



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 2

Implementación del Módulo de Administración del  
Sistema Integrador de Gestión de Vuelo para Aviones  
IL-96/300

Tesis presentada en la opción al grado de Ingeniero en Ciencias Informáticas.

Autor: Yoandry León Pérez.

Tutores: Ing. Leydis E. Garzón Giro.

Ing. Maidelis Milanés Luque.

Ciudad de La Habana, Junio 2008.

Año 50 de la Revolución

## **DECLARACIÓN DE AUTORÍA**

Declaro que soy el único autor de este trabajo y autorizo a la Facultad 2 de la Universidad de las Ciencias Informáticas (UCI) a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los \_\_\_ días del mes de \_\_\_ del año \_\_\_.

---

**Firma del Autor**

Yoandry León Pérez

---

**Firma del Tutor**

Leydis E. Garzón Giro

---

**Firma del Tutor**

Maidelis Milanés Luque

# DEDICATORIA

---

*A mis padres, los mejores del mundo.*

*Para Tatita, Aya, Cucú y Pay, al cumplir mi sueño les cumplo parte de los suyos.*

*A mi hermanito, del que espero muchas y grandes cosas.*

*A mi tía quien provocó al bichito del conocimiento en mí y a mi tío que siempre me ha apoyado.*

# AGRADECIMIENTOS

---

A mis padres porque sin ellos esto no hubiera sido posible.

A mis abuelas y abuelos que siempre han deseado poder ver este día.

A mi novia, Yasmin, por su constante apoyo e incondicionalidad, sin ella hubiera sido mucho más difícil.

A todos mis amigos.

A mis tutores Maidelis, Leydis y Taymí que han contribuido con la preparación del trabajo.

A los que mencioné y a los que se quedaron por mencionar, muchas gracias de todo corazón.

Uno de los principales objetivos de toda empresa, y particularmente las de aviación, es garantizar la seguridad en sus sistemas.

El trabajo consiste en la confección de un módulo de administración para el Sistema Integrador de Gestión de Vuelo para Aviones IL-96/300, que sea capaz de monitorear las trazas de los usuarios en el sistema a partir del chequeo del historial, que controle el acceso de los usuarios al sistema, así como el trabajo dentro de cada módulo o área. También debe posibilitar y facilitar el proceso de gestión de roles y usuarios.

## Contenido

INTRODUCCIÓN.....	1
Capítulo I: Fundamentación Teórica.....	4
Introducción.....	4
1.1 Sistemas automatizados vinculados al campo de acción.....	4
SAPLAV.....	4
AeroPlaner.com.....	5
1.2 Tendencias y tecnologías actuales.....	6
1.2.1 Tecnología .NET.....	6
Arquitectura Framework.NET.....	7
Propiedades de la plataforma .NET.....	8
1.2.2 Lenguaje de programación.....	8
Lenguaje C#.....	8
Características del lenguaje C#.....	8
1.2.3 Arquitectura.....	9
Programación multicapas.....	9
Arquitectura de 3 capas.....	9
Ventajas de la arquitectura de 3 capas.....	10
1.2.4 Proceso de desarrollo de software.....	11
Criterios de selección de metodologías de desarrollo de software.....	11
1.2.5 Metodologías de desarrollo de software.....	13
Extreme Programming (XP).....	13
Microsoft Solution Framework (MSF).....	14
Rational Unified Process (RUP).....	15

1.2.6 UML como lenguaje de modelación visual. ....	16
1.2.7 Herramientas CASE para la modelación del sistema.....	16
Visual Paradigm. ....	17
Rational Rose Enterprise Edition de la Suite 2003. ....	17
1.3 Conclusiones. ....	18
Capítulo II: Características del Sistema. ....	19
Introducción. ....	19
2.1 Modelo de Dominio. ....	19
Glosario de conceptos del Modelo de Dominio. ....	20
2.2 Especificación de los requerimientos.....	21
Requerimientos funcionales.....	21
Requerimientos no funcionales. ....	22
2.3 Actores del Sistema.....	24
2.4 Modelo de Casos de Uso del Sistema.....	25
2.5 Descripción de los Casos de Uso del Sistema. ....	25
Caso de Uso: Autenticar Usuario. ....	25
Caso de Uso: Gestionar Usuario. ....	27
Caso de Uso: Gestionar Rol. ....	31
Caso de Uso: Gestionar Historial. ....	33
2.6 Conclusiones. ....	39
Capítulo III: Análisis y Diseño del Sistema.....	40
Introducción. ....	40
3.1 Análisis.....	40
3.1.1 Diagramas de Clases del Análisis.....	41

Caso de Uso: Autenticar Usuario .....	41
Caso de Uso: Gestionar Rol .....	42
Caso de Uso: Gestionar Usuario .....	43
Caso de Uso: Gestionar Historial .....	44
3.2 Diseño.....	44
3.2.1 Diagramas de Clases del Diseño.....	45
Caso de Uso: Autenticar Usuario .....	45
Caso de Uso: Gestionar Rol .....	46
Caso de Uso: Gestionar Usuario .....	47
Caso de Uso: Gestionar Historial .....	48
3.3 Diseño de la Base de Datos.....	49
3.3.1 Modelo Lógico de Datos.....	49
3.3.2 Modelo Físico de Datos.....	50
3.4 Conclusiones.....	51
Capítulo IV: Implementación.....	52
Introducción.....	52
4.1 Diagrama de Despliegue.....	52
4.2 Diagrama de Componentes.....	52
4.2.1 Diagrama de componentes: CU Autenticar Usuario.....	53
4.2.2 Diagrama de componentes: CU Gestionar Usuario.....	54
4.2.3 Diagrama de componentes: CU Gestionar Rol.....	55
4.2.4 Diagrama de componentes: CU Gestionar Historial.....	56
4.3 Conclusiones.....	56
Capítulo V: Estudio de Factibilidad.....	57

Introducción. ....	57
5.1 Planificación basada en casos de uso.....	57
5.2 Análisis de costos y beneficios. ....	63
5.3 Conclusiones. ....	64
CONCLUSIONES. ....	65
RECOMENDACIONES.....	66
Bibliografía .....	67
ANEXOS. ....	69

## INTRODUCCIÓN.

A partir del año 2005, y gracias al despunte económico que ha tenido nuestro país, se ha decidido por parte del gobierno revitalizar varios sectores de la vida socioeconómica nacional, abarcando ramas como la educación, la salud, el turismo, el deporte y la aeronáutica civil. Esta última, principalmente, ha sido objeto de una fuerte inversión para revitalizar muchos de sus servicios, tanto nacionales como internacionales. En el país se han remodelado varias terminales aéreas, se han modernizado los servicios de envío de paquetes por vía aérea y ha mejorado la calidad de la atención prestada por parte de la tripulación durante el vuelo.

Como resultado de estas inversiones también se han adquirido varias aeronaves de nueva generación, modelo IL-96-300, a través de acuerdos con la República de Rusia. Para que se pueda realizar un vuelo se ejecutan tres procesos fundamentales que abarcan el análisis de pista, la planificación del vuelo y el estudio del peso y balance del avión. Bajo determinadas circunstancias la tripulación tiene que realizar los cálculos relacionados con cada proceso de manera manual. Esto, además de hacer más engorroso el trabajo, puede ocasionar cálculos y resultados erróneos.

Como posible solución a este problema se concibió la construcción del “Sistema Integrador de Gestión de Vuelo para Aviones IL-96/300”, en la Facultad #2, perteneciente a la Universidad de las Ciencias Informáticas, institución que desempeña un papel fundamental en el proceso de informatización que desarrolla el país en varias esferas de la sociedad. Este sistema está constituido por los módulos: “Sistema de Automatización para el Peso y Balance del IL-96-300”, “Sistema Automatizado para la Obtención del Plan de Vuelo del IL-96-300” y “Sistema de Automatización para el Análisis de Pista del IL-96-300”, que ayudan a la tripulación a realizar los cálculos de una manera más sencilla y segura.

En estos momentos no se dispone de un sistema totalmente seguro en lo que respecta al acceso de los diferentes miembros de la tripulación a su área específica de trabajo dentro de cada módulo. Esto trae consigo el peligro de que ocurran inserciones de datos erróneos por parte de personas no autorizadas ó no capacitadas, lo que provocaría tener que realizar una nueva planificación durante el vuelo. Además, actualmente no existe forma de monitorear, a través de un historial o de trazas, las operaciones que se realizan dentro de las áreas de trabajo por los usuarios, lo cual dificulta la adopción de medidas oportunas en caso de que ocurra algún incidente.

Analizando la situación planteada se puede definir como **problema científico**: ¿Cómo controlar el acceso de los usuarios a las distintas áreas de trabajo mediante un módulo de administración del “Sistema Integrador de Gestión de Vuelo para Aviones IL-96/300”?

Con vistas a resolver los problemas planteados anteriormente se define como **objeto de estudio** el proceso de automatización del “Sistema Integrador de Gestión de Vuelo para Aviones IL-96/300”.

Definiendo el **campo de acción** como la administración y el control del acceso de los usuarios al “Sistema Integrador de Gestión de Vuelo para Aviones IL-96-300”.

Se ha determinado como **objetivo general** de la investigación, la implementación de un módulo de administración para la gestión de la seguridad del “Sistema Integrador de Gestión de Vuelo para Aviones IL-96-300”.

De ahí que para lograr la consecución de dicho objetivo general se formulan las siguientes **tareas de investigación**:

- Analizar el estado del arte de módulos de administración en sistemas informáticos similares.
- Realizar entrevistas con el cliente para definir los niveles de acceso de los usuarios a la aplicación y analizar sus necesidades de seguridad.
- Definir procesos a implementar como parte del módulo de administración del “Sistema Integrador de Gestión de Vuelo para Aviones IL-96-300”.
- Obtener el Modelo de Análisis del módulo de administración del “Sistema Integrador de Gestión de Vuelo para Aviones IL-96-300”.
- Obtener el Modelo de Diseño del módulo de administración del “Sistema Integrador de Gestión de Vuelo para Aviones IL-96-300”.

Entre los **resultados esperados** podemos contar la obtención de un módulo, dentro del “Sistema Integrador de Gestión de Vuelo para Aviones IL-96-300”, que facilite: el acceso de los usuarios al sistema según sus privilegios, la inserción y eliminación, ó modificación, de los permisos e información de los usuarios del sistema, así como la generación de reportes de historiales que permitan controlar las actividades dentro del sistema de los mismos.

El presente trabajo consta de 5 capítulos en los cuales se aborda el estudio y desarrollo de la aplicación, quedando organizados de la siguiente manera:

Capítulo 1: Contiene la fundamentación teórica del trabajo. En él se analizan los sistemas existentes, nacional e internacionalmente, que guardan alguna relación con el que se trata en el documento, se detallan las tecnologías, herramientas y metodologías en las que se apoya la solución propuesta realizando un análisis crítico de cada una de ellas.

Capítulo 2: En este capítulo se define como el sistema debe funcionar a partir de los requerimientos funcionales y no funcionales especificados, haciendo referencia al flujo de trabajo de Modelación del Negocio y Captura de Requisitos. Se especifica el modelo de dominio a partir de los conceptos identificados en el negocio.

Capítulo 3: En este capítulo se hace referencia al flujo de Análisis y Diseño. Se definen el modelo de análisis y de clases del diseño, así como los diagramas de interacción por cada caso de uso. También queda estructurado el modelo de datos y el diagrama de clases persistentes.

Capítulo 4: Se hace referencia al flujo de Implementación, especificando los diagramas de componentes y despliegue, que en su conjunto forman el modelo de implementación.

Capítulo 5: Se realiza el estudio de factibilidad del software, especificando los beneficios tangibles e intangibles que se derivan de su implantación e implementación.

En el trabajo las palabras que aparecen en cursiva pertenecen a otro idioma y a las que se les ha colocado un número en superíndice tienen su significado definido al pie de pagina.

## Capítulo I: Fundamentación Teórica.

### Introducción.

En este capítulo se fundamentan cada una de las metodologías de desarrollo de software utilizadas, herramientas, arquitectura elegida, así como el lenguaje de programación sobre el cual se basará la propuesta de solución. Se realiza un bosquejo sobre los módulos de seguridad o sobre los niveles de acceso de los sistemas existentes, nacional e internacionalmente, vinculados al campo de acción.

### 1.1 Sistemas automatizados vinculados al campo de acción.

Analizando las tecnologías y tendencias actuales, a nivel nacional e internacional, de productos y sistemas que brindan solución a problemas similares, y centrándonos en sus módulos de administración o de manejo de permisos y acceso, se han encontrado los siguientes ejemplos de los cuales se brindan sus principales características.

#### SAPLAV.

SAPLAV es un sistema desarrollado por la CUJAE<sup>1</sup> en 1993 con el objetivo de automatizar la obtención de un plan de vuelo para aeronaves del tipo: YAK-40, YAK-42, AN-26 e IL-62. Aunque este sistema aún es utilizado no cumple con los requerimientos actuales de navegación aérea.

El acceso al sistema se divide en 2 categorías o niveles de acceso, donde cada usuario posee una clave única de autenticación. El primer nivel de acceso posee permisos para:

- Crear, eliminar, modificar e invertir, unir y transformar rutas de vuelo.
- Crear, eliminar y modificar puntos de control (actualizar base de puntos de control).
- Introducir, modificar y eliminar aeronave existente.
- Visualizar las aeronaves almacenadas en el sistema con sus respectivos datos y matriculas.
- Confeccionar plan de vuelo.
- Imprimir y archivar plan de vuelo.

---

<sup>1</sup> CUJAE: Ciudad Universitaria José Antonio Echeverría.

Mientras, a los usuarios que poseen nivel de acceso 2 se les restringe:

- No pueden crear ni modificar rutas fija.
- No pueden transformar rutas.
- No pueden modificar valores de rendimiento de los aviones.

## **AeroPlaner.com.**

AeroPlaner.com es un servicio web que permite la confección de un plan de vuelo online.

AeroPlaner.com cuenta con tres niveles de acceso para sus usuarios:

-*Basic Membership.*

-*Premium Membership.*

-*Premium plus Membership*

Estas 3 categorías permiten:

**Basic:** estudiar NOTAMS<sup>2</sup> actual y las condiciones atmosféricas. Descargar las cartas de navegación de manera parcial o completamente.

**Premium:** planear con *Pro Fligth Planner* que proporciona mejoras tales como perfeccionamiento en el plan DUATS<sup>3</sup>, auto-enrutamiento, perfiles del terreno, recubrimiento del radar de tiempo, entre otras. Descargar e imprimir los libros de procedimientos terminales.

**Premium Plus:** permite el libre acceso a los 3 planificadores de vuelo (*Basic Fligth Planner, Pro Fligth Planner, New Pro Fligth Planner*) y libre transferencia de cartas de navegación del tipo WACs<sup>4</sup> e IRF<sup>5</sup>, alto y bajo.

---

<sup>2</sup> NOTAMS: Notices To Airman (Avisos a los aviadores). Publicados en forma de boletines con el objetivo de informar a los pilotos los cambios existentes en cuanto a instalaciones aeronáuticas, servicios, procedimientos o peligros, ya sean temporales o permanentes (Tránsito Aéreo).

<sup>3</sup> DUATS: Direct User Access Terminal Services (Acceso Directo de Usuarios a Servicios de Terminales).

<sup>4</sup> WACs: World Aeronautical Charts (Cartas Aeronáuticas Mundiales).

### Funcionalidades de los planificadores de vuelo.

**Basic Flight Planner:** Permite que los usuarios hagan un plan de vuelo basado en aeropuertos, navais<sup>6</sup>, o ciudades conocidas. Se pueden utilizar las cartas de Sectionals o WACs. Calcula *headings*, los vientos, tiempo y el combustible. Solo debes saber los aeropuertos, VORs<sup>7</sup> o el código postal del país o región a donde deseas volar. Notifica si haz trazado un curso con un área restringida. Genera un archivo en formato pdf con el informe del registro de la navegación listo para imprimir.

**Pro Flight Planner y New Pro Flight Planner:** Agrega muchas más funcionalidades al planificador básico. Permite que un usuario inserte o suprima puntos del centro de registro de la navegación y pone al día la lista automáticamente. Permite crear *waypoints* y anotarlos en la carta, modificar el tamaño del mapa e importar o imprimir los planes de vuelo. Utilizar cartas IFR<sup>8</sup> bajas y altas para tu planeamiento. Incluye la capacidad de exportar tu ruta a una variedad de formatos.

## 1.2 Tendencias y tecnologías actuales.

### 1.2.1 Tecnología .NET.

Microsoft .NET es un conjunto de tecnologías sobre las cuales Microsoft ha venido trabajando en los últimos años, muchas veces de manera aislada y dispersa, a las cuales ha querido unir en un solo paquete. El objetivo de Microsoft con el desarrollo de .NET es el de brindar una plataforma de desarrollo sencilla de usar, potente en la distribución de servicios, que permita la comunicación entre aplicaciones independientemente del lenguaje en el que se han desarrollado.

.NET ofrece un entorno de desarrollo de aplicaciones llamado Visual Studio .NET que consta de varios lenguajes de programación como Visual Basic .NET, Visual C#, Visual FoxPro y Visual C++ .NET. Estos lenguajes combinan las características de los lenguajes existentes con nuevas posibilidades para

---

<sup>5</sup> IRF: Immediate Reaction Force (Fuerza de Reacción Inmediata).

<sup>6</sup> Navais: Navigational Aids (Ayudas de Navegación).

<sup>7</sup> VORs: Very high frequency Omni directional Range (Gama direccional Omni de alta frecuencia).

<sup>8</sup> IFR: Instrument Flight Rules (Reglas de Vuelo por Instrumentos).

proporcionar un potente sistema de desarrollo. A continuación, se detallan algunas de las características de la Arquitectura .NET (Microsoft, Visual Studio Team System).

### **Arquitectura Framework.NET.**

El objetivo principal de esta arquitectura es el de reducir la complejidad en el desarrollo de aplicaciones, proporcionando una fácil comunicación e integración entre los lenguajes de la plataforma .NET.

El diseño del Framework .NET esta enfocado a:

- Proporcionar un entorno coherente de programación orientada a objetos, en el que el código de los objetos se pueda almacenar y ejecutar de forma local, ejecutar de forma local pero distribuida en Internet o ejecutar de forma remota. (Microsoft, MSDN)
- Proporcionar un entorno de ejecución de código que reduzca lo máximo posible la implementación de software y los conflictos de versiones.
- Ofrecer un entorno de ejecución de código que fomente la ejecución segura del mismo, incluso del creado por terceras personas desconocidas o que no son de plena confianza.
- Proporcionar un entorno de ejecución de código que elimine los problemas de rendimiento de los entornos en los que se utilizan secuencias de comandos o intérpretes de comandos.
- Ofrecer al programador una experiencia coherente entre tipos de aplicaciones muy diferentes, como las basadas en Windows o en el Web. Basar toda la comunicación en estándares del sector para asegurar que el código de .NET Framework se puede integrar con otros tipos de código.

El Framework .NET proporciona un entorno de implementación altamente distribuido que favorece la creación de aplicaciones robustas y escalables. Este entorno está compuesto por el CLR<sup>9</sup> y la biblioteca de clases del Framework .NET que reúne un gran conjunto de código orientado a objetos y reutilizable, que se puede emplear tanto en la construcción de aplicaciones como de servicios web.

---

<sup>9</sup> CLR: Common Language Runtime, constituye el fundamento de la tecnología .NET (Microsoft, MSDN).

### **Propiedades de la plataforma .NET.**

**Portabilidad:** es la capacidad que posee un sistema de ser utilizado en varias plataformas. Microsoft continúa con la voluntad de apoyar a su sistema Windows, por lo que Microsoft .NET funciona solamente en plataformas basadas en Win32.

**Escalabilidad:** es la propiedad que posee un sistema de incrementar sus servicios en función del aumento del número de usuarios que lo utilicen. Microsoft .Net ofrece métodos de escalabilidad como la carga balanceada, que permite colaborar a varios servidores y satisfacer la petición de un servicio de manera simultánea.

Microsoft .NET ofrece una solución más barata, con mayor rendimiento, escalabilidad y más fácil de implantar. Tiene en su contra recibir el apoyo de una única empresa – aunque ésta sea Microsoft -, no tener la suficiente madurez en el mercado por lo que no está bastante probado y la falta de portabilidad. (Martín, 2005)

### **1.2.2 Lenguaje de programación.**

#### **Lenguaje C#.**

C# combina los mejores elementos de múltiples lenguajes de amplia difusión como C++, Java, Visual Basic o Delphi (J.Guapo). Con ello quiere aprovechar el alto poder de abstracción de C++ y C con sus amplias posibilidades de aprovechamiento al más bajo nivel. Por otro lado, también pretende limar algunas deficiencias que posee Java en cuanto a portabilidad y eficiencia.

Microsoft puso gran énfasis en que la migración de los programadores a este lenguaje, que forma parte de su plataforma .NET y que está ampliamente sustentado por el Framework .NET, además de ser considerado el lenguaje nativo de la plataforma .NET (Seco, 2007), fuera lo más fácil posible. Debido a ello la sintaxis y estructuración del código en C# es semejante al de C++, mientras que su sencillez y alto nivel de productividad se igualan al de Visual Basic.

#### **Características del lenguaje C#.**

Eficiencia: Restringe el uso de punteros a la necesidad del programador e incluye un conjunto de restricciones que brinda mayor seguridad al código.

Modernidad: C# incorpora en el propio lenguaje elementos que a lo largo de los años ha ido demostrándose son muy útiles para el desarrollo de aplicaciones y que en otros lenguajes como Java o C++ hay que simular.

Orientado a objetos: Como todo lenguaje de programación de propósito general actual, C# es un lenguaje orientado a objetos. Una diferencia de este enfoque orientado a objetos respecto al de otros lenguajes como C++ es que el de C# es más puro, lo que reduce problemas por conflictos de nombres y facilita la legibilidad del código.

Orientado a componentes: La propia sintaxis de C# incluye elementos propios del diseño de componentes que otros lenguajes tienen que simular mediante construcciones más o menos complejas.

Seguridad de tipos: C# incluye mecanismos que permiten asegurar que los accesos a tipos de datos siempre se realicen correctamente.

Gestión automática de memoria: Todo lenguaje de .NET tiene a su disposición el recolector de basura del CLR. Esto tiene el efecto en el lenguaje de que no es necesario incluir instrucciones de destrucción de objetos.

### 1.2.3 Arquitectura.

#### **Programación multicapas.**

En la actualidad se busca con cada software mayor eficiencia y rendimiento, de ahí que en el proceso de desarrollo del mismo se tengan en cuenta cada día más los modelos de desarrollo que permitan la organización de los elementos de la aplicación en partes independientes.

Para alcanzar lo anterior se seguirán los fundamentos de la programación en múltiples capas, debido a que además de brindar mayor rapidez en las conexiones cliente-servidor, permite administrar los componentes de manera independiente.

#### **Arquitectura de 3 capas.**

El fundamento de esta arquitectura lo encontramos en la frase “divide y vencerás”, y se debe a que su aplicación esta muy ligada a la solución de un problema mayor a partir de su descomposición en problemas mas pequeños y de menor complejidad y es de destacar que el éxito, o no, de esta arquitectura estará dado por donde se haya decidido dividir el problema.

Para que una arquitectura de software sea considerada buena debe satisfacer principios como el de reusabilidad, mantenimiento, escalabilidad y robustez. Cuando diseñamos o implementamos un sistema en capas debemos garantizar que entre ellas exista la menor dependencia o cohesión posible, así como que cada una de ellas se ocupe de un conjunto de responsabilidades que posean un alto grado de acoplamiento. Al lograr esto, se garantiza que al tener que modificar alguna responsabilidad o servicio en cualquier capa, se afecte lo menos posible la estructura de las demás y la integridad del sistema.

La arquitectura de 3 capas (ver Fig. 1) plantea la división en las siguientes capas:

Presentación: presenta el sistema al usuario, se comunica únicamente con la capa de negocio.

Negocio: es donde se ejecutan las subrutinas y se establecen todas las reglas que deben cumplirse, se comunica con la capa de presentación y con la capa de datos.

Datos: en esta capa se programa todo lo que tiene que ver con el acceso a la base de datos, es la encargada de tomar la información de la base de datos dada una petición de la capa del Negocio, que a su vez es generada por la capa de Presentación.



Figura 1. 1 Arquitectura de 3 capas (D.E.V.Zapata, 2004).

### Ventajas de la arquitectura de 3 capas.

Permite la reutilización: la aplicación está formada por una serie de componentes que se comunican entre sí a través de interfaces y que cooperan para lograr el comportamiento deseado. Esto permite no solamente que estos componentes puedan ser fácilmente reemplazados por otros, por ejemplo porque se necesita mayor funcionalidad sino también que los mismos puedan ser utilizados para otras aplicaciones (Estrategia Magazine).

Acompaña el crecimiento: cada uno de los componentes de la aplicación pueden colocarse en el mismo equipo o distribuirse a través de una red. De esta manera, proyectos de gran envergadura pueden dividirse en pequeños proyectos más simples y manejables, que se pueden implementar en forma progresiva, agregando nuevos servicios según la medida de crecimiento.

Uso eficiente del hardware: debido a que los componentes pueden ser distribuidos a través de toda la red, se puede hacer un uso más eficiente de los recursos de hardware. En vez de necesitarse grandes servidores que contengan la lógica de negocios y los datos, es posible distribuirlos en varias máquinas más pequeñas, económicas y fáciles de ser reemplazadas.

Encapsula los datos: debido a que las aplicaciones cliente se comunican con los datos a través de peticiones que los servidores responden ocultando y encapsulando los detalles de la lógica de la aplicación, obtenemos un nivel de abstracción que permite un acceso a los datos consistente, seguro y auditable. Con esto se pretende que si hay cambios en la capa de datos, la capa de negocios se haga cargo de administrar tales cambios y el cliente, en la mayor parte de los casos ni se entere.

Mayor calidad en las aplicaciones: como las aplicaciones son construidas en unidades separadas, estas pueden ser probadas independientemente y con mucho más detalle, esto conduce a obtener un producto mucho más sólido.

### **1.2.4 Proceso de desarrollo de software.**

Una metodología está constituida por una serie de pasos ordenados que se han de seguir para cumplir un objetivo. El objetivo de la Ingeniería de Software es el desarrollo de software de alta calidad que satisfaga las necesidades del cliente, por lo que una metodología de ingeniería de software es un conjunto ordenado de pasos a seguir para desarrollar software de alta calidad que cumpla con las necesidades del usuario.

A la hora de seleccionar una metodología de desarrollo es necesario saber que características deben poseer estas para que sean factibles y aplicables en un proyecto.

### **Criterios de selección de metodologías de desarrollo de software.**

- Debe ajustarse a los objetivos: cada aproximación al desarrollo de software está basada en unos objetivos. Por ello la metodología que se elija debe recoger el aspecto filosófico de la

aproximación deseada, es decir que los objetivos generales del desarrollo deben estar implementados en la metodología de desarrollo (Creative Commons, 2007).

- Debe cubrir el ciclo entero de desarrollo del software: para ello la metodología debe cubrir todas las etapas de desarrollo del software.
- Debe integrar las distintas fases del ciclo de desarrollo.
- Debe incluir la realización de validaciones: la metodología debe detectar y corregir los errores cuanto antes. Uno de los problemas más frecuentes y costosos es el aplazamiento de la detección y corrección de problemas en las etapas finales del proyecto. Cuanto más tarde sea detectado el error más caro será corregirlo.
- Debe especificar claramente los responsables de los resultados: debe especificar claramente quienes son los participantes de cada tarea a desarrollar, debe detallar de una manera clara los resultados de los que serán responsables.
- Debe poder enseñarse: cada integrante del equipo de desarrollo debe entender las técnicas específicas de la metodología, los procedimientos organizativos y de gestión que la hacen efectiva, las herramientas automatizadas que soportan la metodología y las motivaciones que subyacen en ella.
- Debe estar soportada por herramientas CASE<sup>10</sup>: la metodología debe estar soportada por herramientas automatizadas que mejoren la productividad, tanto del ingeniero de software en particular, como la del desarrollo en general. El uso de estas herramientas reduce el número de personas requeridas y la sobrecarga de comunicación, además de ayudar a producir especificaciones y diseños con menos errores, más fáciles de probar, modificar y usar.

---

<sup>10</sup> CASE: Computer Aided Software Engineering (Ingeniería de Software Asistida por Ordenador).

### 1.2.5 Metodologías de desarrollo de software.

Actualmente existen una gran variedad de metodologías de desarrollo de software, divididas en 3 tipos de clasificaciones: ágiles, flexibles y robustas. Luego de hacer un estudio de cada tipo se ha seleccionado un ejemplo de cada uno: XP<sup>11</sup>, MSF<sup>12</sup>, RUP<sup>13</sup>.

#### Extreme Programming (XP).

Es una metodología que se aplica con éxito en la actualidad, constituyendo una de las de mejores resultados en proyectos no muy complejos, con plazo de entrega reducido y con escaso equipo de trabajo.

Consiste en realizar una programación rápida teniendo como particularidad que el cliente participa constantemente en el proceso de desarrollo del producto, considerando que es necesaria su participación para que la metodología tenga éxito.

Posee las siguientes características:

- Programación en pares: esta metodología propone que dos desarrolladores participen en el proyecto desde la misma estación de trabajo.
- Pruebas unitarias: propone realizar pruebas a los principales procesos tratando de adelantar en algo las fallas que pudieran ocurrir en el futuro para corregirlas.
- Refabricación: se crean patrones o modelos estándares que faciliten la reutilización de código, posibilitando mayor flexibilidad en el cambio.

XP propone:

- Empieza en pequeño y añade funcionalidad con retroalimentación continua.
- El manejo del cambio se convierte en parte sustantiva del proceso.
- El costo del cambio no depende de la fase o etapa.

---

<sup>11</sup> XP: Extreme Programming.

<sup>12</sup> MSF: Microsoft Solution Framework.

<sup>13</sup> RUP: Rational Unified Process.

- No introduce funcionalidades antes que sean necesarias.

Lo fundamental en este tipo de metodología es la comunicación entre los usuarios y los desarrolladores, la simplicidad al desarrollar y codificar los módulos del sistema y la retroalimentación concreta y frecuente del equipo de desarrollo, el cliente y los usuarios finales (A.Mendoza, 2004).

### **Microsoft Solution Framework (MSF).**

Es una metodología flexible y que está estrechamente ligada a un conjunto de modelos, conceptos y prácticas de uso que permiten gestionar, planificar y controlar los proyectos tecnológicos.

MSF se centra en los modelos de equipo y de proceso y deja en un segundo plano las elecciones tecnológicas. Entre sus principales características se encuentran:

- Adaptable.
- Escalable: puede organizar equipos tan pequeños como de 3 o 4 personas, así como también, proyectos que requieren 50 personas ó más.
- Flexible: es utilizada en el ambiente de desarrollo de cualquier cliente.
- Tecnología agnóstica: porque puede ser usada para desarrollar soluciones basadas sobre cualquier tecnología.

MSF se compone de varios modelos encargados de planificar las diferentes partes implicadas en el desarrollo de un proyecto: Modelo de Arquitectura del Proyecto, Modelo de Equipo, Modelo de Proceso, Modelo de Gestión del Riesgo, Modelo de Diseño de Proceso y finalmente el modelo de Aplicación (A.Mendoza, 2004).

- Modelo de arquitectura del proyecto: diseñado para acortar la planificación del ciclo de vida.
- Modelo de equipo: diseñado para mejorar el rendimiento del equipo de desarrollo.
- Modelo de proceso: diseñado para mejorar el control del proyecto, minimizando el riesgo, y aumentar la calidad acortando el tiempo de entrega.
- Modelo de gestión del riesgo: diseñado para ayudar al equipo a identificar las prioridades, tomar las decisiones estratégicas correctas y controlar las emergencias que puedan surgir.
- Modelo de diseño del proceso: diseñado para distinguir entre los objetivos empresariales y las necesidades del usuario.

- Modelo de aplicación: diseñado para mejorar el desarrollo, el mantenimiento y el soporte, proporciona un modelo de tres niveles para diseñar y desarrollar aplicaciones software. Los servicios utilizados en este modelo son escalables, y pueden ser usados en un solo ordenador o incluso en varios servidores (A.Mendoza, 2004).

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

RUP es una de las metodologías de uso general y más ampliamente usada que existen en la actualidad. Constituye además una de las más fuertes propuestas para el desarrollo orientado a objetos usando UML<sup>14</sup>.

Se caracteriza por:

- Guiado por casos de uso: los casos de uso son el instrumento para validar la arquitectura del software y extraer los casos de prueba.
- Iterativo e incremental: durante todo el proceso de desarrollo se producen versiones incrementales del producto en desarrollo.
- Centrado en la arquitectura: los modelos son proyecciones del análisis y el diseño que constituyen la arquitectura del producto a desarrollar.

RUP propone 4 fases:

- Inicio: el objetivo en esta etapa es determinar la visión del proyecto.
- Elaboración: en esta etapa el objetivo es determinar la arquitectura óptima.
- Construcción: en esta etapa el objetivo es obtener la capacidad operacional inicial.
- Transición: el objetivo es llegar a obtener el *release* del proyecto.

Las fases de inicio y elaboración incluyen las actividades de diseño del ciclo de vida del desarrollo, donde se hace énfasis en el modelado del negocio, requisitos, análisis y diseño. En las fases de construcción y transición el mayor volumen de trabajo se realiza sobre las actividades de implementación, pruebas y despliegue.

---

<sup>14</sup> UML: Unified Model Language (Lenguaje Unificado de Modelado).

Una particularidad de esta metodología es que, en cada ciclo de iteración, se hace exigente el uso de artefactos, siendo por este motivo, una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software (A. Mendoza, 2004).

Luego de analizadas las características fundamentales de cada metodología y teniendo en cuenta que, uno de los factores decisivos que contribuye al éxito de un proyecto de software son las personas calificadas que intervienen durante todo el ciclo de vida del producto, y éstas a su vez, regidas por una metodología capaz de llevar a cabo un satisfactorio desarrollo del mismo (Jacobson, 2000), lo cual es facilitado por RUP, se ha adoptado esta metodología para el modelado y documentación de este sistema.

### **1.2.6 UML como lenguaje de modelación visual.**

El Lenguaje Unificado de Modelado es un lenguaje de modelado visual, muy usado actualmente, utilizado para especificar, documentar y construir artefactos de un sistema de software. Se usa para entender, hojear, controlar, configurar y mantener información sobre el sistema en desarrollo.

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

Entre sus características destacan:

- Permite el modelado orientado a objetos.
- Puede conectarse con lenguajes de programación (Ingeniería directa e inversa).
- Permite documentar todos los artefactos de un proceso de desarrollo (requisitos, arquitectura, pruebas, versiones, etc.).
- Es independiente del proceso, aunque para utilizarlo óptimamente se debería usar en un proceso que fuese dirigido por los casos de uso, centrado en la arquitectura, iterativo e incremental.
- Existe un equilibrio entre expresividad y simplicidad, pues no es difícil de aprender ni de utilizar.

### **1.2.7 Herramientas CASE para la modelación del sistema.**

Las herramientas CASE son consideradas como la aplicación de tecnología informática a las actividades, las técnicas y las metodologías propias de desarrollo. Su objetivo es acelerar el proceso para el

que han sido diseñadas, en este caso, para automatizar o apoyar una o más fases del ciclo de vida del desarrollo de sistemas (Kendall).

La tecnología CASE supone la automatización del desarrollo del software, contribuyendo a mejorar la calidad y la productividad en el desarrollo de sistemas informáticos y se plantean los siguientes objetivos:

- Permitir la aplicación práctica de metodologías estructuradas, con las cuales, al ser realizadas con una herramienta se consigue agilizar el trabajo.
- Facilitar la realización de prototipos y el desarrollo conjunto de aplicaciones.
- Simplificar el mantenimiento de los programas.
- Mejorar y estandarizar la documentación.
- Aumentar la portabilidad de las aplicaciones.
- Facilitar la reutilización de componentes de software.

Para el modelado de este sistema se ha realizado la comparación entre Visual Paradigm y Rational Rose.

### **Visual Paradigm.**

Visual Paradigm ofrece un conjunto de facilidades para el modelado de sistemas informáticos. Tiene la deficiencia de que sus ficheros no pueden exceder el tamaño de 1 MB, algo muy difícil de alcanzar para un proyecto de gran magnitud. Es de gran utilidad para desarrollo sobre lenguajes como PHP pues posee autogeneración de código en dicho lenguaje (2005).

### **Rational Rose Enterprise Edition de la Suite 2003.**

Rational Rose Enterprise Edition permite especificar, analizar y diseñar el sistema antes de codificarlo.

Las herramientas Rational unifican el equipo de proyecto y esta forma parte del conjunto más amplio de herramientas que juntas cubren todo el ciclo de vida del desarrollo de software. Permite completar una gran parte de las disciplinas, ó flujos fundamentales, del RUP e incluye un conjunto de herramientas de ingeniería inversa y generación de código que facilitan el proceso de desarrollar un producto final de buena calidad.

Como herramienta de modelado se ha adoptado Rational Rose debido a que constituye la herramienta líder para este propósito, además de contar con facilidades de autogeneración de código, así como de ingeniería inversa, admite notaciones en UML y la integración con el modelado de datos.

### **1.3 Conclusiones.**

En este capítulo se realizó un estudio profundo de los sistemas vinculados al campo de acción de este trabajo.

Se analizaron algunas de las herramientas, lenguaje de modelado y metodología de desarrollo más usadas actualmente en el desarrollo de aplicaciones, valorando sus ventajas y desventajas, y se llegó a la conclusión de que la propuesta consiste en desarrollar una aplicación usando la metodología RUP; UML como lenguaje para describir todo el proceso de desarrollo; Rational Rose Enterprise Edition 2003 como herramienta de modelado; C# como lenguaje de programación y para el soporte del mismo el entorno de desarrollo que brinda Visual Studio 2005.

## Capítulo II: Características del Sistema.

### **Introducción.**

En este capítulo quedarán plasmadas las funcionalidades con que contará la solución propuesta, a partir de la especificación de los requerimientos funcionales. Se hará referencia al flujo de trabajo Modelamiento del Negocio y se describirá de forma detallada cada caso de uso.

### **2.1 Modelo de Dominio.**

Después de un profundo análisis se llegó a la conclusión de que no se realizaría el modelamiento del negocio a través de su diagrama de casos de uso debido a que no fue posible identificar claramente los procesos del negocio. Solo se han podido identificar conceptos relacionados con la administración y el control de los módulos del sistema.

Es por ello que se ha determinado realizar un Modelo de Dominio, el cual representa un conjunto de conceptos relacionados entre si.

En el siguiente Modelo de Dominio se especifican las relaciones que existen entre los principales conceptos que se gestionan en el Módulo de Administración del Sistema Integrador de Gestión de Vuelo para Aviones Il-96-300 y que están estrechamente ligados con los demás módulos de este sistema.

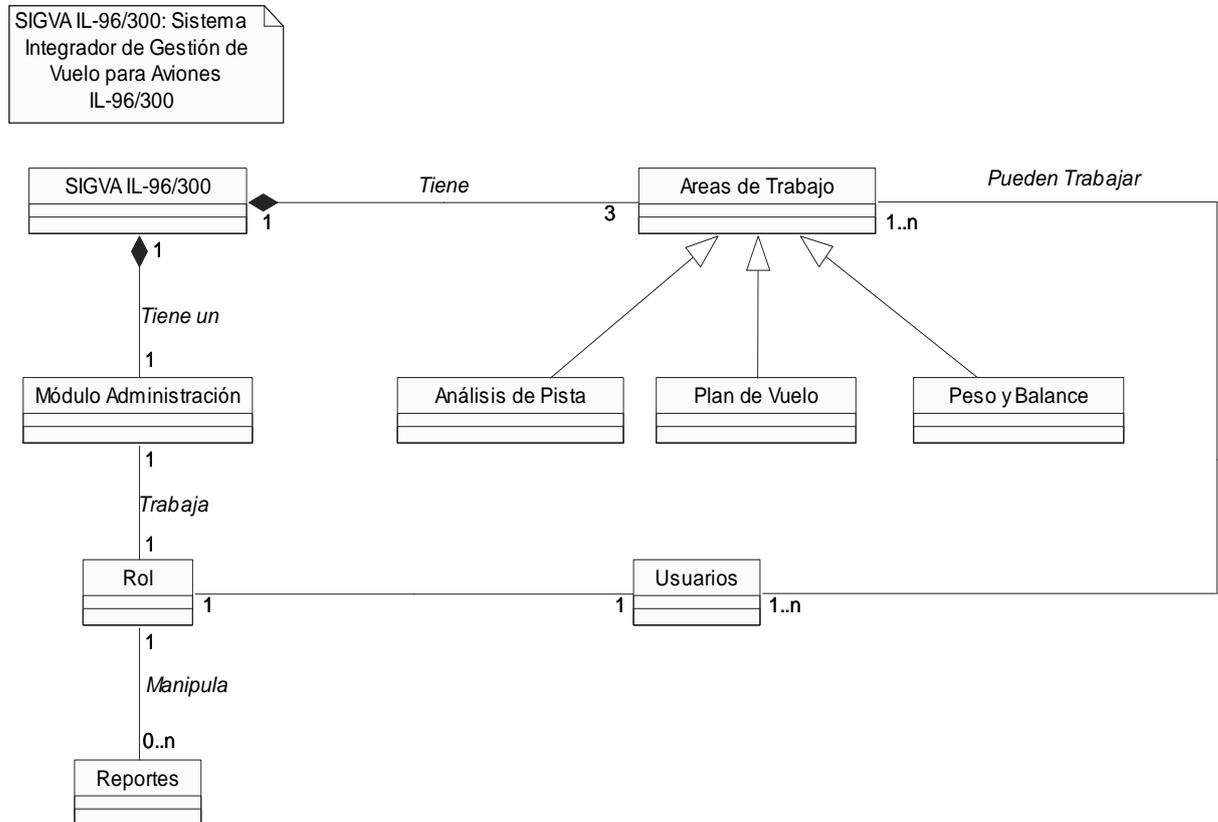


Figura 2. 1 Modelo de Dominio

### Glosario de conceptos del Modelo de Dominio.

- **SIGVA IL-96/300:** Sistema Integrador de Gestión de Vuelo para Aviones IL-96/300, cuenta con 3 áreas de trabajo y está dirigido ó controlado por un módulo de administración.
- **Áreas de trabajo:** En ellas se llevan a cabo los procesos que abarcan las operaciones realizadas dentro del sistema. Estos procesos son el cálculo del Peso y Balance del avión, la gestión del Plan de Vuelo y el Análisis de Pista.
- **Módulo de Administración:** Es un módulo que tiene el SIGVA IL-96/300 para el control de las operaciones realizadas en el sistema y el acceso de los usuarios al mismo.
- **Reportes:** Documentos manipulados y generados por el administrador del sistema.

- **Usuario:** Toda persona que tiene acceso al sistema.
- **Rol:** Es un concepto relacionado con la autorización de los usuarios para la utilización de las funcionalidades que brinda el sistema. Es gestionado a partir del Módulo de Administración.
- **Plan de Vuelo:** Área de trabajo en la que se realizan los cálculos para la confección del Plan de Vuelo del avión IL-96/300.
- **Peso y Balance:** Área de trabajo en la que se realizan los cálculos de Peso y Balance del avión IL-96/300.
- **Análisis de Pista:** Área de trabajo en la que se realizan los cálculos necesarios para el Análisis de Pista.

## 2.2 Especificación de los requerimientos.

### Requerimientos funcionales.

Los Requerimientos funcionales especifican acciones que el sistema debe ser capaz de realizar, sin tomar en consideración ningún tipo de restricción física. Es decir, especifican el comportamiento de entrada y salida del sistema y surgen de la razón fundamental de la existencia del producto.

- **RF 1:** Autenticar usuario.
- **RF 2:** Gestionar usuario.
  - RF 2.1:** Adicionar usuario.
  - RF 2.2:** Eliminar usuario.
  - RF 2.3:** Modificar usuario.
  - RF 2.4:** Listar usuarios.
  - RF 2.5:** Cambiar contraseña.
  - RF 2.6:** Asignar rol.
- **RF 3:** Gestionar historial.

**RF 3.1:** Limpiar historial.

**RF 3.1.1:** Limpiar historial completo.

**RF 3.1.2:** Limpiar historial por fecha.

**RF 3.1.3:** Limpiar historial entre fechas.

**RF 3.1.4:** Limpiar historial por usuario.

**RF 3.2:** Mostrar historial.

**RF 3.2.1:** Mostrar historial completo.

**RF 3.2.2:** Mostrar historial por fecha.

**RF 3.2.3:** Mostrar historial entre fechas.

**RF 3.2.4:** Mostrar historial por usuario.

➤ **RF 4:** Gestionar rol.

**RF 4.1:** Adicionar rol.

**RF 4.2:** Listar roles.

**RF 4.2:** Eliminar rol.

**RF 4.3:** Modificar rol.

### **Requerimientos no funcionales.**

Los requerimientos no funcionales son las propiedades o cualidades que el producto debe tener para que este sea atractivo, usable, rápido y confiable.

➤ **Apariencia o interfaz externa:** El diseño de la interfaz externa del sistema debe ser profesional, sencillo, amigable, de fácil transición, familiar a los usuarios que han usado otras aplicaciones de escritorio en Windows, con el fin de lograr una eficiente interacción con el usuario, pro-

porcionándole en todo momento una sensación de control sobre la aplicación. Para la construcción del sistema se deben seguir las normas convencionales de interfaz de usuario de Windows. Se debe mantener informado al usuario acerca de todo lo que sucede en la aplicación, los mensajes de esta deben estar dirigidos al usuario y por tanto, redactados en su idioma. La información se debe presentar de forma clara.

- **Usabilidad:** El sistema está concebido para ser usado por la tripulación del avión IL\_96 300 por lo tanto la dificultad dependerá del número de pasos, el conocimiento que el usuario debe tener del proceso y las decisiones que este debe tomar en cada paso. Para evitar errores, en los campos que se requiera el sistema brindará la opción de elegir el valor deseado en vez de que el usuario introduzca los datos.

Se seguirán las guías de Interfaz de Usuario para realizar la interfaz.

Se debe informar al usuario en todo momento acerca de lo que sucede en la aplicación por lo que los mensajes deben ser evidentes y personalizados.

- **Rendimiento:** El sistema operará con grandes volúmenes de información, por tanto, se hacen necesarios tiempos de respuestas cortos, al igual que la velocidad de procesamiento de la información.
- **Seguridad y privacidad:** Se mantendrá seguridad y control a nivel de usuarios y contraseñas, garantizando el acceso de los mismos sólo a los niveles establecidos de acuerdo a la función que realizan. Las contraseñas sólo podrán ser cambiadas por el administrador del sistema.

El sistema permitirá registrar todas las acciones que se realizan, llevando el registro de las actividades de cada usuario en todo momento.

- **Soporte:** Debe ser de fácil instalación y mantenimiento. Se debe permitir la inserción de nuevos módulos, sin negar lo realizado o afectar el buen funcionamiento del sistema.

El sistema debe ser sometido a una etapa de prueba donde se realice el adiestramiento de los usuarios, permitiéndole a este familiarizarse con el software y a la vez se puedan detectar posibles errores o posibles cambios en las interfaces de manera que queden complacidos.

- **Portabilidad:** El sistema funcionará sobre plataforma Windows, porque es la exigida por el cliente.
- **Ayuda y documentación:** El sistema contará con la documentación completa de todas las tareas y operaciones que realiza el software, el glosario de términos y las planillas que especifican toda la Ingeniería de Software.

Además poseerá una ayuda que garantice el asesoramiento e información al usuario acerca de los contenidos tratados; donde quedarán claramente reflejadas las funcionalidades del sistema y su manipulación. También contendrá los datos del contacto para negociar futuros mantenimientos del software y solucionar los problemas que se puedan presentar.

El sistema requiere la construcción de un Manual de Usuario que describa detalladamente sus características y uso.

### 2.3 Actores del Sistema

Actores del Sistema	Descripción
Tripulación	Son usuarios del sistema que se autentican y reciben los permisos según los roles que tienen asignados.
Administrador	Es el encargado de gestionar toda la información referente a los usuarios del sistema, historiales, áreas de trabajo, roles y permisos.

## 2.4 Modelo de Casos de Uso del Sistema.

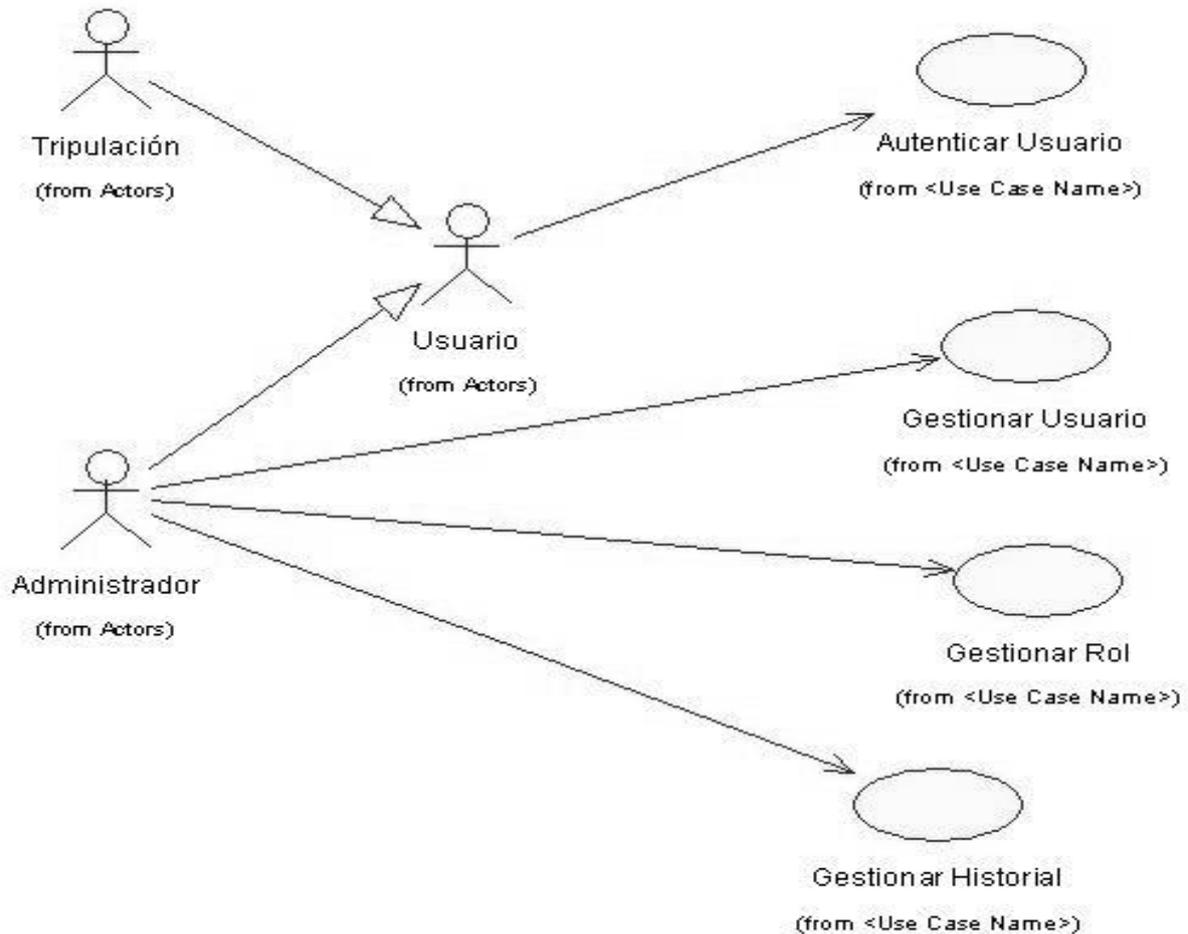


Figura 2. 2 Diagrama de Casos de Uso del Sistema.

## 2.5 Descripción de los Casos de Uso del Sistema.

### Caso de Uso: Autenticar Usuario.

<b>Caso de Uso:</b>	Autenticar Usuario.
<b>Actores:</b>	Administrador, Tripulación.
<b>Propósito:</b>	Permitir el acceso al sistema, otorgando los permisos según los roles que posean los usuarios.

<b>Resumen:</b>	El caso de uso se inicia cuando el administrador o algún miembro de la tripulación introduce en el sistema su usuario y contraseña, el sistema verifica si los datos son válidos y en caso de ser así, se ejecuta la aplicación con las funcionalidades correspondiente a los permisos y roles del usuario autenticado.
<b>Referencias:</b>	RF 1
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- El usuario inicia la aplicación.	1.1 El sistema muestra el formulario de autenticación, solicitando usuario y contraseña.
2- El usuario introduce los datos solicitados.	
3- El usuario presiona el botón Aceptar.	3.1 El sistema verifica que los datos sean correctos y que el usuario posee un rol en un área determinada y se ejecuta la aplicación.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
Acción 3 El usuario presiona el botón Cancelar.	Acción 3.1 El sistema determina que los datos no son correctos, muestra un mensaje de error y solicita introducir usuario y contraseña nuevamente o contactar al administrador.
<b>Poscondiciones:</b>	Usuario Autenticado.

**Caso de Uso: Gestionar Usuario.**

<b>Caso de Uso:</b>	Gestionar Usuario.	
<b>Actores:</b>	Administrador.	
<b>Propósito:</b>	Permitir al administrador insertar, modificar, eliminar, listar, cambiar contraseña y asignar rol a un usuario.	
<b>Resumen:</b>	El caso de uso inicia cuando el administrador selecciona la opción para la gestión de usuario.	
<b>Referencias:</b>	RF 1, RF 2.	
<b>Precondiciones:</b>	Administrador autenticado.	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1- El administrador selecciona la opción Gestionar Usuario del menú.	1.1 El sistema despliega en el menú las opciones referentes a las operaciones sobre los usuarios del sistema. <ul style="list-style-type: none"> <li>a) Adicionar Usuario (Ir a sección Adicionar Usuario).</li> <li>b) Modificar Usuario (Ir a sección Modificar Usuario).</li> <li>c) Cambiar Contraseña (Ir a sección Cambiar Contraseña).</li> <li>d) Eliminar Usuario (Ir a sección Eliminar Usuario).</li> <li>e) Asignar Rol (Ir a sección Asignar Rol).</li> </ul>	
<b>Sección 1: "Adicionar Usuario"</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
2- El administrador introduce los datos solicitados (usuario, contraseña, nom-		

bre, primer apellido, segundo apellido y licencia).	
3- El administrador presiona el botón Adicionar Usuario.	3.1 El sistema comprueba la inserción de los campos obligatorios y muestra mensaje de éxito.
<b>Flujos alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
Acción 3 El administrador presiona el botón Cancelar.	Acción 3.1 El sistema sale de la interfaz Adicionar Usuario.
<b>Sección 2: "Modificar Usuario"</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	4- El sistema muestra un listado de todos los usuarios existentes.
4.1- El administrador selecciona el usuario a modificar.	
4.2- El administrador presiona el botón Seleccionar.	5- El sistema muestra todos los datos del usuario seleccionado.
5.1 – El administrador edita los campos deseados.	
5.2 – El administrador presiona el botón Modificar Usuario.	6- El sistema actualiza los datos del usuario y muestra mensaje de éxito.
<b>Flujos alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>

Acción 5.2- El administrador presiona el botón Cancelar.	Acción 6- El sistema sale de la interfaz Modificar Usuario.
<b>Sección 3: “Cambiar Contraseña”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	7- El sistema muestra un listado de todos los usuarios existentes.
7.1- El administrador selecciona el usuario para cambiar su contraseña.	
7.2- El administrador presiona el botón Seleccionar.	8 – El sistema muestra los campos para cambiar la contraseña.
8.1 – El administrador introduce los datos solicitados.	
8.2 – El administrador presiona el botón Cambiar Contraseña.	9- El sistema actualiza la contraseña del usuario y se muestra mensaje de éxito.
<b>Flujos alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
Acción 8.2- El administrador presiona el botón Cancelar.	Acción 9 – El sistema sale de la interfaz Cambiar Contraseña.
<b>Sección 4: “Eliminar Usuario”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	10- El sistema muestra un listado de todos los usuarios existentes.

10.1- El administrador selecciona el usuario a eliminar.	
10.2 – El administrador presiona el botón Eliminar Usuario.	11 - El sistema elimina el usuario seleccionado y muestra mensaje de éxito.
<b>Flujos alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
Acción 10.2- El administrador presiona el botón Cancelar.	Acción 11- El sistema sale de la interfaz Eliminar Usuario.
<b>Sección 5: “Asignar Rol”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	12- El sistema muestra un listado de todos los usuarios que no tienen un rol asignado.
12.1- El administrador selecciona el usuario para asignarle un rol.	
12.2- El administrador presiona el botón Seleccionar.	13- El sistema muestra las áreas y los roles disponibles en cada una de ellas.
13.1 – El administrador selecciona un área y un rol.	
13.2- El administrador presiona el botón Asignar Rol.	14 – El sistema asigna rol al usuario y muestra mensaje de éxito.
<b>Poscondiciones:</b>	<p>Sección 1: Se adiciona un nuevo usuario.</p> <p>Sección 2: Se modifica un usuario existente.</p> <p>Sección 3: Se establece una nueva contraseña a un usuario existente.</p>

	<p>Sección 4: Se elimina un usuario existente.</p> <p>Sección 5: Se asigna un rol a un usuario.</p>
--	---

### Caso de Uso: Gestionar Rol.

<b>Caso de Uso:</b>	Gestionar Rol.	
<b>Actores:</b>	Administrador.	
<b>Propósito:</b>	Permitir al administrador insertar, modificar, listar y eliminar los roles	
<b>Resumen:</b>	El caso de uso inicia cuando el administrador selecciona la opción para la gestión de roles.	
<b>Referencias:</b>	RF 1, RF 4	
<b>Precondiciones:</b>	Administrador autenticado.	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1- El administrador selecciona la opción Gestionar Rol del menú.	1.1 El sistema despliega en el menú las opciones referentes a las operaciones sobre los roles. <ul style="list-style-type: none"> <li>a) Adicionar Rol (Ir a sección Adicionar Rol).</li> <li>b) Modificar Rol (Ir a sección Modificar Rol).</li> <li>c) Eliminar Rol (Ir a sección Eliminar Rol).</li> </ul>	
<b>Sección 1: "Adicionar Rol"</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
2- El administrador introduce los datos solicitados (nombre, descripción, per-		

miso).	
3- El administrador presiona el botón Adicionar Rol.	3.1 El sistema comprueba que los datos insertados estén en el formato correcto y muestra mensaje de éxito.
<b>Flujos alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
Acción 3 El administrador presiona el botón Cancelar.	Acción 3.1 El sistema sale de la interfaz Adicionar Rol.
<b>Sección 2: "Modificar Rol"</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	4- El sistema muestra un listado de todos los roles existentes.
4.1- El administrador selecciona el rol a modificar.	
4.2- El administrador presiona el botón Seleccionar.	5- El sistema muestra todos los datos del rol seleccionado.
5.1 – El administrador edita los campos deseados.	
5.2 – El administrador presiona el botón Modificar Rol.	6- El sistema actualiza los datos del rol y muestra mensaje de éxito.
<b>Flujos alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>

Acción 5.2- El administrador presiona el botón Cancelar.	Acción 6- El sistema sale de la interfaz Modificar Rol.
<b>Sección 3: "Eliminar Rol"</b>	
Acción del Actor	Respuesta del Sistema
	7- El sistema muestra un listado de todos los roles existentes.
7.1- El administrador selecciona el rol a eliminar.	
7.2 – El administrador presiona el botón Eliminar Rol.	8 - El sistema elimina el rol seleccionado y muestra mensaje de éxito.
<b>Flujos alternos</b>	
Acción del Actor	Respuesta del Sistema
Acción 7.2- El administrador presiona el botón Cancelar.	Acción 8- El sistema sale de la interfaz Eliminar Usuario.
<b>Poscondiciones:</b>	Sección 1: Se adiciona un nuevo rol. Sección 2: Se modifica un rol existente. Sección 3: Se elimina un rol existente.

### Caso de Uso: Gestionar Historial.

<b>Caso de Uso:</b>	Gestionar Historial
<b>Actores:</b>	Administrador
<b>Propósito:</b>	Permitir al administrador gestionar el historial.

<b>Resumen:</b>	El caso de uso se inicia cuando el administrador solicita mostrar ó limpiar el historial según distintos criterios.	
<b>Referencias:</b>	RF 1, RF 3.	
<b>Precondiciones:</b>	Administrador autenticado.	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1- El administrador selecciona la opción Gestionar Historial del menú.	1.1 El sistema despliega en el menú las opciones referentes a las operaciones sobre el historial. <ul style="list-style-type: none"> <li>a) Mostrar Historial Completo (Ir a sección Mostrar Historial Completo).</li> <li>b) Mostrar Historial Por Fecha (Ir a sección Mostrar Historial Por Fecha).</li> <li>c) Mostrar Historial Entre Fechas (Ir a sección Mostrar Historial Entre Fechas).</li> <li>d) Mostrar Historial Por Usuario (Ir a sección Mostrar Historial Por Usuario).</li> <li>e) Limpiar Historial Completo (Ir a sección Limpiar Historial Completo).</li> <li>f) Limpiar Historial Por Fecha (Ir a sección Limpiar Historial Por Fecha).</li> <li>g) Limpiar Historial Entre Fechas (Ir a sección Limpiar Historial Entre Fechas).</li> <li>h) Limpiar Historial Por Usuario (Ir a sección Limpiar Historial Por Usuario).</li> </ul>	
<b>Sección 1: “Mostrar Historial Completo”</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	

2- El administrador selecciona la opción Mostrar Historial Completo.	2.1- El sistema muestra el historial completo, finalizando así el caso de uso.
<b>Flujos alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<b>Sección 2: “Mostrar Historial Por Fecha”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
3- El administrador selecciona la opción Mostrar Historial Por Fecha.	3.1- El sistema muestra una interfaz con los criterios de búsqueda de la opción seleccionada.
4- El administrador selecciona la fecha deseada.	
5- El administrador presiona el botón Ver Historial.	5.1- Muestra el historial.
<b>Flujos alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
Acción 5 El administrador presiona el botón Cancelar.	Acción 5.1 El sistema sale de la interfaz Mostrar Historial Por Fecha.
<b>Sección 3: “Mostrar Historial Entre Fechas”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
6- El administrador selecciona la opción Mostrar Historial Entre	6.1- El sistema muestra una interfaz con los criterios de búsqueda de la opción seleccionada.

Fechas.	
7- El administrador selecciona las fechas deseadas.	
8- El administrador presiona el botón Ver Historial.	8.1- Muestra el historial.
<b>Flujos alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
Acción 8 El administrador presiona el botón Cancelar.	Acción 8.1 El sistema sale de la interfaz Mostrar Historial Entre Fechas.
<b>Sección 4: “Mostrar Historial Por Usuario”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
9- El administrador selecciona la opción Mostrar Historial Por Usuario.	9.1- El sistema muestra una interfaz con los criterios de búsqueda de la opción seleccionada.
10-El administrador selecciona el usuario deseado.	
11-El administrador presiona el botón Ver Historial.	11.1- Muestra el historial.
<b>Flujos alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
Acción 11 El administrador presiona el botón Cancelar.	Acción 11.1 El sistema sale de la interfaz Mostrar Historial Por Usuario.
<b>Sección 5: “Limpiar Historial Completo”</b>	

<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
12-El administrador selecciona la opción Limpiar Historial Completo.	12.1- El sistema limpia el historial y muestra mensaje de éxito finalizando así el caso de uso.
<b>Flujos alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<b>Sección 6: "Limpiar Historial Por Fecha"</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
13-El administrador selecciona la opción Limpiar Historial Por Fecha.	13.1- El sistema muestra una interfaz con los criterios de búsqueda de la opción seleccionada.
14-El administrador selecciona la fecha deseada.	
15-El administrador presiona el botón Limpiar Historial.	15.1- Limpia el historial y muestra mensaje de éxito, finalizando así el caso de uso.
<b>Flujos alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
Acción 15 El administrador presiona el botón Cancelar.	Acción 15.1 El sistema sale de la interfaz Limpiar Historial Por Fecha.
<b>Sección 7: "Limpiar Historial Entre Fechas"</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
16-El administrador selecciona la opción Limpiar Historial Entre	16.1- El sistema muestra una interfaz con los criterios de búsqueda de la opción seleccionada.

Fechas.	
17-El administrador selecciona las fechas deseadas.	
18-El administrador presiona el botón Limpiar Historial.	18.1- Limpia el historial y muestra mensaje de éxito, finalizando así el caso de uso.
<b>Flujos alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
Acción 18 El administrador presiona el botón Cancelar.	Acción 18.1 El sistema sale de la interfaz Limpiar Historial Entre Fechas.
<b>Sección 8: “Limpiar Historial Por Usuario”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
19-El administrador selecciona la opción Limpiar Historial Por Usuario.	1.1- El sistema muestra una interfaz con los criterios de búsqueda de la opción seleccionada.
20-El administrador selecciona el usuario deseado.	
21-El administrador presiona el botón Limpiar Historial.	21.1- Limpia el historial y muestra mensaje de éxito, finalizando así el caso de uso.
<b>Flujos alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
Acción 21 El administrador presiona el botón Cancelar.	Acción 21.1 El sistema sale de la interfaz Limpiar Historial Por Usuario.
<b>Poscondiciones:</b>	Sección 1: Se muestra el Historial Completo.

	Sección 2: Se muestra el Historial Por Fecha.
	Sección 3: Se muestra el Historial Entre Fechas.
	Sección 4: Se muestra el Historial Por Usuario.
	Sección 5: Se limpia el Historial Completo.
	Sección 6: Se limpia el Historial Por Fecha.
	Sección 7: Se limpia el Historial Entre Fechas.
	Sección 8: Se limpia el Historial Por Usuario.

### 2.6 Conclusiones.

En este capítulo se hizo una descripción de las características del sistema a través de la confección del modelo de dominio a partir de los conceptos identificados; se plantearon los requisitos funcionales correspondiente al módulo y los no funcionales de la aplicación informática que se va a desarrollar y se modeló la misma en términos de casos de uso de sistema, de los cuales se realizó su descripción textual, se identificaron los actores y trabajadores que intervendrán en el sistema.

## Capítulo III: Análisis y Diseño del Sistema.

### Introducción.

En este capítulo se muestran los resultados obtenidos en el desarrollo del flujo de trabajo Análisis y Diseño del Sistema, utilizando para su modelado el Lenguaje Unificado de Modelado (UML), donde se elaborarán los diagramas de clases del Análisis y del Diseño, así como los diagramas de interacción que fueron necesarios utilizar para obtener una mayor claridad a la hora de elaborar la solución que se propone.

### 3.1 Análisis.

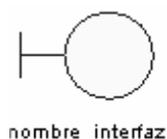
El objetivo del análisis es comprender perfectamente los requisitos del software y no precisar cómo se implementará la solución, por esto no se toman en cuenta el lenguaje de programación a usar en la construcción, la plataforma en la que se ejecutará la aplicación, los componentes prefabricados o re-usables de otras aplicaciones, entre otras características que afectan al sistema. El análisis consiste en obtener una visión del sistema que sólo se interesa por los requisitos funcionales.

Para la elaboración de los diagramas de clases del análisis se han utilizado tres tipos de clases:

- Entidad: Para modelar información que posee larga vida y que es a menudo persistente.



- Interfaz: Para modelar la interacción entre el sistema y sus actores.



- Control: Para coordinar la realización de los casos de uso coordinando las actividades de los objetos que implementan la funcionalidad del caso de uso.



### 3.1.1 Diagramas de Clases del Análisis.

#### Caso de Uso: Autenticar Usuario

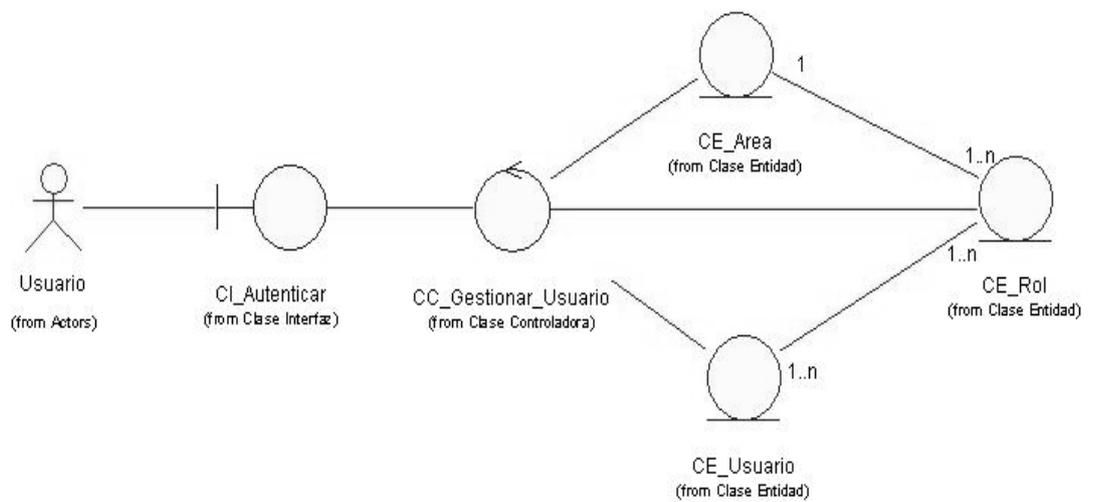


Figura 3. 1 Diagrama de clases del análisis del caso de uso Autenticar.

## Caso de Uso: Gestionar Rol

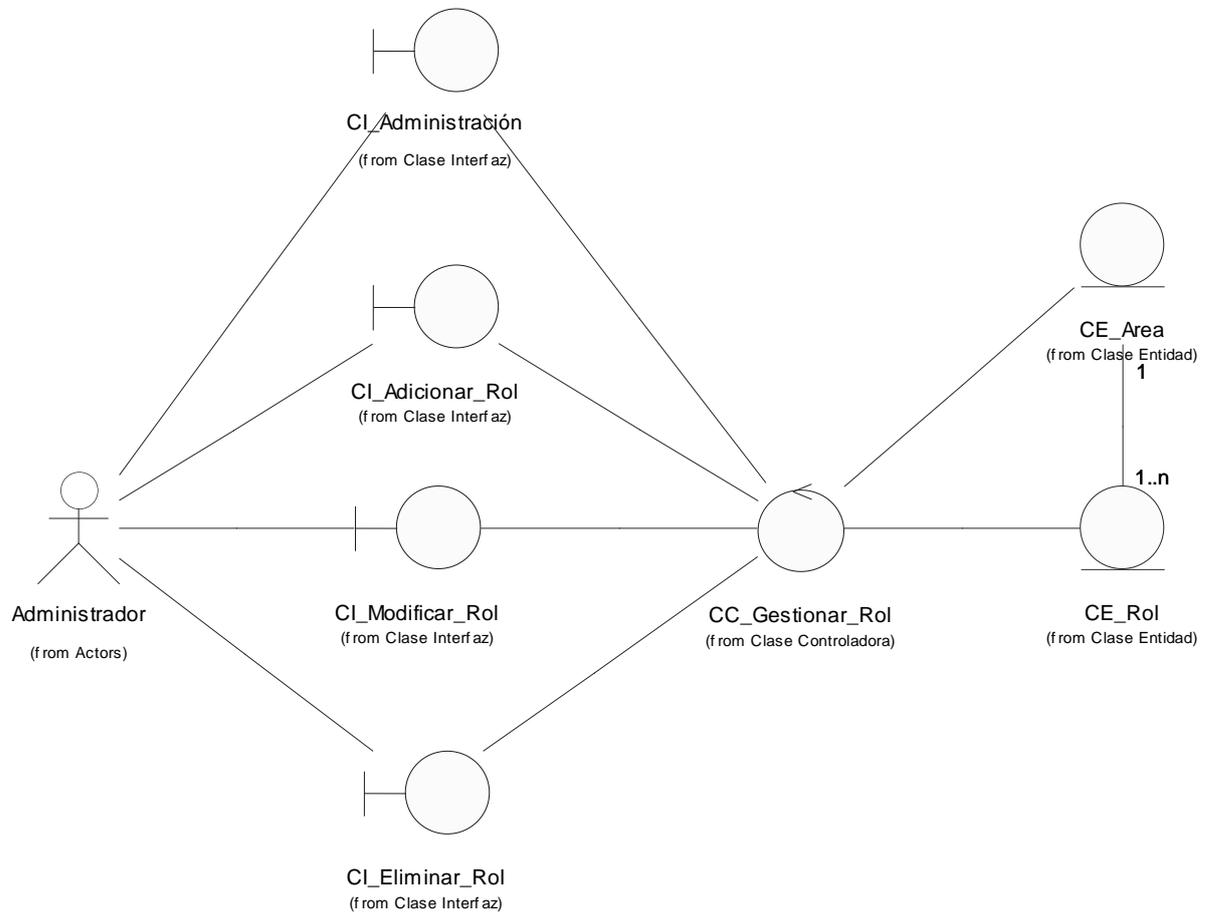


Figura 3. 2 Diagrama de clases del análisis del caso de uso Gestionar Rol.

## Caso de Uso: Gestionar Usuario

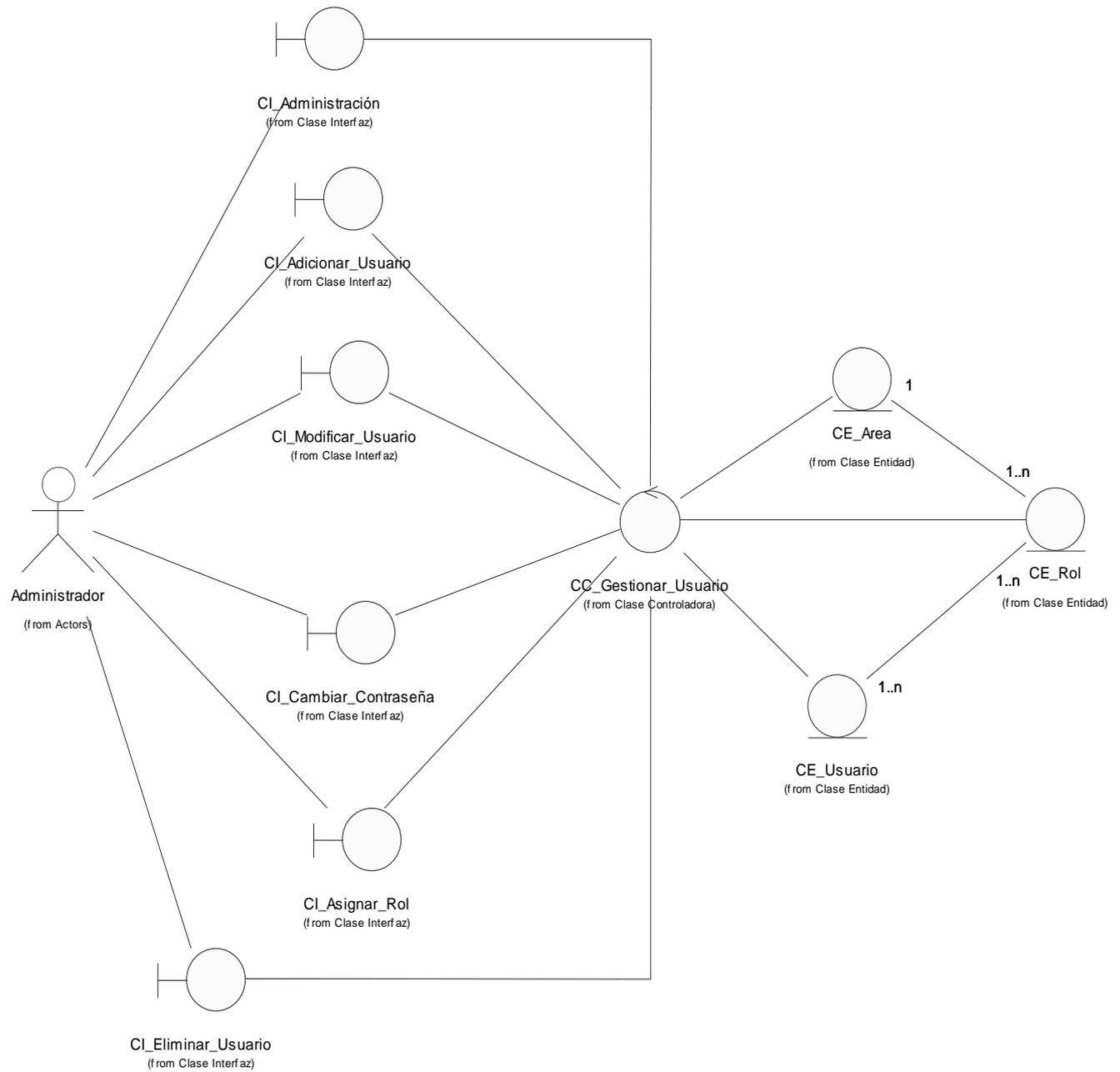


Figura 3. 3 Diagrama de clases del análisis del caso de uso Gestionar Usuario.

## Caso de Uso: Gestionar Historial

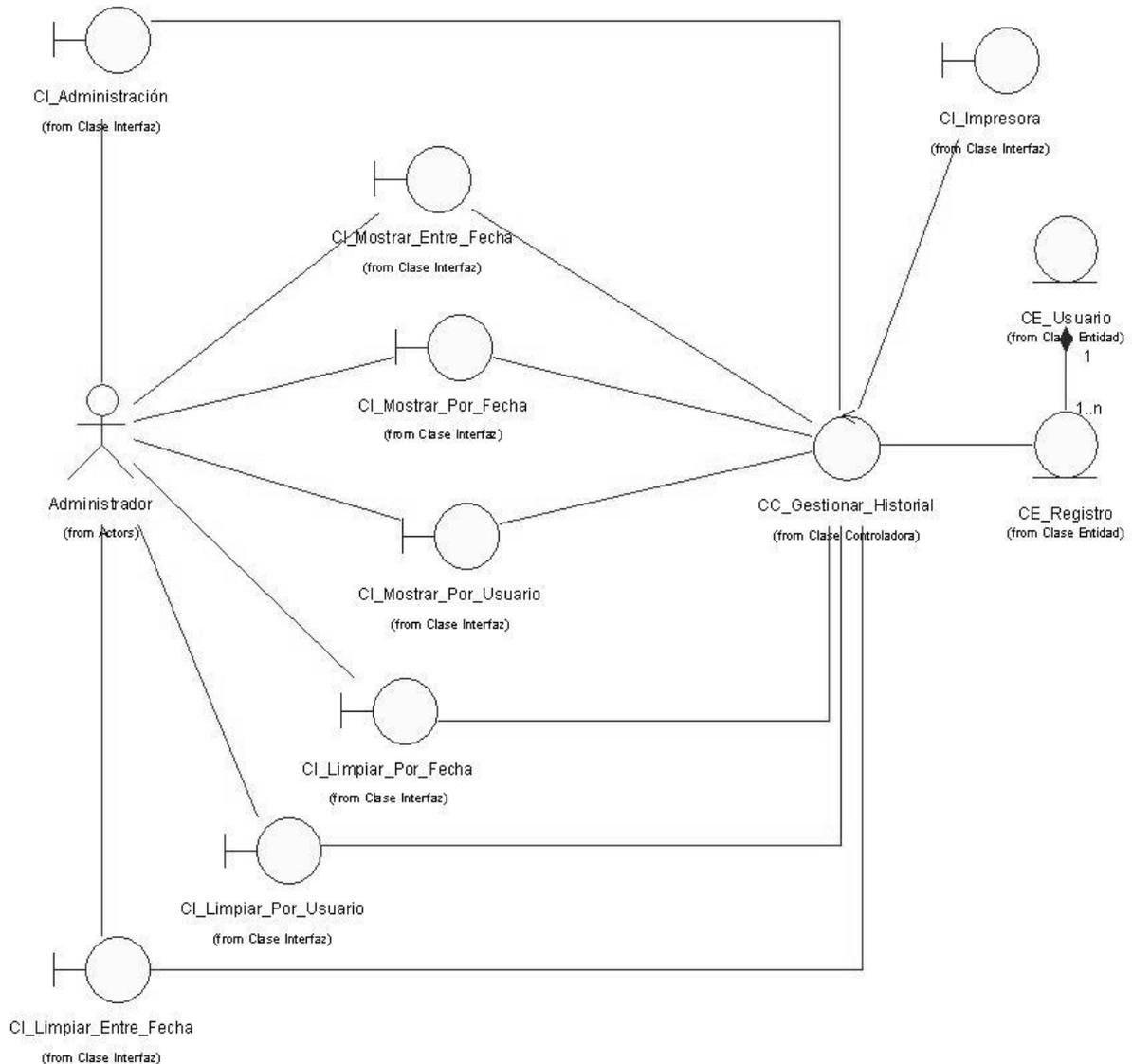


Figura 3. 4 Diagrama de clases del análisis del caso de uso Gestionar Historial.

### 3.2 Diseño.

Por otro lado el diseño es un refinamiento del análisis que tiene en cuenta los requisitos no funcionales, en definitiva cómo cumple el sistema sus objetivos. El grueso del esfuerzo del diseño está en las

últimas iteraciones de la fase de elaboración y las primeras de construcción. El diseño debe ser suficiente para que el sistema pueda ser implementado sin ambigüedades.

### 3.2.1 Diagramas de Clases del Diseño.

#### Caso de Uso: Autenticar Usuario

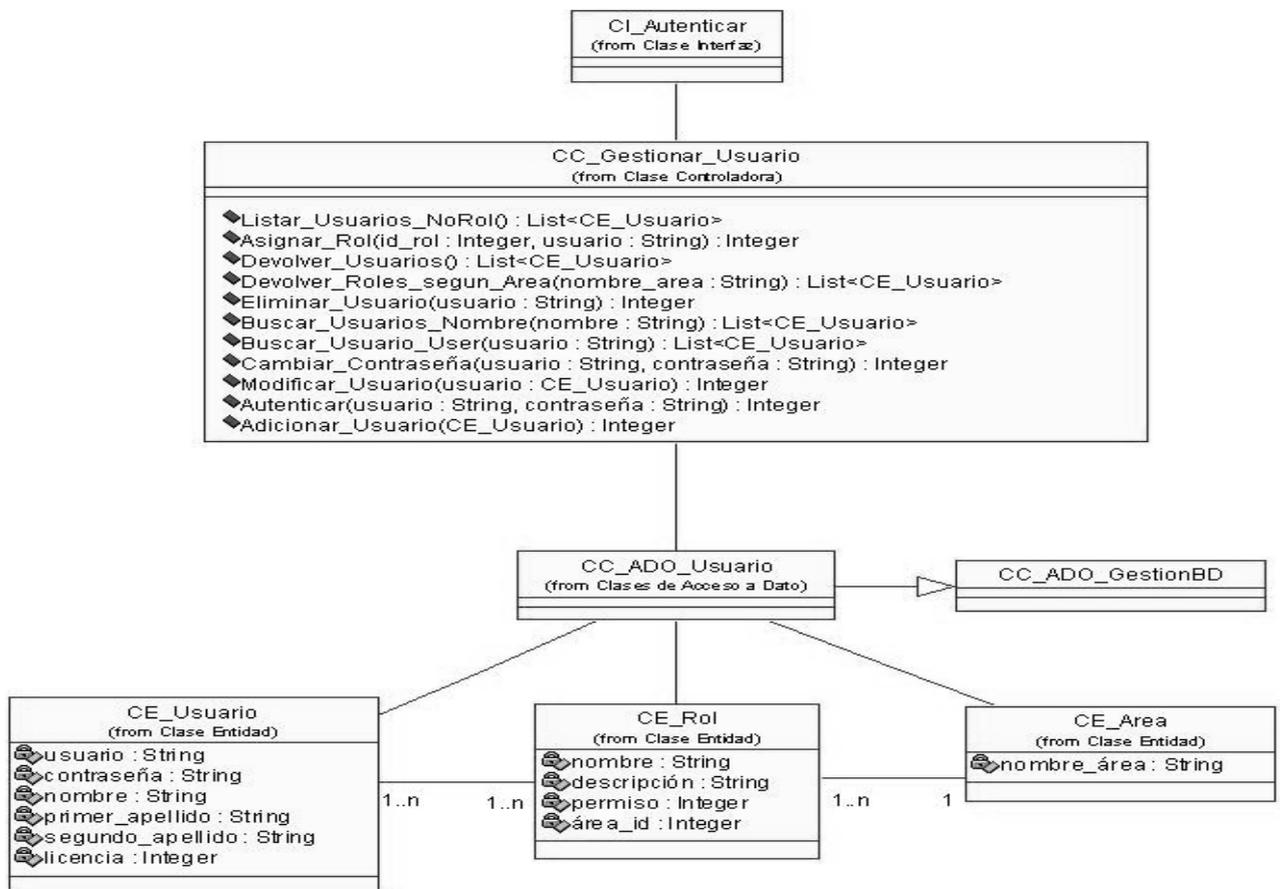


Figura 3. 5 Diagrama de clases del diseño del caso de uso Autenticar.

## Caso de Uso: Gestionar Rol

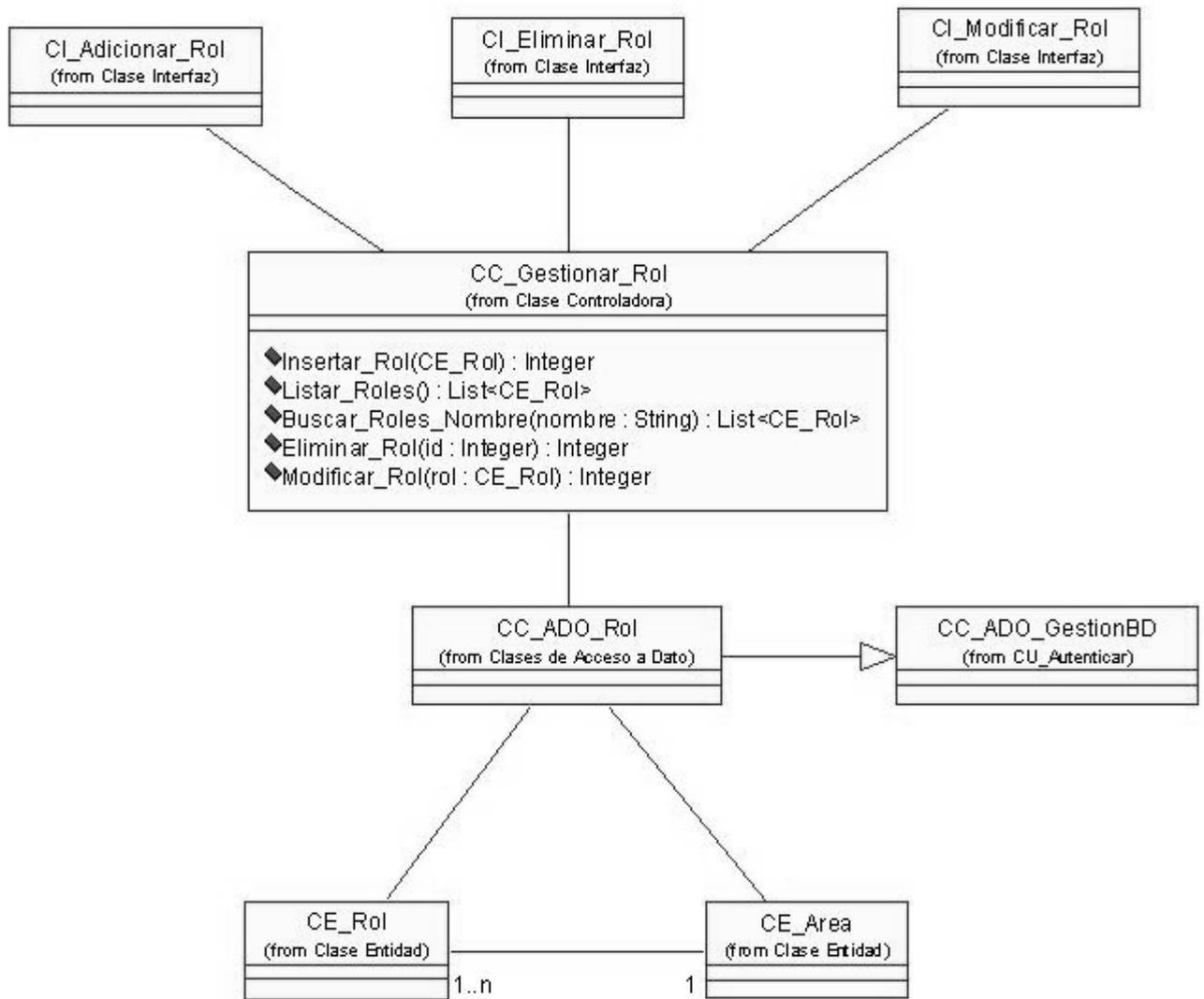


Figura 3. 6 Diagrama de clases del diseño del caso de uso Gestionar Rol.

## Caso de Uso: Gestionar Usuario

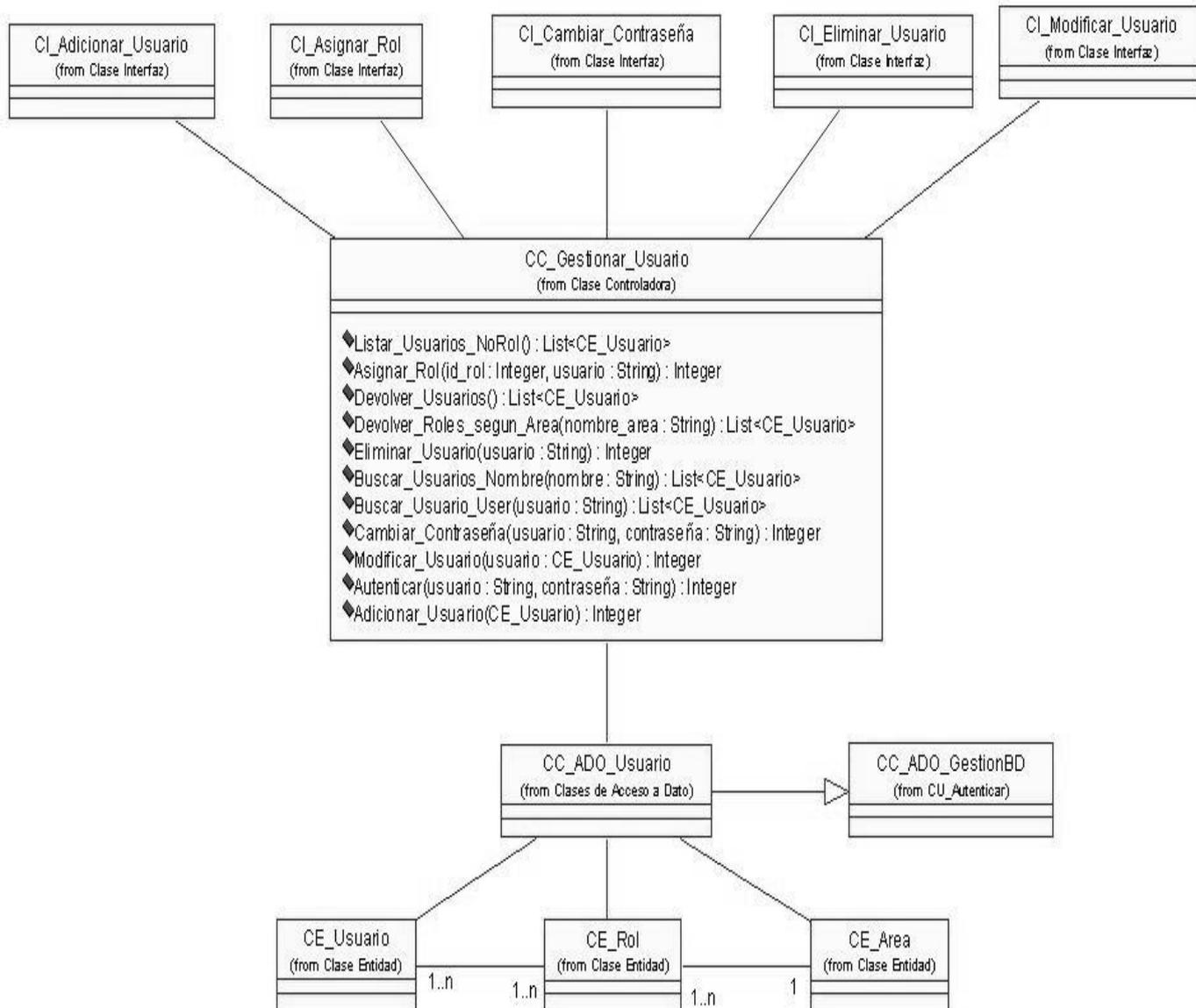


Figura 3. 7 Diagrama de clases del diseño del caso de uso Gestionar Usuario.

### Caso de Uso: Gestionar Historial

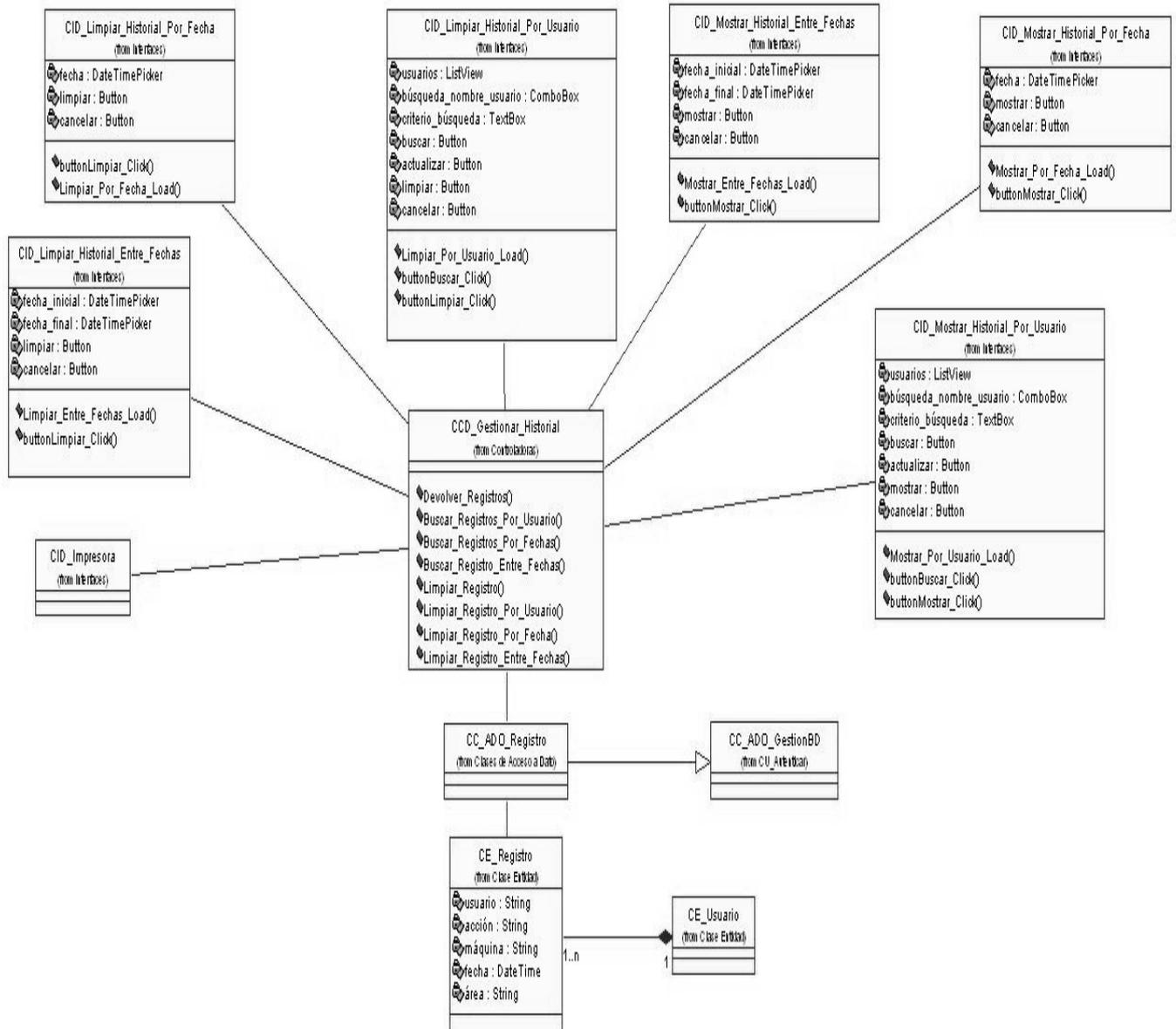


Figura 3. 8 Diagrama de clases del diseño del caso de uso Gestionar Historial.

## 3.3 Diseño de la Base de Datos.

### 3.3.1 Modelo Lógico de Datos.

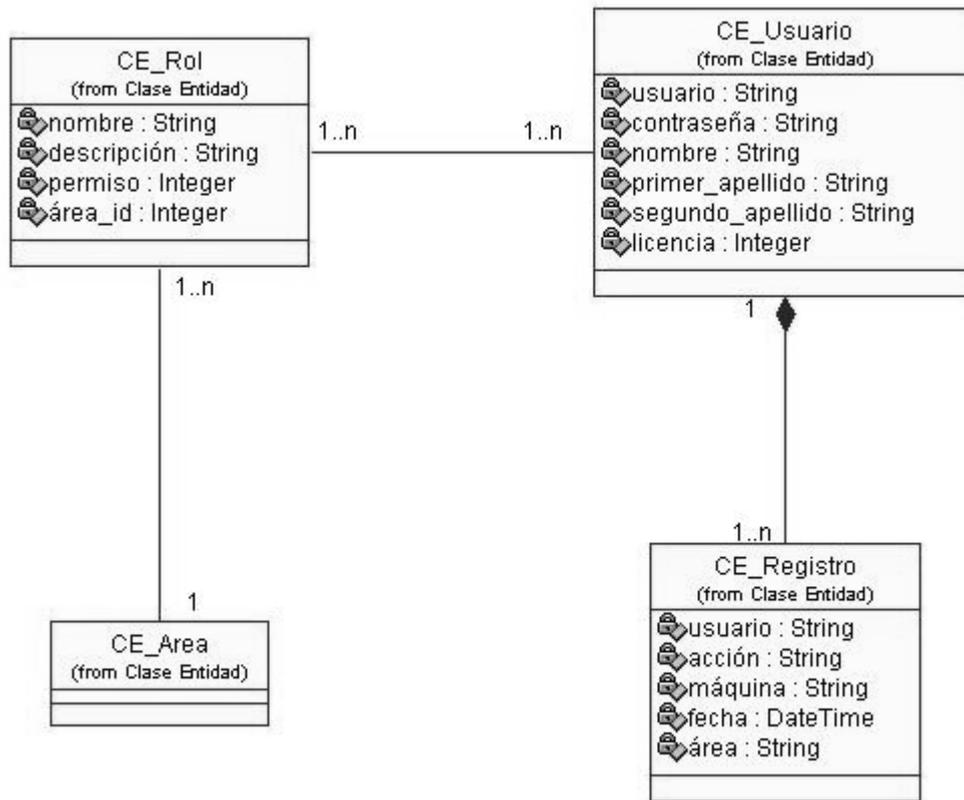


Figura 3. 9 Modelo lógico de datos.

## 3.3.2 Modelo Físico de Datos.

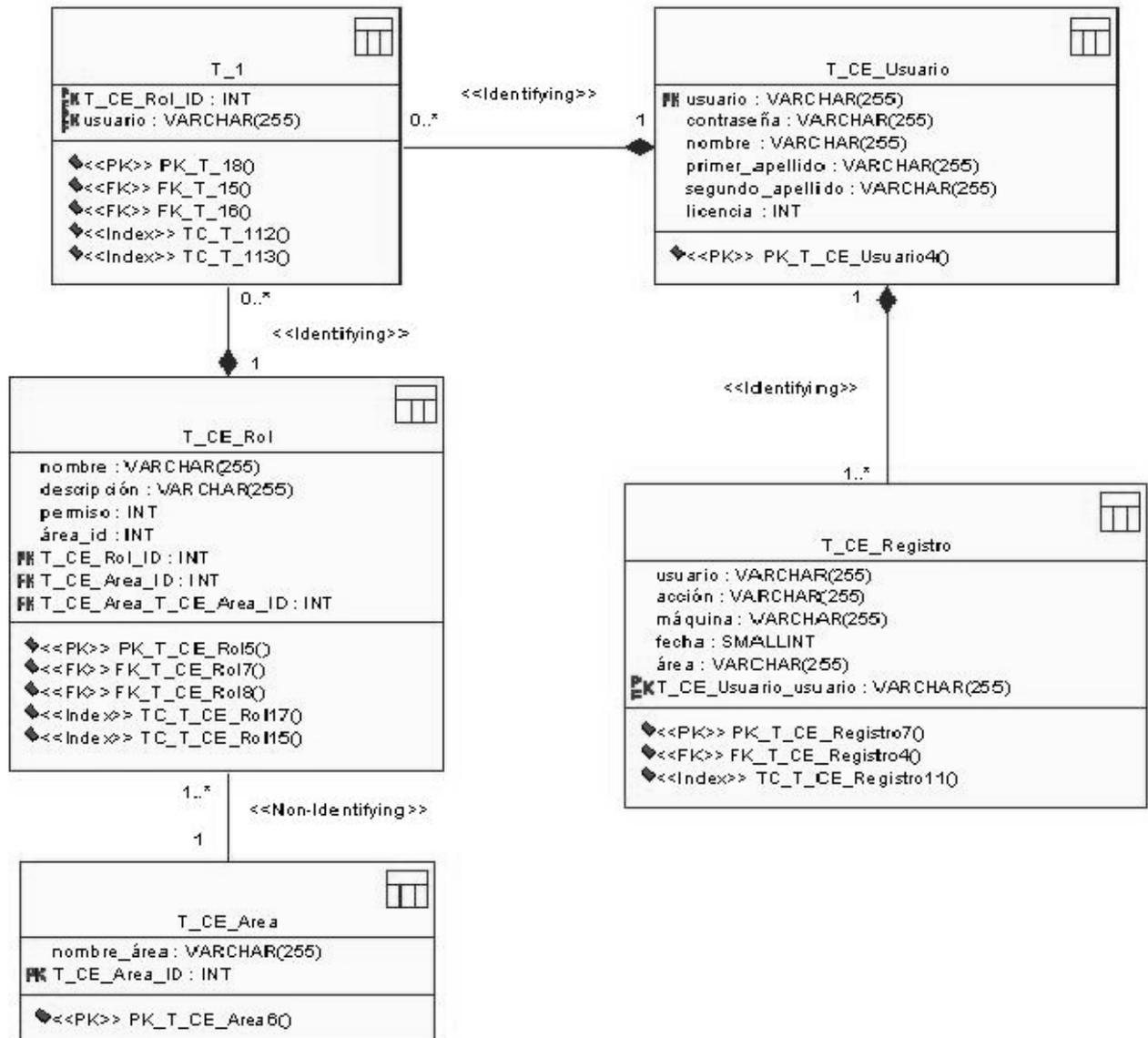


Figura 3. 10 Modelo físico de datos.

### **3.4 Conclusiones.**

En este capítulo se modelaron los diagramas de clases del análisis, representando las clases y las relaciones entre ellas, con el fin de comprender de una mejor manera las funcionalidades del sistema. También se obtuvieron los diagramas de clases del diseño, donde se modela la forma en que el sistema debe darle solución a los requerimientos funcionales y no funcionales.

## Capítulo IV: Implementación.

### Introducción.

En este capítulo se hace referencia al flujo de trabajo de implementación, modelándose los artefactos que se obtienen durante el mismo que son el diagrama de despliegue y los diagramas de componentes, los cuales serán de gran utilidad para la posterior construcción del sistema propuesto.

### 4.1 Diagrama de Despliegue.

El Diagrama de Despliegue es un tipo de diagrama del Lenguaje Unificado de Modelado que se utiliza para modelar el hardware utilizado en la implementación de sistemas y las relaciones entre sus componentes.

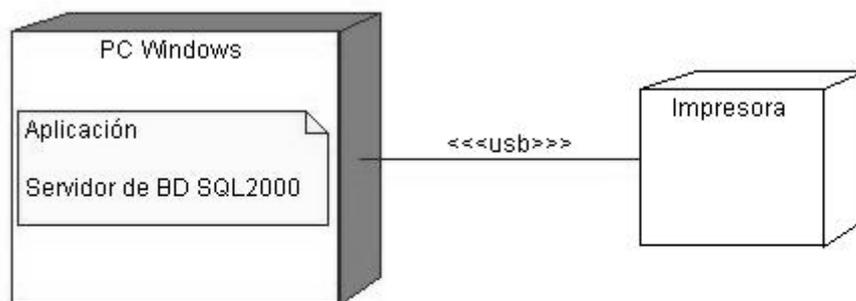


Figura 4. 1 Diagrama de despliegue.

### 4.2 Diagrama de Componentes.

Un diagrama de componentes representa la separación de un sistema de *software* en componentes físicos y muestra las dependencias entre estos componentes.

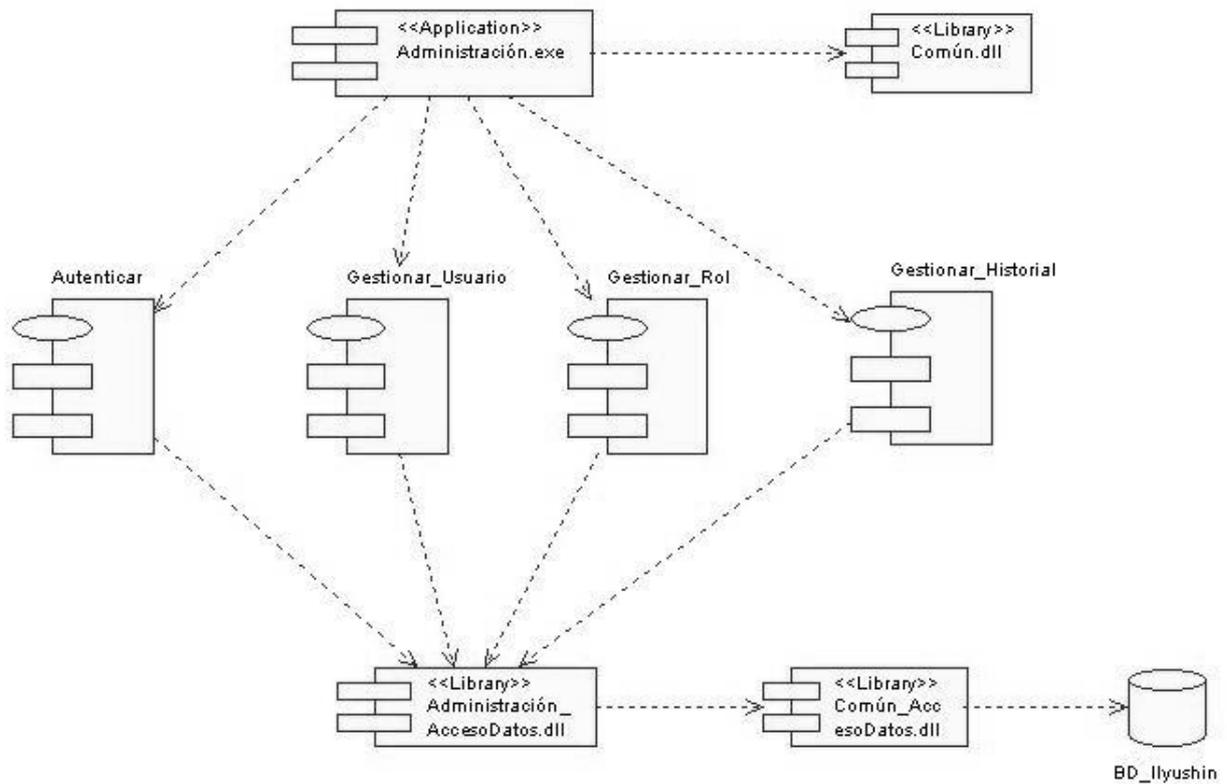


Figura 4. 2 Diagrama de componentes del Módulo de Administración.

### 4.2.1 Diagrama de componentes: CU Autenticar Usuario.

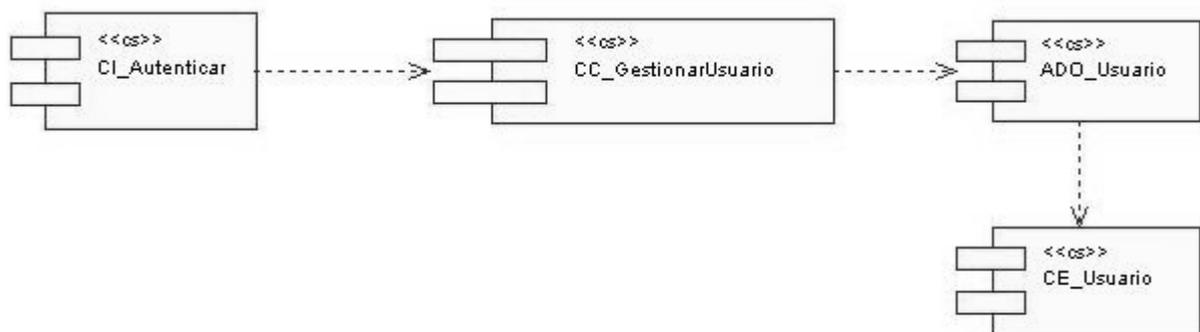


Figura 4. 3 Diagrama de componentes del caso de uso Autenticar.

### 4.2.2 Diagrama de componentes: CU Gestionar Usuario.

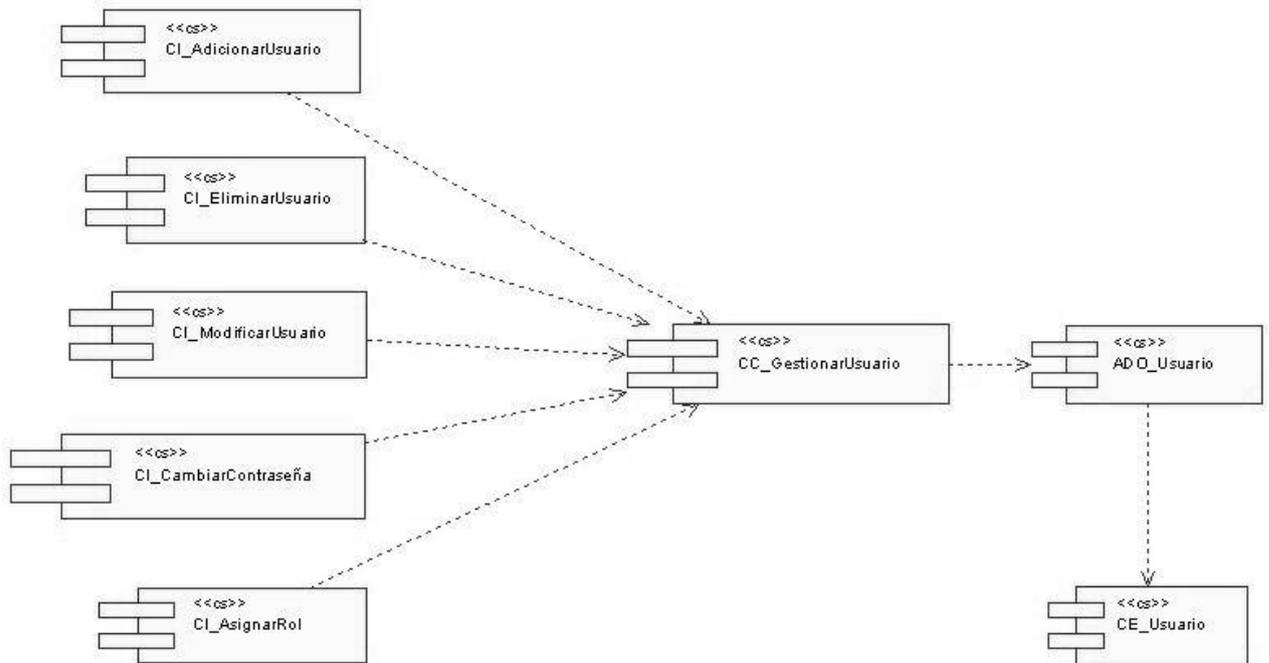


Figura 4. 4 Diagrama de componentes del caso de uso Gestionar Usuario.

### 4.2.3 Diagrama de componentes: CU Gestionar Rol.

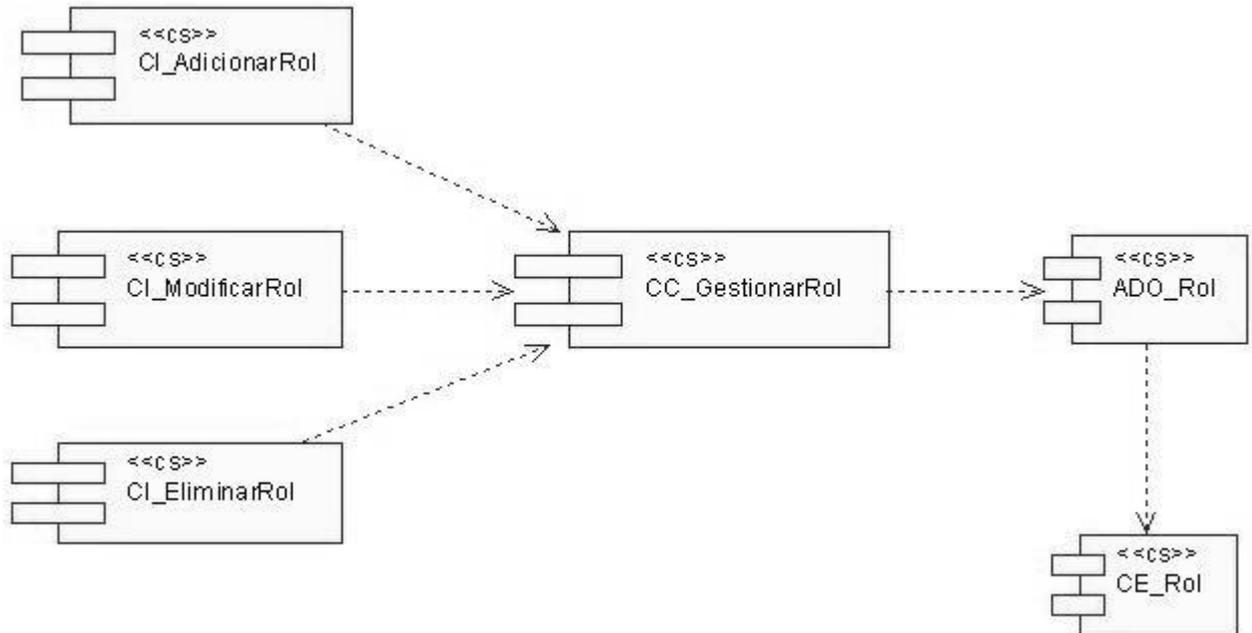


Figura 4. 5 Diagrama de componentes del caso de uso Gestionar Rol.

### 4.2.4 Diagrama de componentes: CU Gestionar Historial.

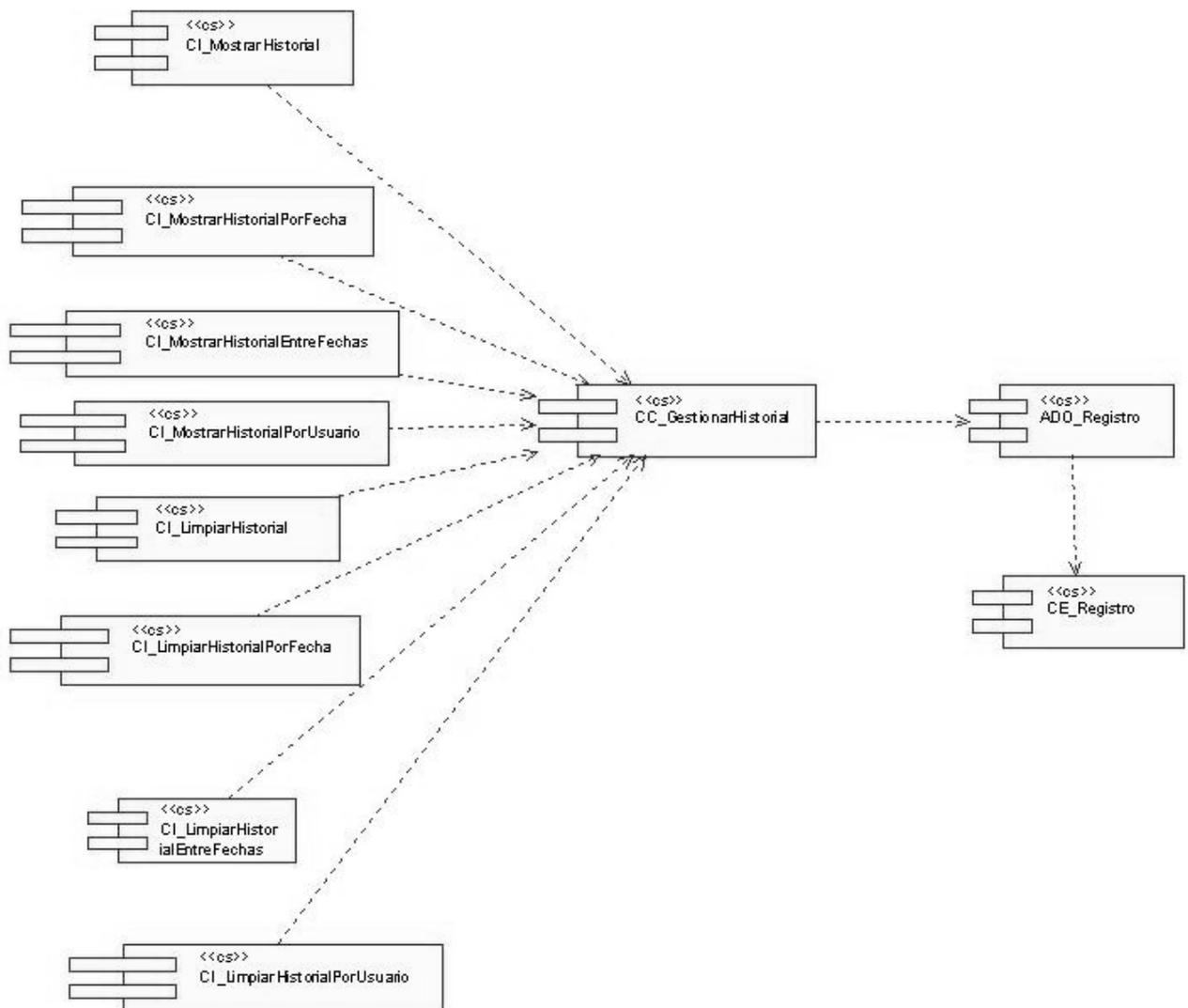


Figura 4. 6 Diagrama de componentes del caso de uso Gestionar Historial.

### 4.3 Conclusiones.

En este capítulo se mostraron los diagramas de despliegue y de componentes, lográndose así una visión de la aplicación como entidad física, y dotando al desarrollador de una base sólida para un desarrollo eficiente de la solución propuesta.

## Capítulo V: Estudio de Factibilidad.

### Introducción.

Este capítulo aborda el tema relacionado al estudio de factibilidad del producto, presentándose la planificación y los costos del proyecto, basados en la técnica de estimación de esfuerzo por punto de casos de uso, la que permitirá obtener una idea del esfuerzo necesario para llevar adelante el mismo.

En este capítulo se analizan también los beneficios tangibles e intangibles obtenidos con el desarrollo del sistema, realizando una evaluación de factibilidad mediante el análisis de costos y beneficios.

### 5.1 Planificación basada en casos de uso.

#### a) Cálculo de los puntos de casos de uso desajustados.

$$UUCP = UAW + UUCW$$

Donde:

UUCP: Puntos de Casos de Uso sin ajustar.

UAW: Factor de Peso de los Actores sin ajustar.

UUCW: Factor de Peso de los Casos de Uso sin ajustar.

#### Factor de Peso de los Actores sin ajustar (UAW).

Tipo de actor	Descripción	Factor de peso	Actores	Total
Simple	Otro sistema que interactúa con el sistema a desarrollar mediante una interfaz de programación.	1	0	0
Medio	Otro sistema que interactúa con el sistema a desarrollar mediante un protocolo o una interfaz basada en texto.	2	0	0

Complejo	Una persona que interactúa con el sistema mediante una interfaz gráfica.	3	2	6
Total				6

**Tabla 5.1 Factor de peso de los actores sin ajustar.**

UAW =  $\Sigma$  cantidad actores \* peso.

UAW =  $2*3=6$ .

Factor de Peso de los Casos de Uso sin ajustar (UUCW).

Tipo de CU	Descripción	Peso	Cantidad de CU	Total
Simple	El CU tiene de 1 a 3 transacciones.	5	1	5
Medio	El CU tiene de 4 a 7 transacciones	10	1	10
Complejo	El CU tiene más de 8 transacciones	15	2	30

**Tabla 5.2 Factor de peso de los CU sin ajustar.**

UUCW =  $\Sigma$  cantidad CU \* Peso.

UUCW = 45.

UUCP = 6+45.

UUCP = 51.

**b) Cálculo de los puntos de casos de uso ajustados.**

$UCP = UUCP * TCF * EF$ .

Donde:

UCP: Puntos de Casos de Uso ajustados.

UUCP: Puntos de Casos de Uso sin ajustar.

TCF: Factor de complejidad técnica.

EF: Factor de ambiente.

### Cálculo de complejidad técnica (TCF).

El factor de complejidad técnica es el resultado de un conjunto de factores que determinan la complejidad técnica del sistema. Cada uno de los factores se cuantifica con un valor de 0 a 5, donde 0 significa un aporte irrelevante y 5 un aporte muy importante.

Significado de los valores:

0: No presente o sin influencia.

1: Influencia incidental o presencia incidental.

2: Influencia moderada o presencia moderada.

3: Influencia media o presencia media.

4: Influencia significativa o presencia significativa.

5: Fuerte influencia o fuerte presencia.

Factor	Descripción	Peso	Valor asignado	Total
T1	Sistema distribuido.	2	0	0
T2	Objetivos de performance o tiempo de respuesta.	1	5	5
T3	Eficiencia del usuario final.	1	5	5

T4	Procesamiento interno complejo.	1	2	2
T5	El código debe ser reutilizable.	1	5	5
T6	Facilidad de instalación.	0.5	3	1.5
T7	Facilidad de uso.	0.5	5	2.5
T8	Portabilidad.	2	5	10
T9	Facilidad de cambio.	1	3	3
T10	Concurrencia.	1	1	1
T11	Incluye objetivos especiales de seguridad.	1	5	5
T12	Provee acceso directo a terceras partes.	1	0	0
T13	Se requieren facilidades especiales de entrenamiento a usuarios.	1	0	0
Total				40

**Tabla 5.3 Factor de complejidad técnica.**

$$TCF = 0.6 + 0.01 * \Sigma (\text{peso} * \text{valor asignado})$$

$$TCF = 0.6 + 0.01 * 40$$

$$TCF = 1$$

### Cálculo de factor de ambiente (EF).

Las habilidades y el entrenamiento del grupo involucrado en el desarrollo tienen un gran impacto en las estimaciones de tiempo. Estos factores son los que se contemplan en el cálculo del factor de ambiente. El cálculo del mismo es similar al cálculo del Factor de complejidad técnica, es decir, se trata de un conjunto de factores que se cuantifican con valores de 0 a 5.

Factor	Descripción	Peso	Valor asignado	Total
E1	Familiaridad con el modelo de proyecto utilizado.	1.5	0	0
E2	Experiencia en la aplicación.	0.5	1	0.5
E3	Experiencia en orientación a objetos.	1	4	4
E4	Capacidad del analista líder.	0.5	3	1.5
E5	Motivación.	1	5	5
E6	Estabilidad de los requerimientos.	2	3	6
E7	Personal <i>part-time</i> .	-1	2	-2
E8	Dificultad del lenguaje de programación.	-1	2	-2
Total				13

**Tabla 5.4 Factor de ambiente.**

$$EF = 1.4 - 0.03 * \sum (\text{Peso}_i * \text{Valor}_i)$$

$$EF = 1.4 - 0.03 * 13$$

$$EF = 1.01$$

$$UCP = UUCP * TCF * EF$$

$$UCP = 51 * 1 * 1.01$$

$$UCP = 51.51$$

### c) Estimación de esfuerzo a través de los puntos de casos de uso.

$$E = UCP * CF$$

Donde:

E: Esfuerzo estimado en horas hombres.

UCP: Punto de casos de usos ajustados.

CF: Factor de conversión.

#### Cálculo de factor de conversión (CF).

Factor de convención (CF) se cuentan cuántos valores del factor ambiente están por debajo de la media (3) de E1 a E6, y cuántos están por encima de la media en E7 y E8. Si el total es 2 o menos se utiliza el factor de convención 20 Horas/Hombre. Si el total es 3 o 4 se utiliza 28 Horas/Hombre. Si el total es mayor a 5 se recomienda efectuar cambios en el proyecto ya que tiende a perecer en corto tiempo. En este caso se puede afirmar que el total de valores por debajo de la media entre E1 y E6, y por encima de la media entre E7 y E8 es 2, por lo tanto:

$$CF = 20 \text{ Horas/Hombre.}$$

$$E = 51.51 * 20$$

$$E = 1030.2 \text{ Horas/Hombre}$$

#### Esfuerzo de todo el proyecto.

Actividad.	Porcentaje %	Horas/Hombre
Análisis.	20	515.1

Diseño.	30	772.65
Implementación.	40	1030.2
Pruebas.	0	0
Sobrecarga (otras actividades).	10	257.55
Total.	100	2575.5

**Tabla 5.5 Esfuerzo del proyecto.**

Si el esfuerzo total (ET) es igual a 2575.5 Horas/hombre y se estima que una persona trabaja como promedio 240 horas al mes, se obtendría ET = 10.73 Mes/Hombre ~11 Mes/Hombre.

Costo del proyecto.

En el caso de pagar un salario mensual de \$150.00.

Sean:

CH: cantidad de hombre.

Tiempo: tiempo total del proyecto.

$$\text{Costo} = \text{CH} * \text{salario} * \text{ET/CH}$$

$$\text{Costo} = 1 * 150 * 11/1$$

$$\text{Costo} = \$ 1650.$$

De los resultados obtenidos se interpreta que con un hombre el proyecto tiene un tiempo de duración de aproximadamente 11 meses con un costo total de \$1650.

**5.2 Análisis de costos y beneficios.**

El desarrollo de todo producto de software requiere de una inversión de capital, el cual se distribuye en gastos de personal, y de materiales necesarios para su elaboración, por lo que la viabilidad del proyecto depende de la medida en que los beneficios del mismo puedan saldar lo invertido en su desarrollo, ya sea, social o económicamente.

En la realización de este proyecto se usará software propietario para el cual el cliente garantizó la licencia del producto por lo que este gasto no está reflejado en el estudio de factibilidad.

Al implantar el sistema se tendrá como resultado el aumento de la eficiencia del proceso de gestión de las operaciones de cada usuario dentro de la aplicación, se alcanzará una mayor seguridad en la ejecución de los diferentes procesos así como también se garantizará la posibilidad de monitoreo de cada acción ejecutada en el software, esto elevará sustancialmente los niveles de seguridad de la aplicación.

Es de destacar también el ahorro monetario logrado a partir de la posibilidad de no tener que contratar servicios de empresas extranjeras para la realización del sistema.

A partir del análisis de los costos del proyecto y de los beneficios que la realización del mismo aporta, se puede concluir que su implementación es totalmente factible.

### **5.3 Conclusiones.**

En este capítulo se realizó el estudio de factibilidad del sistema propuesto, considerando el costo de realización del mismo así como los beneficios que reportará su implementación.

## CONCLUSIONES.

Con la integración exitosa del Módulo de Administración al Sistema Integrador de Gestión de Vuelo para Aviones IL-96/300 se dará solución a los problemas que con respecto a seguridad, control y monitoreo de usuarios presentaba dicho sistema.

Para la realización del Módulo de Administración se recorrieron los flujos de trabajo del RUP de negocio, análisis, diseño, administración de proyectos e implementación, respondiendo así al desarrollo iterativo e incremental propuesto por esta metodología y desarrollando en cada fase los artefactos correspondientes.

Como consecuencia de la selección de herramientas se determinó utilizar Rational Rose como herramienta de modelado del sistema, SQL2000 como gestor de base de datos y como lenguaje de programación se escogió C# con su plataforma .NET.

Con el estudio de factibilidad llevado a cabo se determinó la posibilidad real de desarrollar el Módulo de Administración, luego de analizar también los beneficios tangibles e intangibles que representa su implementación e implantación dentro del Sistema Integrador de Gestión de Vuelo para Aviones IL-96/300, lo cual arrojó un saldo favorable para su desarrollo.

## **RECOMENDACIONES.**

Se recomienda a todo el que vaya mejorar o continuar con este trabajo:

- Realizar las pruebas necesarias del Módulo de Administración del Sistema Integrador de Gestión de Vuelo para Aviones IL-96/300.
- Incorporar nuevas funcionalidades que permitan el manejo de las áreas o módulos de trabajo, ya sea inserción, modificación o eliminación de alguna de ellas, lo cual implicaría también el trabajo con la configuración de las operaciones y acciones dentro de las mismas.
- Desarrollar la ayuda del Módulo de Administración.
- Analizar la posibilidad de migrar el Módulo de Administración como parte del Sistema Integrador de Gestión de Vuelo para Aviones IL-96/300 a software libre.

## Bibliografía

1. **Seco, J.A.G.** Tema 2: Introducción a C#. [En línea] abril de 2007. [Citado el: 6 de diciembre de 2007.] <http://www.programacion.com/tutorial/csharp/3/>.
2. **Martín, M. S., E. G. Guruceaga.** Salazarweb. [En línea] 16 de noviembre de 2005. [Citado el: 5 de diciembre de 2007.] <http://www.salazarweb.com/antbuspre.asp>.
3. **Kendall, Kendall &.** *Analisis y deiseño de sistemas*. Tercera edicion. págs. 15-18.
4. **Jacobson, I. G. Booch.** *Proceso Unificado de Desarrollo de Software*. Madrid : Adison-Wesley, 2000.
5. **J.Guapo.** Desarrollo Web. [En línea] [Citado el: 6 de Diciembre de 2007.] <http://www.desarrolloweb.com/articulos/561.php>.
6. **Garlan, M.S.Y.D.** *Software Architecture: Perspectives on an emerging discipline*. Upper Saddle River : Prentice Hall, 1996.
7. **D.E.V.Zapata.** Introduccion a la programacion multicapas. [En línea] 28 de mayo de 2004. [Citado el: 2007 de diciembre de 6.] [http://www.elguille.info/colabora/puntoNET/jevergara\\_Multitier.htm](http://www.elguille.info/colabora/puntoNET/jevergara_Multitier.htm).
8. **D.E.Perry.** *Software Architecture and its relevance for Software Engineering*. 1997.
9. **Callao, J.A. Hernandez, E.M.Mendoza y Y.G.Laina.** *Concepción y Desarrollo del Módulo de Administración del Proyecto TeleBanca*. Ciudad de la Habana : s.n., julio 2007. pág. 26, Trabajo de Diploma.
10. **A.Mendoza, Sanchez Maria.** Metodologías de desarrollo de software. *Revista peruana de informática*. [En línea] 7 de junio de 2004. [Citado el: 8 de diciembre de 2007.] [http://www.informatizate.net/articulos/metodologias\\_de\\_desarrollo\\_de\\_software](http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software).
11. **Microsoft, Visual Studio Team System.** Visual Studio Team System. [En línea] [Citado el: 3 de diciembre de 2007.] <http://msdn2.microsoft.com/en-us/teamsystem/default.aspx>.
12. Programación en castellano. *Visual Paradigm for UML*. [En línea] 2005. [Citado el: 10 de diciembre de 2007.] <http://www.programacion.net/noticia/1363/>.
13. **Microsoft, MSDN.** MSDN. [En línea] [Citado el: 3 de diciembre de 2007.] <http://msdn.microsoft.com/library/spa/default.asp>.
14. **Creative Commons.** Metodologías usadas en ingeniería de software. [En línea] 10 de mayo de 2007. [Citado el: 8 de diciembre de 2007.] <http://www.um.es/docencia/barzana/IAGP/lagp3.html#BM2>.
15. **Monografias.** Diseño y Modelación de un Proyecto de Software Utilizando el lenguaje UML. *Monografia.com - Tesis, Documentos, Publicaciones y Recursos Educativos*. [En línea] [Citado el: 10 de diciembre de 2007.] <http://www.monografias.com/trabajos28/proyecto-software/proyectosoftware.shtml>.

16. **Tránsito Aéreo.** Direccion de Tránsito Aéreo. *Republica Argentina*. [En línea] [Citado el: 14 de diciembre de 2007.] <http://www.cra.gov.ar/dta/ais/notam.php>.
17. **Estrategia Magazine.** Año 3- Num. 54- Sección tecnológica. [En línea] [Citado el: 6 de diciembre de 2007.] <http://www.e-estrategia.com.ar>.
18. **SHAW, M.** *Abstraction Techniques in Modern Programming Languages*. IEEE Software, 1984.pp.10-26 p.
19. **ROBERT MONROE, A. K., RALPH MELTON Y DAVID GARLAN.** *Stylized architecture, design patterns, and objects*. [En línea] [Citado el: 15 de diciembre de 2007.] <http://citeseer.nj.nec.com/monroe96stylized.html>,<http://citeseer.nj.nec.com/monroe96stylized.html>.
20. **MOLPECERES, A.** *Procesos de desarrollo: RUP, XP y FDD*. [En línea] [Citado el: 15 de diciembre de 2007.] [www.willydev.net/descargas/articulos/general/cualxpfdrup.PDF](http://www.willydev.net/descargas/articulos/general/cualxpfdrup.PDF), 2003.

## ANEXOS.

### Anexo 1: Diagramas de Secuencia.

#### CU Autenticar.

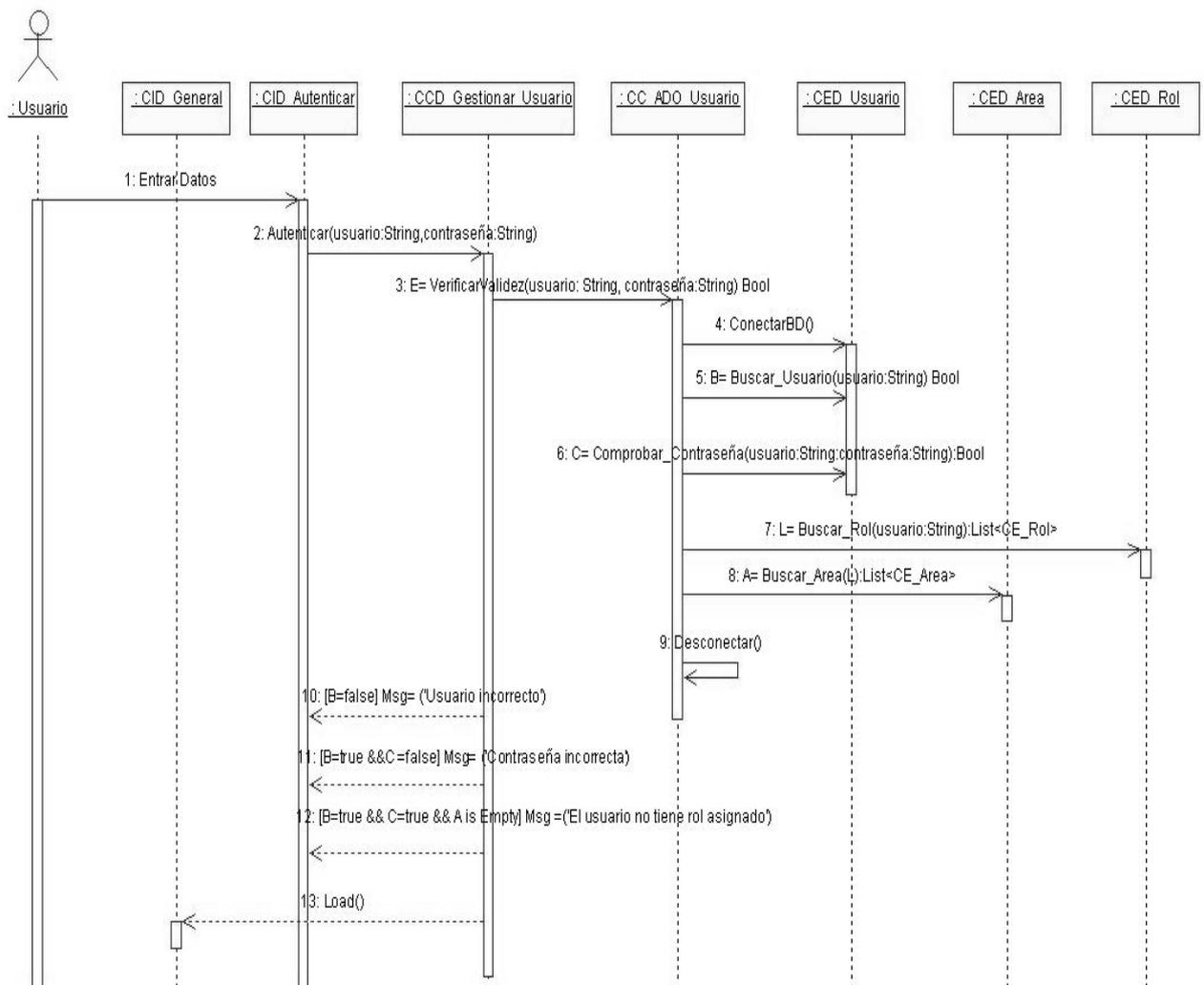


Figura Anexo 1. 1 Diagrama de secuencia del caso de uso Autenticar.

## CU Gestionar Usuario: Adicionar Usuario.

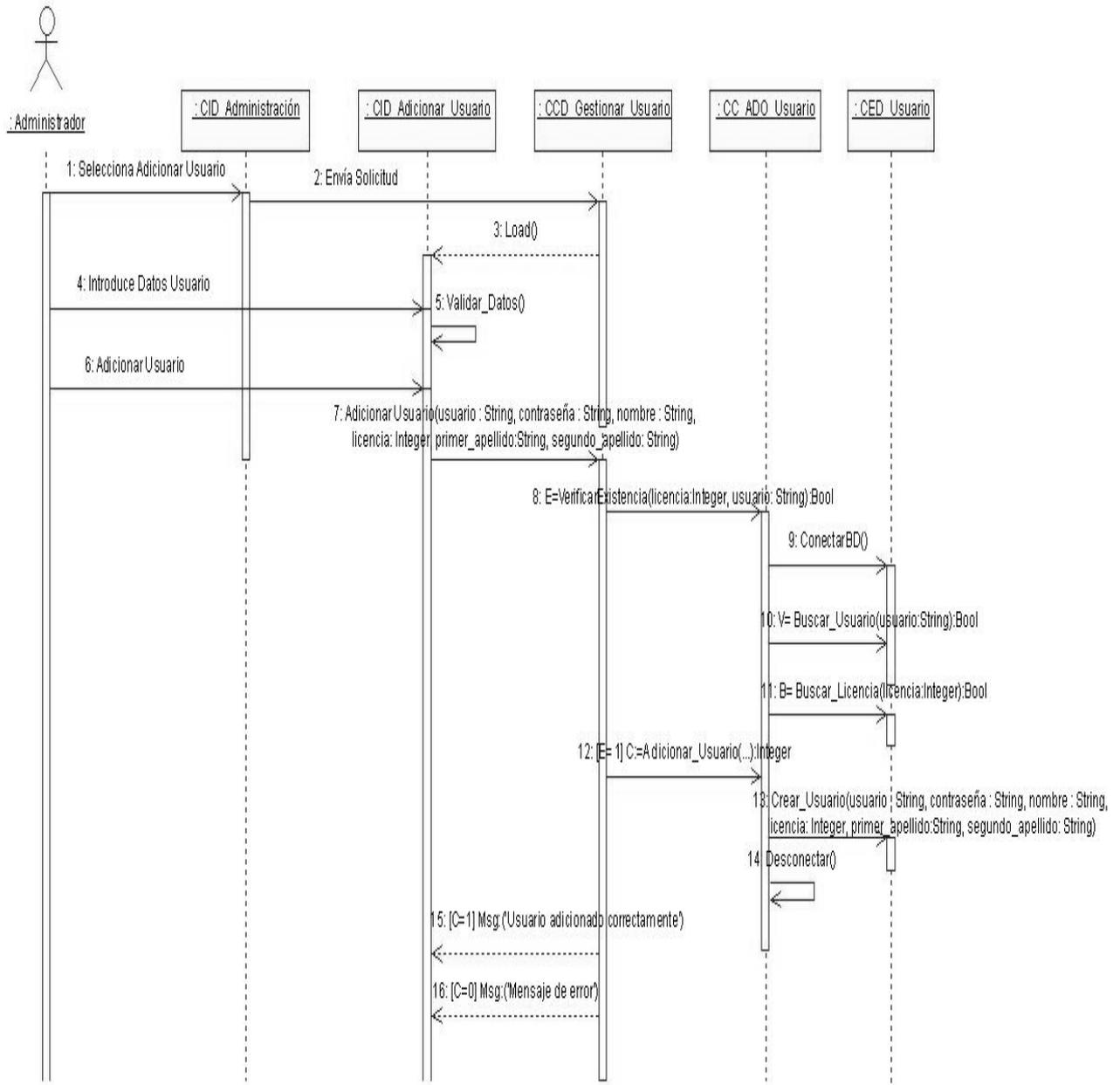


Figura Anexo 1. 2 Diagrama de secuencia del Adicionar Usuario, perteneciente al CU Gestionar Usuario.

## CU Gestionar Usuario: Eliminar Usuario.

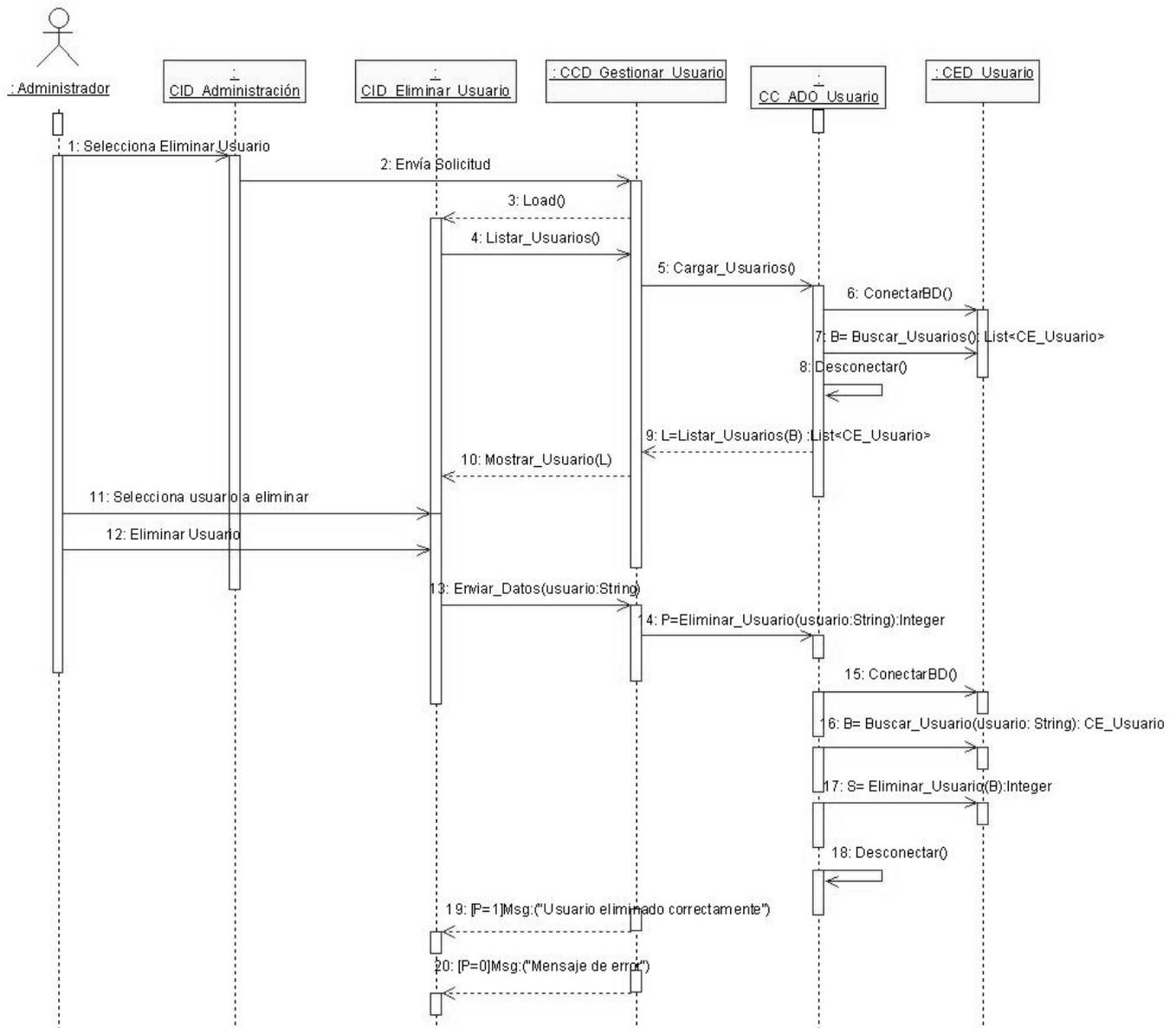


Figura Anexo 1. 3 Diagrama de secuencia del Eliminar Usuario, perteneciente al CU Gestionar Usuario.

## CU Gestionar Usuario: Cambiar Contraseña.

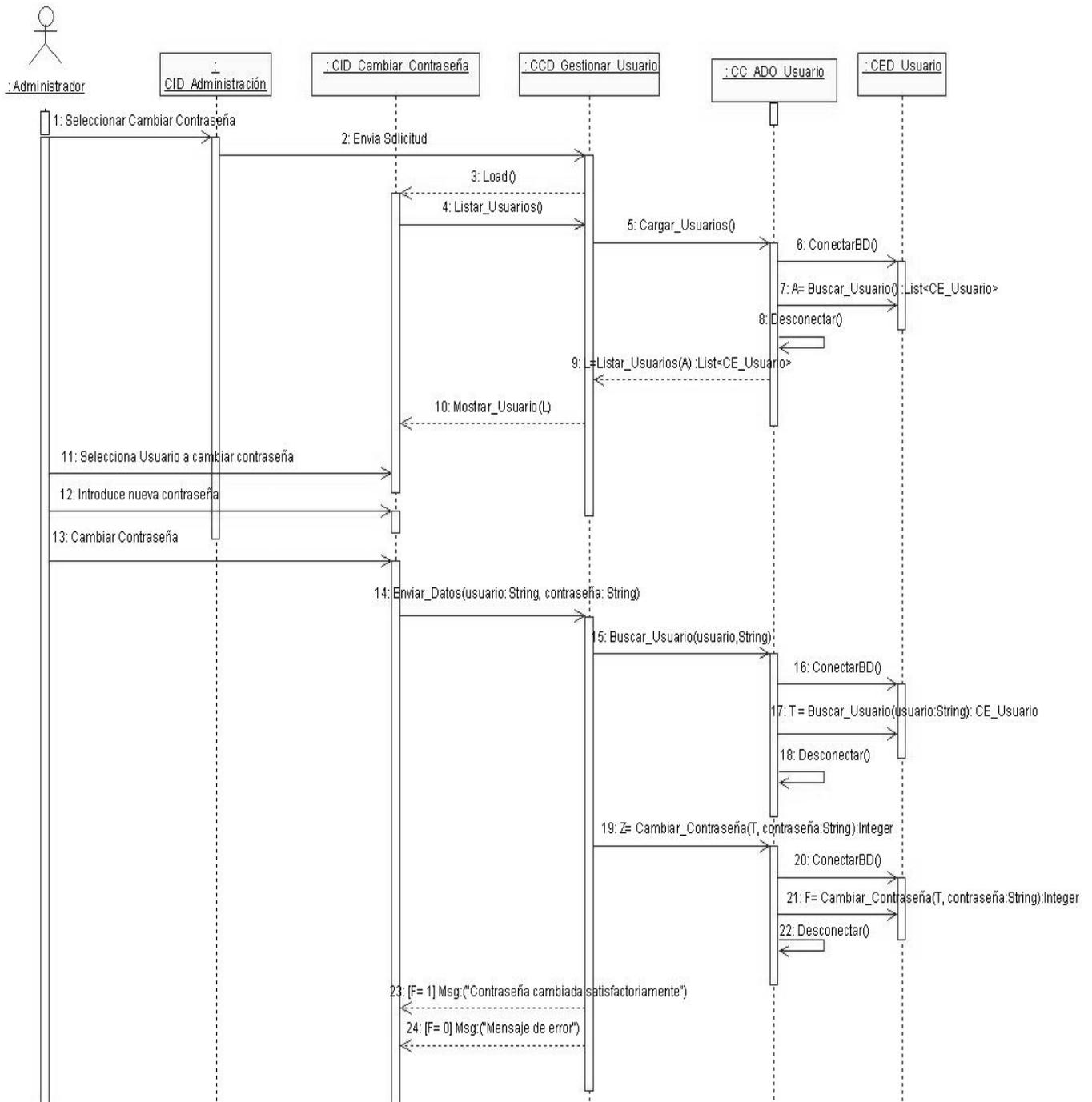


Figura Anexo 1. 4 Diagrama de secuencia del Cambiar Contraseña, perteneciente al CU Gestionar Usuario.

## CU Gestionar Rol: Adicionar Rol.

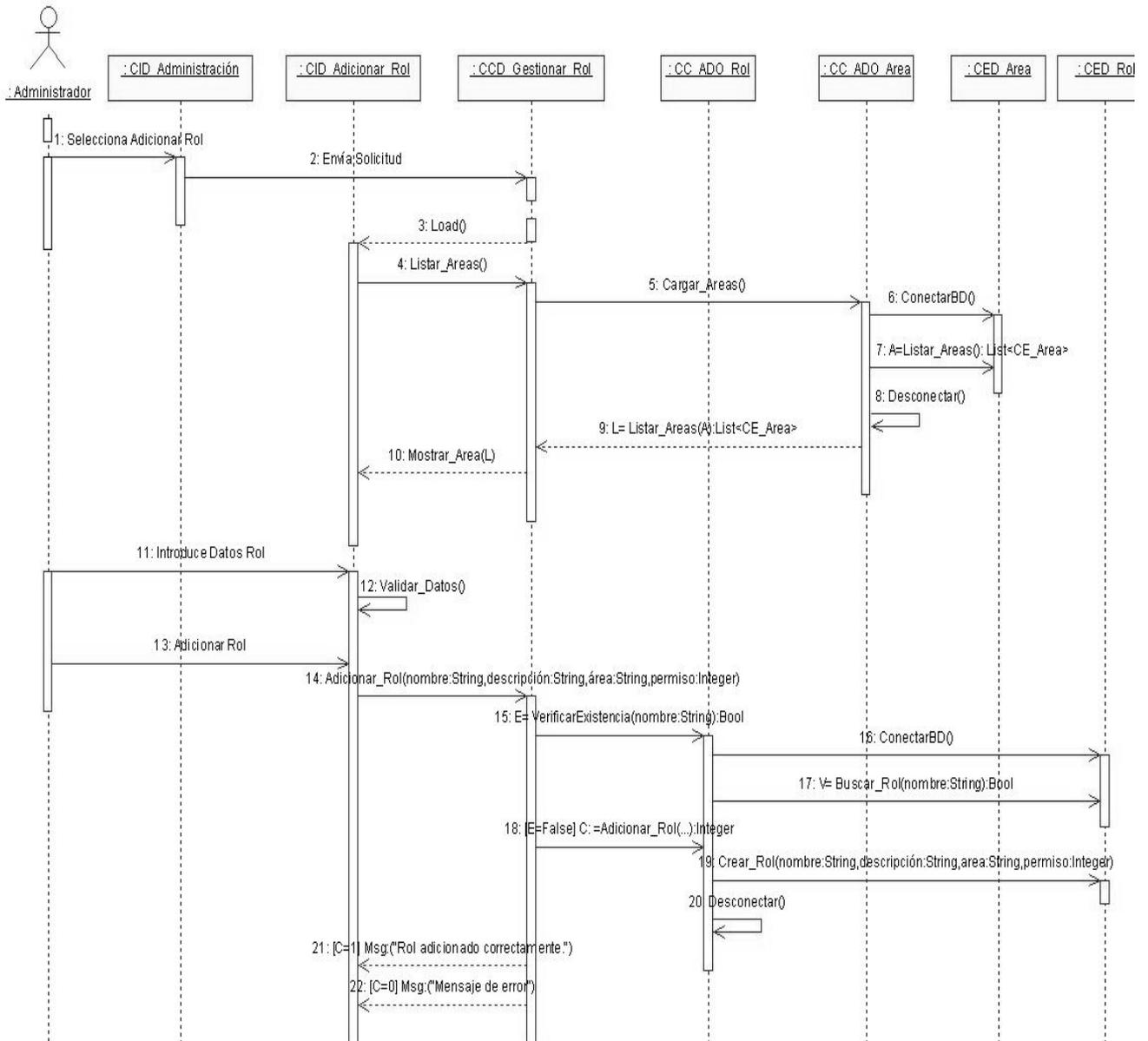


Figura Anexo 1. 5 Diagrama de secuencia del Adicionar Rol, perteneciente al CU Gestionar Rol.

## CU Gestionar Rol: Eliminar Rol.

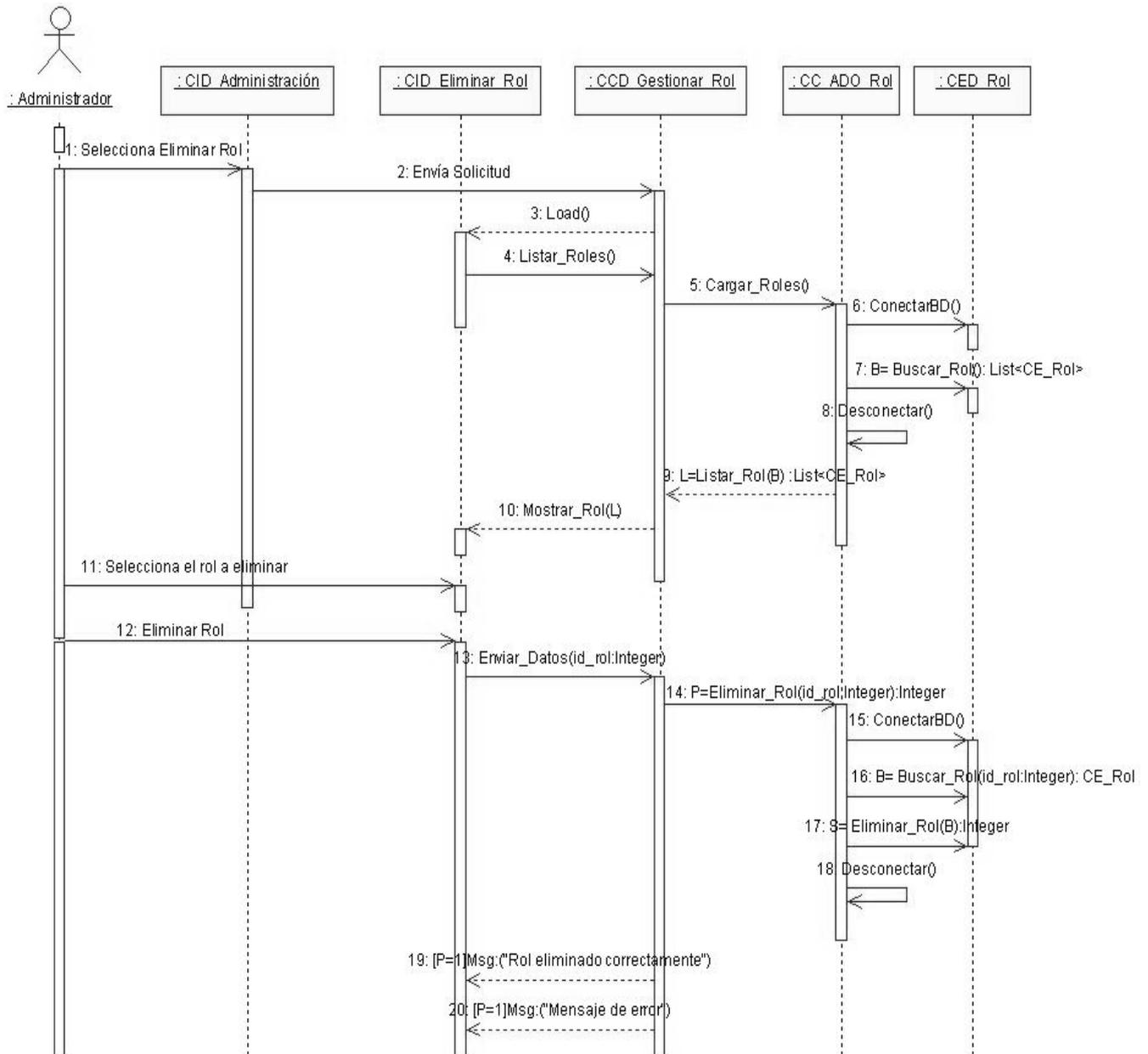


Figura Anexo 1. 6 Diagrama de secuencia del Eliminar Rol, perteneciente al CU Gestionar Rol.

## CU Gestionar Historial: Limpiar Historial Completo.

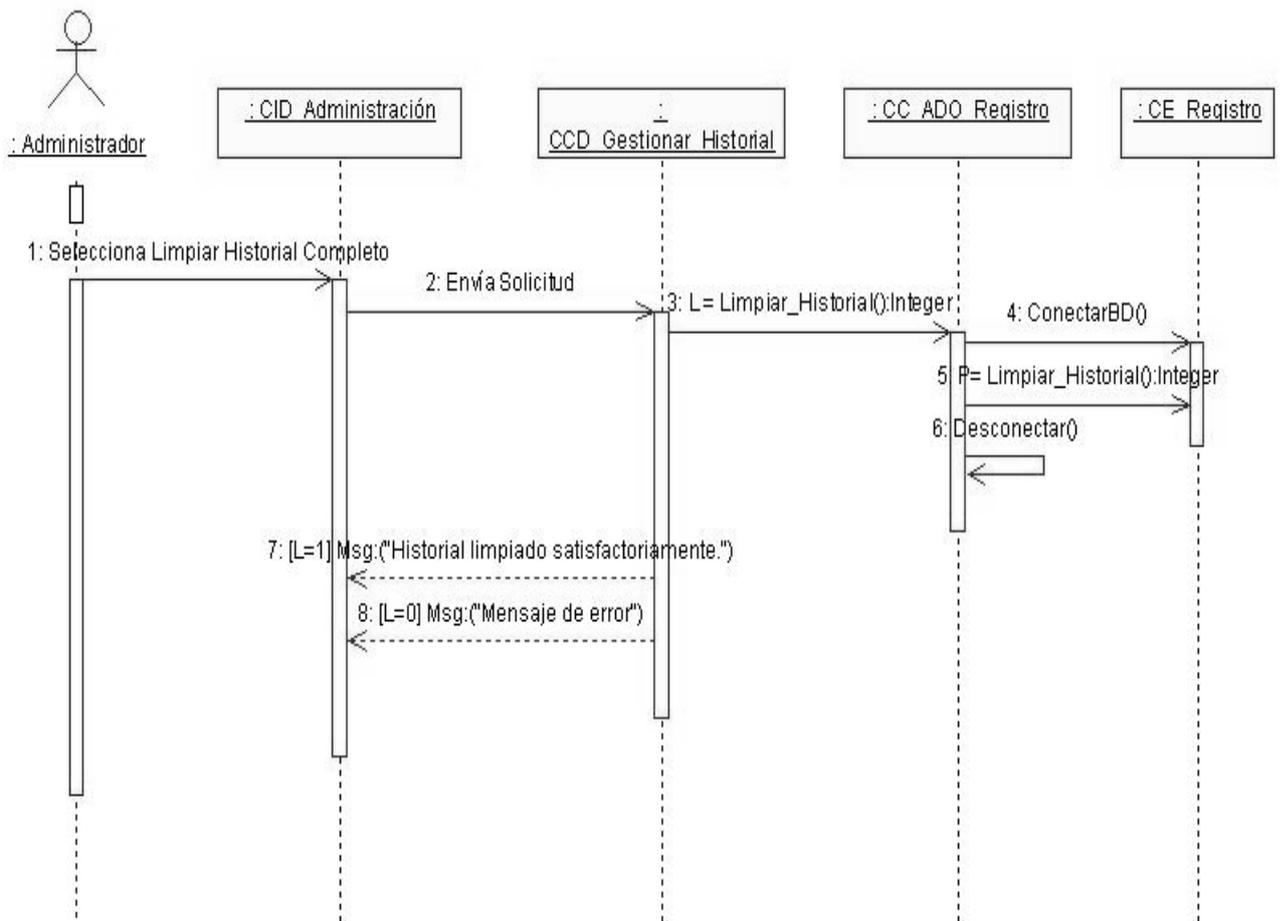


Figura Anexo 1. 7 Diagrama de secuencia del Limpiar Historial Completo, perteneciente al CU Gestionar Historial.

## CU Gestionar Historial: Limpiar Historial Por Fecha.

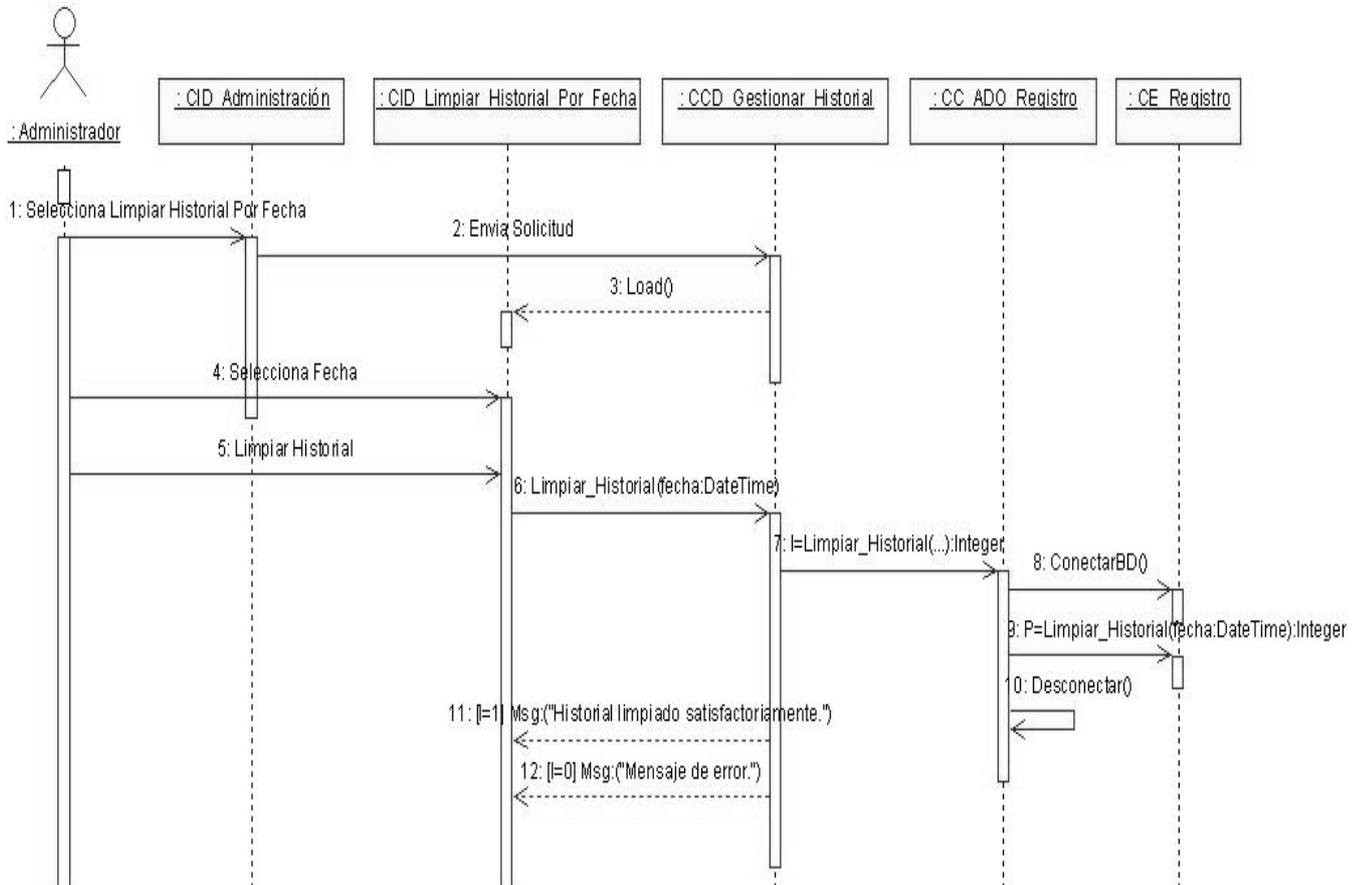


Figura Anexo 1. 8 Diagrama de secuencia del Limpiar Historial Por Fecha, perteneciente al CU Gestionar Historial.

## CU Gestionar Historial: Limpiar Historial Entre Fechas.

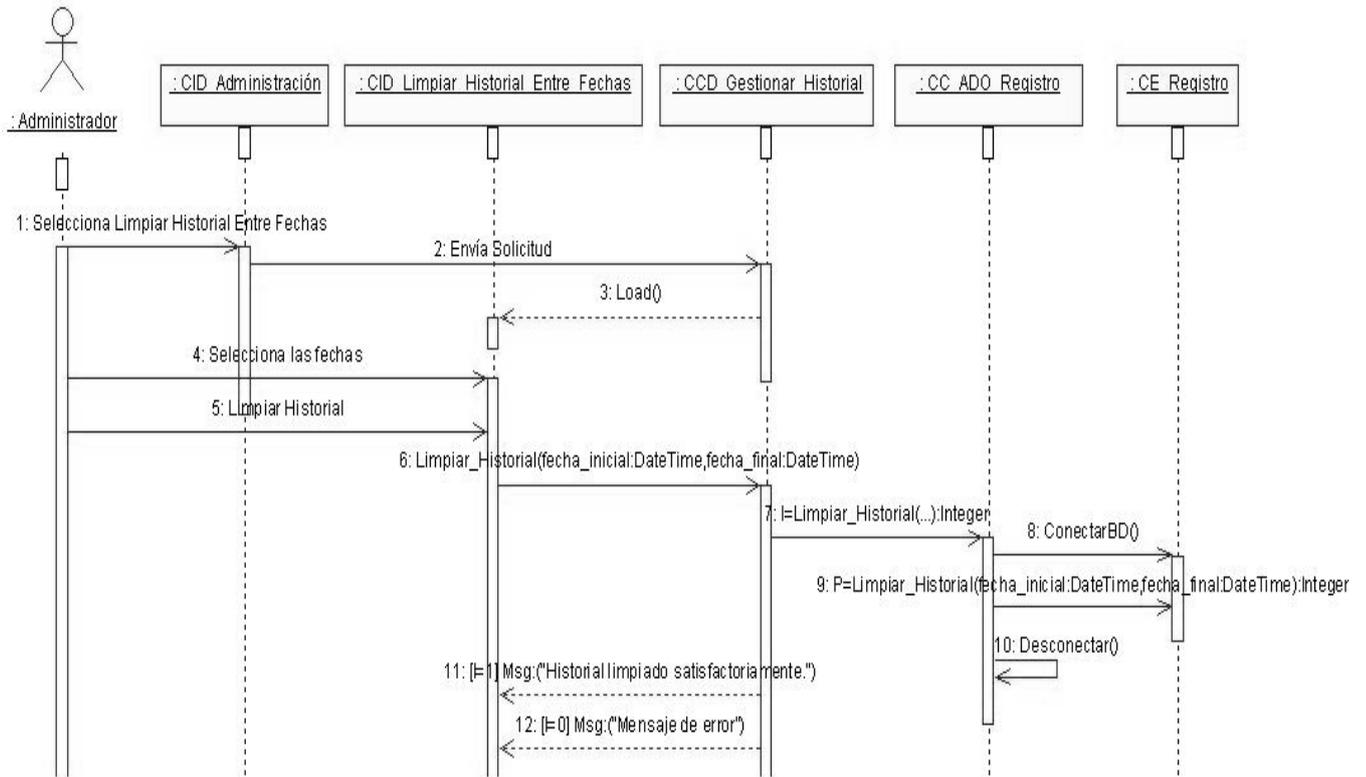


Figura Anexo 1. 9 Diagrama de secuencia del Limpiar Historial Entre Fechas, perteneciente al CU Gestionar Historial.

## CU Gestionar Historial: Mostrar Historial Completo.

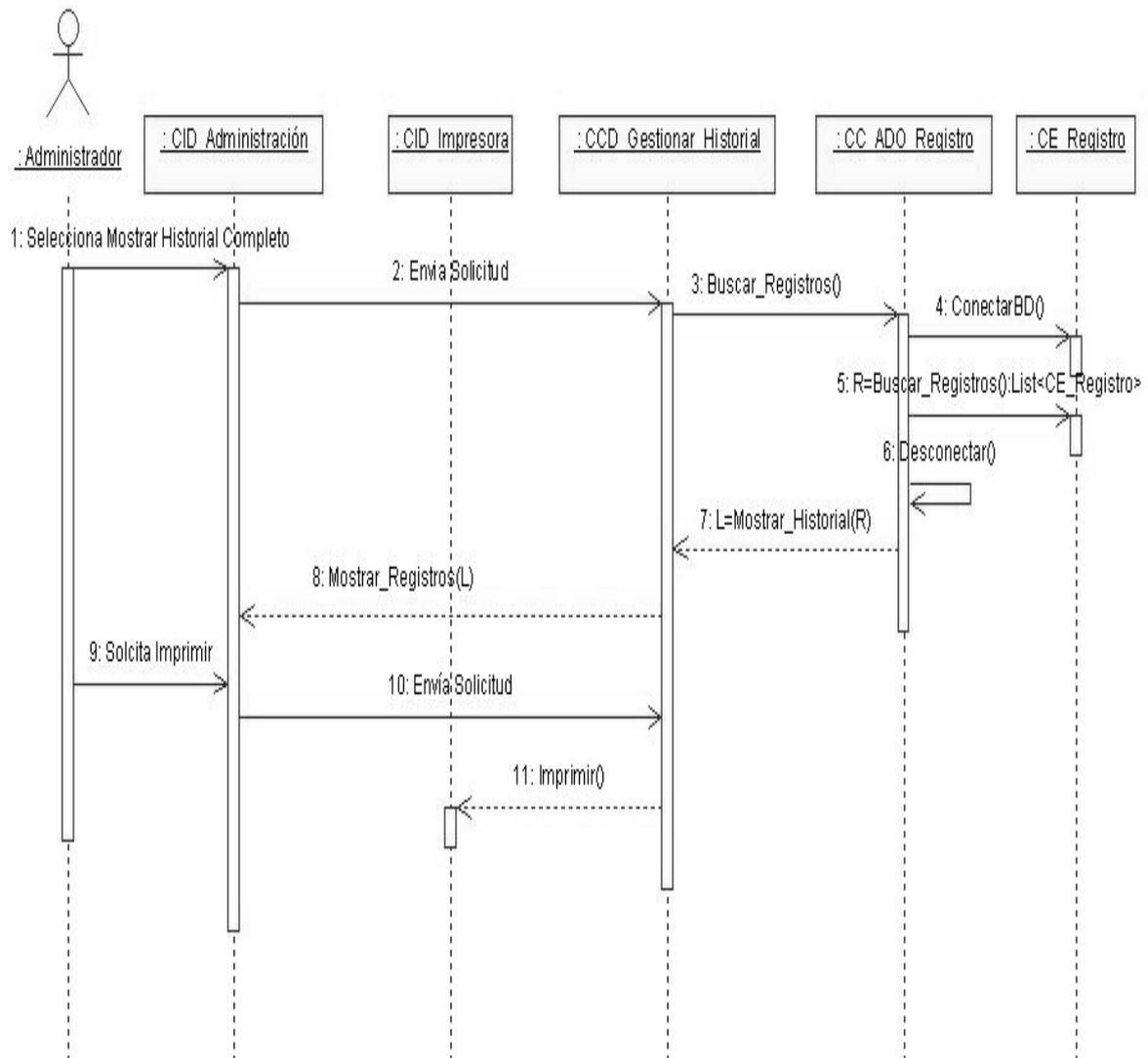


Figura Anexo 1. 10 Diagrama de secuencia del Mostrar Historial Completo, perteneciente al CU Gestionar Historial.

## CU Gestionar Historial: Mostrar Historial Por Fecha.

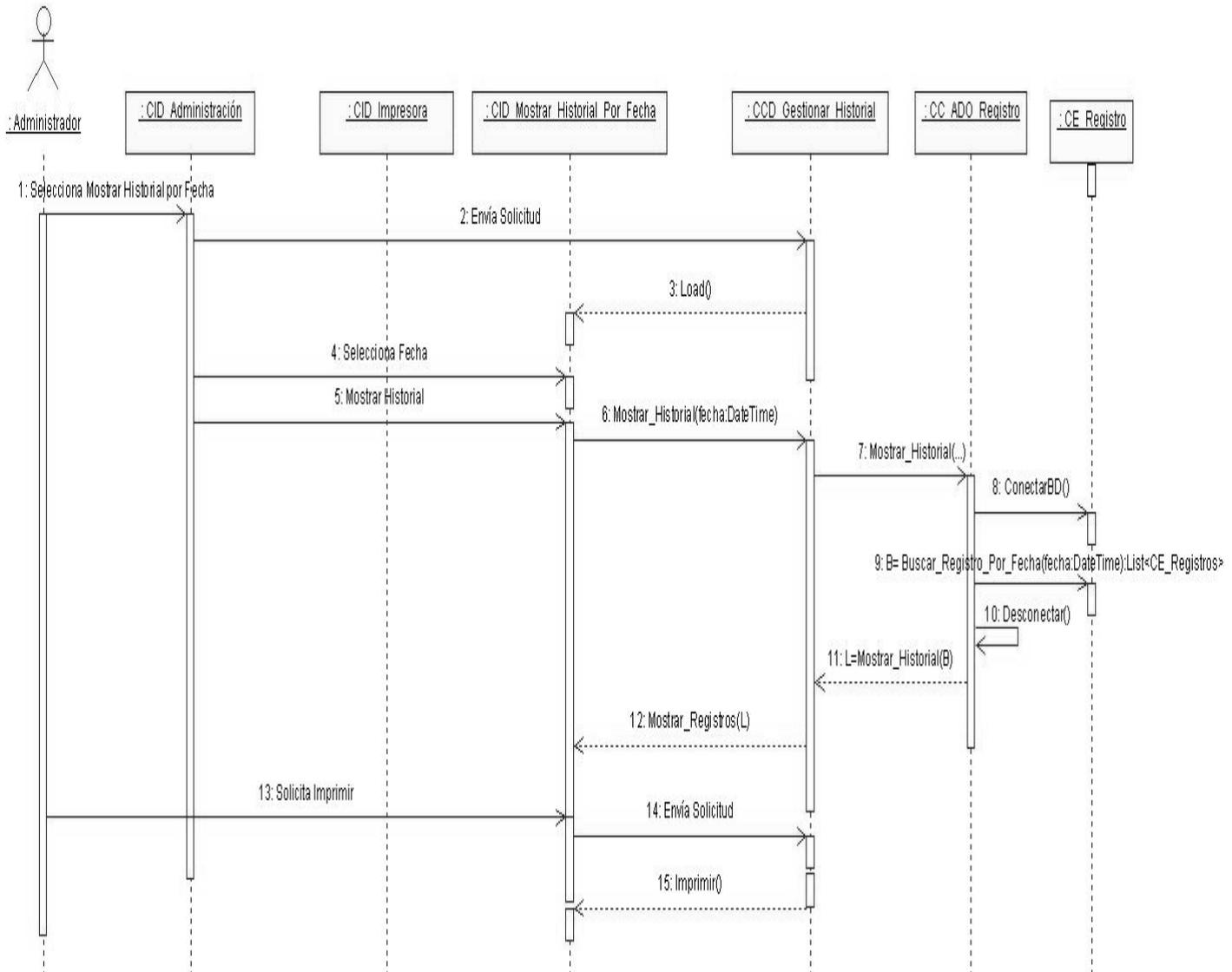


Figura Anexo 1. 11 Diagrama de secuencia del Mostrar Historial Por Fecha, perteneciente al CU Gestionar Historial.

## CU Gestionar Historial: Mostrar Historial Entre Fechas.

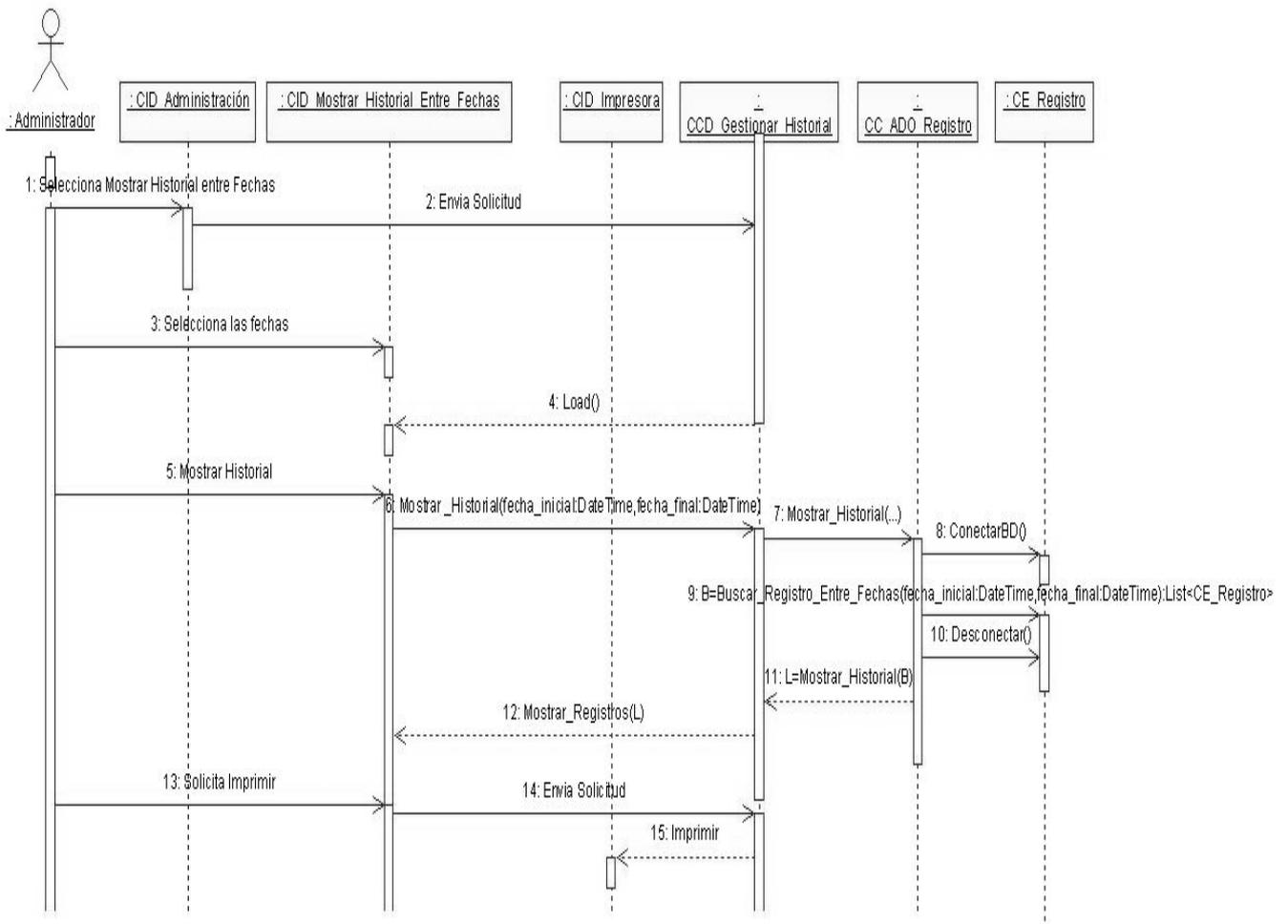


Figura Anexo 1. 12 Diagrama de secuencia del Mostrar Historial Entre Fechas, perteneciente al CU Gestionar Historial.

## Anexo 1: Interfaces.

### CU Autenticar.



Figura Anexo 2. 1 Interfaz del caso de uso Autenticar.

## CU Gestionar Usuario: Adicionar Usuario.



Adicionar Usuario

Crear Usuario

Usuario

Contraseña

Nombre

Primer Apellido

Segundo Apellido

Licencia

Adicionar Usuario Cancelar

Figura Anexo 2. 2 Interfaz del Adicionar Usuario, perteneciente al caso de uso Gestionar Usuario.

## CU Gestionar Usuario: Eliminar Usuario.

Eliminar Usuario

Eliminar Usuario

Lista de Usuarios

Usuario	Nombre	Primer Apellido	Segundo Apellido
ernesto	Ernesto	Avila	Domenech

Buscar Actualizar

Eliminar Usuario Cancelar

Figura Anexo 2. 3 Interfaz del Eliminar Usuario, perteneciente al caso de uso Gestionar Usuario.

## CU Gestionar Usuario: Modificar Usuario.

The screenshot shows a window titled "Modificar Usuario" with a close button in the top right corner. The window is divided into two main sections: "Lista de Usuarios" and "Datos de Usuario".

**Lista de Usuarios:** A table with four columns: "Usuario", "Nombre", "Primer Apellido", and "Segundo Apellido". The first two rows contain data: "yleonpe" (Yoandry Leon Perez) and "ernesto" (Ernesto Avila Domenech). Below the table is a "Seleccionar" button. To the right of the table are two empty input fields, a "Buscar" button, and an "Actualizar" button.

**Datos de Usuario:** A form with the following fields and values: "Usuario" (ernesto), "Primer Apellido" (Avila), "Contraseña" (masked with six asterisks), "Segundo Apellido" (Domenech), "Nombre" (Ernesto), and "Licencia" (4). At the bottom are "Modificar Usuario" and "Cancelar" buttons.

Usuario	Nombre	Primer Apellido	Segundo Apellido
yleonpe	Yoandry	Leon	Perez
ernesto	Ernesto	Avila	Domenech

Figura Anexo 2. 4 Interfaz del Modificar Usuario, perteneciente al caso de uso Gestionar Usuario.

## CU Gestionar Usuario: Cambiar Contraseña.

Cambiar Contraseña

Cambiar Contraseña

Lista de Usuarios

Usuario	Nombre	Primer Apellido	Segundo Apellido
yleonpe	Yoandry	Leon	Perez
ernesto	Ernesto	Avila	Domenech

Buscar Actualizar

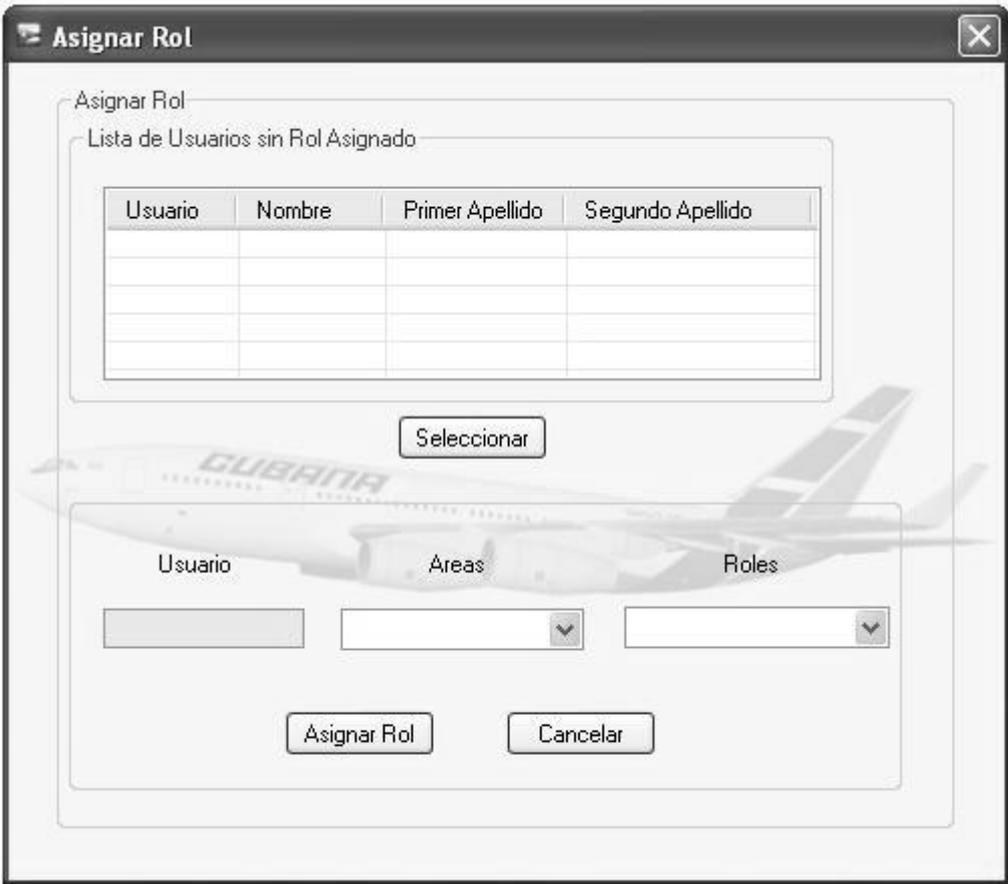
Seleccionar

Usuario Nueva Contraseña Repetir Contraseña

Cambiar Contraseña Cancelar

Figura Anexo 2. 5 Interfaz del Cambiar Contraseña, perteneciente al caso de uso Gestionar Usuario.

## CU Gestionar Usuario: Asignar Rol.



Asignar Rol

Asignar Rol

Lista de Usuarios sin Rol Asignado

Usuario	Nombre	Primer Apellido	Segundo Apellido

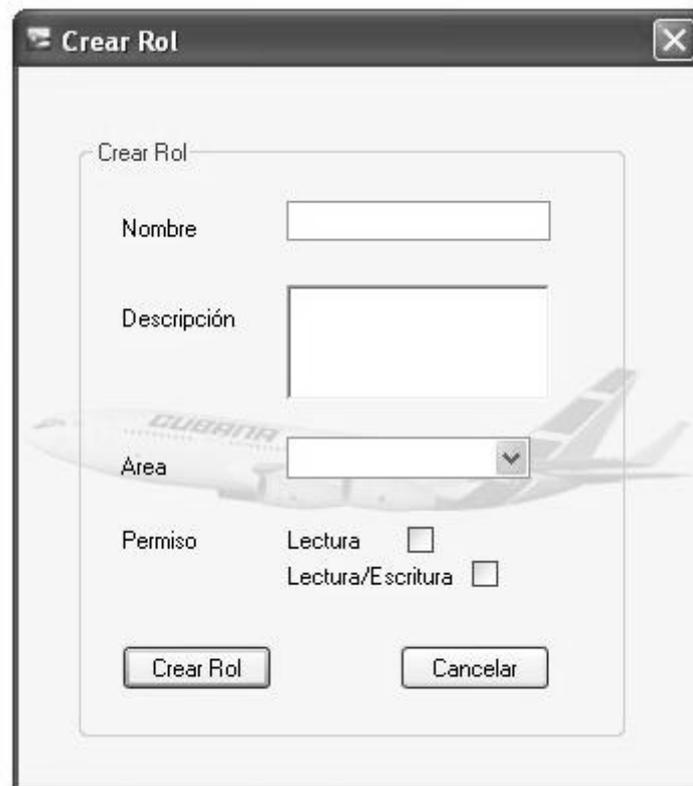
Seleccionar

Usuario Areas Roles

Asignar Rol Cancelar

Figura Anexo 2. 6 Interfaz del Asignar Rol, perteneciente al caso de uso Gestionar Usuario.

## CU Gestionar Rol: Crear Rol.



Crear Rol

Nombre

Descripción

Area

Permiso Lectura

Lectura/Escritura

Crear Rol Cancelar

Figura Anexo 2. 7 Interfaz del Crear Rol, perteneciente al caso de uso Gestionar Rol.

## CU Gestionar Rol: Modificar Rol.

Modificar Rol

Lista de Roles

Nombre	Descripcion
Administrador	Acceso total

Nombre de Rol

Buscar Actualizar

Seleccionar

Figura Anexo 2. 8 Interfaz del Modificar Rol, perteneciente al caso de uso Gestionar Rol.

## CU Gestionar Rol: Eliminar Rol.

Eliminar Rol

Lista de Roles

Nombre	Descripcion
Administrador	Acceso total

Nombre de Rol

Buscar Actualizar

Eliminar Rol Cancelar

Figura Anexo 2. 9 Interfaz del Eliminar Rol, perteneciente al caso de uso Gestionar Rol.

## CU Gestionar Historial: Mostrar Historial por Usuario.



The screenshot shows a window titled "Por Usuario" with a close button in the top right corner. Inside the window, there is a section labeled "Lista de Usuarios" containing a table with the following data:

Usuario	Nombre	Primer Apellido	Segundo Apellido
yleonpe	Yoandry	Leon	Perez

To the right of the table is a search area with a dropdown menu, an empty text input field, and two buttons labeled "Buscar" and "Actualizar". At the bottom of the window are two buttons labeled "Ver Historial" and "Cancelar".

Figura Anexo 2. 10 Interfaz del Mostrar Historial por Usuario, perteneciente al caso de uso Gestionar Historial.

## CU Gestionar Historial: Mostrar por Fecha.



The screenshot shows a window titled "Por Fecha" with a close button in the top right corner. Inside the window, there is a section labeled "Reportes por Fecha" containing a date selection dropdown menu showing "Saturday , June 07, 2008". Below the dropdown are two buttons labeled "Ver Historial" and "Cancelar".

Figura Anexo 2. 11 Interfaz del Mostrar Historial por Fecha, perteneciente al caso de uso Gestionar Historial.

## CU Gestionar Historial: Mostrar entre Fechas.

Entre Fechas

Reportes entre Fechas:

7/2008 6/ 7/2008

Fecha Inicial Fecha Final

Ver Historial Cancelar

Figura Anexo 2. 12 Interfaz del Mostrar Historial entre Fechas, perteneciente al caso de uso Gestionar Historial.

## CU Gestionar Historial: Limpiar Historial por Usuario.

Eliminar por Usuarios

Lista de Usuarios

Usuario	Nombre	Primer Apellido	Segundo Apellido
yleonpe	Yoandry	Leon	Perez

Buscar Actualizar

Limpiar Historial Cancelar

Figura Anexo 2. 13 Interfaz del Limpiar Historial por Usuario, perteneciente al caso de uso Gestionar Historial.

## CU Gestionar Historial: Limpiar Historial por Fecha.



Figura Anexo 2. 14 Interfaz del Limpiar Historial por Fecha, perteneciente al caso de uso Gestionar Historial.

## CU Gestionar Historial: Limpiar Historial entre Fechas.



Figura Anexo 2. 15 Interfaz del Limpiar Historial entre Fechas, perteneciente al caso de uso Gestionar Historial.