IdExamen	Varchar(50)	Identificador de la tabla.
Fechalni	Varchar(50)	Fecha de comienzo del examen.
Horalni	Varchar(50)	Hora de comienzo del examen.
FechaFin	Varchar(50)	Fecha de terminado el examen.
HoraFin	Varchar(50)	Hora de terminado el examen.
Nivel	Varchar(50)	Nivel de complejidad del ejercicio(básico, medio, avanzado)

Tabla 5. Descripción de la tabla Variable.

Nombre:	Variable	
Descripción:	En esta tabla se almacenan los datos correspondientes a las variables de cada ejercicio.	
Atributo	Tipo	Descripción
NombreVar	Varchar(50)	Identificador de la tabla.

Tabla 6. Descripción de la tabla Nomenclador Tipo

Nombre:	Nomenclador Tipo	
Descripción:	En esta tabla se almacenan los ejercicios a realizar por los usuarios.	
Atributo	Tipo	Descripción

IdNomen	Varchar(50)	Identificador de la tabla.
NombreNomen	Varchar(50)	Nombre de cada ejercicio.

Tabla 7. Descripción de la tabla NomencladorTipoVariable

Nombre:	NomencladorTipoVariable	
Descripción:	En esta tabla se almacenan los id de las variables con el id del nomenclador a que corresponde cada una.	
Atributo	Tipo	Descripción
IdNomen	Varchar(50)	Identificador de la tabla.
NombreVar	Varchar(50)	Identificador de la tabla.

Tabla 8. Descripción de la tabla ExamenVariable

Nombre:	ExamenVariable	
Descripción:	En esta tabla se almacenan los datos que toman las variables con la realización de un ejercicio.	
Atributo	Tipo	Descripción
IdExamen	Varchar(50)	Identificador de la tabla.
NombreVar	Varchar(50)	Identificador de la tabla.
Valor	Varchar(50)	El valor que toma la variable.