

Universidad de las Ciencias Informáticas
Facultad 4



Título: Análisis y Diseño de un sistema para la planificación automatizada del Horario Docente de la facultad 4.

Trabajo de Diploma para optar por el título de
Ingeniero Informático

Autora: Maylen Ortiz Ramos

Tutor: Ing. Yuliesky Torres Iglesias

Ciudad de la Habana, junio 2008

“Año del 50 Aniversario del triunfo de la Revolución”

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Maylen Ortiz Ramos

Ing. Yuliesky Torres Iglesias

Firma del Autor

Firma del Tutor

DATOS DE CONTACTO

Tutor: Ing. Yuliesky Torres Iglesias

- ✓ Graduado de Ingeniero Informático en el 2005.
- ✓ Profesor instructor con tres años de experiencia docente en la Universidad de las Ciencias Informáticas (UCI).

Correo electrónico: ytorresi@uci.cu

AGRADECIMIENTOS

Muchos han sido los que me han ayudado en el transcurso de estos cinco años, de una u otra forma, para llegar hasta aquí. Desafortunadamente no tengo espacio para mencionarlos a todos de forma individual, pero quiero que sepan que estaré eternamente agradecida, muy en especial:

A mis padres por ser los mejores del mundo, por darme los mejores consejos, por quererme, por confiar siempre en mí, por luchar y sacrificarse tanto para verme graduada, los amo mucho.

A mis hermanitos por estar presentes en todo momento, los adoro.

A mi abuela Biche por cuidarme, por preocuparse por mí y por darme siempre lo mejor y a mi abuela Mita que aunque ya no está presente se que estará muy orgullosa de mí.

A yaya, tine, keke, vive y nany que son mi segunda familia, por cuidarme mucho desde que era una niña y por todo el amor y el cariño que siempre me han brindado.

A mi prima Arianna por ser mi hermana, mi amiga y por estar siempre presente cuando más la necesito.

A Rosy, Rubén y mi tía Teté por demostrarme siempre que podía contar con ellos y por su ayuda.

A mi familia en general.

A mis amigos del pre (Diana y Pepe), por todos los momentos que pasamos juntos, por su eterna amistad.

A las chicas del 12 26 (Delmys, Lyly, Yudi, Carmen y Yuri), las otras locas (Cary, Annelis, Yadira, y Mairelis) y a todos mis compañeros de aula, por toda la ayuda prestada, por todos los momentos lindos que pasamos, los voy a extrañar mucho.

A Giselle y Alionuská por marcar la diferencia, por ser tan buenas amigas y por su apoyo incondicional, las quiero mucho.

A mi novio por enseñarme a vivir cada momento, por confiar en mí, por estar presente cuando más lo necesito y por comprenderme siempre, te amo papi. A su familia por todo el apoyo que me han dado y por aceptarme tal como soy.

A Hermes por toda la ayuda prestada en la realización de mi tesis, por dedicarme tanto tiempo y a Ortiz que en tan poco tiempo de amistad ha logrado ocupar un lugar muy importante en mi corazón, gracias por todo.

A Mireya y a su familia por brindarme su apoyo y por cuidar a mi papá, eso me hace muy feliz.

A mi tutor por todas las horas de dedicación.

A la Revolución, a nuestro comandante Fidel y a la UCI por permitirme formar parte del proyecto futuro y por formarme como profesional.

“A todos, Muchas Gracias”

DEDICATORIA

A mis padres que han esperado este día tanto como yo, por todo el amor que me dan y porque sé que están muy orgullosos de mi, los quiero más que a nada en el mundo.

RESUMEN

Nuestra universidad juega un papel importante en el desarrollo de la Industria Cubana del Software, y en la materialización de los proyectos asociados al programa cubano de informatización. Uno de los sectores más importantes en Cuba es la educación, es por ello que en nuestra universidad se está trabajando en base a la informatización del mismo y de todos los componentes necesarios, para proporcionarle a las escuelas del país un mejor funcionamiento.

Es por esto que en la facultad 4 surge la idea de crear un sistema capaz de generar automáticamente el horario docente, que a su vez ayude a la toma de decisiones en este ámbito y que permita realizar una distribución certera de los locales docentes, regular los horarios de los profesores y estudiantes, además de tener en cuenta las frecuencias docentes de cada asignatura, puesto que todos estos factores atentan contra la buena planificación de la docencia en la facultad 4.

En el presente trabajo se realizó un estudio sobre los sistemas de planificación de horarios docentes así como de las tecnologías y herramientas utilizadas en su desarrollo. Se realizó el análisis y diseño del sistema de planificación de horarios utilizando RUP como metodología de desarrollo.

Con el desarrollo de este sistema se espera que los involucrados en este proceso cuenten con una herramienta de trabajo que sea capaz de realizar una rápida planificación del horario docente y que ayude en la toma de decisiones en este ámbito.

PALABRAS CLAVE

Planificación, Horario Docente, Análisis, Diseño.

TABLA DE CONTENIDOS

INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	3
Introducción	3
1.1 Sistemas de Planificación	3
1.2 Planificación de horarios	4
1.2.1 Estudio de Sistemas de planificación de Horarios.	4
1.3 Herramientas para el desarrollo del sistema	6
1.3.1 Estudios de Plataformas de Desarrollo	6
1.3.2 Estudio de algunos lenguajes de programación	9
1.3.3 Herramientas de desarrollo	11
1.3.4 ¿Qué metodología de desarrollo debo utilizar para el desarrollo del software?	12
1.3.5 Lenguaje de Modelado	16
1.3.6 Herramienta de modelación visual.....	16
1.3.7 Sistema de gestión de Base Datos.....	17
Conclusiones	19
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA	20
Introducción	20
2.1 Descripción del negocio actual	20
2.2 Reglas del negocio	21
2.3 Actores del negocio	22
2.4 Diagrama de casos de uso del negocio	22
2.5 Trabajadores del negocio	22
2.6 Realización de los Casos de uso del negocio	23
2.7 Requisitos	26
2.7.1 Requisitos funcionales.....	26
2.7.2 Requisitos no funcionales.....	29
2.8 Actores del sistema	30
2.9 Diagrama de Casos de Uso del Sistema	30
2.10 Lista de Casos de Uso del Sistema y su Prioridad	31
2.10.1 Descripciones de los Casos de Uso del Sistema.....	32
Conclusiones	39
CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA	40

Introducción	40
3.2 Análisis del Sistema	41
3.2.1 Diagramas de clases del análisis.....	41
3.3 Diseño del Sistema	43
3.3.1 Diagramas de Clases de Diseño	43
3.3.2 Diagramas de Interacción.....	45
3.3.3 Diagrama de Clases del Dominio	48
3.4 Diseño de la Base de Datos	49
3.4.1 Modelo Físico de Datos	49
3.4.2 Descripción de las tablas de la base de datos.	49
3.5 Interfaz de usuario	55
3.6 Tratamiento de errores	56
Conclusiones	56
CONCLUSIONES	57
RECOMENDACIONES	58
BIBLIOGRAFÍA	59
TRABAJOS CITADOS	60
ANEXOS	62
Anexo 1: Descripción de los casos de uso del sistema.	62
Anexo 2: Diagramas de clases del análisis.	72
Anexo 3: Diagramas de clases del diseño.	75
Anexo 4: Diagramas de secuencia.	78
GLOSARIO	87

INTRODUCCIÓN

Una sociedad que aplique la informatización en todas sus esferas y procesos será más eficaz, eficiente y competitiva. Es evidente que para los países subdesarrollados resulta un reto el logro de este propósito, ya que su problemática fundamental está en lograr la supervivencia de sus pueblos. (MIC, 2002).

Cuba ha identificado desde muy temprano la conveniencia y necesidad de dominar e introducir en la práctica social las Tecnologías de la Información y las Comunicaciones (TIC); y lograr una cultura digital como una de las características imprescindibles del hombre nuevo, lo que facilitaría a nuestra sociedad acercarse más hacia el objetivo de un desarrollo sostenible. (MIC, 2002).

La Industria Cubana del Software (ICSW) está destinada a convertirse en una significativa fuente de ingresos para el país, como resultado del correcto aprovechamiento de las ventajas del considerable capital humano disponible. Nuestra universidad juega un papel importante en el desarrollo de la Industria Cubana del Software, y en la materialización de los proyectos asociados al programa cubano de informatización. Uno de los sectores más importantes en Cuba es la educación, es por ello que en nuestra universidad se está trabajando en base a la informatización del mismo y de todos los componentes necesarios, para proporcionarle a las escuelas del país un mejor funcionamiento.

Es por esto que en la facultad 4 surge la idea de crear un sistema capaz de generar automáticamente el horario docente, que a su vez ayude a la toma de decisiones en este ámbito y que permita realizar una distribución certera de los locales docentes, regular los horarios de los profesores y estudiantes, además de tener en cuenta las frecuencias docentes de cada asignatura, puesto que todos estos factores atentan contra la buena planificación de la docencia en la facultad 4.

Este trabajo surge como necesidad de resolver el siguiente **problema científico** ¿cómo realizar el análisis y diseño de un sistema para la planificación automática del horario docente en la facultad 4 que permita realizar una certera distribución de los locales docentes y regular los horarios de los profesores y estudiantes?

Con vista a la solución del problema antes planteado se define como **objeto de estudio** sistemas de planificación de horarios docentes.

Definiendo así el **campo de acción** como los procesos de planificación del horario docente de la facultad 4.

El **objetivo general** del presente trabajo es desarrollar el análisis y diseño de una herramienta informática que permita realizar, de forma automática, la planificación del horario docente de la facultad 4.

Objetivos específicos:

- ✓ Realizar una investigación sobre los procesos de planificación del horario docente de la facultad 4.
- ✓ Definir los procesos del sistema de planificación del horario docente.
- ✓ Realizar el Análisis y Diseño del sistema de planificación del horario docente.

Para llevar a cabo la investigación se proponen las siguientes **tareas de la investigación:**

- ✓ Realizar una investigación sobre los procesos de planificación del horario docente de la facultad 4.
- ✓ Realizar un estudio del arte sobre la planificación de los horarios.
- ✓ Realizar un estudio de las metodologías de desarrollo de software.
- ✓ Analizar los factores de los que depende la planificación del horario docente.
- ✓ Realizar el Análisis y Diseño de una herramienta informática que le dé solución al problema a resolver.

Entre los **aportes prácticos esperados** con el desarrollo de este trabajo se espera que se realice un buen análisis y diseño para su posterior implementación para que los involucrados en este proceso cuenten con una herramienta de trabajo que sea capaz de realizar una rápida planificación del horario docente y que ayude en la toma de decisiones en este ámbito.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Introducción

La Informatización de la Sociedad es el proceso de utilización ordenada y masiva de las Tecnologías de la Información y las Comunicaciones en la vida cotidiana, para satisfacer las necesidades de todas las esferas de la sociedad (MIC, 2002), incluyendo la educación, que constituye una de las más importantes hoy en día.

Partiendo del estudio de la planificación de los horarios docentes desde diferentes puntos de vista se hace necesaria la realización del análisis y diseño de un sistema capaz de realizar una serie de funciones para el mejor desempeño de esta importante esfera, utilizando para ello las herramientas adecuadas para su mejor funcionamiento.

1.1 Sistemas de Planificación

¿Qué es un sistema?

Un sistema es un conjunto de elementos que interactúan entre sí con el fin de apoyar las actividades de una empresa o negocio. (Peralta)

El concepto de sistema tiene dos usos muy diferenciados, que se refieren respectivamente a los sistemas de conceptos y a los objetos reales más o menos complejos y dotados de organización.

¿Qué es planificación?

Es una actividad humana que expresa la voluntad de acción sobre el futuro, determina prioridades, orientaciones y objetivos, y establece vías que conduzcan al cumplimiento de metas previamente trazadas.

El hecho de planificar deriva una serie de ventajas:

- ✓ Contribuye a actividades ordenadas y con un propósito. Todos los esfuerzos están apuntados hacia los resultados deseados y se logra una secuencia efectiva y sincronizada de tales esfuerzos.
- ✓ Señala la necesidad de cambios futuros. Dirige la atención hacia los objetivos. Ayuda a determinar funcionalidades necesarias.
- ✓ Proporciona una base para el control. La planificación y el control son inseparables, ya que son como los gemelos de la administración. Cualquier intento de controlar sin planes carece

de sentido, puesto que no hay forma de que las personas sepan si van en la dirección correcta, a no ser que primero tengan en claro a donde ir. Así, los planes proporcionan los estándares de control.

- ✓ Ayuda a tener siempre presente, por parte de todos los componentes de la organización, los objetivos de esta y la adecuación de ellos al medio, cuando es necesario.

¿Qué es un sistema de planificación?

Podemos decir entonces que un sistema de planificación es un conjunto de elementos que interactúan y se relacionan entre sí, en ellos se determinan prioridades, orientaciones y objetivos que se realizan con el fin de apoyar las actividades de una empresa o negocio.

1.2 Planificación de horarios.

La planificación de la docencia está orientada al desarrollo de Proyectos formativos, a organizar nuestra actuación no como un conjunto de acciones imprevisibles y desconectadas entre sí, sino como la puesta en práctica de un plan bien pensado y articulado. Es justamente por eso que planificar es uno de los compromisos más importantes que se deben asumir en el ámbito universitario. (BERAZA, 2004)

1.2.1 Estudio de Sistemas de planificación de Horarios.

La realización de una buena planificación del horario docente constituye un factor determinante para el buen funcionamiento de un centro docente durante el período de clases, es sumamente importante que este se realice con suficiente tiempo de antelación al comienzo del curso escolar para que todos los estudiantes y profesores tengan conocimiento previo del mismo y puedan planificar sus tareas docentes y extra-docentes.

Generador de Horarios Docentes 7.0.

Generador de Horarios para Centros Docentes es un administrador muy sencillo para la realización de horarios escolares, todo ello atendiendo a preferencias e imposiciones propias.

Para generalizar podemos decir que es un potente motor que genera combinaciones hasta encontrar una válida con nuestras prioridades válidas. También será posible retocar esos horarios una vez hechas las combinaciones, con lo que es difícil que no se amolde a cualquier centro docente.

El Planificador de Horarios consta de tres procesos que se ejecutan en el sistema de forma independiente:

- ✓ **Planificador:** Configura los datos iniciales.
- ✓ **Motor:** Genera y optimiza resultados
- ✓ **Editor:** Presenta y edita resultados

La mayor dificultad que presenta este Generador de Horario es que está limitado a 10 profesores, para tener una mayor cantidad de profesores se debe comprar una licencia. En este sistema se realiza la planificación de la guardia que no es objetivo de nuestro trabajo.

EAH (Elaborador Automático de Horarios)

Es un sistema experto que reproduce el conocimiento adquirido tras años de elaborar los horarios de la Escuela Superior de Informática de la Universidad Europea de Madrid (UEM). El conjunto de restricciones que impone la filosofía de la Universidad introduce complejas limitaciones y conflictos entre asignaturas.

Mejora los mecanismos genéricos de elaboración de horarios por cuanto que permite la introducción de restricciones concretas que determinan la forma del horario resultante.

Los resultados obtenidos cuentan con la aprobación de un experto y se espera obtener en el futuro cercano una versión revisada para ser utilizada en la práctica.

Alternativa para el proceso de planificación de horarios docentes de una Universidad.

Es una alternativa que mejora los procesos manuales de confección de un calendario docente, basada en un modelo computacional que utiliza un algoritmo de búsqueda por satisfacción de restricciones, que es capaz de encontrar una asignación óptima no solo guiándose por restricciones explícitas, sino siguiendo además normas y principios de higiene de la actividad docente que deben cumplirse.

En él se toman en cuenta una serie de requisitos para una mayor aceptación del horario dentro de las que se encuentran:

- ✓ Determinar el nivel de complejidad de las asignaturas.
- ✓ Ubicar las asignaturas de mayor complejidad en horas de alta capacidad. La capacidad se clasifica en: baja durante el primer turno, aparición de fatiga en los últimos turnos y elevada en los intermedios.

- ✓ Alternar clases que provocan una alta tensión intelectual con clases de baja tensión o dirigidas fundamentalmente al sistema muscular: cambio intelectual-físico.
- ✓ Utilización de las horas de la tarde para actividades prácticas, de auto preparación, extra docentes, puesto que el ritmo biológico tiende a disminuir.

En este proceso de planificación, las restricciones en los horarios de los profesores, restricciones en los horarios de los estudiantes, restricciones de locales y dosificaciones no compensadas, son algunos de los factores que influyen en la elaboración del calendario docente y hacen, junto con los aspectos anteriores, que sea un proceso altamente complejo, lento y propenso a fallas de todo género.

Siguiendo este algoritmo realizado por ingenieros de la Universidad de Holguín, se creó un sistema que se puso en práctica en dicha universidad aunque en estos momentos no se está utilizando.

Planificador de Horario en la Universidad de las Ciencias Informáticas.

El profesor David Silva de la facultad # 9 está confeccionando un planificador de horario con el objetivo de disminuir un tanto el trabajo del encargado de confeccionar el horario docente, el mismo ya cuenta con una serie de funcionalidades, incluso en versiones anteriores ya ha sido capaz de generar algún que otro horario, el sistema no está terminado, todavía se trabaja en él para lograr un sistema potente.

Este sistema aun no se ha presentado en ningún evento científico.

1.3 Herramientas para el desarrollo del sistema

Existen muchas herramientas que se utilizan para el desarrollo de sistemas, algunas de ellas han alcanzado mayor popularidad que otras en el mundo de la informática por sus preciadas ventajas que favorecen la realización del software, a continuación se explican algunas:

1.3.1 Estudios de Plataformas de Desarrollo

Plataforma Java

Java es toda una tecnología orientada al desarrollo de software con el cual podemos realizar cualquier tipo de programa. Hoy en día, la tecnología Java ha cobrado mucha importancia en el ámbito de Internet gracias a su plataforma J2EE. Pero Java no se queda ahí, ya que en la industria para dispositivos móviles también hay una gran acogida para este lenguaje.

La tecnología Java está compuesta básicamente por 2 elementos: el lenguaje Java y su plataforma. Con plataforma nos referimos a la máquina virtual de Java (Java Virtual Machine).

Una de las principales características que favoreció el crecimiento y difusión del lenguaje Java es su capacidad de que el código funcione sobre cualquier plataforma de software y hardware. Esto significa que nuestro mismo programa escrito para Linux puede ser ejecutado en Windows sin ningún problema. Además es un lenguaje orientado a objetos que resuelve los problemas en la complejidad de los sistemas, entre otras. (Ciberaula, 2006)

Plataforma .NET

La plataforma .NET es una capa de software que se coloca entre el Sistema Operativo (SO) y el programador y que abstrae los detalles internos del SO. Las características fundamentales de esta plataforma son las siguientes: (Unai Extremo Baigorri, 2002)

- ✓ **Portabilidad:** Debido a la abstracción del programador respecto al SO, una aplicación .NET puede ser ejecutada en cualquier SO de cualquier máquina que disponga de una versión de la plataforma. En estos momentos la plataforma .NET solamente está disponible para la familia Windows aunque se está desarrollando una versión para Linux. (Unai Extremo Baigorri, 2002)
- ✓ **Multilenguaje:** Cualquier lenguaje de programación puede adaptarse a la plataforma .NET y ejecutarse en ella. (Unai Extremo Baigorri, 2002)
- ✓ **Interoperabilidad:** La interoperabilidad entre diferentes trozos de código escritos en diferentes lenguajes es total. (Unai Extremo Baigorri, 2002)

.NET es el encargado de proveer lo que se llama código administrado, es decir, un entorno que provee servicios automáticos al código que se ejecuta. Los servicios son variados:

- ✓ **Cargador de clases:** Permite cargar en memoria las clases.
- ✓ **Compilador MSIL a nativo:** Transforma código intermedio de alto nivel independiente del hardware que lo ejecuta a código de máquina propio del dispositivo que lo ejecuta.
- ✓ **Administrador de código:** Coordina toda la operación de los distintos subsistemas del Common Language Runtime.
- ✓ **Recolector de basura:** Elimina de memoria objetos no utilizados.

Plataforma Mono

El proyecto Mono es una iniciativa comunitaria respaldada por Novell, que hace aún más fácil la construcción y el desarrollo de aplicaciones en Linux y otras plataformas, permitiendo a los desarrolladores llevar el software al mercado más rápidamente y más efectivamente en función del costo. (Novell, 2004)

Mono incluye un compilador C#, un entorno de ejecución (runtime) compatible con .NET y dos conjuntos de APIs (application programming interfaces): un conjunto Mono, que toma las principales ventajas de los servidores y escritorios Linux, y un conjunto de APIs compatible con Microsoft .NET Framework 1.1 que brinda soporte para ASP.NET, ADO.NET y otros componentes. (Novell, 2004)

Adicionalmente el proyecto Mono ofrece:

- ✓ Un entorno avanzado de desarrollo para escribir aplicaciones Linux con productividad sin precedentes. (Novell, 2004)
- ✓ APIs exhaustivos para la entrega de completas aplicaciones para cliente, aplicaciones para servicios Web y para servidores. (Novell, 2004)
- ✓ Herramientas altamente efectivas que facilitan la creación de APIs de productos y kits de desarrollo de software (SDK) que son independientes de lenguaje entre sistemas operativos múltiples. (Novell, 2004)
- ✓ Soporte para varios lenguajes tales como VisualBasic, Python, JScript y Java. Mono incluye soporte Java, mediante el proyecto de código abierto iKVM, haciendo de este popular lenguaje un vehículo capaz de apalancar la tecnología Mono. (Novell, 2004)

¿Por qué Plataforma Mono?

Mono es una plataforma eficaz y sencilla para el desarrollo de aplicaciones, para portar soluciones ya existentes para Windows al mundo Linux, y en general como una excelente herramienta para el desarrollo de programas multiplataforma y servicios Web. Es sin duda una plataforma que será crucial para que las empresas puedan migrar sus aplicaciones al mundo Linux y para que nuevas aplicaciones puedan ser desarrolladas con el menor coste posible.

1.3.2 Estudio de algunos lenguajes de programación

C#

C#, es actualmente uno de los lenguajes de programación más populares en informática y comunicaciones.

Permite a los programadores abordar el desarrollo de aplicaciones complejas con facilidad y rapidez. Es como si tomáramos todas las cosas buenas de Visual Basic y las añadiéramos a C++, aunque recortando algunas de las tradiciones más ocultas y difíciles de conocer de C y C++. (Sierra, 2001)

Con C# no sólo se pueden escribir programas para la Web, sino que también permite desarrollar aplicaciones de propósito general, finalidad de esta obra (Sierra, 2001). Se basa en las lecciones aprendidas de los lenguajes C, C++, Java y Visual Basic. Por ello se trata de un lenguaje que combina todas las cualidades que se pueden esperar de un lenguaje moderno (orientación a objetos, gestión automática de memoria, etc.) a la vez que proporciona un gran rendimiento. (Cañar)

Algunas de las principales características de C# se explican a continuación.

- ✓ **Sencillez:** C# elimina muchos elementos que otros lenguajes incluyen y que son innecesarios en .NET. (Seco, 2002)
- ✓ **Orientación a objetos:** Como todo lenguaje de programación de propósito general actual, C# es un lenguaje orientado a objetos. (Seco, 2002)
- ✓ **Eficiente:** En principio, en C# todo el código incluye numerosas restricciones para asegurar su seguridad y no permite el uso de punteros. Sin embargo, y a diferencia de Java, en C# es posible saltarse dichas restricciones manipulando objetos a través de punteros. (Seco, 2002)
- ✓ **Compatible:** Para facilitar la migración de programadores, C# no sólo mantiene una sintaxis muy similar a C, C++ o Java que permite incluir directamente en código escrito en C# fragmentos de código escrito en estos lenguajes, sino que el CLR también ofrece, la posibilidad de acceder a código nativo escrito como funciones sueltas no orientadas a objetos tales como las DLLs de la API Win32. (Seco, 2002)

Java

Java es toda una tecnología orientada al desarrollo de software con el cual podemos realizar cualquier tipo de programa. Hoy en día, la tecnología Java ha cobrado mucha importancia en el ámbito de Internet gracias a su plataforma J2EE. Pero Java no se queda ahí, ya que en la industria para dispositivos móviles también hay una gran acogida para este lenguaje. (Ciberaula, 2006)

Una de las principales características que favoreció el crecimiento y difusión del lenguaje Java es su capacidad de que el código funcione sobre cualquier plataforma de software y hardware. Esto significa que nuestro mismo programa escrito para Linux puede ser ejecutado en Windows sin ningún problema. Además es un lenguaje orientado a objetos que resuelve los problemas en la complejidad de los sistemas, entre otras. (Ciberaula, 2006)

C

C es un lenguaje de programación de propósito general que ofrece economía sintáctica, control de flujo y estructuras sencillas y un buen conjunto de operadores. No es un lenguaje de muy alto nivel y más bien un lenguaje pequeño, sencillo y no está especializado en ningún tipo de aplicación. Esto lo hace un lenguaje potente, con un campo de aplicación ilimitado y sobre todo, se aprende rápidamente. En poco tiempo, un programador puede utilizar la totalidad del lenguaje. (Sequera)

Este lenguaje ha sido estrechamente ligado al sistema operativo UNIX, puesto que fueron desarrollados conjuntamente. Sin embargo, este lenguaje no está ligado a ningún sistema operativo ni a ninguna máquina concreta. Se le suele llamar lenguaje de programación de sistemas debido a su utilidad para escribir compiladores y sistemas operativos, aunque de igual forma se puede desarrollar cualquier tipo de aplicación. (Sequera)

¿Por qué C# como lenguaje de programación?

La razón fundamental de por qué se eligió C# es porque es capaz de utilizar al máximo nivel todo su potencial. La programación en este lenguaje es mucho más sencilla que en otros ya que por ejemplo C# carece de elementos heredados innecesarios de .NET. La sintaxis y estructuración de C# es de fácil entendimiento, además de que se tiene conocimiento de experiencias anteriores.

Además de que ya existe un compilador implementado, el que provee que se generen programas para una plataforma diferente como Linux.

1.3.3 Herramientas de desarrollo

SharpDevelop

SharpDevelop constituye un entorno de desarrollo integrado (IDE, en inglés) para aplicaciones .NET Framework. Soporta el desarrollo de aplicaciones escritas en los lenguajes C, Visual Basic.NET. Esta herramienta de código abierto está elaborada en C. Proporciona todas las características demandadas para un entorno de programación de Windows, como el autocompletado de código, plantillas de proyecto, depurador integrado o diseñador de formularios. Es compatible con Visual Studio Express y Visual Studio 2005, emplea el mismo tipo de formato para los ficheros de proyecto y código fuente.

- ✓ Diseñador de formularios para los lenguajes mencionados anteriormente.
- ✓ Soporte de los Frameworks de .Net 1.1 y 2.0.
- ✓ Depurador de código integrado.
- ✓ Soporte para la edición de XML.
- ✓ Auto inserción de código.
- ✓ Interfaz en múltiples idiomas.
- ✓ Soporte de plantillas de código.
- ✓ Capacidad para escribir código en C#, ASP.NET, ADO.NET, XML y HTML.

MonoDevelop

MonoDevelop es un IDE diseñado principalmente para C# y otros .NET. MonoDevelop permite a los desarrolladores escribir rápidamente aplicaciones de escritorio y Web, como ASP.NET en Linux. MonoDevelop facilita a los desarrolladores realizar aplicaciones .NET creada hasta ahora con Visual Studio y portada a Linux y Mac OS X para mantener una base de código único para las tres plataformas. (Group, 2006)

KDevelop

KDevelop es un entorno de desarrollo integrado que permite la creación y el desarrollo de aplicaciones estándar GNU de una manera fácil, incluso para los principiantes.

El Proyecto KDevelop surgió en 1998 con el fin de desarrollar un IDE (Entorno de desarrollo integrado) fácil de usar para KDE. Desde entonces, el IDE KDevelop está públicamente disponible bajo la GPL y soporta muchos lenguajes de programación.

Es un editor de interfaces gráficas y un asistente que nos ayuda a crear nuestro proyecto con sus esqueletos predefinidos.

¿Por qué MonoDevelop?

Monodevelop es un bonito ambiente de desarrollo específicamente diseñado para la plataforma Mono. Lo novedoso de esta nueva versión es que incluye el paquete Stetic completamente integrado a Monodevelop. Stetic es un módulo que permite crear pantallas en. La diferencia es que Stetic fue diseñado específicamente para Mono y ahora al integrarse a Monodevelop le da a este IDE un ambiente que le resulta bastante familiar a los usuarios de Visual Studio. Monodevelop sigue estando en desarrollo.

1.3.4 ¿Qué metodología de desarrollo debo utilizar para el desarrollo del software?

Todo desarrollo de software es riesgoso y difícil de controlar, pero si no llevamos una metodología de por medio, lo que obtenemos es clientes insatisfechos con el resultado y desarrolladores aún más insatisfechos (Sanchez, 2004). Se debe tener en cuenta además utilizar una metodología adecuada, es decir que sea la más apropiada para el caso que vaya a desarrollar.

Para dar una idea de qué metodología podemos utilizar y cuál se adapta más a nuestro medio, mencionaré tres de ellas de las que se consideran las más importantes, tal como: RUP, XP y MSF.

Rational Unified Process (RUP)

El Proceso Unificado Rational (Rational Unified Process) es la metodología para el desarrollo y construcción de software basado íntegramente en UML como soporte a la metodología.

El ciclo de vida de RUP se caracteriza por:

- ✓ **Dirigido por casos de uso:** Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. A partir de aquí los casos de uso guían el proceso de desarrollo ya que los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso (cómo se llevan a cabo) (UCI, 2007).

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- ✓ **Centrado en la arquitectura:** La arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente. (UCI, 2007)
- ✓ **Iterativo e Incremental:** RUP propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros. Las iteraciones hacen referencia a pasos en los flujos de trabajo, y los incrementos, al crecimiento del producto. (UCI, 2007)

Está compuesto por 4 fases:

- ✓ **Inicio:** Se describe el negocio y se delimita el proyecto describiendo sus alcances con la identificación de los casos de uso del sistema. (UCI, 2007)
- ✓ **Elaboración:** En esta etapa el objetivo es determinar la arquitectura óptima. (UCI, 2007)
- ✓ **Construcción:** Se obtiene un producto listo para su utilización que está documentado y tiene un manual de usuario. Se obtiene 1 o varios release del producto que han pasado las pruebas. Se ponen estos release a consideración de un subconjunto de usuarios. (UCI, 2007)
- ✓ **Transición:** El release ya está listo para su instalación en las condiciones reales. Puede implicar reparación de errores cada una de estas fases es desarrollada mediante el ciclo de iteraciones, la cual consiste en reproducir el ciclo de vida en cascada a menor escala. Los Objetivos de una iteración se establecen en función de la evaluación de las iteraciones precedentes. (UCI, 2007)

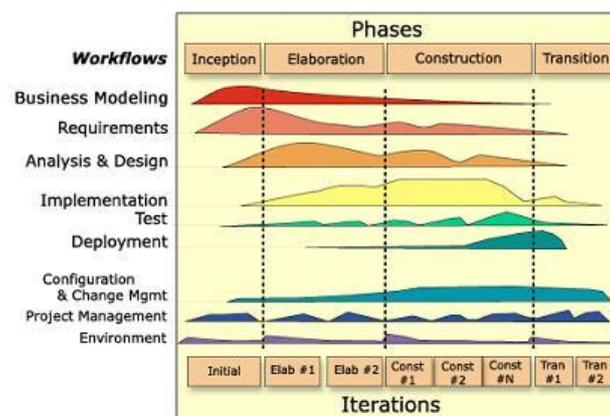


Figura 1.1. Fases e Iteraciones de la Metodología RUP.

Extreme Programming (XP)

Es una de las metodologías de desarrollo de software más exitosas en la actualidad utilizada para proyectos de corto plazo. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto. (Sanchez, 2004)

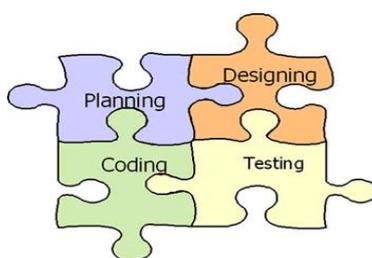


Figura 1.2. Metodología Extreme Programming

Características de XP, la metodología se basa en:

- ✓ **Pruebas Unitarias:** Se basa en las pruebas realizadas a los principales procesos, de tal manera que adelantándonos en algo hacia el futuro, podamos hacer pruebas de las fallas que pudieran ocurrir. Es como si nos adelantáramos a obtener los posibles errores. (Sanchez, 2004)
- ✓ **Refabricación:** Se basa en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio. (Sanchez, 2004)
- ✓ **Programación en pares:** Una particularidad de esta metodología es que propone la programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento. Es como el chofer y el copiloto: mientras uno conduce, el otro consulta el mapa. (Sanchez, 2004)

Microsoft Solution Framework (MSF)

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

centra en los modelos de proceso y de equipo dejando en un segundo plano las elecciones tecnológicas. (Sanchez, 2004)



Figura 1.3. Metodología MSF

MSF tiene las siguientes características:

- ✓ **Adaptable:** Es parecido a un compás, usado en cualquier parte como un mapa, del cual su uso es limitado a un específico lugar. (Sanchez, 2004)
- ✓ **Escalable:** Puede organizar equipos tan pequeños entre 3 o 4 personas, así como también, proyectos que requieren 50 personas a más. (Sanchez, 2004)
- ✓ **Flexible:** Es utilizada en el ambiente de desarrollo de cualquier cliente. (Sanchez, 2004)
- ✓ **Tecnología:** Agnóstica: porque puede ser usada para desarrollar soluciones basadas sobre cualquier tecnología. (Sanchez, 2004)

Dado el estudio anterior de de las diferentes metodologías de desarrollo del software podemos concluir diciendo que:

- ✓ La Metodología RUP es más adaptable para proyectos de largo plazo.
- ✓ La Metodología XP en cambio, se recomienda para proyectos de corto plazo.
- ✓ La Metodología MSF se adapta a proyectos de cualquier dimensión y de cualquier tecnología.

Podemos concluir además, que lo más importante antes de elegir la metodología que se debe usar para la implementación de un determinado software, es determinar el alcance que tendrá y luego de ahí ver cuál es la que más se acomoda en tu aplicación.

¿Por qué RUP?

Decidimos utilizar RUP ya que este proceso está dividido en 4 fases, las mismas se dividen en iteraciones, los artefactos son el objetivo principal de cada actividad, es basado en roles, posee mucha documentación, muy organizativo. Viendo así de una forma más descriptiva el papel que juega cada

objeto en el sistema. Por todo lo antes dicho y basándonos en lo mismo podemos decir que este proceso de desarrollo es el idóneo para llevar a cabo la descripción de este software.

1.3.5 Lenguaje de Modelado

UML

Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables.

Es importante resaltar que UML es un "lenguaje" para especificar y no para describir métodos o procesos. Se utiliza para definir un sistema de software, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo.

¿Por qué UML?

La elección del lenguaje para modelar depende de muchos factores. UML es una notación, es decir, una serie de reglas y recomendaciones para representar modelos. Nos permite documentar y especificar los elementos creados mediante un lenguaje común describiendo modelos. UML surge como respuesta al primer problema identificado para contar con un lenguaje estándar y poder escribir planos de software. Además de que RUP propone como lenguaje de modelado a UML.

1.3.6 Herramienta de modelación visual

Rational Rose Enterprise Edition

Rational Rose Enterprise es el producto más completo de la familia Rational Rose. Todos los productos Rational Rose incluyen soporte Unified Modeling Language (UML). Es la mejor elección para el ambiente de modelado. Proporciona un lenguaje común de modelado para el equipo que facilita la creación de software de calidad más rápidamente. (INNOVA)

Algunas características:

- ✓ Posee una capacidad de análisis de calidad de código.
- ✓ Provee visualización, modelado y las herramientas para desarrollar aplicaciones de Web.
- ✓ Posee un modelado UML para trabajar en diseños de base de datos, con capacidad de representar la integración de los datos y los requerimientos de aplicación a través de diseños lógicos y físicos.

Visual Paradigm

Visual Paradigm ofrece:

- ✓ Entorno de creación de diagramas para UML 2.0.
- ✓ Diseño centrado en casos de uso y enfocado al negocio que genera un software de mayor calidad.
- ✓ Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- ✓ Capacidades de ingeniería directa (versión profesional) e inversa.
- ✓ Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- ✓ Disponibilidad de múltiples versiones, para cada necesidad.
- ✓ Disponibilidad en múltiples plataformas.

¿Por qué Rational Rose?

Rational Rose tiene todas las características que los desarrolladores, analistas, y arquitectos exigen: soporte UML incomparable, completo soporte al equipo, desarrollo basado en componentes con soporte para arquitecturas líderes en la industria, facilidad de uso, integración optimizada, entre otras características que conllevan a que Rational sea reconocido como el líder tecnológico por su rol en el desarrollo del UML.

1.3.7 Sistema de gestión de Base Datos.

Los Sistemas de gestión de base de datos son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta.

El propósito general de los sistemas de gestión de base de datos es el de manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante, para un buen manejo de datos.

Entre los tipos de base de datos se encuentran:

MySQL

MySQL es un sistema de administración relacional de bases de datos. Una base de datos relacional archiva datos en tablas separadas en vez de colocar todos los datos en un gran archivo. Esto permite velocidad y flexibilidad. Las tablas están conectadas por relaciones definidas que hacen posible combinar datos de diferentes tablas sobre pedido.

MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones

En concreto, usar MySQL tiene ventajas adicionales: (Pozo, 2004)

- ✓ Escalabilidad: Es posible manipular bases de datos enormes, del orden de seis mil tablas y alrededor de cincuenta millones de registros, y hasta 32 índices por tabla.
- ✓ MySQL está escrito en C y C++ y probado con multitud de compiladores y dispone de APIs para muchas plataformas diferentes.
- ✓ Es multihilo, con lo que puede beneficiarse de sistemas multiprocesador.
- ✓ Permite manejar multitud de tipos para columnas.
- ✓ Permite manejar registros de longitud fija o variable.

PostgreSQL

PostgreSQL es un servidor de base de datos objeto relacional libre, es un motor de base de datos. PostgreSQL es multiplataforma, diseñado para ambientes de alto volumen. Está disponible en casi cualquier Unix (34 plataformas en la última versión estable), usa una estrategia de almacenamiento de filas llamada MVCC para conseguir una mejor respuesta en ambientes de grandes volúmenes. Presenta la desventaja de consumir bastante recursos y carga más el sistema.

Posee alta concurrencia, claves ajenas también denominadas llaves ajenas o llaves foráneas (foreign keys), disparadores, vistas, integridad transaccional, herencia de tablas.

SQL Server

SQL Server 2000 es un potente motor de bases de datos de alto rendimiento capaz de soportar millones de registros por tabla con un interface intuitivo y con herramientas de desarrollo integradas como Visual Studio 6.0 o .NET, además incorpora un modelo de objetos totalmente programable (SQL-DMO) con el que podemos desarrollar cualquier aplicación que manipule componentes de SQL Server, es decir, hacer aplicación para crear bases de datos, tablas, backups, etc., todo lo que se puede hacer desde el administrador del SQL Server y podemos hacerlo no solo en Visual C++ sino también en Visual Basic, ASP y por supuesto en .NET. (Netveloper, 2002)

Proporciona un rendimiento, fiabilidad y escalabilidad excepcionales; facilita la migración y ofrece ventajas de Coste Total de Propiedad para las más duras condiciones de carga de trabajo de las empresas actuales.

¿Por qué MySQL?

El principal objetivo de diseño de MySQL fue la velocidad, consume muy pocos recursos, tanto de CPU como de memoria, además de ser rápido, robusto, multitarea, multiusuario tiene un enfoque relacional. Es seguro y usa Listas de Control de acceso para todas las conexiones, consultas y otras operaciones. Trabaja en diferentes plataformas, soporta múltiples idiomas. Está completo y optimizado. Tiene una buena integración con MonoDevelop, también tiene un control de acceso de los de los usuarios bastante amplio. No tiene límites en el tamaño de los registros. Posee extraordinarias utilidades de administración como backup y recuperación de errores.

Conclusiones

Después de realizar un estudio sobre los Sistemas de Planificación Docentes a nivel mundial y nacional, teniendo en cuenta el estudio realizado de algunos sistemas de Planificación de Horarios Docentes, además de realizar las investigaciones pertinentes para escoger las herramientas, metodologías y plataforma adecuadas para la realización del software, se puede concluir que lo estudiado ayuda a determinar las responsabilidades que debe incluir nuestro sistema, contando de esta manera con los elementos necesarios para realizar la propuesta que se presente a los usuarios, con el objetivo de que sea ajustada al sistema que se desea administrar.

Finalmente para el desarrollo del sistema se utilizarán las siguientes tecnologías por las características fundamentadas con anterioridad: la plataforma Mono, como lenguaje de programación (C#), como metodología de desarrollo (RUP), lenguaje de modelado (UML), herramienta de modelado (Racional Rose) y como base datos (MySQL).

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Introducción

En este capítulo se realiza un análisis del proceso que se lleva a cabo para realizar la planificación del horario docente de la facultad 4, para facilitar una mayor comprensión del mismo.

Se presenta el modelo del negocio con el objetivo de entender el plano en que se emplaza el sistema y por tanto contribuir a la comprensión de los requisitos que se desprenden de este contexto. Se determinan los actores y trabajadores del negocio y del sistema; se plantean las principales reglas del negocio a tener en cuenta en el sistema a desarrollar; se muestran los casos de uso del negocio detallados y una representación del modelo de objetos; así como una especificación de los requisitos y casos de usos del sistema generados a raíz de la captura de requisitos.

2.1 Descripción del negocio actual

Con suficiente tiempo de antelación al comienzo de cada semestre la vicedecana de la facultad 4 solicita al planificador la confección del Horario Docente.

Para la confección del horario docente, la vicedecana de Formación entrega al planificador, el Balance de Carga de las Asignaturas, la cantidad de aulas y laboratorios con que cuenta la facultad, la distribución de los profesores por asignaturas y grupos a los que impartirán clases, las sesiones en que recibirá clases cada grupo y cualquier otra afectación que pueda influir en la planificación del horario (los días de Preparación Metodológica de cada disciplina, Consejo Docente, Consejo de Dirección, Consejo de Producción, Consejo de Extensión Universitaria y Beca, Reunión de la Facultad, Reunión del Sindicato, etc.).

Los Jefes de Departamentos de la facultad le entregan a la vicedecana de Formación la distribución de profesores por asignaturas y la cantidad de grupos a los que impartirá clases cada profesor. Teniendo en cuenta estos datos la vicedecana de Formación asigna los grupos a los profesores y entrega esta información al planificador así como las afectaciones de estos.

Después de tener todos estos datos, se realiza la planificación del horario y se envía a la vicedecana de Formación para que sea revisado y apruebe su publicación. Si la vicedecana considera que ya está listo lo envía a los profesores y estudiantes. En caso contrario, el planificador realiza una nueva versión del horario.

2.2 Reglas del negocio

Las reglas de negocio describen políticas que deben cumplirse o condiciones que deben satisfacerse, por lo que regulan algún aspecto del negocio. (UCI, 2007)

En el sistema se han definido las reglas del negocio que se mencionan a continuación:

1. La vicedecana de Formación es la encargada de entregarle al planificador:
 - 1.1. El Balance de Carga de las Asignaturas.
 - 1.2. La relación de profesores por grupos y las asignaturas que imparten así como sus afectaciones.
 - 1.3. Afectaciones que puedan influir sobre la planificación (los días de Preparación Metodológica de cada disciplina, Consejo Docente, Consejo de Dirección, Consejo de Producción, Consejo de Extensión universitaria, Reunión de la Facultad, Reunión del Sindicato, etc.).
 - 1.4. La sesión en que cada grupo debe recibir clases.
2. El Jefe de Departamento es el encargado de entregar a la vicedecana de Formación la distribución de profesores por asignaturas y grupos a los que impartirá clases así como las afectaciones de cada uno de ellos.
3. El Profesor es el encargado de entregarle al Jefe de Dpto. las afectaciones que puedan influir en la planificación del horario.
4. Para poder planificar el horario el Planificador debe tener:
 - 4.1. El Balance de Carga de las Asignaturas por año.
 - 4.2. Los días de Preparación Metodológica de cada disciplina.
 - 4.3. Las afectaciones de los profesores.
 - 4.4. La relación de profesores por grupos y las asignaturas que imparten.
 - 4.5. La sesión en que cada grupo recibirá clases.
 - 4.6. Otras afectaciones que puedan influir sobre la planificación.
5. Todos los turnos de clases de una asignatura no pueden planificarse para un mismo día.
6. Los días de preparación metodológica de una disciplina no pueden ser afectados por turnos de clases de las asignaturas de esa disciplina.
7. No puede planificarse el 3er y 4to turno de clases, el mismo día, a profesores o grupos; pues debe respetarse el horario de almuerzo.
8. Los profesores adjuntos sólo pueden tener clases planificadas los días que su empresa se lo permita.

2.3 Actores del negocio

Un actor del negocio es cualquier individuo, grupo, entidad, organización, máquina o sistema de información externos; con los que el negocio interactúa. Lo que se modela como actor es el rol que se juega cuando se interactúa con el negocio para beneficiarse de sus resultados. (UCI, 2007)

En el negocio analizado existe un actor que se beneficia con el negocio: la vicedecana (Tabla 2.1).

Nombre del Actor	Descripción
Vicedecana	Persona interesada en que se confeccione el horario.

Tabla 2.1. Actores del negocio.

2.4 Diagrama de casos de uso del negocio

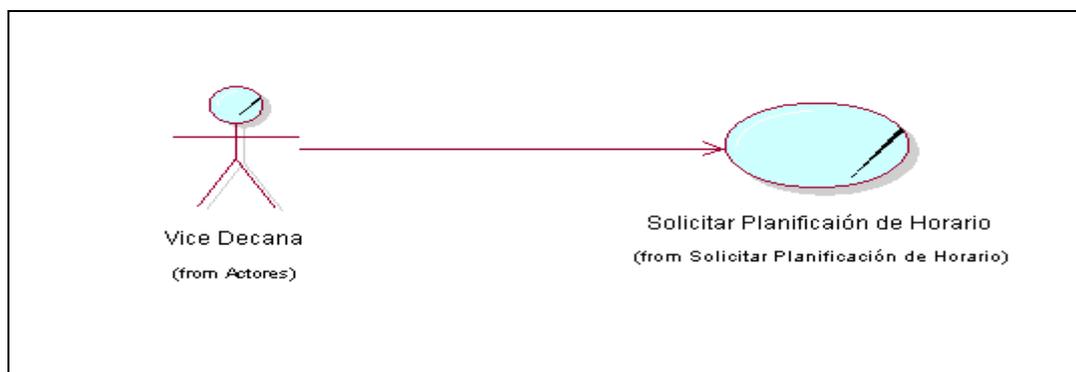


Figura 2.1. Diagrama de casos de uso del negocio.

2.5 Trabajadores del negocio

Un trabajador del negocio es una abstracción de una persona (o grupo de personas), una máquina o un sistema automatizado; que actúa en el negocio realizando una o varias actividades, interactuando con otros trabajadores del negocio y manipulando entidades. Representa un rol. (UCI, 2007)

A continuación se muestra la descripción de los trabajadores identificados en el negocio.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Nombre del Trabajador	Descripción
Planificador	Es el encargado de recopilar toda la información necesaria para confeccionar el horario, realizar la planificación del mismo y entregarlo a la vicedecana de Formación para que esta lo revise y lo envíe a los estudiantes y profesores.
Jefe de Dpto.	Es el encargado de entregar a la vicedecana de Formación la distribución de profesores por asignaturas y los grupos a los que impartirá clases, cada profesor así como las afectaciones de cada uno de ellos.

Tabla 2.2. Trabajadores del negocio.

2.6 Realización de los Casos de uso del negocio

La Realización de un Caso de Uso de negocio muestra como colaboran los trabajadores y entidades del negocio para ejecutar el proceso. Se considera que con una descripción textual y los diagramas de actividades, es suficiente para describir completamente el proceso de negocio.

Con el propósito de lograr una mejor comprensión de los procesos, se especifica la descripción del caso de uso del negocio (Tabla 2.3) y el diagrama de actividades correspondiente al mismo (Figura 2.2).

Descripción del caso de uso del negocio Solicitar Planificación de Horario

Nombre del Caso de uso	Solicitar Planificación de Horario
Actores	Vicedecana (Inicia)
Trabajadores	Planificador y Jefe de Dpto.
Propósito	Facilitarle a la planificadora del horario docente de la facultad 4 una vía más cómoda y eficaz para realizar su trabajo.
Resumen	
El caso de uso se inicia cuando la vicedecana solicita que se confeccione el horario docente. El planificador reúne todos los datos que necesita para confeccionar el horario, lo confecciona y lo	

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

<p>entrega la vicedecana de Formación para que lo revise. La vicedecana de Formación lo revisa y si considera que el horario esta correcto lo envía a los estudiantes y profesores, si no, el planificador confecciona una nueva versión de este.</p>	
Precondiciones	El planificador debe contar con todos los documentos necesarios para la confección del horario.
Casos de Uso Asociados	No tiene
Flujo Normal de Eventos	
Acción del Actor	Respuesta del negocio
1. La vicedecana solicita la confección del horario docente.	2. El planificador solicita Información necesaria.
3. La vicedecana solicita al Jefe de Dpto. las afectaciones y grupos de los profesores y la asignatura que imparten.	4. El Jefe de Dpto. entrega la Información solicitada.
5. La vicedecana recibe la información solicitada. 6. Entrega la información a la planificadora.	7. El planificador recibe la información solicitada. 8. El planificador confecciona el horario docente. 9. El planificador entrega el horario a la vicedecana para que lo revise.
10. La vicedecana recibe el horario. 11. La vicedecana revisa el horario, si está correcto lo envía a los estudiantes y profesores.	
Flujos Alternos	
	11. Si la vicedecana considera que el horario está mal confeccionado le comunica al planificador que debe realizar una nueva versión.
Poscondiciones	Queda confeccionado el horario docente.
Mejoras	La automatización de las actividades llevadas a cabo en este proceso les permitirá al planificador una vía más eficaz para realizar su trabajo.

Tabla 2.3. Descripción del caso de uso del negocio Solicitar Planificación de Horario.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

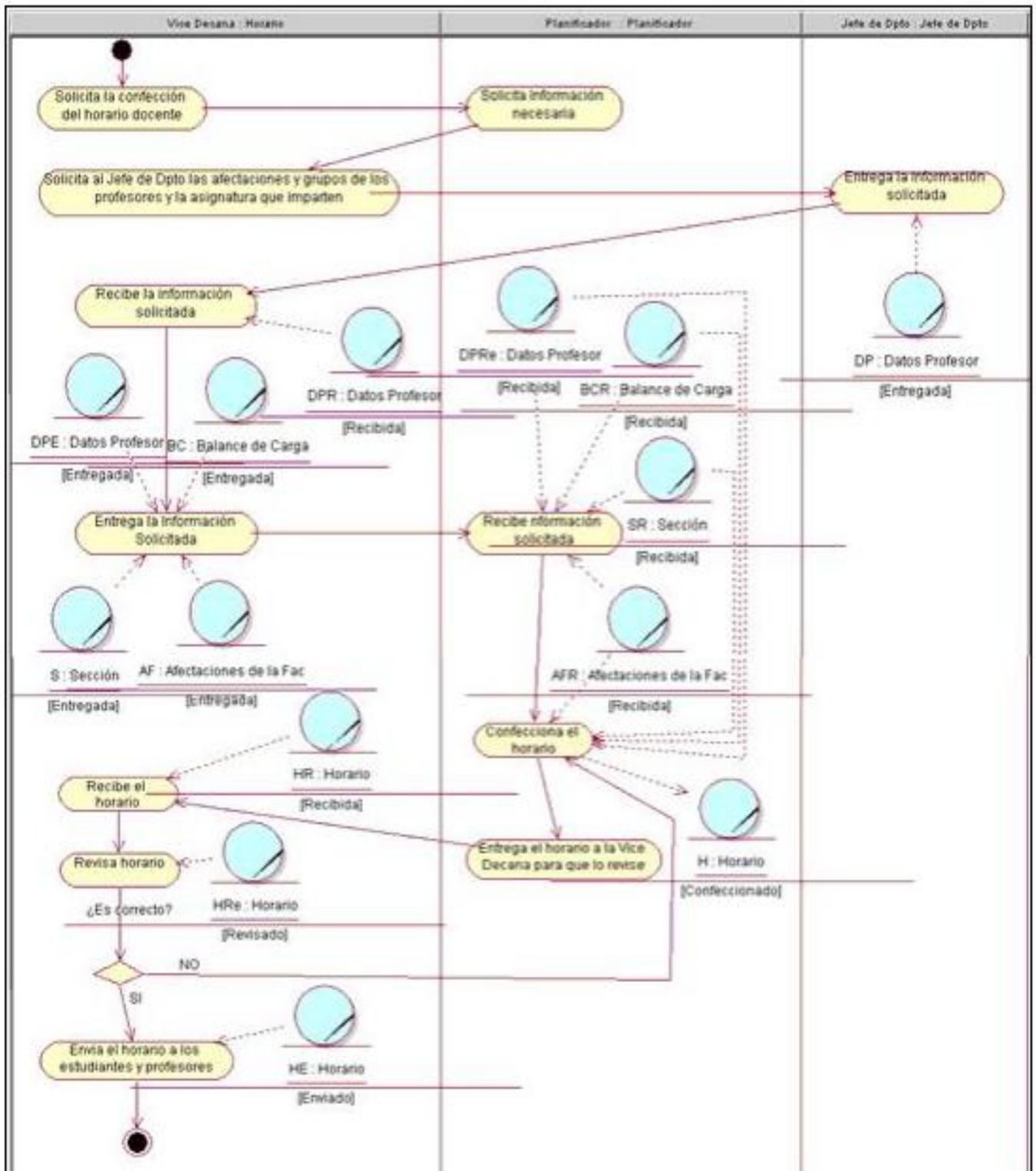


Figura 2.2 Diagrama de actividades del caso de uso del negocio Solicitar Planificación de Horario.

2.7 Requisitos

Condición o capacidad que tiene que ser alcanzada o poseída por un sistema o componente de un sistema para satisfacer un contrato, estándar, u otro documento impuesto formalmente. (UCI, 2007)

Los requisitos se pueden clasificar en: funcionales y no funcionales.

Requisitos funcionales

Son capacidades o condiciones que el sistema debe cumplir.

Los requerimientos funcionales se mantienen invariables sin importar con que propiedades o cualidades se relacionen. (UCI, 2007)

Requisitos no funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. (UCI, 2007)

En la aplicación se definen los siguientes requisitos funcionales.

2.7.1 Requisitos funcionales.

1. Gestionar Información de los profesores

*De los profesores se registrarán los siguientes datos: solapín, nombre, Dpto., categoría (si es interno, externo o adjunto), asignatura que imparte, grupos que imparte clases, fecha afectación, hora inicio afectación, hora fin afectación, tipo de clase.

1.1. Insertar profesor. El sistema debe ser capaz de permitir adicionar un nuevo profesor.

1.2. Modificar profesor. El sistema debe ser capaz de permitir modificar los datos del profesor anteriormente creado.

1.3. Eliminar profesor. El sistema debe ser capaz de permitir eliminar un profesor anteriormente creado.

Seguimiento: CU_Gestionar Profesor.

2. Gestionar Información de las asignaturas

*De cada Asignatura se registran los siguientes datos: id _ asignatura, nombre, nombre corto, año, nivel (alto, medio, bajo), bloqueada, semana que inicia la asignatura bloqueada, cantidad de semanas que se va a impartir esa asignatura bloqueada.

- 2.1. Insertar asignatura. El sistema debe ser capaz de permitir adicionar una nueva asignatura.
- 2.2. Modificar asignatura. El sistema debe ser capaz de permitir modificar los datos de la asignatura anteriormente creada.
- 2.3. Eliminar asignatura. El sistema debe ser capaz de permitir eliminar una asignatura anteriormente creada.

Seguimiento: CU_Gestionar Asignatura.

3. Gestionar información de los locales

*De cada local se registran los siguientes datos: id _ local, nombre, tipo de local (salón, laboratorio o aula), capacidad, piso.

- 3.1. Insertar local. El sistema debe ser capaz de permitir adicionar un nuevo local.
- 3.2. Modificar local. El sistema debe ser capaz de permitir modificar los datos de un local anteriormente creado.
- 3.3. Eliminar local. El sistema debe ser capaz de permitir eliminar un local anteriormente creado.

Seguimiento: CU_Gestionar Local.

4. Gestionar información de los grupos

*De cada grupo se registran los siguientes datos: id _ grupo, nombre, cantidad de estudiantes, año.

- 4.1. Insertar grupo. El sistema debe ser capaz de permitir adicionar un nuevo grupo.
- 4.2. Modificar grupo. El sistema debe ser capaz de permitir modificar los datos de un grupo anteriormente creado.
- 4.3. Eliminar grupo. El sistema debe ser capaz de permitir eliminar un grupo anteriormente creado.

Seguimiento: CU_Gestionar Grupo.

5. Gestionar información de los Dpto.

*De cada Dpto. se registran los siguientes datos: id _ departamento, nombre, día de preparación metodológica.

- 5.1. Insertar Dpto. El sistema debe ser capaz de permitir adicionar un nuevo Dpto.
- 5.2. Modificar Dpto. El sistema debe ser capaz de permitir modificar los datos de un Dpto. anteriormente creado.
- 5.3. Eliminar Dpto. El sistema debe ser capaz de permitir eliminar un Dpto. anteriormente creado.

Seguimiento: CU_Gestionar Dpto.

6. Gestionar información de la semana

*De cada semana se registran los siguientes datos: id _ semana, número, número del semestre.

- 6.1. Insertar semana. El sistema debe ser capaz de permitir adicionar una nueva semana.
- 6.2. Modificar semana. El sistema debe ser capaz de permitir modificar los datos de una semana anteriormente creada.
- 6.3. Eliminar semana. El sistema debe ser capaz de permitir eliminar una semana anteriormente creada.

Seguimiento: CU_Gestionar Semana.

7. Gestionar Balance de Carga

*De cada balance de carga se registran los siguientes datos: asignatura, semana, frecuencias semanales, tipo de clase (clase práctica, laboratorio, conferencia), orden.

- 7.1. Insertar balance de carga. El sistema debe ser capaz de permitir adicionar un nuevo balance de carga.
- 7.2. Modificar balance de carga. El sistema debe ser capaz de permitir modificar los datos del balance de carga anteriormente creado.

Seguimiento: CU_Gestionar Balance de Carga.

8. Gestionar Horario

*Para confeccionar el horario se registran los siguientes datos: profesores, asignaturas, grupos, semanas, locales, balance de carga.

- 8.1. Comprobar la existencia de todas las informaciones existentes en la base de datos necesarias para la confección del horario.
- 8.2. Generar planificación de horario.
- 8.3. Almacenar en la base de datos la planificación del horario.
- 8.4. Mostrar horario.

Seguimiento: CU_Gestionar Horario.

2.7.2 Requisitos no funcionales.

Apariencia o interfaz externa

- ✓ La interfaz debe ser sencilla, para que el usuario que no sea una persona expertas en la rama de la informática, puedan utilizarlo fácilmente.
- ✓ La interfaz debe ser agradable, que favorezca el estado de ánimo del cliente y que combine correctamente los colores, tipo de letra y tamaño.
- ✓ Deben utilizarse interfaces con un mismo estilo para que el usuario no se sienta perdido dentro del sistema.

Usabilidad

- ✓ El sistema debe ser de fácil manejo para los usuarios que tengan niveles básicos sobre la computación.

Rendimiento

- ✓ El sistema debe estar concebido para el consumo mínimo de recursos.

Software

- ✓ El sistema será programado en C#.
- ✓ Se utilizará MySQL como gestor de BD.

Seguridad

- ✓ Mantener la integridad de la información, es decir que no se pierda durante su almacenamiento.
- ✓ Permitir que cuando se borre cualquier documento o información pueda existir una opción de advertencia antes realizar la acción.

Disponibilidad

- ✓ El sistema deberá estar disponible cuando se necesite.

Confiabilidad

- ✓ La información manejada por el sistema debe estar protegida.

Integridad

- ✓ La información manejada por el sistema será objeto de cuidadosa protección contra cualquier ataque.

Fiabilidad

- ✓ Debe garantizarse el resguardo de la información, de modo que estén duplicados, así como la grabación periódica de la Base de Datos.

Legales

- ✓ Algunas de las herramientas de desarrollo son libres y del resto, las licencias están avaladas.

2.8 Actores del sistema

Un actor puede representar personas o cosas externas al sistema que de alguna forma interactúan con él.

A continuación se describen los actores del sistema que debe ser desarrollado (Tabla 2.4).

Nombre del Actor	Descripción
Planificador	Es el encargado de confeccionar y actualizar el horario; así como gestionar las informaciones necesarias para la planificación del mismo.

Tabla 2.4. Actores del sistema.

2.9 Diagrama de Casos de Uso del Sistema.

El modelo de Casos de Uso del sistema representa las funcionalidades deseadas y el entorno del sistema a través de actores y casos de uso, además sirve como un contrato entre los clientes y los desarrolladores. Este sienta las bases necesarias para el desarrollo del análisis y el diseño del sistema.

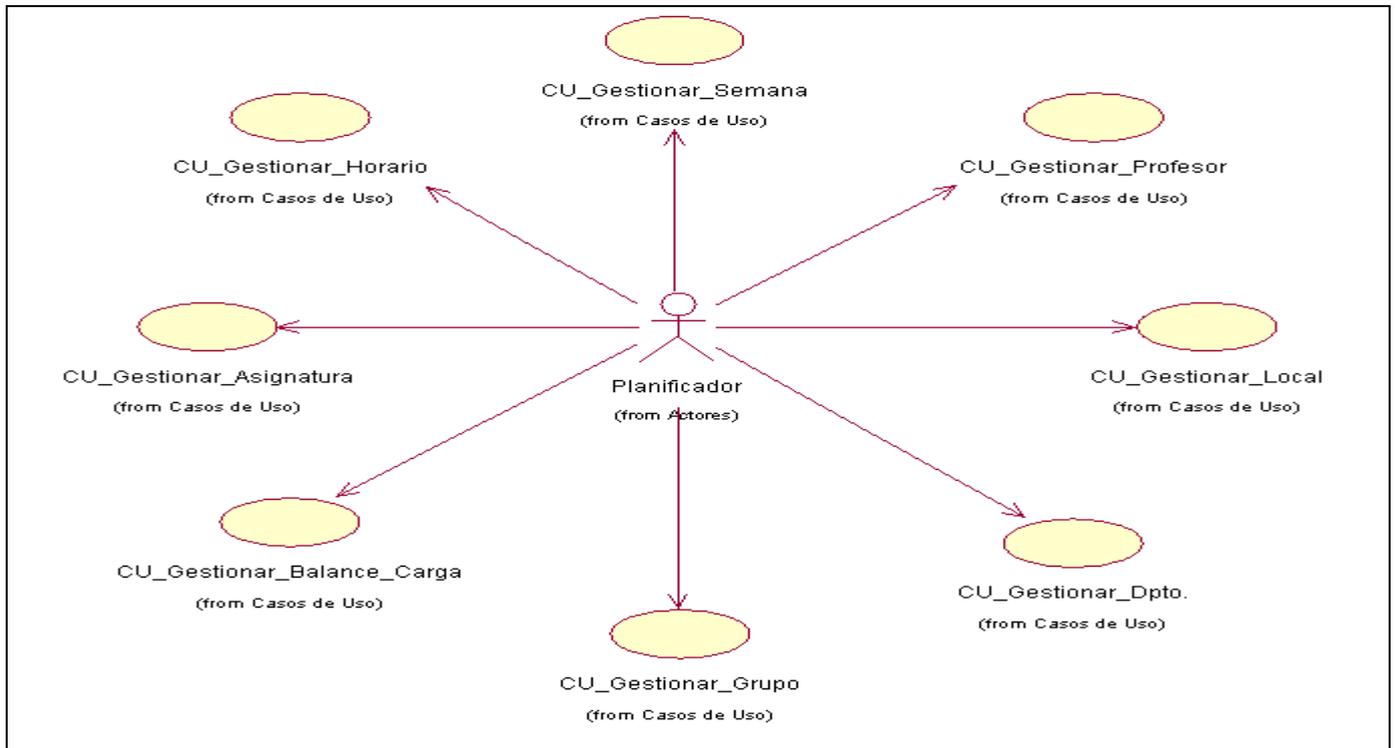


Figura 2.3. Diagrama de casos de uso del sistema.

2.10 Lista de Casos de Uso del Sistema y su Prioridad

Referencia	Caso de uso	Prioridad
CUS 1	CU_Gestionar Profesor	Crítico
CUS 2	CU_Gestionar Asignatura	Crítico
CUS 3	CU_Gestionar Local	Crítico
CUS 4	CU_Gestionar Grupo	Crítico
CUS 5	CU_Gestionar Dpto.	Crítico
CUS 6	CU_Gestionar Semana	Crítico
CUS 7	CU_Gestionar Balance de Carga	Crítico
CUS 8	CU_Gestionar Horario	Crítico

Tabla 2.5. Lista de casos de uso del sistema y su prioridad.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

2.10.1 Descripciones de los Casos de Uso del Sistema.

Con el propósito de lograr una mejor comprensión de los procesos a automatizar, se especifican los casos de uso del sistema mediante la descripción textual de los mismos.

A continuación se muestran las descripciones de algunos de los casos de uso del sistema (tabla 2.6, tabla 2.7, tabla 2.8), los demás pueden observarse en el Anexo 1.

Descripción del caso de uso CU_Gestionar Profesor

Nombre del Caso de uso	CU_Gestionar Profesor
Actores	Planificador
Propósito	Permitir registrar los datos de los profesores en el sistema.
Resumen	
El caso de uso inicia cuando el planificador selecciona la opción “Gestionar Profesores”, la cual permite insertar en el sistema los datos correspondientes a los profesores.	
Precondiciones	El planificador debe contar con todos los datos correspondientes al profesor.
Casos de Uso Asociados	No tiene
Referencia	1.1, 1.2, 1.3
Flujo Normal de Eventos	
Acción del Actor	Respuesta del sistema
1. El planificador selecciona la opción “Gestionar Profesores”.	1.1 El sistema despliega un menú con las opciones para la gestión de profesores. a) Para insertar profesores ver sección Insertar. b) Para eliminar profesores ver sección Eliminar. c) Para modificar profesores ver sección Modificar.
Sección Insertar	
1. El planificador selecciona la opción “Insertar Profesores”.	1.1 El sistema muestra la interfaz para entrar los datos del profesor.
2. El planificador entra el nombre del profesor, escoge la categoría y el departamento.	2.1 El sistema activa el campo perteneciente a la asignatura, este muestra las asignaturas que pertenecen al departamento escogido.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

3. El planificador escoge la asignatura que va a impartir el profesor.	3.1 El sistema activa el campo para escoger los grupos a los que les va a impartir clase el profesor, que muestra los grupos que pertenezcan al mismo año de la asignatura escogida.
4. El planificador escoge los grupos de uno en uno y va presionando el botón "Aceptar"	4.1 El sistema le inserta los grupos al profesor.
5. El planificador escoge el tipo de clase que va a impartir el profesor, si son todas marca la opción "Todas", si no escoge el tipo.	
6. El planificador inserta los datos de las afectaciones, si la afectación es únicamente de ese profesor presiona el botón "Aceptar", si es una afectación general de la facultad se marca "Aplicar a todos los profesores".	6.1 El sistema le inserta las afectaciones al profesor.
7. Después de entrar todos los datos del profesor el planificador presiona el botón "Registrar".	7.1 El sistema valida los datos entrados. 7.2 Se crea la matriz de afectaciones del profesor. 7.3 El sistema registra en la base de datos los datos del profesor.
Flujo Alternativo de Eventos	
	7.1 Si ya existe el profesor en la base de datos el sistema envía un mensaje de error.
7. El planificador presiona el botón "Cancelar".	7.1 El sistema pasa a la acción 1.
Sección Eliminar	
1. El planificador selecciona la opción "Eliminar Profesores".	1.1 El sistema muestra la interfaz para escoger el profesor que se desea eliminar.
2. El planificador escoge el departamento del cual se va a eliminar el profesor.	2.1 El sistema activa el campo para escoger el profesor, este muestra todos los profesores pertenecientes al

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

	departamento escogido.
3. El planificador selecciona los profesores que eliminará y presiona el botón "Eliminar".	3.1 El sistema envía un mensaje de confirmación " Desea eliminar la asignatura"
4. El planificador presiona el botón "Aceptar".	4.1 El sistema elimina los profesores de la base de datos. 4.2 El sistema modifica la matriz de afectaciones del profesor.
Flujo Alternativo de Eventos	
	2.1 Si no existe ningún profesor en la base de datos, el sistema mostrará un mensaje de error.
4. El planificador presiona el botón "Cancelar"	4.1 El sistema vuelve a la acción 1.1.
Sección Modificar	
1. El planificador selecciona la opción "Modificar Profesores".	1.1 El sistema muestra la interfaz para escoger el profesor que se desea modificar.
2. El planificador escoge el departamento del cual se va a modificar el profesor.	2.1 El sistema activa el campo para escoger el profesor, que muestra todos los profesores pertenecientes al departamento escogido.
3. El planificador selecciona el profesor que se modificará.	3.1 El sistema muestra los datos del profesor para modificarlos.
3. El planificador hace los cambios y presiona el botón "Modificar".	3.1 El sistema actualiza los datos del profesor en la base de datos. 3.2 El sistema modifica la matriz de afectaciones del profesor.
Flujo Alternativo de Eventos	
	2.1 Si no existe ningún profesor en la base de datos, el sistema mostrará un mensaje de error.
3. El planificador presiona el botón "Cancelar".	3.1 El sistema pasa a la acción 1.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Poscondiciones	Se almacenan en la base de datos las informaciones y se actualiza la matriz de afectaciones del profesor.
Prioridad	Crítico

Tabla 2.6. Descripción del caso de uso del sistema Gestionar Profesor.

Descripción del caso de uso CU_Gestionar Asignatura

Nombre del Caso de uso	CU_Gestionar Asignatura
Actores	Planificador
Propósito	Permitir registrar los datos de las asignaturas en el sistema.
Resumen	
El caso de uso inicia cuando el planificador selecciona la opción “Gestionar Asignatura”, la cual permite insertar en el sistema los datos correspondientes a las asignaturas.	
Precondiciones	El planificador debe contar con todos los datos correspondientes a la asignatura.
Casos de Uso Asociados	No tiene
Referencia	2.1, 2.2, 2.3
Flujo Normal de Eventos	
Acción del Actor	Respuesta del sistema
1. El planificador selecciona la opción “Gestionar Asignatura”.	1.1 El sistema despliega un menú con las opciones para la gestión de asignaturas. a) Para insertar asignaturas ver sección Insertar. b) Para eliminar asignaturas ver sección Eliminar. c) Para modificar asignaturas ver sección Modificar.
Sección Insertar	
1. El planificador selecciona la opción “Insertar Asignaturas”.	1.1 El sistema muestra la interfaz para entrar los datos de la asignatura.
2. El planificador entra el nombre y el nombre corto de la asignatura y escoge el departamento, nivel y año, y presiona el	2.1 El sistema valida los datos entrados. 2.2 El sistema registra en la base de datos la asignatura.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

botón “Registrar”.	
Flujo Alternativo de Eventos	
2. Si la asignatura está bloqueada el planificador marca la opción “bloqueada” e inserta los datos correspondientes a la misma (semana que inicia y la cantidad de semanas) y presiona el botón “Registrar”.	2.1 El sistema valida los datos entrados. 2.2 El sistema registra en la base de datos la asignatura.
	2.1 En caso de que ya la asignatura exista el sistema envía un mensaje de error, volviendo a la acción 1.1.
2. El planificador presiona el botón “Cancelar”	2.1 El sistema pasa a la acción 1
Sección Eliminar	
1. El planificador selecciona la opción “Eliminar Asignatura”.	1.1 El sistema muestra la interfaz para escoger la asignatura que se desea eliminar.
2. El planificador escoge el departamento del cual se va a eliminar la asignatura.	2.1 El sistema activa el campo para escoger la asignatura, este muestra todas las asignaturas pertenecientes al departamento escogido.
3. El planificador selecciona la asignatura que eliminará y presiona el botón “Eliminar”.	3.1 El sistema envía un mensaje de confirmación ” Desea eliminar la asignatura”
4. El planificador presiona el botón “Aceptar”.	4.1 El sistema elimina la asignatura de la base de datos.
Flujo Alternativo de Eventos	
	2.1 Si no existe ninguna asignatura en la base de datos, el sistema mostrará un mensaje de error.
4. El planificador presiona el botón “Cancelar”	4.1 El sistema vuelve a la acción 1.1.
Sección Modificar	
1. El planificador selecciona la opción “Modificar Asignatura”.	1.1 El sistema muestra la interfaz para escoger la asignatura que se desea modificar.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

2. El planificador escoge el departamento del cual se va a modificar la asignatura.	2.1 El sistema activa el campo para escoger la asignatura, este muestra todas las asignaturas pertenecientes al departamento escogido.
3. El planificador selecciona la asignatura que se modificará.	3.1 El sistema muestra los datos de la asignatura.
3. El planificador hace los cambios y presiona el botón "Modificar".	3.1 El sistema actualiza los datos de la asignatura en la base de datos.
Flujo Alternativo de Eventos	
	2.1 Si no existe ninguna asignatura en la base de datos, el sistema mostrará un mensaje de error.
3. El planificador presiona el botón "Modificar".	3.1 El sistema pasa a la acción 1.
Poscondiciones	Se almacenan en la base de datos las informaciones.
Prioridad	Crítico

Tabla 2.7. Descripción del caso de uso del sistema Gestionar Asignatura.

Descripción del caso de uso CU_Gestionar Horario

Nombre del Caso de uso	CU_ Gestionar Horario
Actores	Planificador
Propósito	Confeccionar el horario docente de la facultad 4.
Resumen	
El caso de uso inicia cuando el Planificador selecciona la opción "Gestionar Horario", comprueba que estén todos los datos necesarios para su confección, luego genera el horario y este se almacena en la base de datos.	
Precondiciones	El Planificador debe contar con todos los datos necesarios para realizar la planificación del horario.
Casos de Uso Asociados	No tiene
Referencia	8.1, 8.2, 8.3, 8.4
Flujo Normal de Eventos	

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Acción del Actor	Respuesta del sistema
1. El planificador selecciona la opción "Gestionar Horario".	1.1 El sistema despliega un menú con las opciones para la gestión del horario. a) Para crear horario ver sección Confeccionar Horario. b) Para mostrar horario ver sección Mostrar.
Sección Confeccionar Horario	
1. El planificador selecciona la opción "Confeccionar Horario".	1.1 El sistema muestra la interfaz para su confección.
2. El planificador escoge el año para el cual se va a generar el horario y presiona el botón "Generar Horario".	2.1 El sistema comprueba que existan los datos requeridos para su confección. 2.1 El sistema genera el horario especificado. 2.2 El sistema archiva el horario en la base de datos.
Flujo Alternativo de Eventos	
	1.1. Si falta algún dato necesario para la confección del horario, el sistema mostrará el mensaje correspondiente.
Sección Mostrar Horario	
1. El planificador escoge la opción "Mostrar Horario"	1.1 El sistema muestra la interfaz para mostrar el horario.
2. El planificador escoge el año para el cual se va a mostrar el horario y presiona el botón "Mostrar".	2.1 El sistema muestra el horario.
3. El planificador presiona el botón "Exportar Horario".	3.1 Se exporta el horario para un Excel.
Poscondiciones	Se almacenan en la base de datos las informaciones.
Prioridad	Crítico

Tabla 2.8. Descripción del caso de uso del sistema Confeccionar Horario.

Conclusiones

En este capítulo se abordó todo lo relacionado con los eventos más comunes en el contexto del Negocio para lograr una mejor comprensión del mismo, también se realizó la captura de los requerimientos funcionales y no funcionales del sistema, el diagrama de casos de uso del sistema y la especificación detallada de los casos de uso.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

Introducción

En el presente capítulo analizamos los requisitos definidos anteriormente, refinándolos y estructurándolos, se realizan los diagramas de clases del análisis, los diagramas de clases de diseño y los diagramas de secuencia. Además se realiza el diseño de la base de datos mediante los diagramas lógico y físico de los datos y se abordan otros aspectos relacionados con el diseño, la seguridad y el tratamiento de errores en la aplicación.

3.1 Descripción de la Arquitectura

El objetivo de la Arquitectura del Software es aportar elementos que ayuden a la toma de decisiones y, al mismo tiempo, proporcionar conceptos y un lenguaje común que permitan la comunicación entre los equipos que participen en un proyecto. Para conseguirlo, la Arquitectura del Software construye abstracciones, materializándolas en forma de diagramas comentados.

Para realizar la propuesta de análisis y diseño del sistema que le de cumplimiento a la situación problemática se utilizará la Arquitectura Tres Capas. La misma define cómo organizar el modelo de diseño en capas, donde los componentes de una capa sólo pueden hacer referencia a componentes en capas inmediatamente inferiores. Este patrón es importante porque simplifica la comprensión y la organización del desarrollo de sistemas complejos, reduciendo las dependencias de forma tal que las capas más bajas no son conscientes de ningún detalle o interfaz de las superiores. Las tres capas están compuestas por:

- ✓ **Capa de Presentación:** Esta capa contiene las clases interfaces de la aplicación con las que va a interactuar el usuario, estas son las encargadas de gestionar todas las peticiones que el mismo realice.
- ✓ **Capa de Negocio:** Esta capa contiene toda la lógica del negocio, en ella se encuentran las clases controladoras, en este caso se realizó una clase controladora por cada caso de uso identificado en el sistema.
- ✓ **Capa de Datos:** En esta capa se guardan todos los datos necesarios para la aplicación, es donde se encuentra la base de datos que esta contiene una clase por cada entidad.

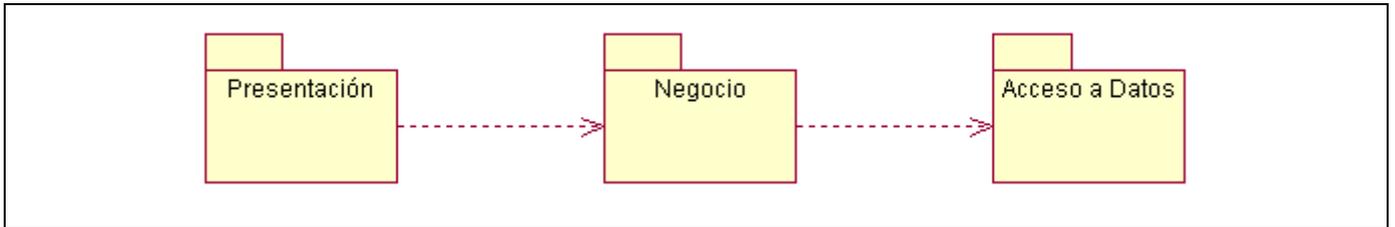


Figura 3.1. Arquitectura Tres Capas.

3.2 Análisis del Sistema

El análisis consiste en obtener una visión del sistema que se preocupa de ver que hace, o sea, su objetivo es comprender perfectamente los requisitos de software y no precisar como se implementará la solución, interesándose únicamente por los requisitos funcionales y utilizando como lenguaje el modelo de análisis.

3.2.1 Diagramas de clases del análisis.

A continuación se muestran algunos de los diagramas de clases del análisis (figura 3.2, figura 3.3, figura 3.4), los demás se pueden ver en el Anexo 2.

Diagrama de análisis del CU_Gestionar Profesor

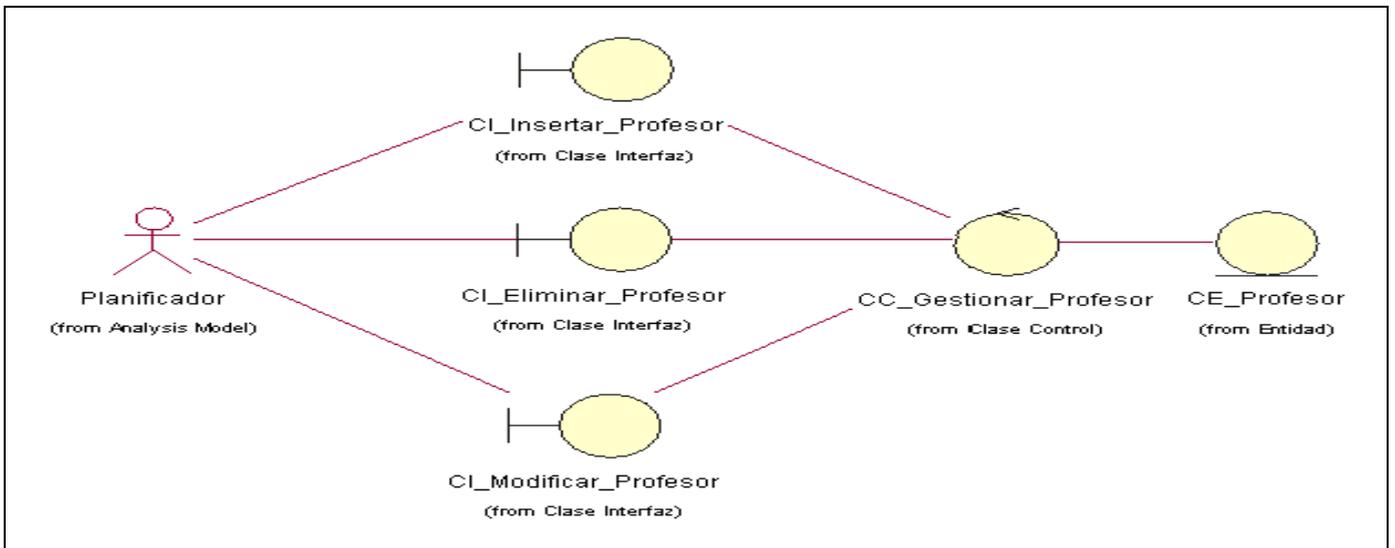


Figura 3.2. Diagrama de análisis del CU_Gestionar Profesor.

Diagrama de análisis del CU_Gestionar Asignatura

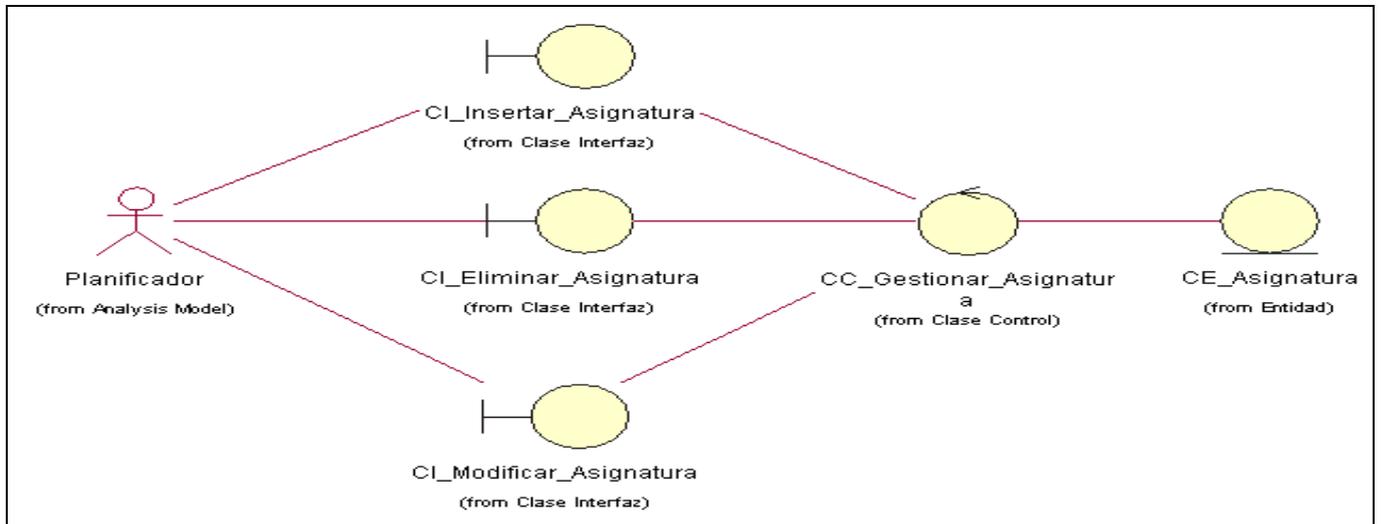


Figura 3.3. Diagrama de análisis del CU_Gestionar Asignatura.

Diagrama de análisis del CU_Confeccionar Horario

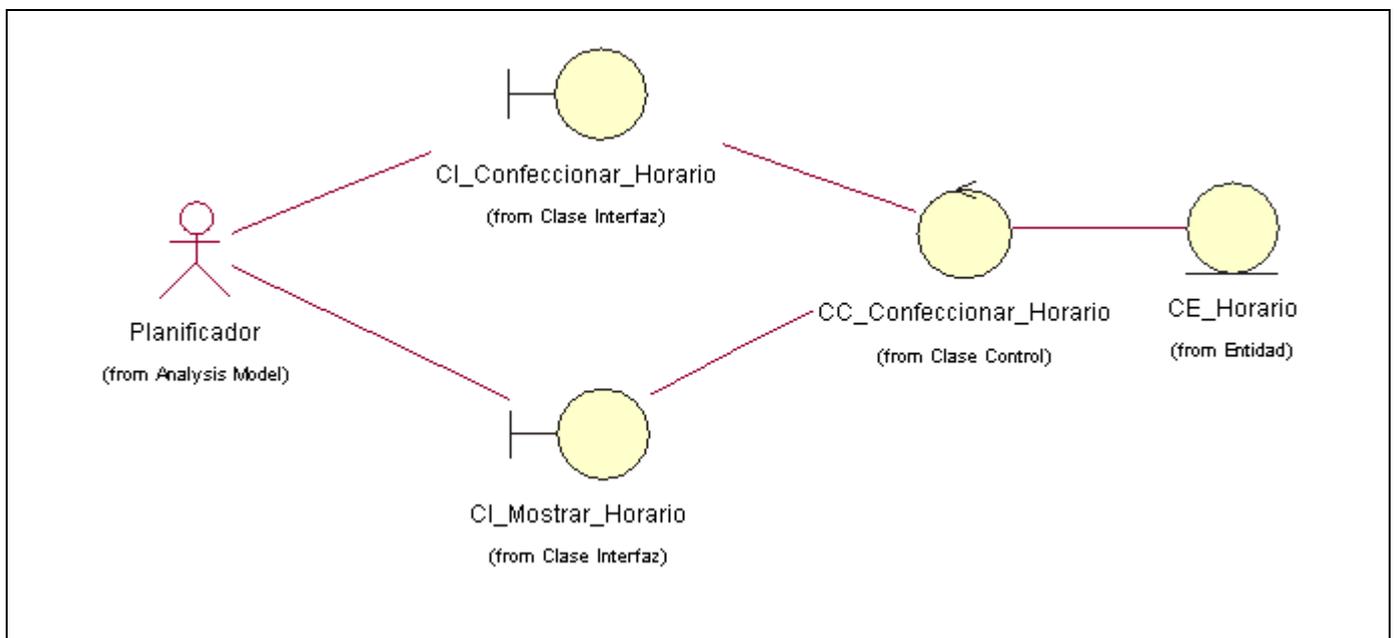


Figura 3.4. Diagrama de clases del análisis del CU_Confeccionar Horario.

3.3 Diseño del Sistema

El Diseño es un refinamiento del análisis que tiene en cuenta los requisitos no funcionales y restricciones relacionadas principalmente con el lenguaje de programación, además debe crear una entrada apropiada y un punto de partida para las actividades de implementación y utilizando como manifiesto el modelo de diseño, quien contiene estereotipos físicos específicos para una implementación.

3.3.1 Diagramas de Clases de Diseño

A continuación se muestran algunos los diagramas de diseño de algunos casos de uso (figura 3.5, figura 3.6, figura 3.7), los demás pueden observarse en el Anexo 3.

Diagrama de diseño del CU_Gestionar Profesor

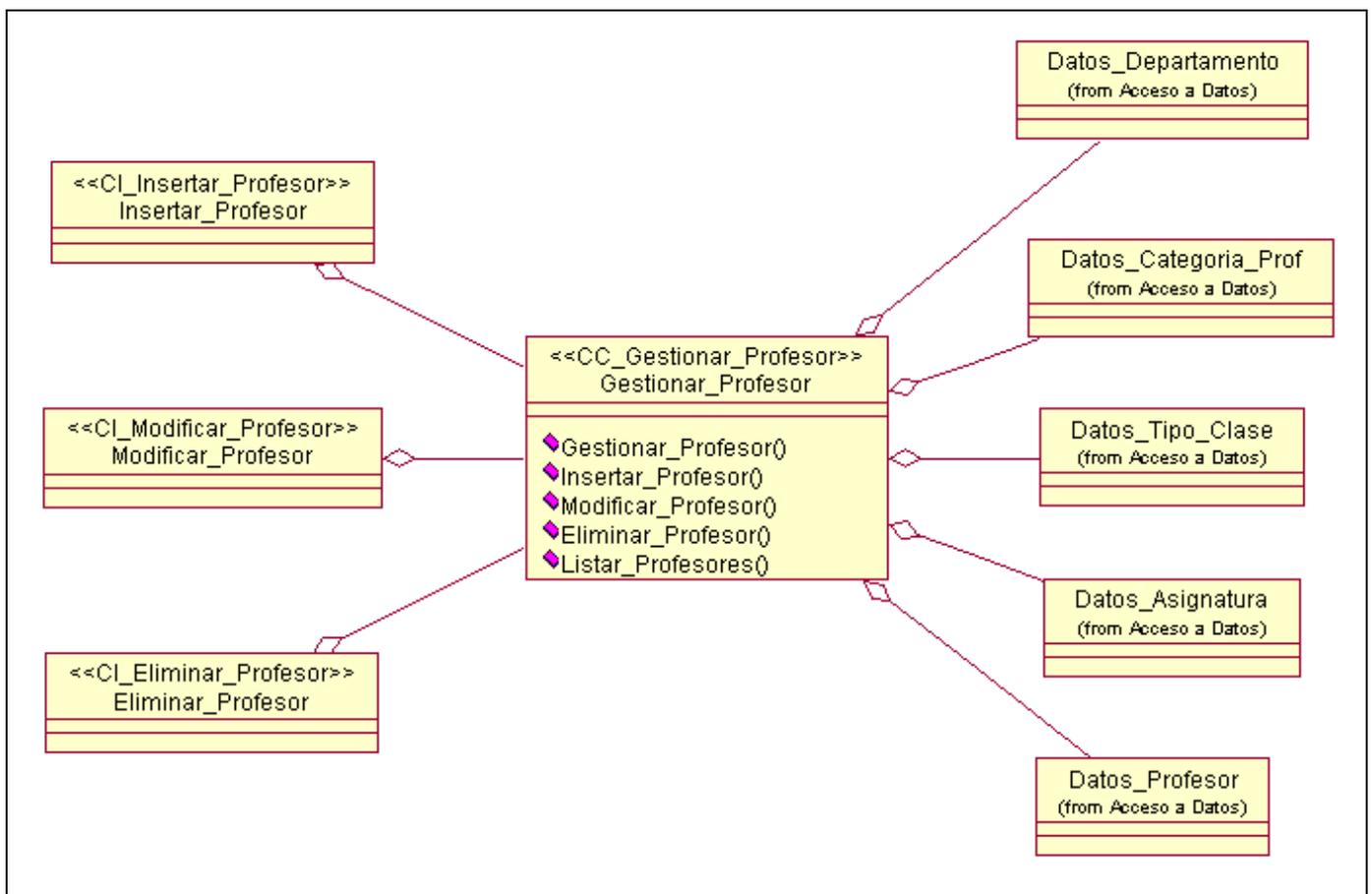


Figura 3.5. Diagrama de diseño del CU_Gestionar Profesor.

Diagrama de diseño del CU_Gestionar Asignatura

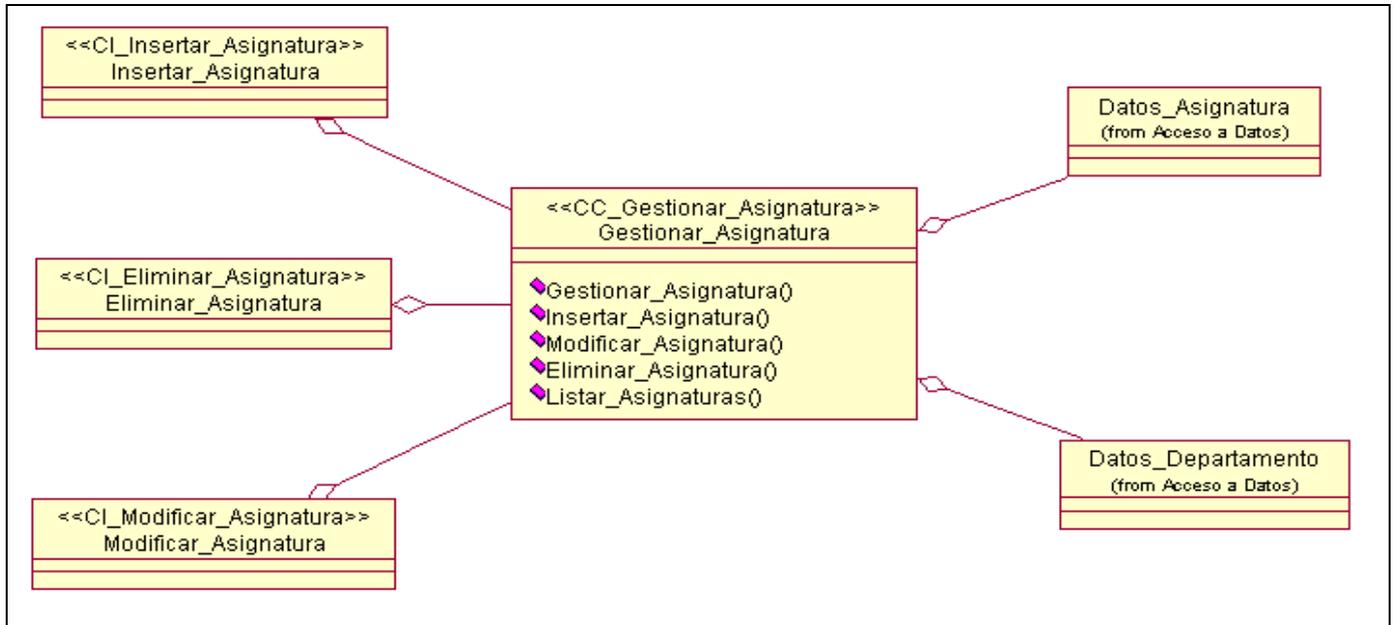


Figura 3.6. Diagrama de diseño del CU_Gestionar Asignatura.

Diagrama de diseño del CU_Confeccionar Horario

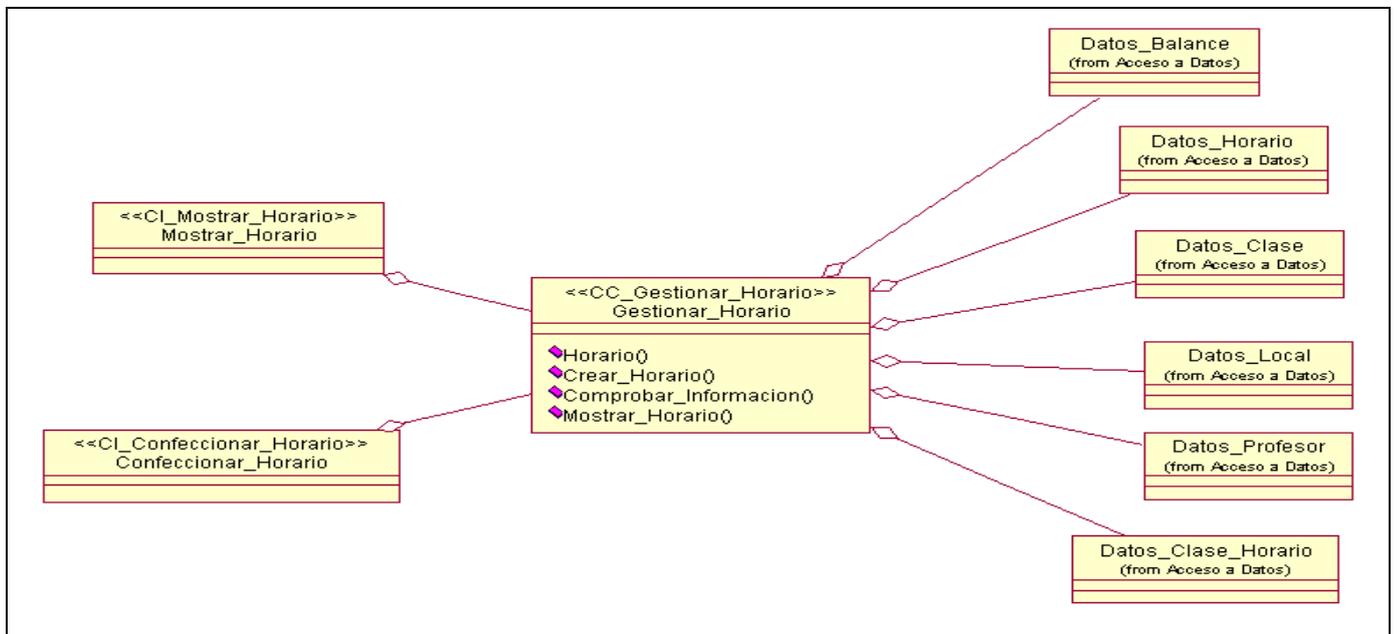


Figura 3.7. Diagrama de diseño del CU_Confeccionar Horario.

3.3.2 Diagramas de Interacción

Los diagramas de interacción muestran cómo se comunican los objetos en una interacción. Existen dos tipos de diagramas de interacción: el Diagrama de Colaboración y el Diagrama de Secuencia.

A continuación se muestran algunos los diagramas de secuencia para el CU_Confeccionar Horario y el CU_Gestionar Profesor (figura 3.8, figura 3.9, figura 3.10, 3.11, 3.12), los demás se pueden observar en el Anexo 4.

Diagrama de secuencia del CU_Gestionar Profesor

Sección Insertar

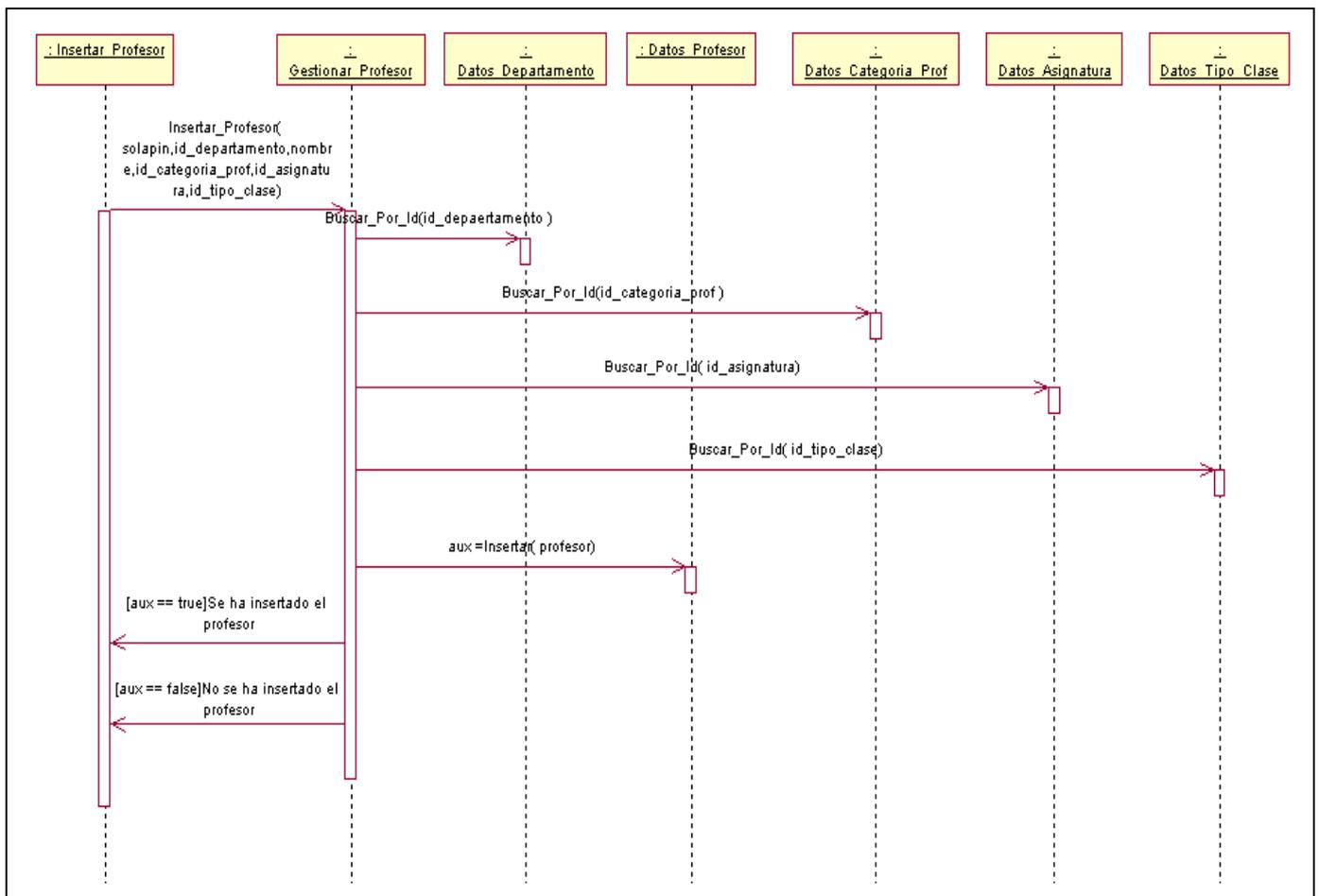


Figura 3.8. Diagrama de secuencia del CU_Gestionar Profesor: Sección Insertar.

Sección Eliminar

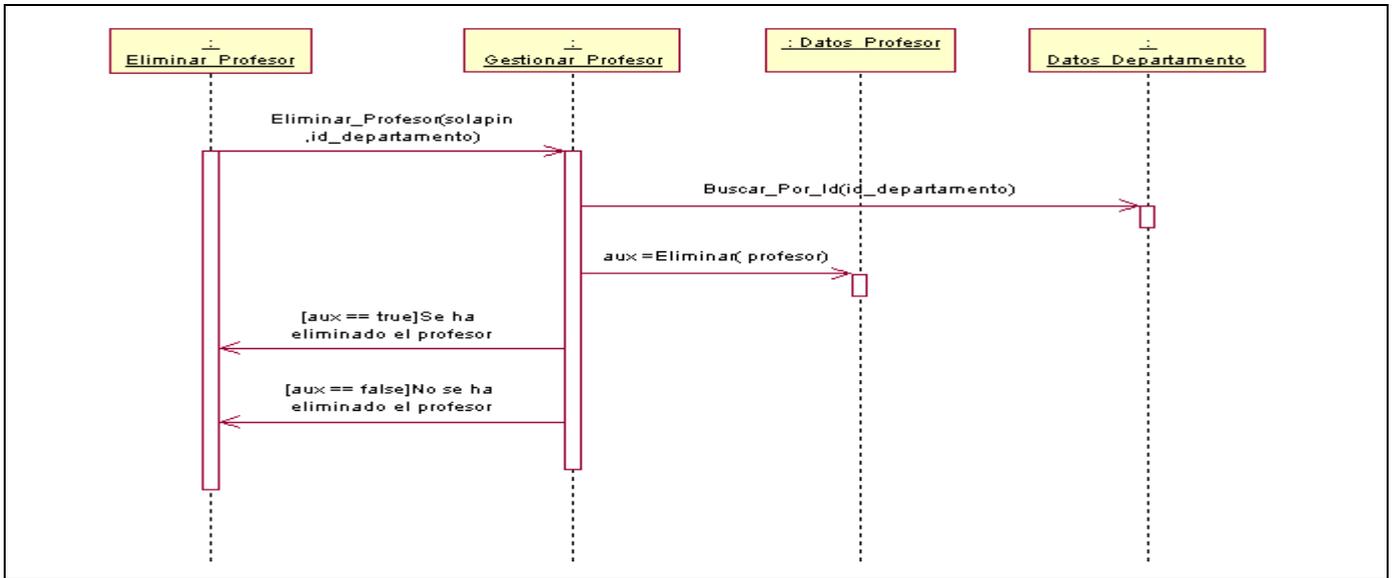


Figura 3.9. Diagrama de secuencia del CU_Gestionar Profesor: Sección Eliminar.

Sección Modificar

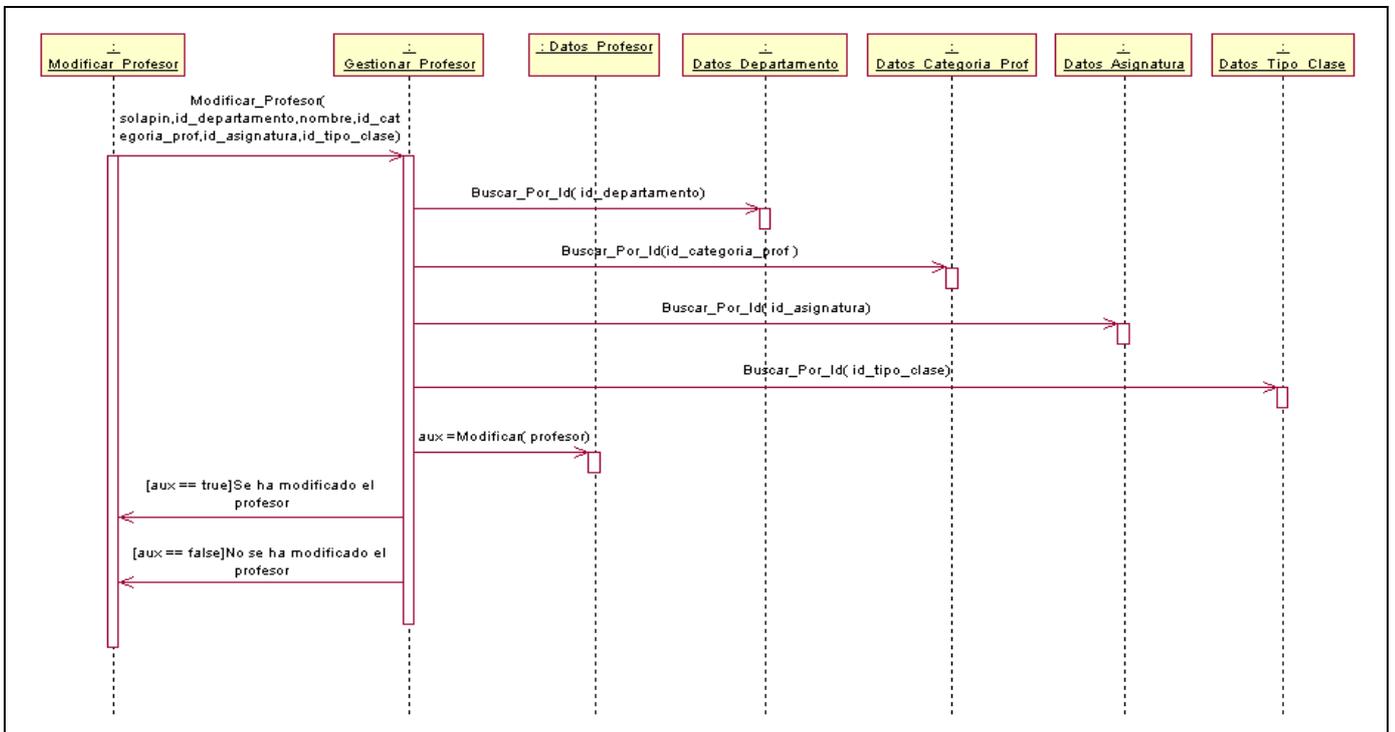


Figura 3.10. Diagrama de secuencia del CU_Gestionar Profesor: Sección Modificar.

Diagrama de secuencia del CU_Gestionar Horario

Sección Confeccionar Horario

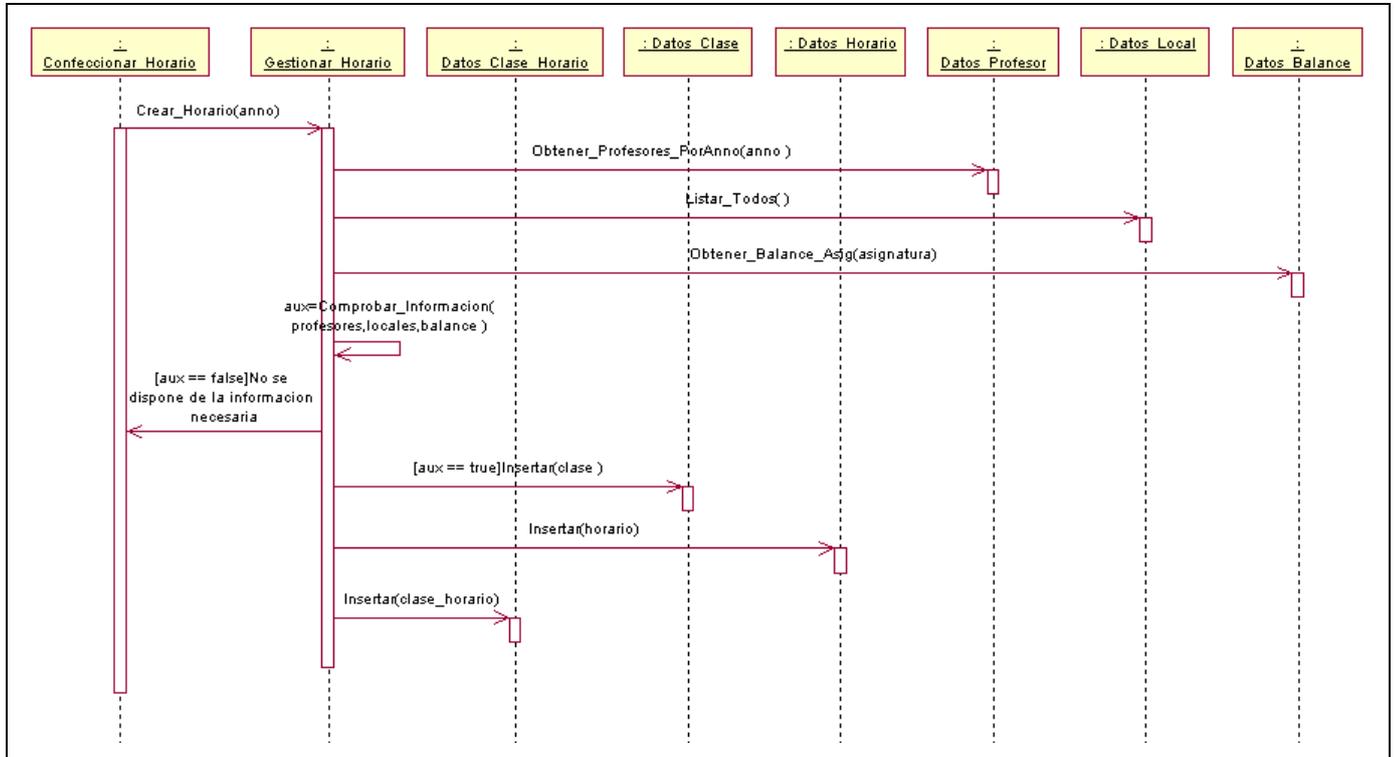


Figura 3.11. Diagrama de secuencia del CU_Gestionar Horario: Sección Confeccionar Horario.

Sección Mostrar Horario

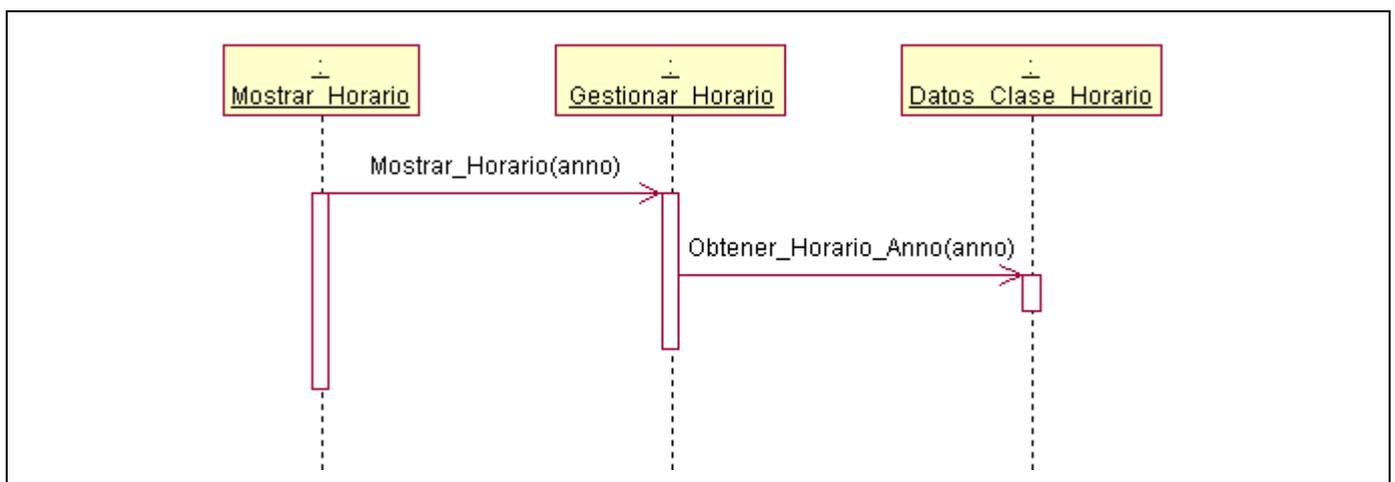


Figura 3.12. Diagrama de secuencia del CU_Gestionar Horario: Sección Mostrar.

3.3.3 Diagrama de Clases del Dominio

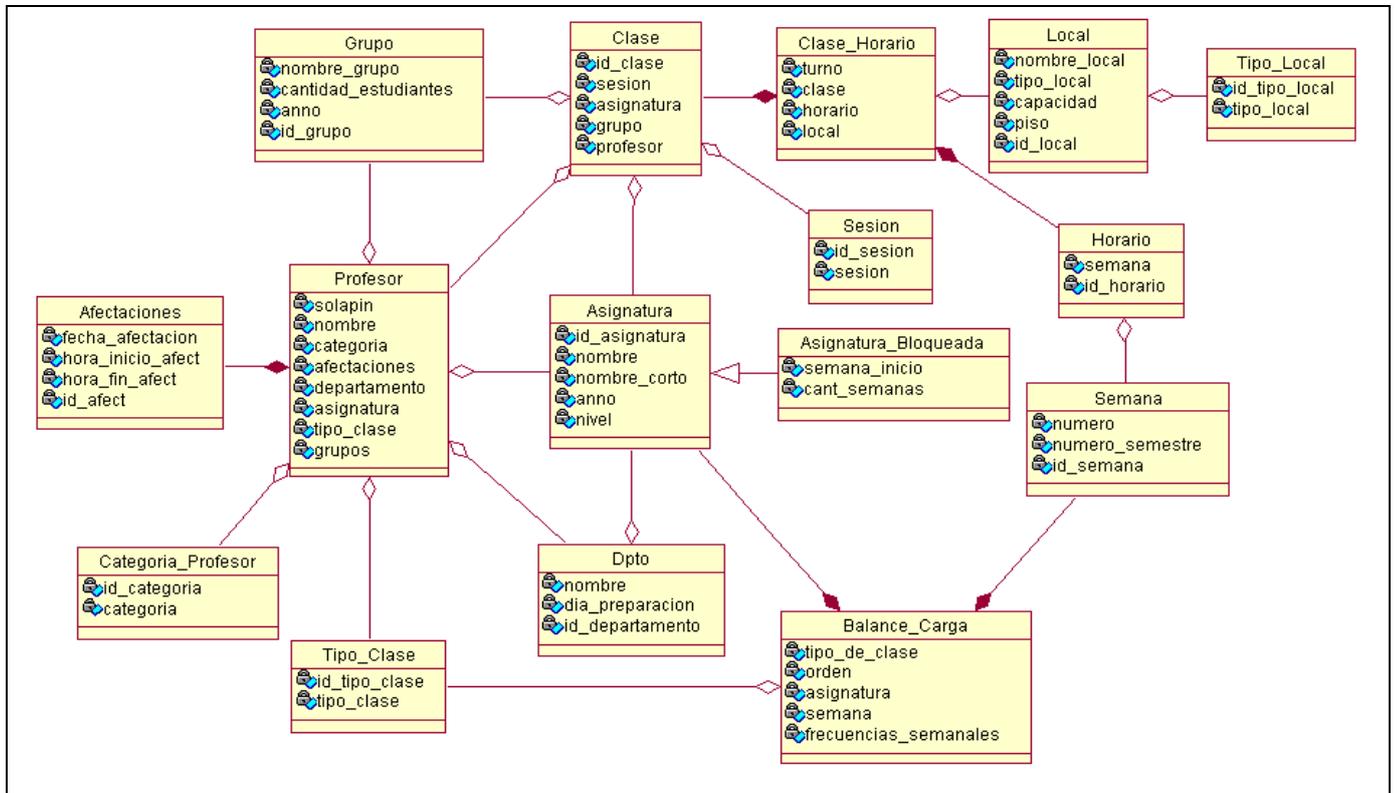


Figura 3.13. Diagrama de Clases del Dominio.

3.4 Diseño de la Base de Datos

3.4.1 Modelo Físico de Datos

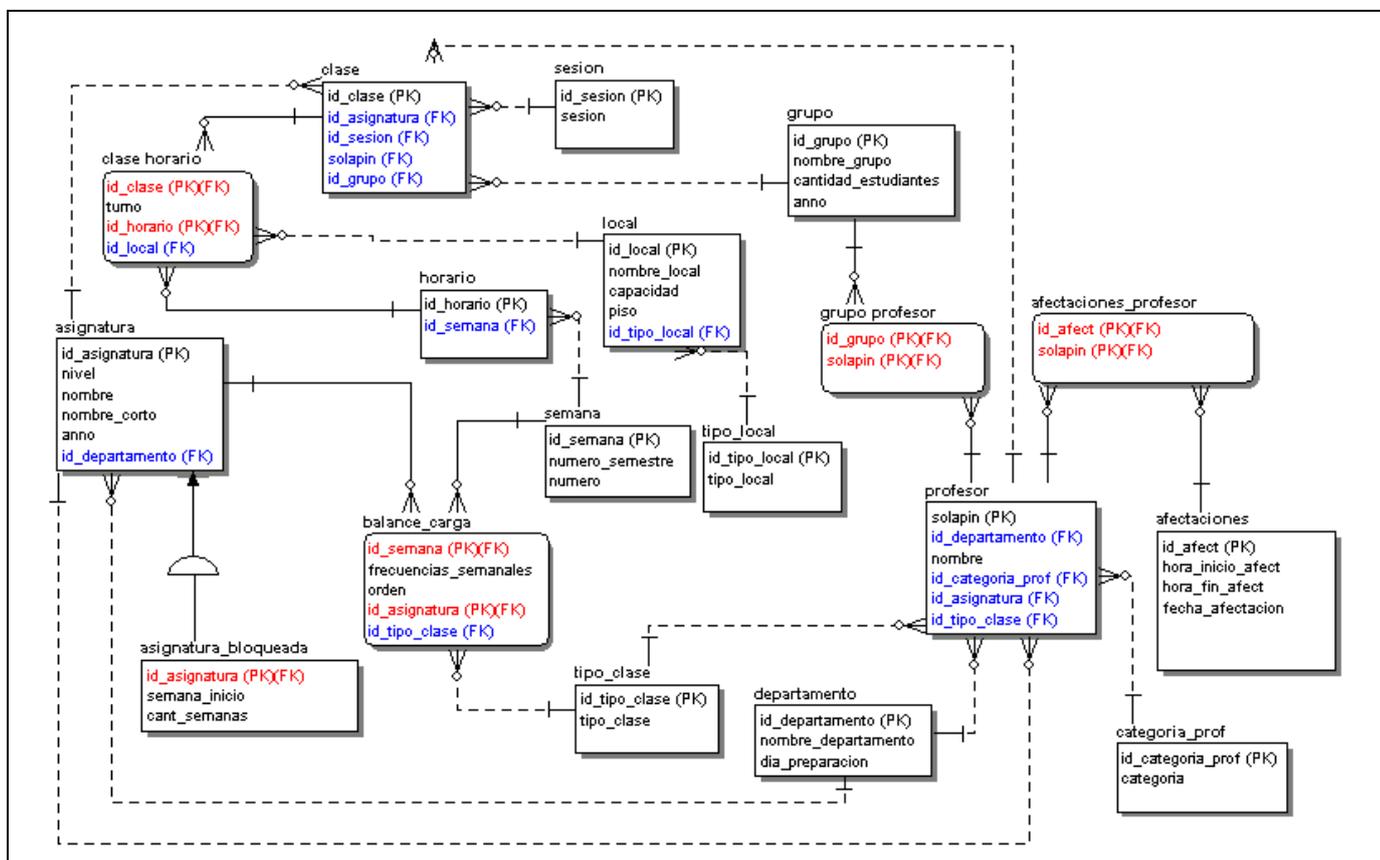


Figura 3.14. Modelo Físico de Datos.

3.4.2 Descripción de las tablas de la base de datos.

Nombre: departamento		
Descripción: En esta tabla se almacena la información de cada departamento		
Atributo	Tipo	Descripción
id _ departamento	int(50)	Identificador del departamento.
nombre _ departamento	varchar(50)	Nombre del departamento.
día _ preparación	int(50)	Día de preparación metodológica del departamento.

Tabla 3.1. Descripción de la tabla departamento.

Nombre: balance _ carga		
Descripción: En esta tabla surge de la relación de la tabla asignatura y la tabla semana, en ella se almacenan los datos del balance de carga de cada asignatura.		
Atributo	Tipo	Descripción
id _ semana	int(50)	Identificador de la semana.
id _ asignatura	int(50)	Identificador de la asignatura.
id_tipo_clase	int(50)	Identificador del tipo de clase.
orden	int(50)	Orden en que se van a impartir las clases (conferencia, clase práctica, laboratorio) en una semana.
frecuencias _ semanales	int(50)	Frecuencia de cada asignatura en una semana.

Tabla 3.2. Descripción de la tabla balance _ carga.

Nombre: asignatura		
Descripción: En esta tabla se almacenan los datos de cada asignatura.		
Atributo	Tipo	Descripción
id_asignatura	int(50)	Identificador de la asignatura.
id_ departamento	int(50)	Identificador del departamento.
nombre	varchar(50)	Nombre de la asignatura.
nombre _ corto	varchar(50)	Nombre corto de la asignatura.
anno	varchar(50)	Año de la asignatura.

Tabla 3.3. Descripción de la tabla asignatura.

Nombre: horario		
Descripción: En esta tabla se almacenan todos los datos necesarios para la confección del horario.		
Atributo	Tipo	Descripción
id _ horario	int(50)	Identificador del horario que se va a crear.
id _ semana	int(50)	Identificador de la semana correspondiente a cada horario.

Tabla 3.4. Descripción de la tabla horario.

Nombre: local		
Descripción: En esta tabla se almacenan los datos de cada local en el que se impartirá clases.		
Atributo	Tipo	Descripción
id _ local	int(50)	Identificador del local.
nombre _ local	varchar(50)	Nombre del local.
tipo _ local	varchar(50)	Tipo de local (aula, laboratorio o salón de conferencias).
capacidad	int(50)	Capacidad de cada local.
piso	varchar(50)	Piso en que se encuentra cada local.

Tabla 3.5. Descripción de la tabla local.

Nombre: profesor		
Descripción: En esta tabla se almacena los datos de cada profesor.		
Atributo	Tipo	Descripción
solapín	int(50)	Identificador del profesor.
id _ asignatura	int(50)	Identificador de la asignatura que imparte el profesor.
id_ departamento	int(50)	Identificador del departamento al cual pertenece la asignatura que imparte el profesor.
id_categoria_profesor	int(50)	Identificador de la categoría del profesor (interno, externo, adjunto).
nombre	varchar(50)	Nombre del profesor.

Tabla 3.6. Descripción de la tabla profesor.

Nombre: afectaciones _ profesor		
Descripción: En esta tabla surge de la relación entre la tabla profesor y afectaciones, se almacenan los datos de cada afectación que tenga un profesor.		
Atributo	Tipo	Descripción
id_afect	int(50)	Identificador de la afectación del profesor.
solapín	int(50)	Identificador del profesor

Tabla 3.7. Descripción de la tabla afectaciones _ profesor.

Nombre: afectaciones		
Descripción: En esta tabla se almacenan los datos de cada afectación que tenga un profesor.		
Atributo	Tipo	Descripción
id_afect	int(50)	Identificador de la afectación del profesor.
fecha_afectación	date	Fecha de la afectación.
hora_inicio_afect	varchar(50)	Hora que comienza la afectación.
hora_fin_afect	varchar(50)	Hora que termina la afectación.

Tabla 3.8. Descripción de la tabla afectaciones.

Nombre: grupo		
Descripción: En esta tabla se almacena a información de cada grupo docente.		
Atributo	Tipo	Descripción
id_grupo	int(50)	Identificador del grupo.
nombre_grupo	varchar(50)	Nombre del grupo.
cantidad_estudiantes	int(50)	Cantidad de estudiantes de cada grupo.
anno	varchar(50)	Año de cada grupo.

Tabla 3.9. Descripción de la tabla grupo.

Nombre: grupo profesor		
Descripción: En esta tabla se almacenan los datos de los grupos que se le asignan al profesor.		
Atributo	Tipo	Descripción
id_grupo	int(50)	Identificador del grupo que se le asigna al profesor.
solapín	int(50)	Identificador del profesor.

Tabla 3.10. Descripción de la tabla grupo profesor.

Nombre: semana		
Descripción: En esta tabla se almacenan los datos de cada semana del semestre.		
Atributo	Tipo	Descripción
id _ semana	int(50)	Identificador de la semana.
numero	int(50)	Numero de la semana.
numero _ semestre	int(50)	Numero del semestre.

Tabla 3.11. Descripción de la tabla semana.

Nombre: clase horario		
Descripción: Esta tabla se crea de la relación entre las tablas clase y horario.		
Atributo	Tipo	Descripción
id _ clase	int(50)	Identificador de la clase.
id _ horario	int(50)	Identificador del horario.
id _ local	int(50)	Identificador del local.
turno	int(50)	Turno en el que se va a ubicar la clase.

Tabla 3.12. Descripción de la tabla clase horario.

Nombre: sesión		
Descripción: En esta tabla se almacenan los datos de cada sesión de clase.		
Atributo	Tipo	Descripción
id _ sesión	int(50)	Identificador de la sesión.
sesión	varchar(50)	Nombre de la sesión (mañana o tarde).

Tabla 3.13. Descripción de la tabla sesión.

Nombre: categoría_prof		
Descripción: En esta tabla se almacenan los datos de la categoría de un profesor.		
Atributo	Tipo	Descripción
id _ categoría_prof	int(50)	Identificador de la categoría.
categoría	varchar(50)	Nombre de la categoría (interno, externo, adjunto).

Tabla 3.14. Descripción de la tabla categoría_prof.

Nombre: clase		
Descripción: En esta tabla se almacenan los datos correspondientes a cada clase que se imparte en un grupo determinado.		
Atributo	Tipo	Descripción
id _ clase	int(50)	Identificador de la clase.
id _ asignatura	int(50)	Identificador de la asignatura.
id _ sesión	int(50)	Identificador de la sesión.
solapín	int(50)	Identificador del profesor.
id _ horario	int(50)	Identificador del horario que se va a crear.

Tabla 3.15. Descripción de la tabla clase.

Nombre: tipo _ clase		
Descripción: En esta tabla se almacenan el tipo de clase de una asignatura.		
Atributo	Tipo	Descripción
id _ tipo _ clase	int(50)	Identificador del tipo de clase.
tipo _ clase	int(50)	Nombre del tipo de clase (conferencia, clase práctica, laboratorio).

Tabla 3.16. Descripción de la tabla tipo _ clase.

Nombre: tipo _ local		
Descripción: En esta tabla se almacenan el tipo de local.		
Atributo	Tipo	Descripción
id _ tipo _ local	int(50)	Identificador del tipo de local.
tipo _ local	varchar(50)	Nombre del tipo de local (aula, salón, laboratorio).

Tabla 3.17. Descripción de la tabla tipo _ local.

Nombre: asignatura _ bloqueada		
Descripción: En esta tabla se almacenan los datos de cada asignatura _ bloqueada.		
Atributo	Tipo	Descripción
id_asignatura	int(50)	Identificador de la asignatura.
semana _ inicio	int(50)	Semana que inicia la signatura bloqueada.
cant_semanas	int(50)	Cantidad de semanas que dura la asignatura bloqueada.

Tabla 3.18. Descripción de la tabla asignatura _ bloqueada.

3.5 Interfaz de usuario

La interfaz de usuario es la forma en que los usuarios pueden comunicarse con una computadora, y comprende todos los puntos de contacto entre el usuario y el equipo. Sus principales funciones son:

- ✓ Manipulación de archivos y directorios.
- ✓ Herramientas de desarrollo de aplicaciones.
- ✓ Comunicación con otros sistemas.
- ✓ Información de estado.
- ✓ Configuración de la propia interfaz y entorno.
- ✓ Intercambio de datos entre aplicaciones.
- ✓ Control de acceso
- ✓ Sistema de ayuda interactivo.

Para facilitar la utilización de la aplicación, la misma presentará una interfaz sencilla y de fácil uso. Se ha tenido especial interés en no sobrecargar las mismas, cuenta con barra de menú con diferentes opciones se encontrará en la parte superior de la aplicación de forma horizontal, donde se incluyen otros servicios a los que puede acceder el usuario.

Gracias a los aspectos anteriormente descritos se garantiza una rápida y fácil adaptación por parte de los usuarios con el área de trabajo del sistema.

3.6 Tratamiento de errores

Identificar y controlar los posibles errores que se pueden presentar a la hora de interactuar con el software es de vital importancia para así garantizar un correcto funcionamiento del sistema. El sistema se propone prevenir al máximo los posibles errores, sobre todo los que ocurren por parte del usuario, para ello solo se le brindan las opciones a las que tiene acceso, las demás se mantendrán inhabilitadas. Se insistió en que el usuario realizara la menor cantidad posible de entrada de datos por lo que se aprovecharon al máximo los componentes visuales de selección. Al introducir información en un formulario se verifican los campos obligatorios, la veracidad y tipo de los datos. En caso de error se emite un mensaje de alerta, especificando claramente en que consiste para su fácil comprensión por el usuario. Se tratan estos errores de forma tal que las interacciones con la base de datos se realicen de forma correcta. Se mostraran mensajes de confirmación en el caso de las inserciones y eliminaciones de manera que no se realice una operación no deseada.

Conclusiones

Tras obtener el Modelo de Análisis como entrada fundamental para las actividades de diseño, conservar en las mismas todo lo posible la estructura que este define, estructurar la base de datos que soportará al sistema, y profundizar en características de la interfaz de usuario y la aplicación de forma general, se ha creado una entrada apropiada y un punto de partida para actividades de implementación y prueba.

CONCLUSIONES

Después de la realización del presente trabajo se llegó a la conclusión de que:

- ✓ Se cumplieron todos los objetivos propuestos.
- ✓ Se realizó un estudio del arte sobre la planificación de horarios docentes y de las metodologías de desarrollo de software y se identificaron las más idóneas para darle solución al problema propuesto.
- ✓ Con la una investigación de los procesos de planificación del horario docente de la facultad 4 y el análisis de los factores de los que depende el mismo se pudieron identificar los más importantes a considerar para la confección del horario docente y posteriormente realizar el Análisis y Diseño de una herramienta informática que le dé solución al problema a resolver.

RECOMENDACIONES

Después de la realización de este trabajo se recomienda:

- ✓ Realizar la implementación de dicho sistema para que los involucrados en este proceso cuenten con una herramienta de trabajo que sea capaz de realizar una rápida planificación del horario docente y que ayude en la toma de decisiones en este ámbito.
- ✓ Estudiar las técnicas de Inteligencia Artificial para una mejora y optimización del sistema generador del Horario Docente.

BIBLIOGRAFÍA

1. **BERAZA, Prof. Miguel A. ZABALZA. 2004.** Guía de guías. [Online] 2004.
2. <http://www.fv.ulpgc.es/ficheros/guiadeguias.pdf>.
3. **INNOVA, GRUPO SOLUCIONES.** Rational Rose Enterprise. [Online]
<http://www.rational.com.ar/herramientas/roseenterprise.html>.
4. **Novell. 2004.** [Online] 2004. <http://www.channelplanet.com/index.php?idcategoria=12724>.
5. **Pablo Gervás, Beatriz San Miguel. 1999.** Un sistema experto basado en reglas para la automatización del horario. [Online] 1999.
<http://www.fdi.ucm.es/profesor/pgervas/papers/gervasCAEPIA1999.pdf> .
6. **PLANIFICACIÓN DEL ESTUDIO.** [Online]
http://perso.wanadoo.es/angel.saez/pagina_nueva_145.htm.
7. **Sanchez, María A. Mendoza. 2004.** Metodologías De Desarrollo De Software. [Online] 2004.
http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html
8. **Seco, José Antonio González. 2002.** EL LENGUAJE DE PROGRAMACIÓN C#. [Online] 2002. <http://www.programacion.com/tutorial/csharp/3/>.
9. **Sergio Cleger Tamayo, Marisol Pérez Campaña, Félix Rodríguez Expósito. 2007.** Alternativa para el proceso de planificación de horarios docentes de una Universidad. [Online] 2007.
<http://www.ciencias.holguin.cu/2007/Diciembre/articulos/ARTI6.htm>.
10. **Unai Extremo Baigorri, Borja Sotomayor Basili. 2002.** La plataforma .NET:¿El futuro de la Web? [Online] 2002. <http://people.cs.uchicago.edu/~borja/pubs/revistaeside2002.pdf>.
11. **Web. 2005.** Generador de Horarios para Centros Docentes. [Online] 2005. <http://generador-de-horarios-para-centros-docentes.pro.softonic.com/>.

TRABAJOS CITADOS

1. **BERAZA, Prof. Miguel A. ZABALZA. 2004.** Guía de guías. [Online] 2004.
<http://www.fv.ulpgc.es/ficheros/guiadeguias.pdf>.
2. **Cañar, David.** [Online] http://es.wikibooks.org/wiki/Programaci%C3%B3n:C_sharp_NET.
3. **Ciberaula. 2006.** ¿Qué es Java? [Online] 2006. http://java.ciberaula.com/articulo/que_es_java/.
4. **Group, Linus Party. 2006.** Linus Party Group. [Online] 2006. <http://www.linux-party.com/modules.php?name=News&file=article&sid=3055>.
5. **INNOVA, GRUPO SOLUCIONES.** Rational Rose Enterprise. [Online]
<http://www.rational.com.ar/herramientas/roseenterprise.html>.
6. **MIC, Ministerio de la Informática y las Comunicaciones. 2002.** Informatización de la Sociedad. [Online] 2002. <http://www.mic.gov.cu/hinfosoc.aspx>.
7. **Netveloper. 2002.** SQL Server 2000. [Online] 2002.
http://www.netveloper.com/contenido2.aspx?IDC=64_0.
8. **Novell. 2004.** [Online] 2004. <http://www.channelplanet.com/index.php?idcategoria=12724>.
9. **Pablo Gervás, Beatriz San Miguel. 1999.** Un sistema experto basado en reglas para la automatización del horario. [Online] 1999.
<http://www.fdi.ucm.es/profesor/pgervas/papers/gervasCAEPIA1999.pdf>.
10. **Peralta, Manuel.** Sistema de Información. [Online]
<http://www.monografias.com/trabajos7/sisinf/sisinf.shtml>.
11. **PLANIFICACIÓN DEL ESTUDIO.** [Online]
http://perso.wanadoo.es/angel.saez/pagina_nueva_145.htm.
12. **Pozo, Salvador. 2004.** My SQL. [Online] 2004. <http://mysql.conclase.net/curso/index.php>.
13. **Sanchez, María A. Mendoza. 2004.** Metodologías De Desarrollo De Software. [Online] 2004.
http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html
14. **Seco, José Antonio González. 2002.** EL LENGUAJE DE PROGRAMACIÓN C#. [Online] 2002. <http://www.programacion.com/tutorial/csharp/3/>.
15. **Sequera, Richard A.** Lenguaje C . [Online]
<http://www.monografias.com/trabajos4/lenguajec/lenguajec.shtml>.
16. **Sierra, Fco. Javier Ceballos. 2001.** EL LENGUAJE DE PROGRAMACIÓN C#. [Online] 2001. <http://www.agapea.com/El-lenguaje-de-programacion-C--n11766i.htm>.

17. **UCI. 2007.** [Online] 2007. <http://teleformacion.uci.cu/mod/resource/view.php?id=8865>.
18. —. **2007.** Teleformación. [Online] 2007.
<http://teleformacion.uci.cu/mod/resource/view.php?id=6680>.
19. **UCI, Universidad de las Ciencias Infomaticas. 2007.** [Online] 2007.
<http://teleformacion.uci.cu/mod/resource/view.php?id=6655>.
20. **Unai Extremo Baigorri, Borja Sotomayor Basili. 2002.** La plataforma .NET:¿El futuro de la Web? [Online] 2002. <http://people.cs.uchicago.edu/~borja/pubs/revistaeside2002.pdf>.
21. **Web. 2005.** Generador de Horarios para Centros Docentes. [Online] 2005. <http://generador-de-horarios-para-centros-docentes.pro.softonic.com/>.

ANEXOS

Anexo 1: Descripción de los casos de uso del sistema.

Descripción del caso de uso Gestionar Local

Nombre del Caso de uso	CU_Gestionar Local
Actores	Planificador
Propósito	Permitir registrar los datos de los locales en el sistema.
Resumen	
El caso de uso inicia cuando el planificador selecciona la opción "Gestionar Local", la cual permite insertar en el sistema los datos correspondientes a los locales.	
Precondiciones	El planificador debe contar con todos los datos correspondientes al local.
Casos de Uso Asociados	No tiene
Referencia	3.1, 3.2, 3.3
Flujo Normal de Eventos	
Acción del Actor	Respuesta del sistema
1. El planificador selecciona la opción "Gestionar Local".	1.1 El sistema despliega un menú con las opciones para la gestión del local. a) Para insertar local ver sección Insertar. b) Para eliminar local ver sección Eliminar. c) Para modificar local ver sección Modificar.
Sección Insertar	
1. El planificador selecciona la opción "Insertar Local".	1.1 El sistema muestra la interfaz para entrar los datos del local.
2. El planificador entra los datos del local (id _ local, nombre, tipo de local, capacidad, piso) y presiona el botón "Registrar".	2.1 El sistema valida los datos entrados. 2.2 El sistema registra en la base de datos el local.
Flujo Alternativo de Eventos	
	2.1 En caso de que ya exista el local en la base de datos el sistema muestra un mensaje de error.

2. El planificador presiona el botón "Cancelar".	2.1 El sistema pasa a la acción 1
Sección Eliminar	
1. El planificador selecciona la opción "Eliminar Local".	1.1 El sistema muestra la interfaz para escoger el local que se desea eliminar.
2. El planificador selecciona el local que eliminará y presiona el botón "Eliminar".	2.1 El sistema muestra un mensaje de confirmación "Desea eliminar el local"
3. El planificador presiona el botón "Aceptar".	3.1 El sistema elimina el local de la base de datos.
Flujo Alternativo de Eventos	
	1.1 Si no existe ningún local en la base de datos, el sistema mostrará un mensaje de error.
3. El planificador presiona el botón "Cancelar".	3.1 El sistema pasa a la acción 1.1.
Sección Modificar	
1. El planificador selecciona la opción "Modificar Local".	1.1 El sistema muestra la interfaz para escoger el local que se desea modificar.
2. El planificador selecciona el local que se modificará.	2.1 El sistema muestra los datos del local.
3. El planificador hace los cambios y presiona el botón "Modificar".	3.1 El sistema actualiza los datos del local en la base de datos.
Flujo Alternativo de Eventos	
	1.1 Si no existe ningún local en la base de datos, el sistema mostrará un mensaje de error.
3. El planificador presiona el botón "Cancelar".	3.1 El sistema pasa a la acción 1.
Poscondiciones	Se almacenan en la base de datos las informaciones.
Prioridad	Crítico

Tabla 2.9. Descripción del caso de uso del sistema Gestionar Local

Descripción del caso de uso Gestionar Grupo

Nombre del Caso de uso	CU_Gestionar Grupo
Actores	Planificador
Propósito	Permitir registrar los datos de los grupos en el sistema.
Resumen	
El caso de uso inicia cuando el planificador selecciona la opción "Gestionar Grupo", la cual permite insertar en el sistema los datos correspondientes a los grupos.	
Precondiciones	El planificador debe contar con todos los datos correspondientes al grupo.
Casos de Uso Asociados	No tiene
Referencia	4.1, 4.2, 4.3
Flujo Normal de Eventos	
Acción del Actor	Respuesta del sistema
1. El planificador selecciona la opción "Gestionar Grupo".	1.1 El sistema despliega un menú con las opciones para la gestión del grupo. a) Para insertar grupo ver sección Insertar. b) Para eliminar grupo ver sección Eliminar. c) Para modificar grupo ver sección Modificar.
Sección Insertar	
1. El planificador selecciona la opción "Insertar Grupo".	1.1 El sistema muestra la interfaz para entrar los datos del grupo.
2. El planificador entra los datos del grupo (id _ grupo, nombre, cantidad de estudiantes, año) y presiona el botón "Registrar".	2.1 El sistema valida los datos entrados. 2.2 El sistema registra en la base de datos el grupo.
Flujo Alternativo de Eventos	
	2.1 En caso de que ya exista el grupo en la base de datos el sistema muestra un mensaje de error.
2. El planificador presiona el botón "Cancelar".	2.1 El sistema pasa a la acción 1

Sección Eliminar	
1. El planificador selecciona la opción "Eliminar Grupo".	1.1 El sistema muestra la interfaz para escoger el grupo que se desea eliminar.
2. El planificador escoge el año del cual va a eliminar el grupo.	2.1 El sistema activa el campo para escoger el grupo que se va a eliminar
3. El planificador selecciona el grupo que eliminará y presiona el botón "Eliminar".	3.1 El sistema muestra un mensaje de confirmación "Desea eliminar el grupo".
4. El planificador presiona el botón "Aceptar"	4.1 El sistema elimina el grupo de la base de datos.
Flujo Alternativo de Eventos	
	1.1 Si no existe ningún grupo en la BD, el sistema mostrará un mensaje de error.
4. El planificador presiona el botón "Cancelar"	4.1 El sistema pasa la acción 1.1.
Sección Modificar	
1. El planificador selecciona la opción "Modificar Grupo".	1.1 El sistema muestra la interfaz para escoger el grupo que se desea modificar.
2. El planificador escoge el año del cual va a modificar el grupo.	2.1 El sistema activa el campo para escoger el grupo que se va a modificar.
3. El planificador selecciona el grupo que se modificará.	3.1 El sistema muestra los datos del grupo.
4. El planificador hace los cambios y presiona el botón "Modificar".	3.1 El sistema actualiza los datos del grupo en la base de datos.
Flujo Alternativo de Eventos	
	1.1 Si no existe ningún grupo en la base de datos, el sistema mostrará un mensaje de error.
3. El planificador presiona el botón "Cancelar".	3.1 El sistema pasa a la acción 1.
Poscondiciones	Se almacenan en la base de datos las informaciones.
Prioridad	Crítico

Tabla 2.10. Descripción del caso de uso del sistema Gestionar Grupo.

Descripción del caso de uso Gestionar Dpto.

Nombre del Caso de uso	CU_Gestionar Dpto.
Actores	Planificador
Propósito	Permitir registrar los datos de los Dpto. en el sistema.
Resumen	
El caso de uso inicia cuando el planificador selecciona la opción “Gestionar Dpto.”, la cual permite insertar en el sistema los datos correspondientes a los Dpto.	
Precondiciones	El planificador debe contar con todos los datos correspondientes al Dpto.
Casos de Uso Asociados	No tiene
Referencia	5.1, 5.2, 5.3
Flujo Normal de Eventos	
Acción del Actor	Respuesta del sistema
1. El planificador selecciona la opción “Gestionar Dpto.”.	1.1 El sistema despliega un menú con las opciones para la gestión del Dpto. a) Para insertar Dpto. ver sección Insertar. b) Para eliminar Dpto. ver sección Eliminar. c) Para modificar Dpto. ver sección Modificar.
Sección Insertar	
1. El planificador selecciona la opción “Insertar Dpto.”.	1.1 El sistema muestra la interfaz para entrar los datos del Dpto.
2. El planificador entra los datos del Dpto. (id _ departamento, nombre, día de preparación metodológica) y presiona el botón “Registrar”.	2.1 El sistema valida los datos entrados. 2.2 El sistema registra en la base de datos el Dpto.
Flujo Alternativo de Eventos	
	2.1 En caso de que el Dpto. ya exista en la base de datos el sistema muestra un mensaje de error.
2. El planificador presiona el botón “Cancelar”	2.1 El sistema pasa a la acción 1

Sección Eliminar	
1. El planificador selecciona la opción "Eliminar Dpto."	1.1 El sistema muestra la interfaz para escoger el Dpto. que se desea eliminar.
2. El planificador selecciona el Dpto. que eliminará y presiona el botón "Eliminar".	2.1 El sistema muestra un mensaje de confirmación "Desea eliminar el Dpto."
3. El planificador presiona el botón "Aceptar"	3.1 El sistema elimina el grupo de la base de datos.
Flujo Alternativo de Eventos	
	1.1 Si no existe ningún Dpto. en la base de datos, el sistema mostrará un mensaje de error.
3. El planificador presiona el botón "Cancelar"	3.1 El sistema pasa a la acción 1.1.
Sección Modificar	
1. El planificador selecciona la opción "Modificar Dpto."	1.1 El sistema muestra la interfaz para escoger el Dpto. que se desea modificar.
2. El planificador selecciona el Dpto. que se modificará.	2.1 El sistema muestra los datos del Dpto.
3. El planificador hace los cambios y presiona el botón "Modificar".	3.1 El sistema actualiza los datos del Dpto. en la base de datos.
Flujo Alternativo de Eventos	
	1.1 Si no existe ningún Dpto. en la base de datos, el sistema mostrará un mensaje de error.
3. El planificador presiona el botón "Cancelar".	3.1 El sistema pasa a la acción 1.
Poscondiciones	Se almacenan en la base de datos las informaciones.
Prioridad	Crítico

Tabla 2.11. Descripción del caso de uso del sistema Gestionar Dpto.

Descripción del caso de uso Gestionar Semana.

Nombre del Caso de uso	CU_Gestionar Semana.
Actores	Planificador
Propósito	Permitir registrar los datos de la semana en el sistema.
Resumen	
El caso de uso inicia cuando el planificador selecciona la opción "Gestionar Semana", la cual permite insertar en el sistema los datos correspondientes a la semana.	
Precondiciones	El planificador debe contar con todos los datos correspondientes a la semana
Casos de Uso Asociados	No tiene
Referencia	6.1, 6.2, 6.3
Flujo Normal de Eventos	
Acción del Actor	Respuesta del sistema
1. El planificador selecciona la opción "Gestionar Semana".	1.1 El sistema despliega un menú con las opciones para la gestión de la semana. a) Para insertar semana ver sección Insertar. b) Para eliminar semana ver sección Eliminar. c) Para modificar semana ver sección Modificar.
Sección Insertar	
1. El planificador selecciona la opción "Insertar Semana".	1.1 El sistema muestra la interfaz para entrar los datos de la semana
2. El planificador entra los datos de la semana (id _ semana, número, número semestre) y presiona el botón "Registrar".	2.1 El sistema valida los datos entrados. 2.2 El sistema registra en la base de datos los datos de la semana.
Flujo Alterno de Eventos	
	2.1 En caso de que ya exista la semana en la base de datos el sistema muestra un mensaje de error.
2. El planificador presiona el botón "Cancelar"	2.1 El sistema pasa a la acción 1.
Sección Eliminar	
1. El planificador selecciona la opción "Eliminar Semana".	1.1 El sistema muestra la interfaz para escoger la semana que se desea eliminar.

2. El planificador selección la semana que eliminará y presiona el botón "Eliminar".	2.1 El sistema muestra un mensaje de confirmación "Desea eliminar la semana".
3. El planificador presiona el botón "Aceptar".	3.1 El sistema elimina la semana de la base de datos.
Flujo Alternativo de Eventos	
	1.1 Si no existe ninguna semana en la base de datos, el sistema mostrará un mensaje de error.
3. El planificador presiona el botón "Cancelar".	3.1 El sistema pasa a la acción 1.1.
Sección Modificar	
1. El planificador selecciona la opción "Modificar Semana".	1.1 El sistema muestra la interfaz para escoger la semana que se desea modificar.
2. El planificador selecciona la semana que se modificará.	2.1 El sistema muestra los datos de la semana.
3. El planificador hace los cambios y presiona el botón "Modificar".	3.1 El sistema actualiza los datos de la semana en la base de datos.
Flujo Alternativo de Eventos	
	1.1 Si no existe ninguna semana en la base de datos, el sistema mostrará un mensaje de error.
3. El planificador presiona el botón "Cancelar".	3.1 El sistema pasa a la acción 1.
Poscondiciones	Se almacenan en la base de datos las informaciones.
Prioridad	Crítico

Tabla 2.12. Descripción del caso de uso del sistema Gestionar Semana

Descripción del caso de uso Gestionar Balance de Carga

Nombre del Caso de uso	CU_Gestionar Balance de Carga
Actores	Planificador
Propósito	Permitir registrar los datos del Balance de Carga en el sistema.

Resumen	
El caso de uso inicia cuando el planificador selecciona la opción "Gestionar Balance de Carga", la cual permite insertar en el sistema los datos correspondientes al Balance de Carga.	
Precondiciones	El planificador debe contar con todos los datos del Balance de Carga.
Casos de Uso Asociados	No tiene
Referencia	7.1, 7.2
Flujo Normal de Eventos	
Acción del Actor	Respuesta del sistema
1. El planificador selecciona la opción "Gestionar Balance de Carga".	1.1 El sistema despliega un menú con las opciones para la gestión del Balance de Carga. a) Para insertar Balance de Carga ver sección Insertar. b) Para modificar Balance de Carga ver sección Modificar.
Sección Insertar	
1. El planificador selecciona la opción "Insertar Balance de Carga".	1.1 El sistema muestra la interfaz para entrar los datos del Balance de Carga.
2. El planificador entra los datos del Balance de Carga (semana, asignatura, frecuencias semanales, tipo de clase y orden) y presiona el botón "Registrar".	2.1 El sistema valida los datos entrados. 2.2 El sistema registra en la base de datos el Balance de Carga.
Flujo Alternativo de Eventos	
	2.1 En caso que alguno de los datos entrados no sea correcto el sistema muestra un mensaje de error informando cuál es el dato incorrecto, pasando a la acción 1.1
2. El planificador presiona el botón "Cancelar"	2.1 El sistema pasa a la acción 1.
Sección Modificar	
1. El planificador selecciona la opción "Modificar Balance de Carga".	1.1 El sistema muestra la interfaz para escoger la asignatura que se le va a modificar el Balance de Carga.

2. El planificador escoge la asignatura que se modificará.	2.1 El sistema muestra los datos del Balance de Carga.
3. El planificador hace los cambios y presiona el botón "Modificar".	3.1 El sistema actualiza los datos del Balance de Carga en la base de datos.
Flujo Alternativo de Eventos	
	1.1 Si no existe el Balance de Carga de esa asignatura en la base de datos, el sistema mostrará un mensaje de error.
3. El planificador presiona el botón "Cancelar".	3.1 El sistema pasa a la acción 1.
Poscondiciones	Se almacenan en la base de datos las informaciones.
Prioridad	Crítico

Tabla 2.13. Descripción del caso de uso del sistema Balance de Carga.

Anexo 2: Diagramas de clases del análisis.

Diagrama de análisis del CU_Gestionar Balance de Carga

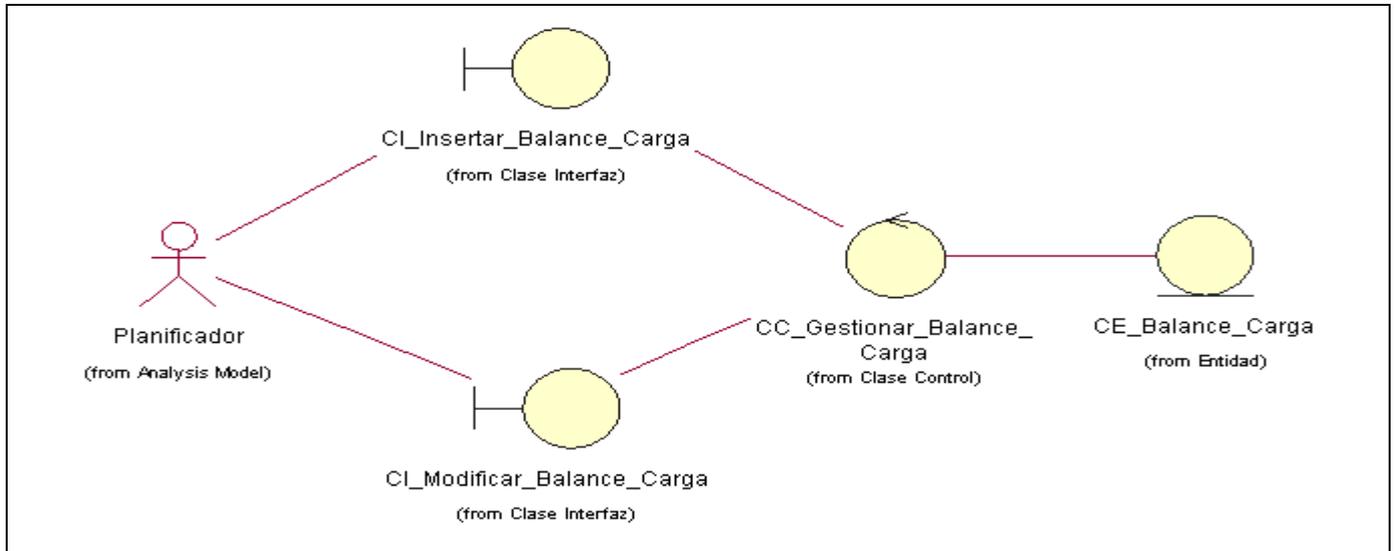


Figura 3.15. Diagrama de análisis del CU_Gestionar Balance de Carga

Diagrama de análisis del CU_Gestionar Dpto.

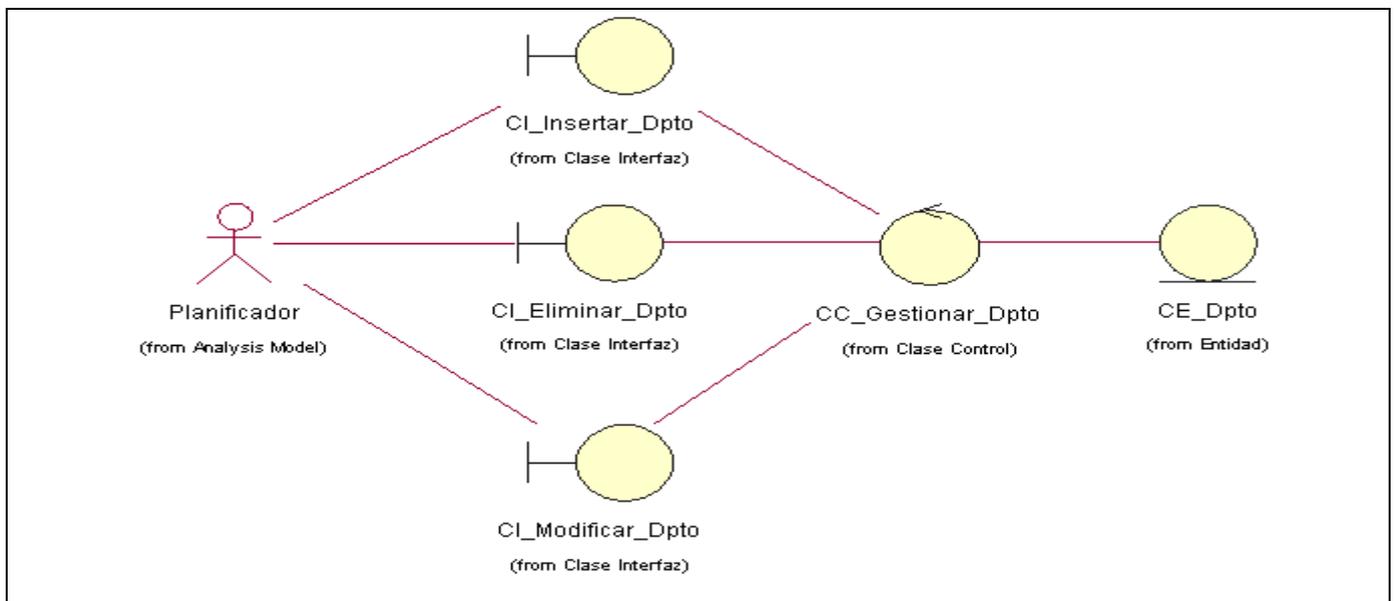


Figura 3.16. Diagrama de análisis del CU_Gestionar Dpto.

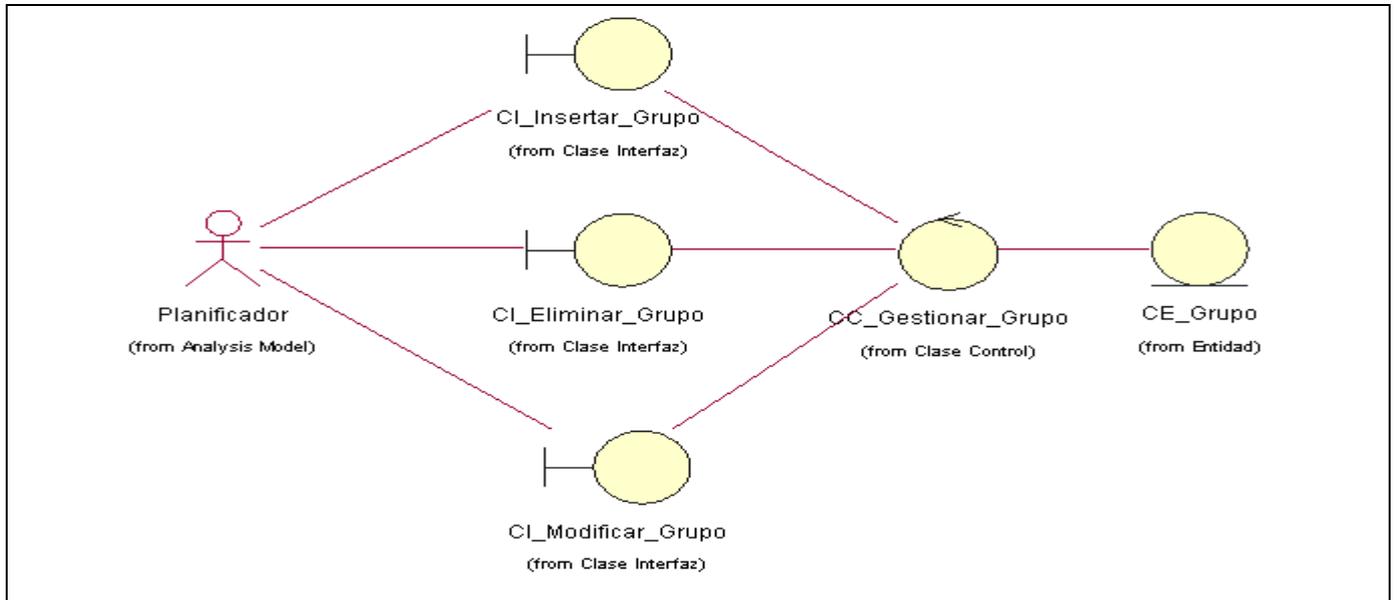
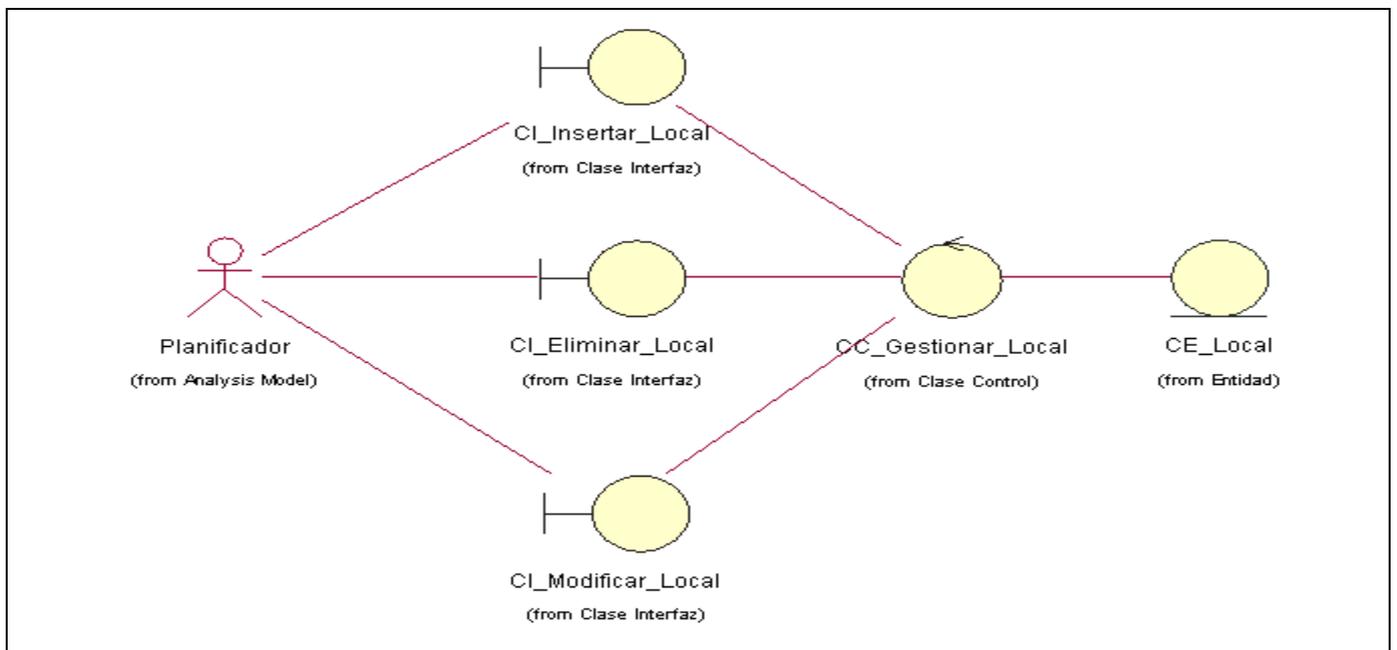
Diagrama de análisis del CU_Gestionar Grupo**Figura 3.17. Diagrama de análisis del CU_Gestionar Grupo****Diagrama de análisis del CU_Gestionar Local****Figura 3.18. Diagrama de análisis del CU_Gestionar Local**

Diagrama de análisis del CU_Gestionar Semana

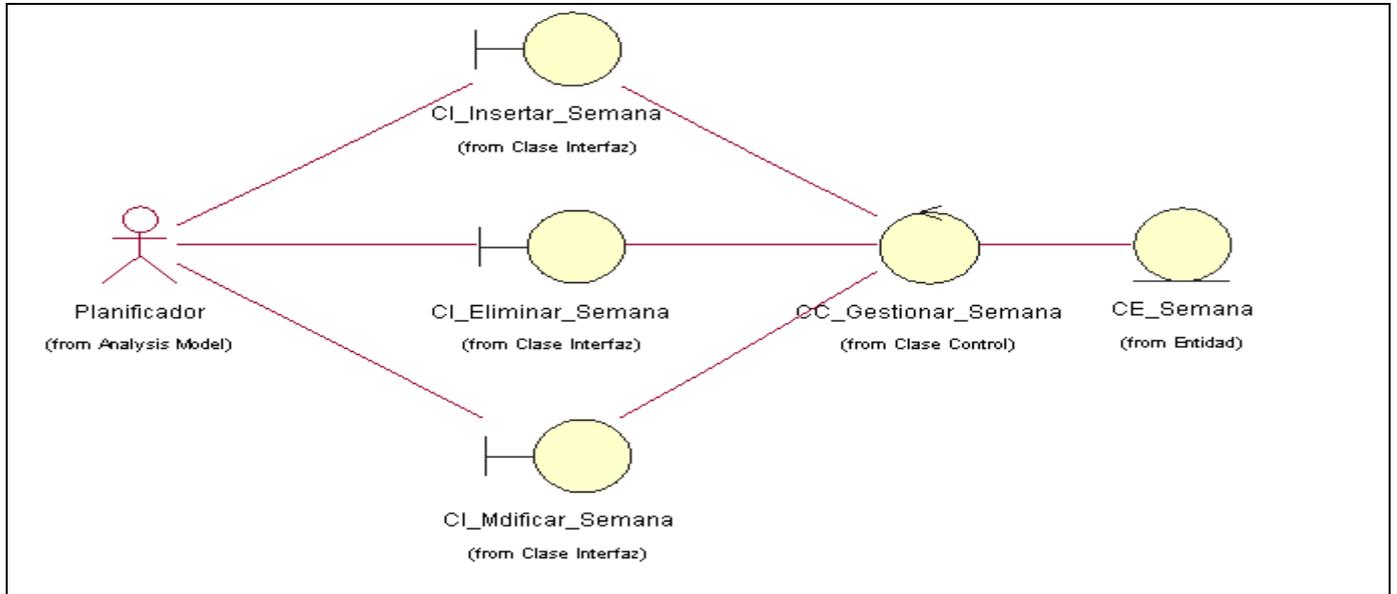


Figura 3.19. Diagrama de análisis del CU_Gestionar Semana

Anexo 3: Diagramas de clases del diseño.

Diagrama de diseño del CU_Gestionar Balance Carga

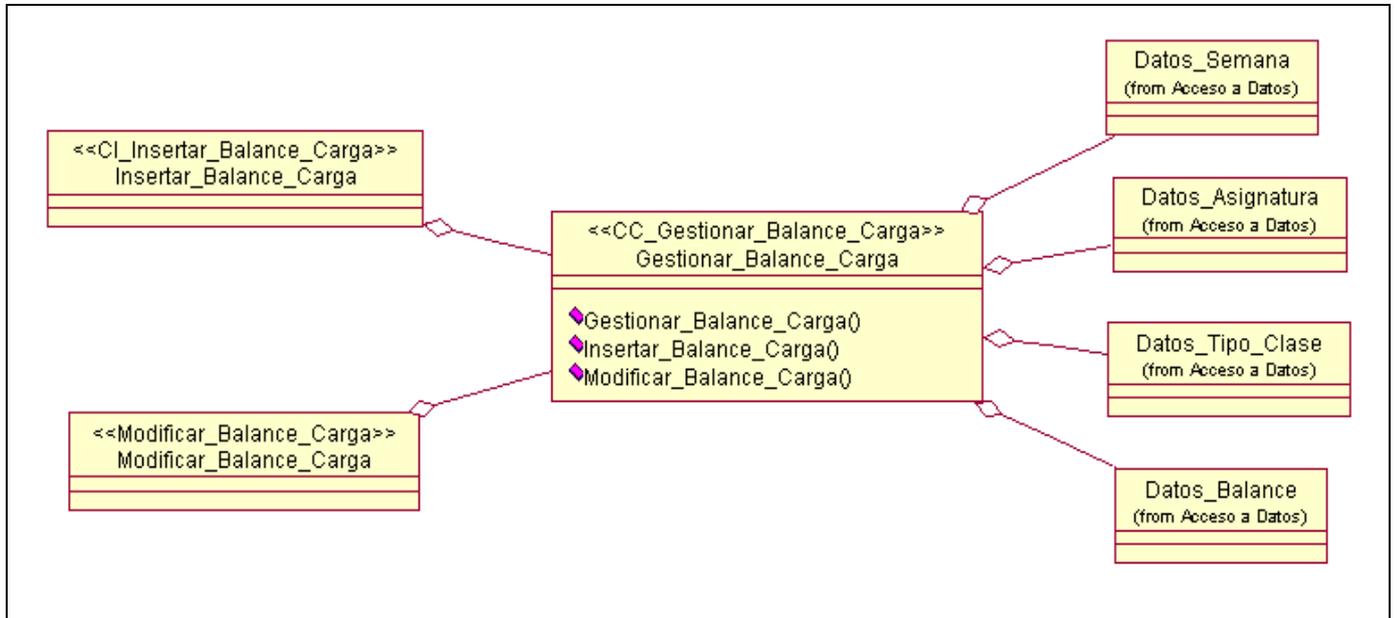


Figura 3.20. Diagrama de diseño del CU_Gestionar Balance Carga

Diagrama de diseño del CU_Gestionar Dpto.

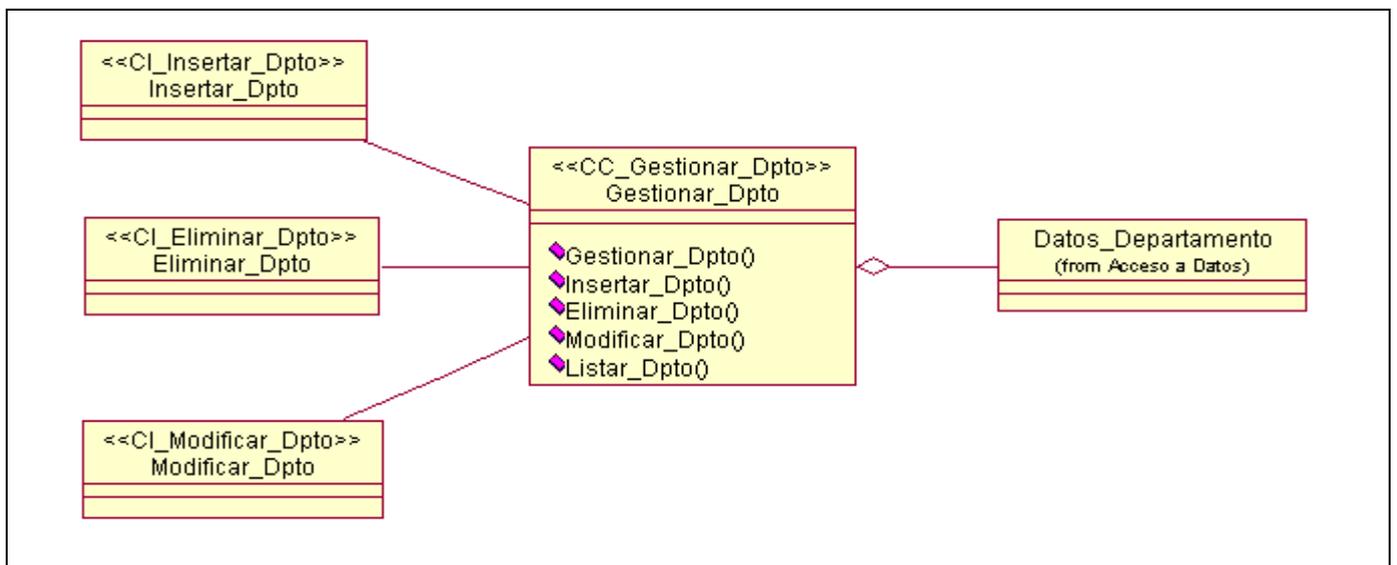


Figura 3.21. Diagrama de diseño del CU_Gestionar Dpto.

Diagrama de diseño del CU_Gestionar Grupo

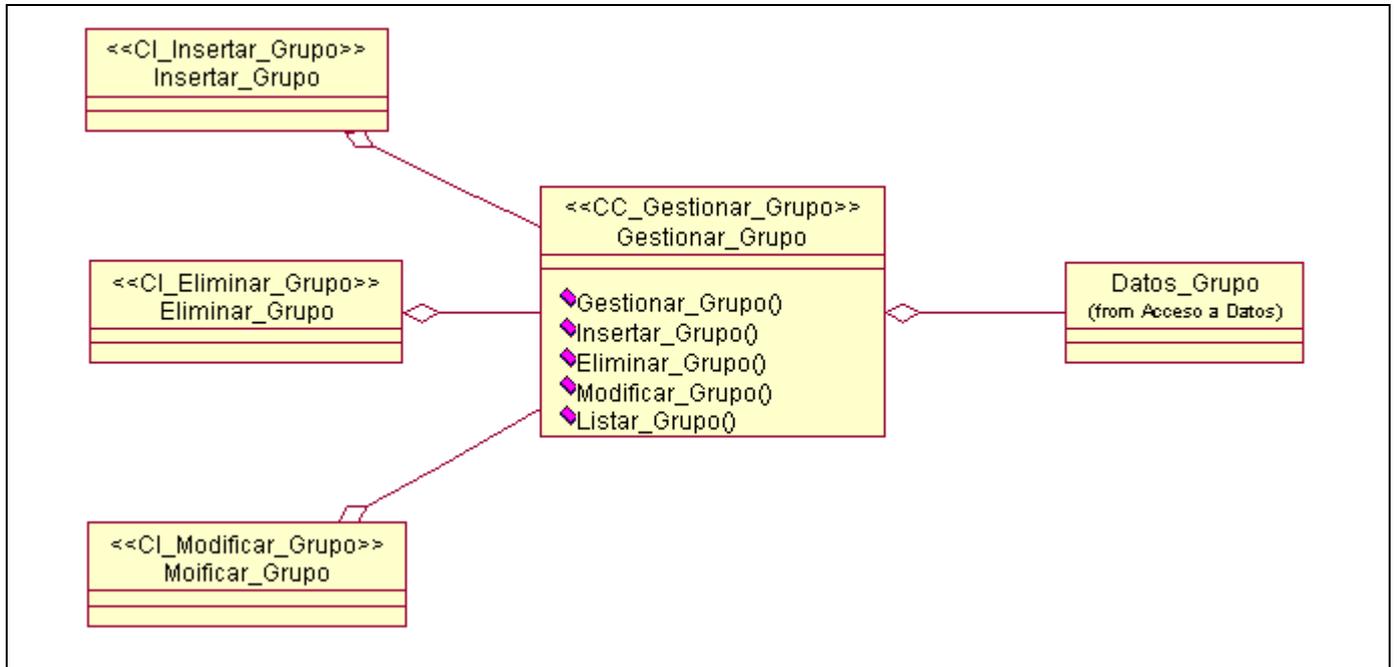


Figura 3.22. Diagrama de diseño del CU_Gestionar Grupo

Diagrama de diseño del CU_Gestionar Local

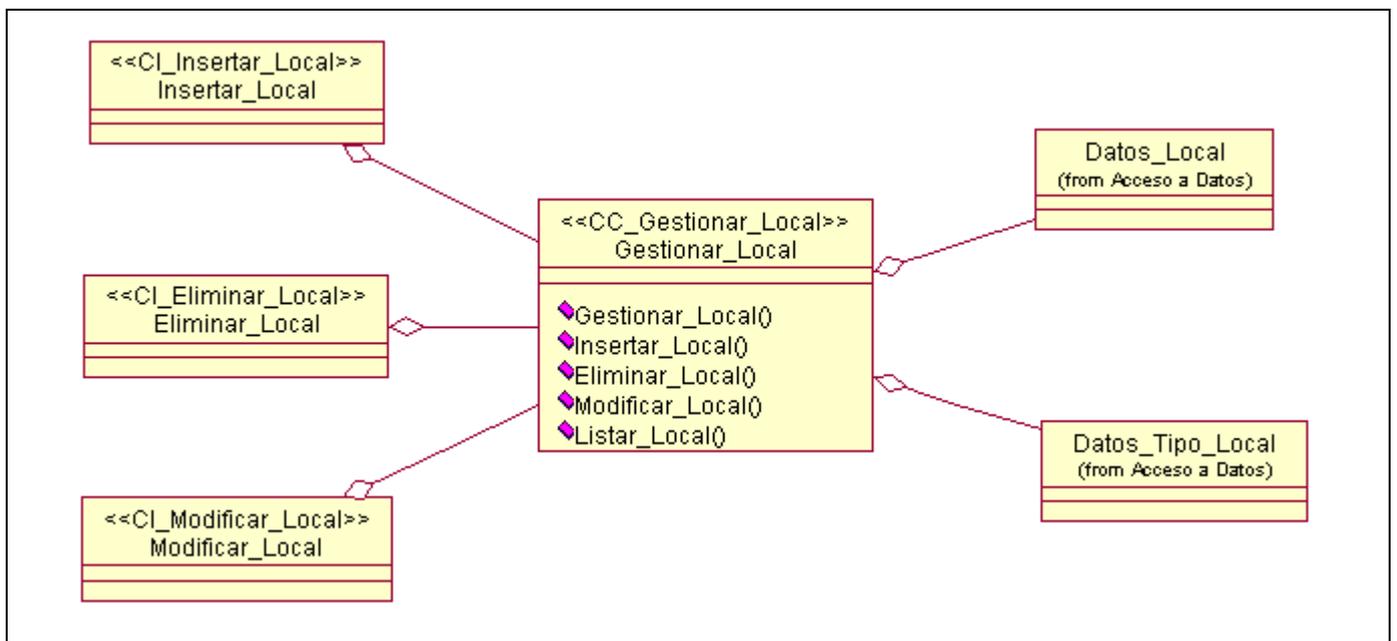
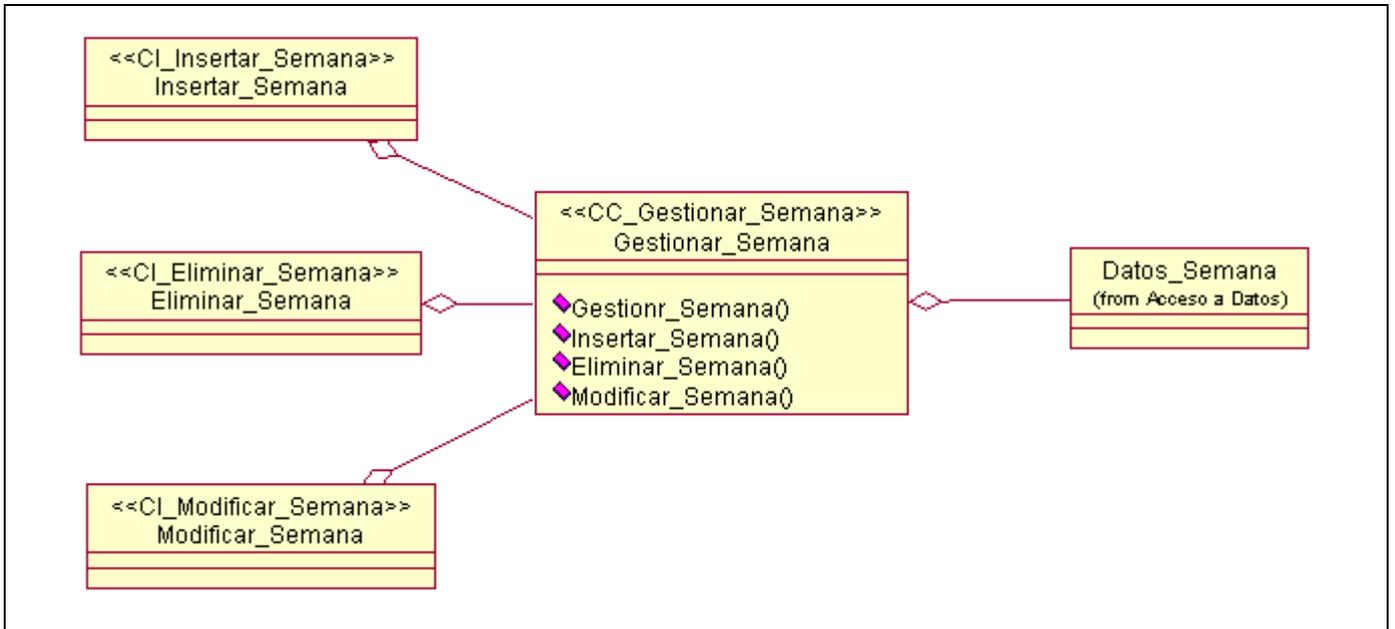


Figura 3.23. Diagrama de diseño del CU_Gestionar Local

Diagrama de diseño del CU_Gestionar Semana**Figura 3.24. Diagrama de diseño del CU_Gestionar Semana**

Anexo 4: Diagramas de secuencia

Diagrama de secuencia del CU_Gestionar Asignatura

Sección Insertar

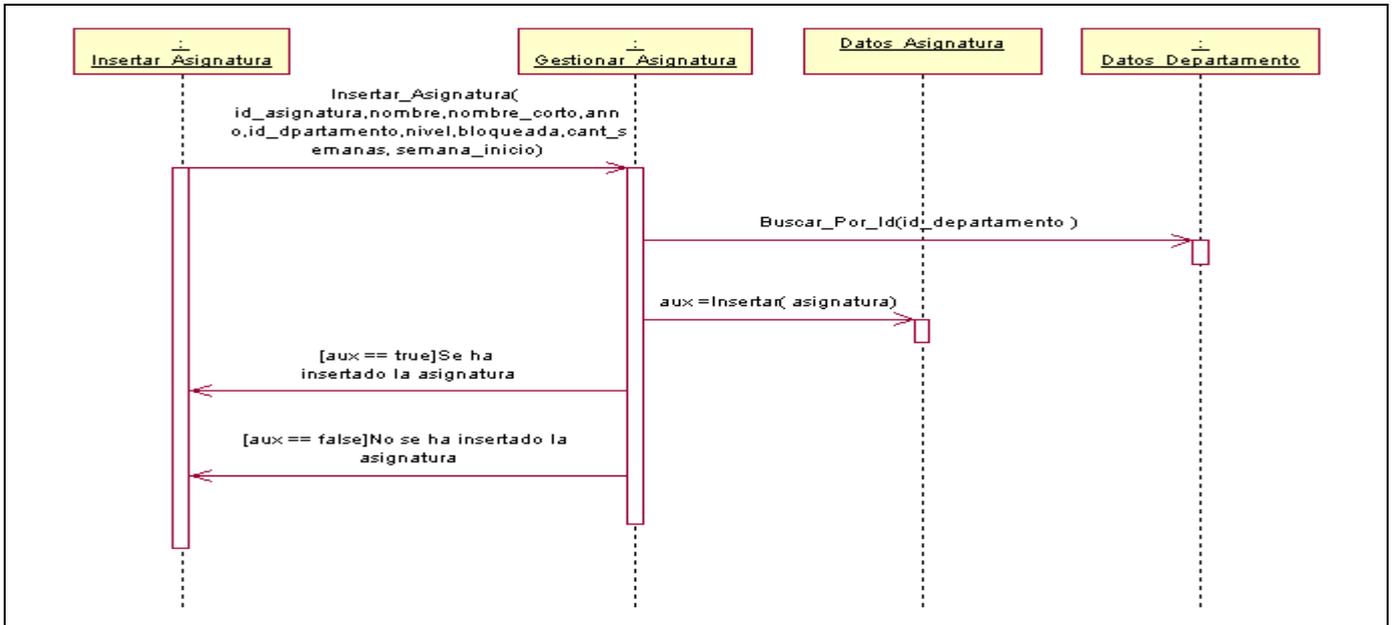


Figura 3.25. Diagrama de secuencia del CU_Gestionar Asignatura: Sección Insertar

Sección Eliminar

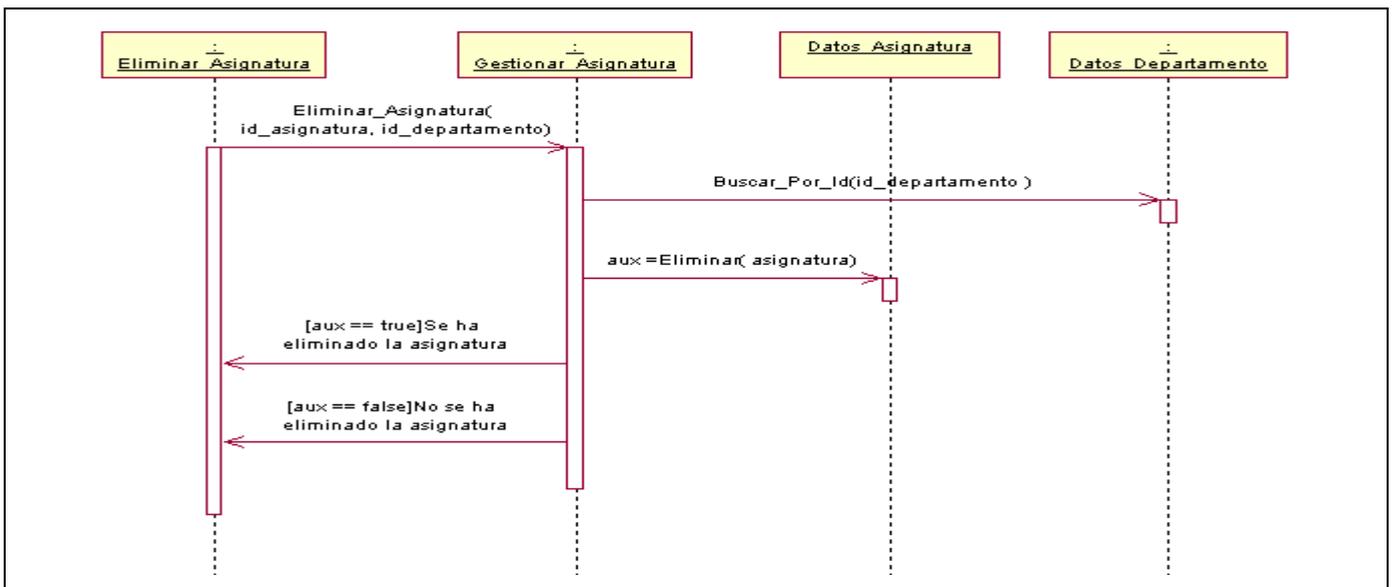


Figura 3.26. Diagrama de secuencia del CU_Gestionar Asignatura: Sección Eliminar

Sección Modificar

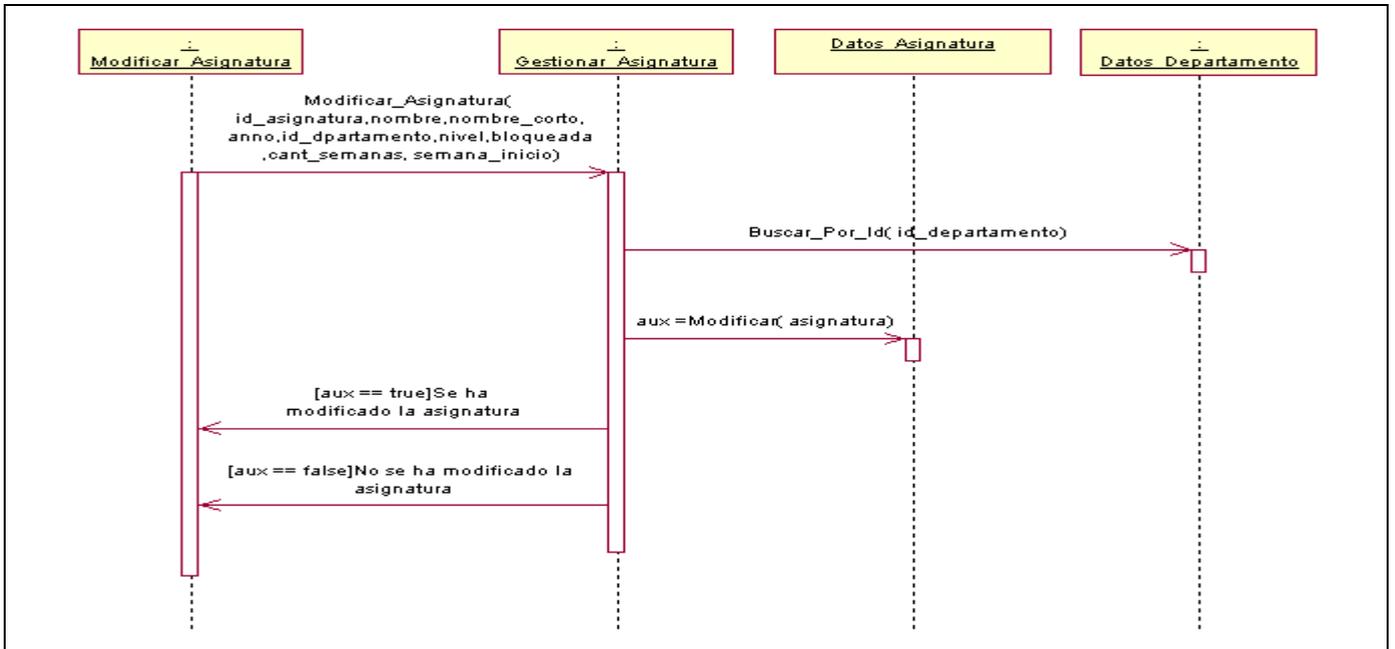


Figura 3.27. Diagrama de secuencia del CU_Gestionar Asignatura: Sección Modificar

Diagrama de secuencia del CU_Gestionar Balance Carga: Sección Insertar

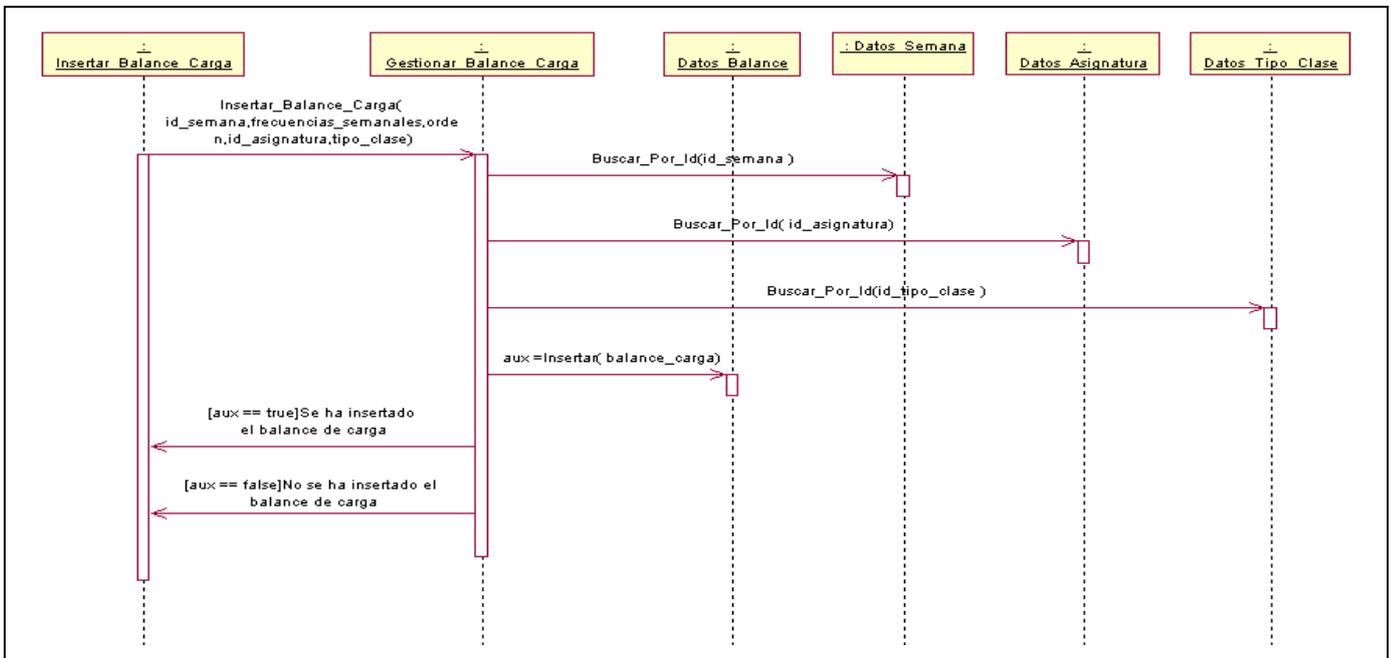


Figura 3.28. Diagrama de secuencia del CU_Gestionar Balance Carga: Sección Insertar

Sección Modificar

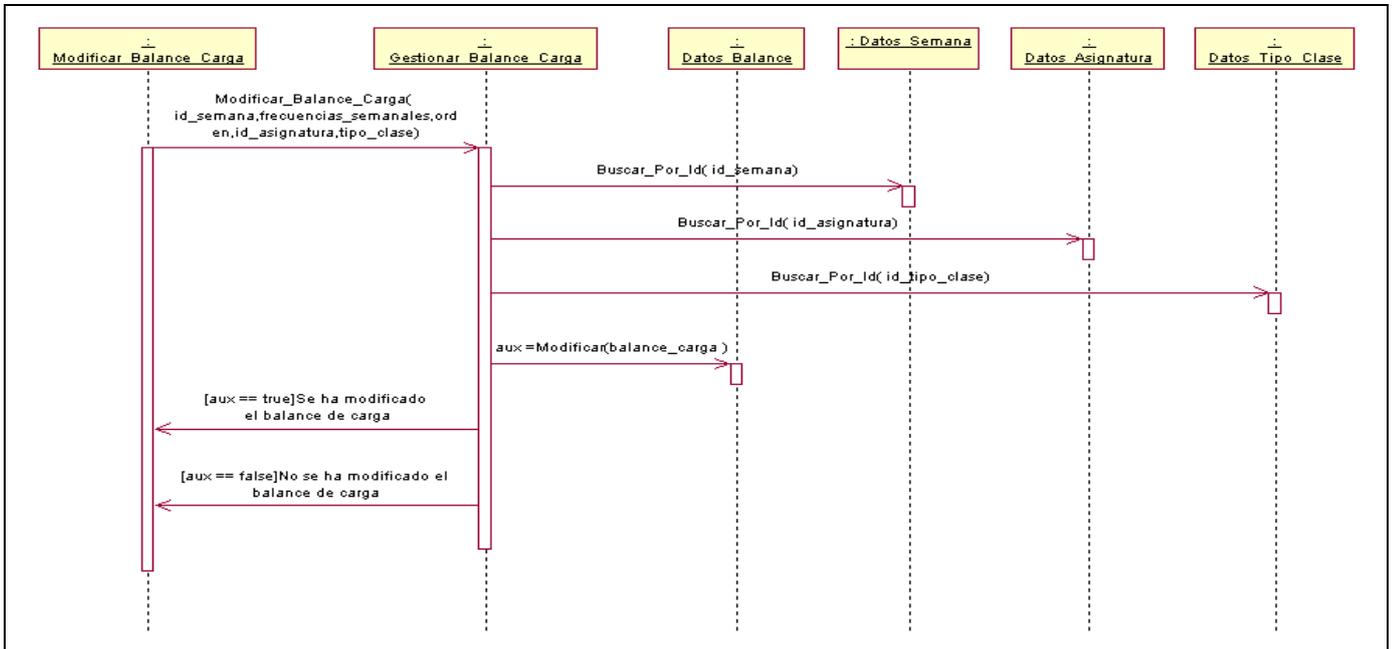


Figura 3.29. Diagrama de secuencia del CU_Gestionar Balance Carga: Sección Modificar

Diagrama de secuencia del CU_Gestionar Dpto: Sección Insertar

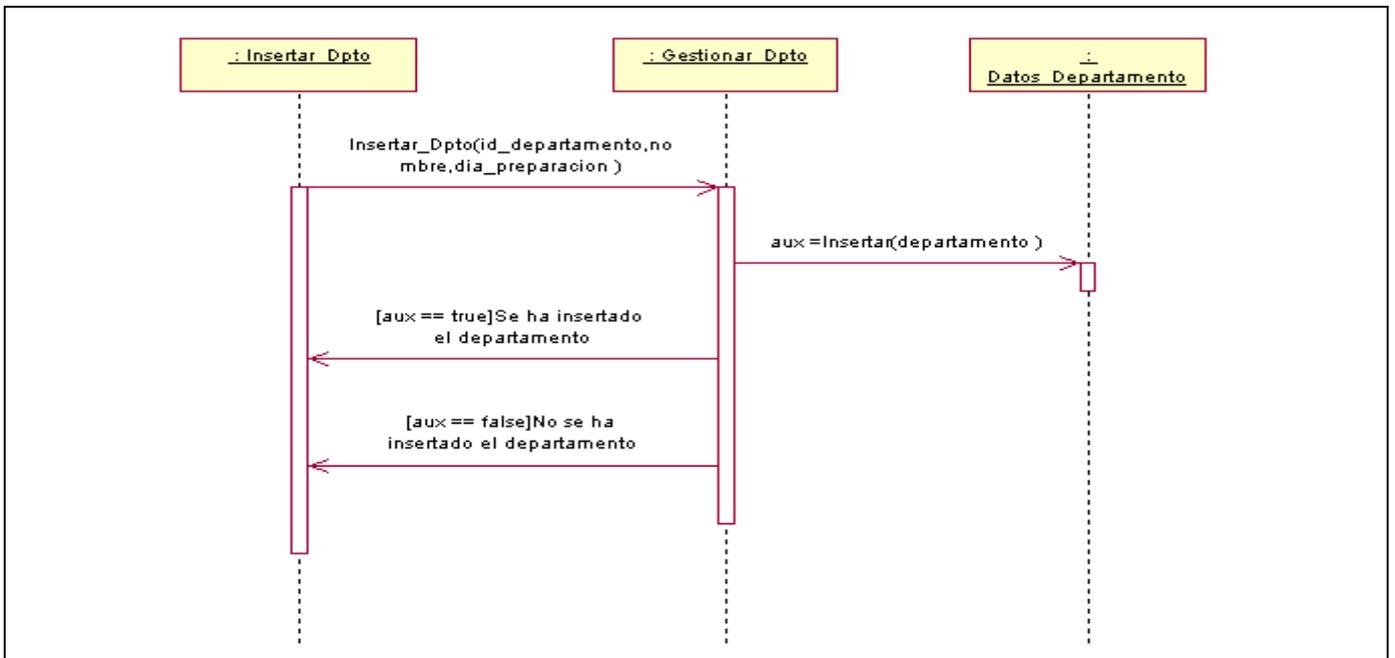


Figura 3.30. Diagrama de secuencia del CU_Gestionar Dpto.: Sección Insertar

Sección Eliminar

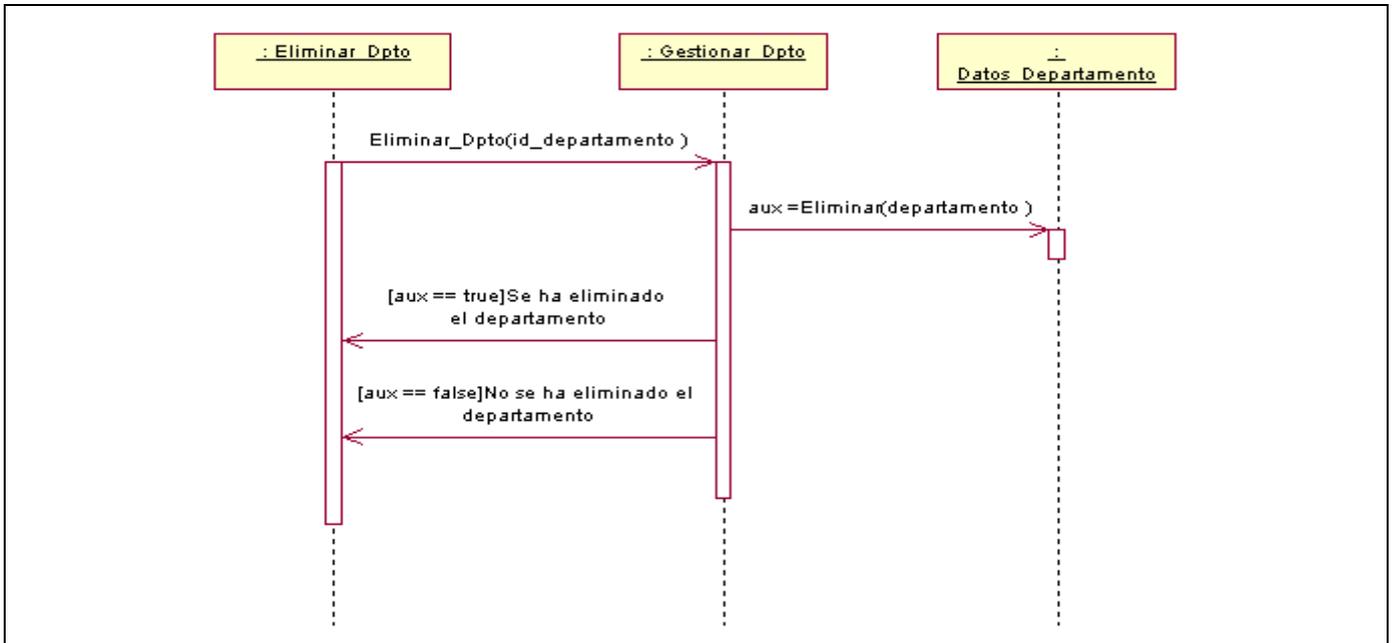


Figura 3.31. Diagrama de secuencia del CU_Gestionar Dpto.: Sección Insertar

Sección Modificar

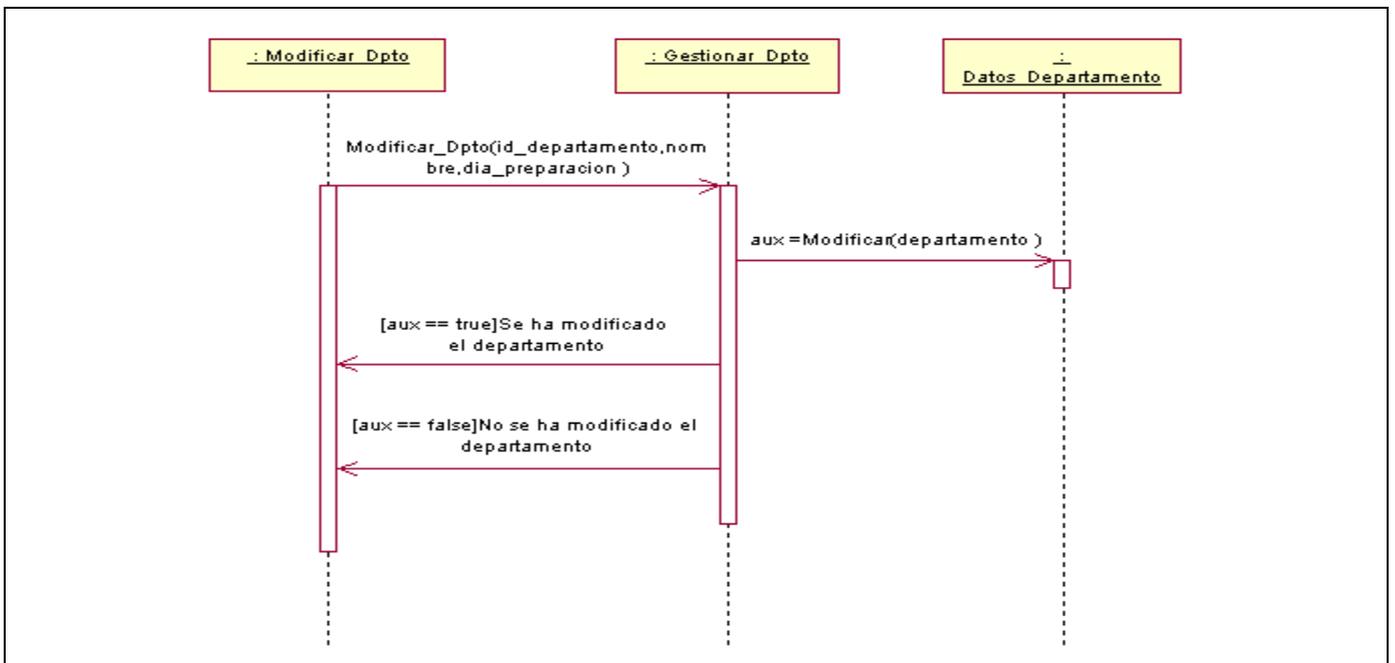


Figura 3.32. Diagrama de secuencia del CU_Gestionar Dpto.: Sección Modificar

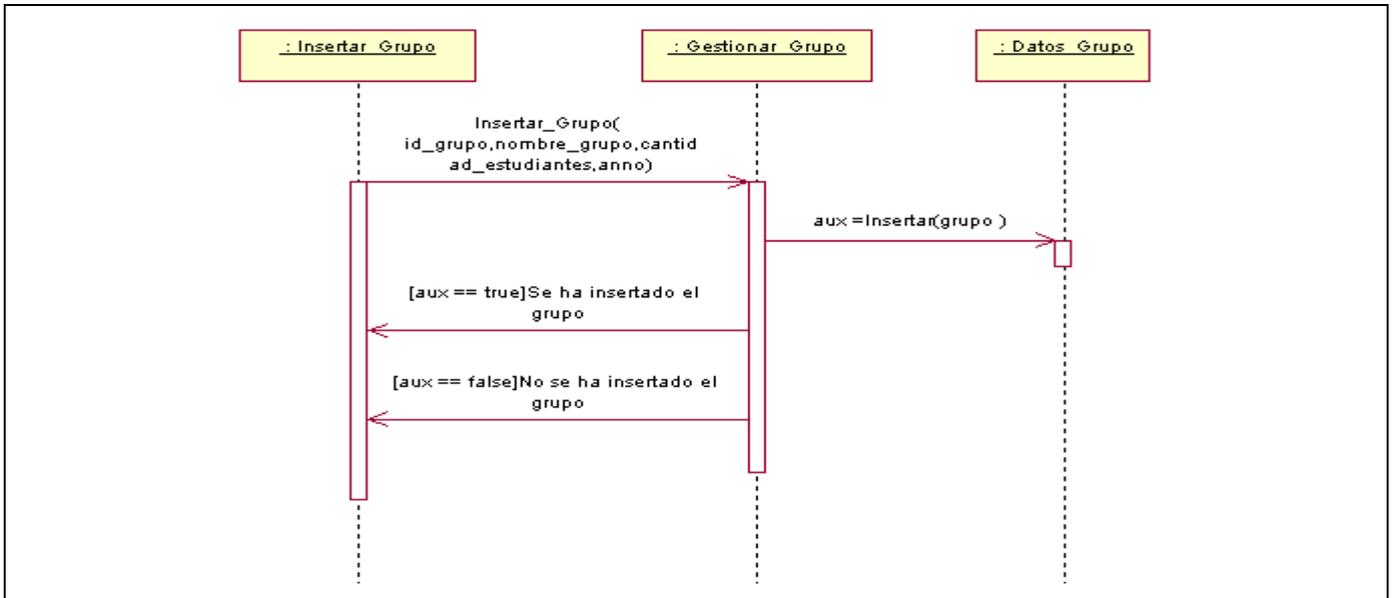
Diagrama de secuencia del CU_Gestionar Grupo: Sección Insertar

Figura 3.33. Diagrama de secuencia del CU_Gestionar Grupo: Sección Insertar

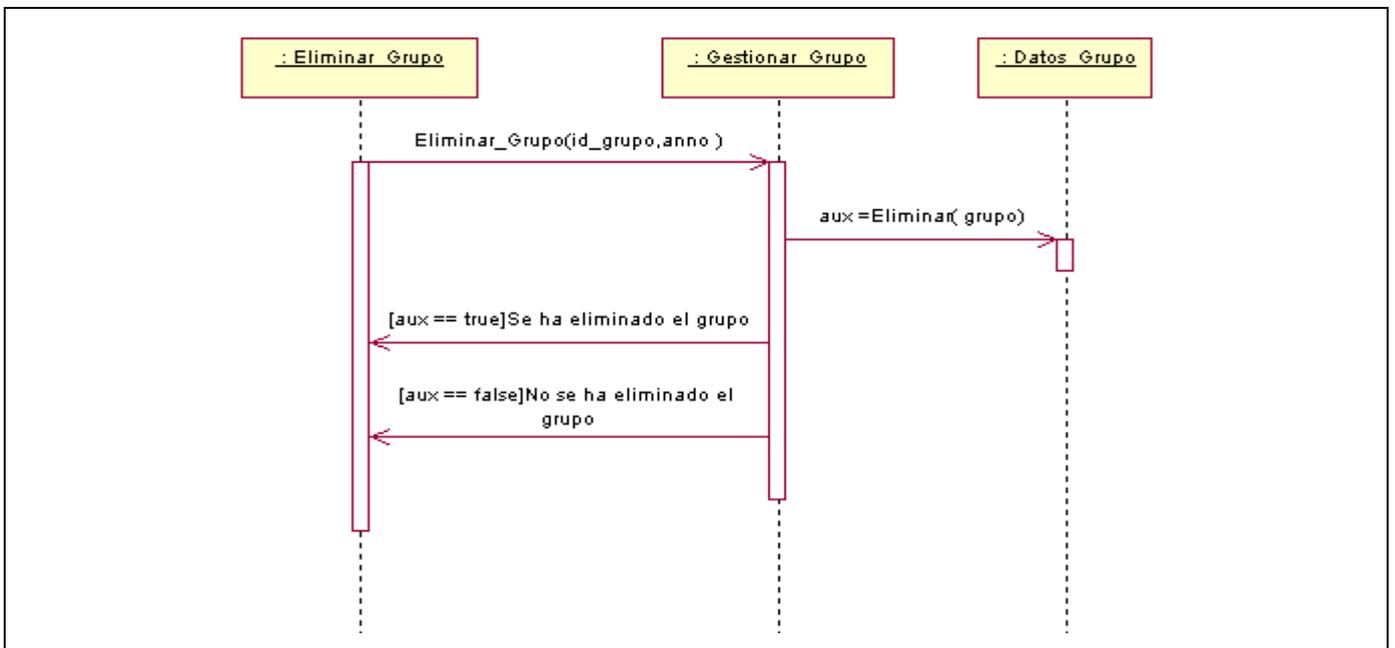
Sección Eliminar

Figura 3.34. Diagrama de secuencia del CU_Gestionar Grupo: Sección Eliminar

Sección Modificar

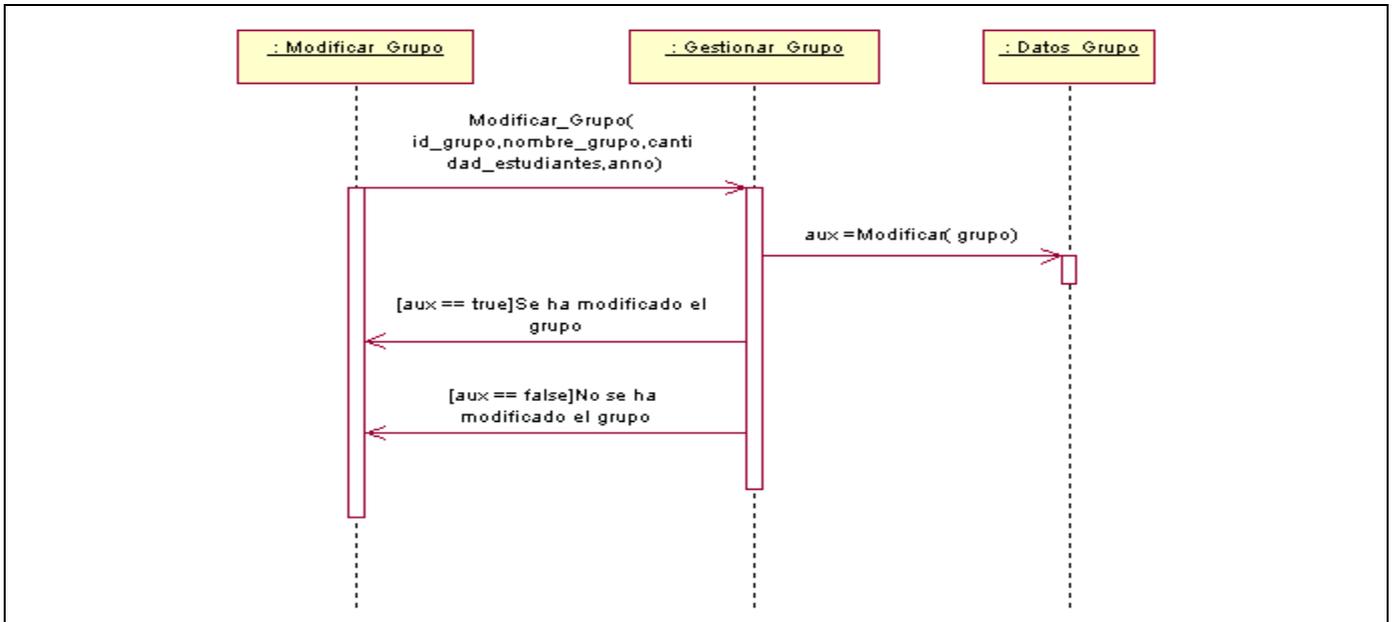


Figura 3.35. Diagrama de secuencia del CU_Gestionar Grupo: Sección Modificar

Diagrama de secuencia del CU_Gestionar Local: Sección Insertar

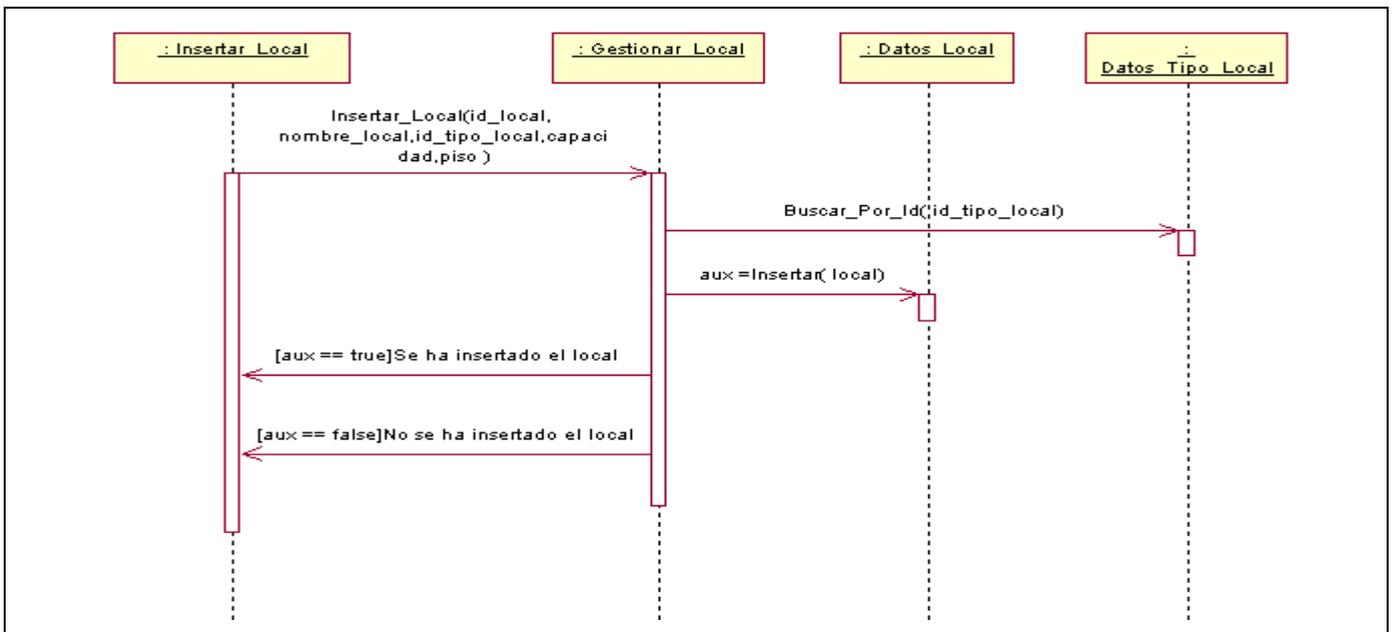


Figura 3.36. Diagrama de secuencia del CU_Gestionar Local: Sección Modificar Insertar

Sección Eliminar

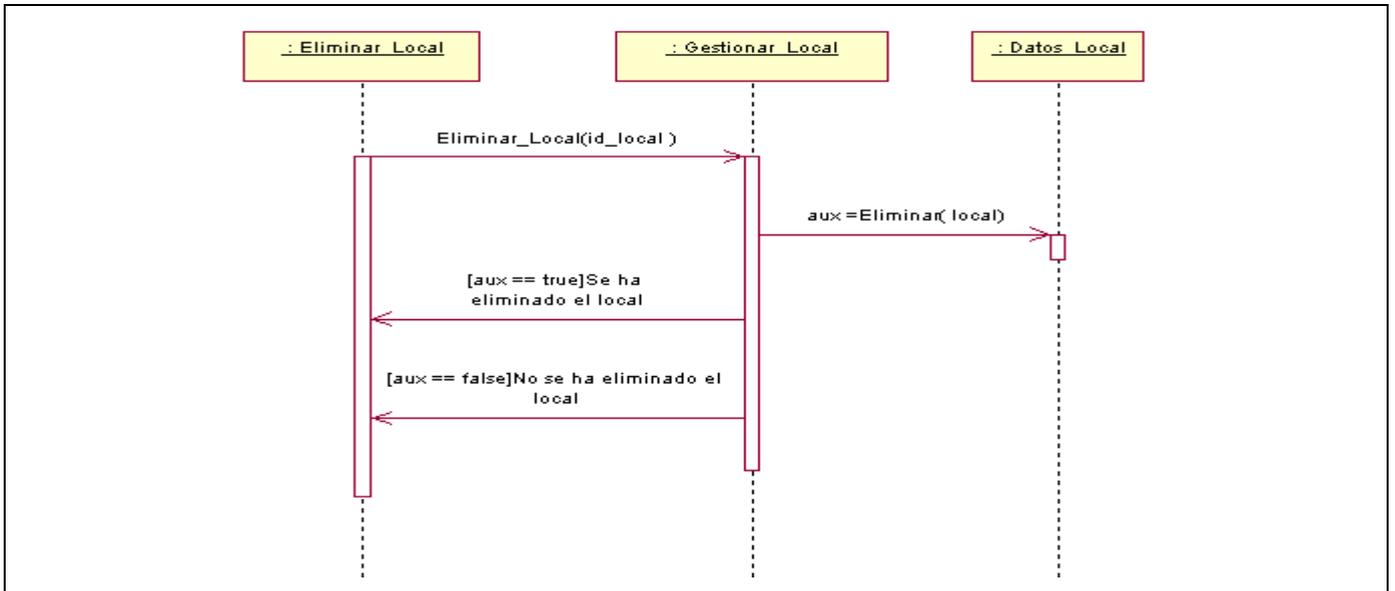


Figura 3.37. Diagrama de secuencia del CU_Gestionar Local: Sección Modificar Eliminar

Sección Modificar

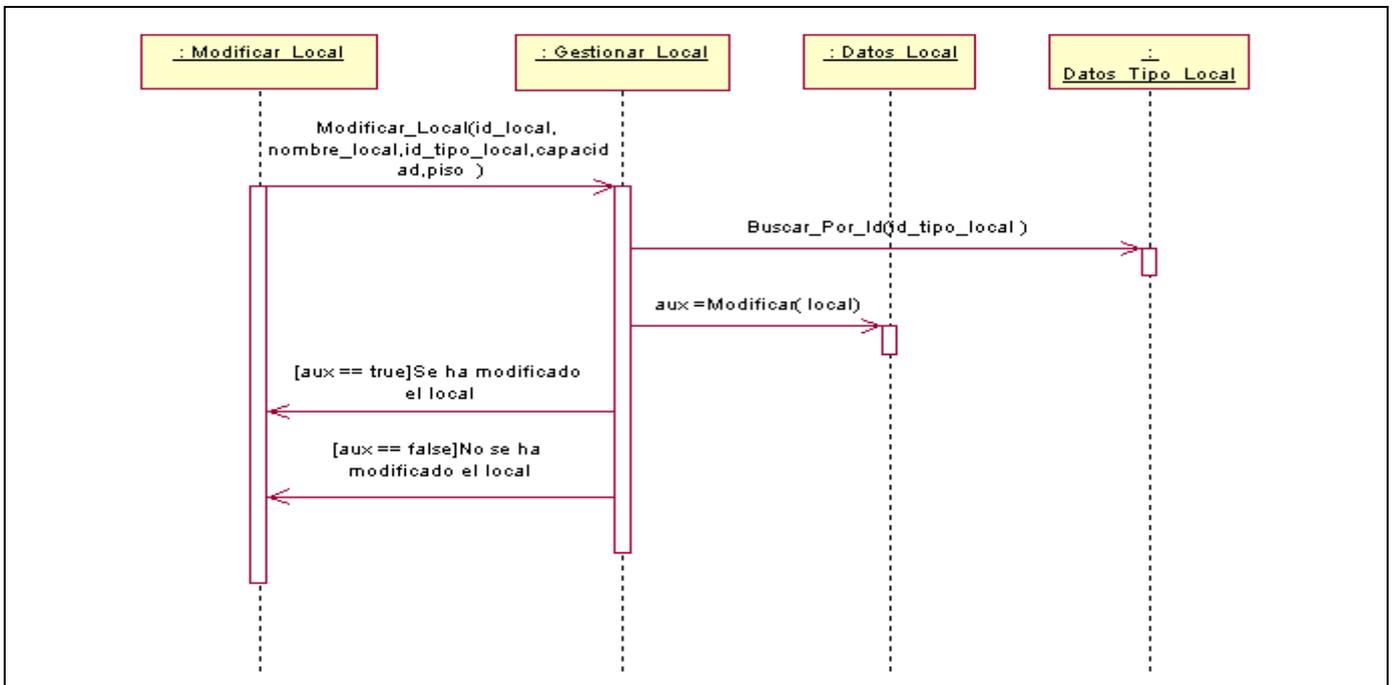
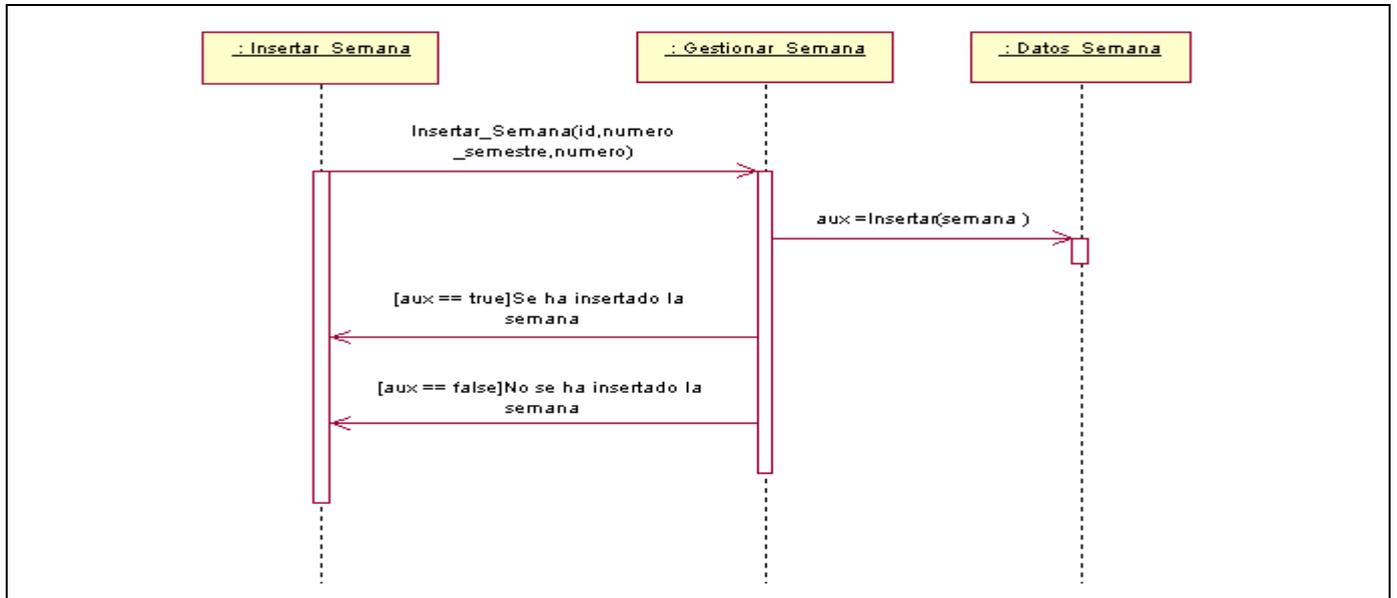
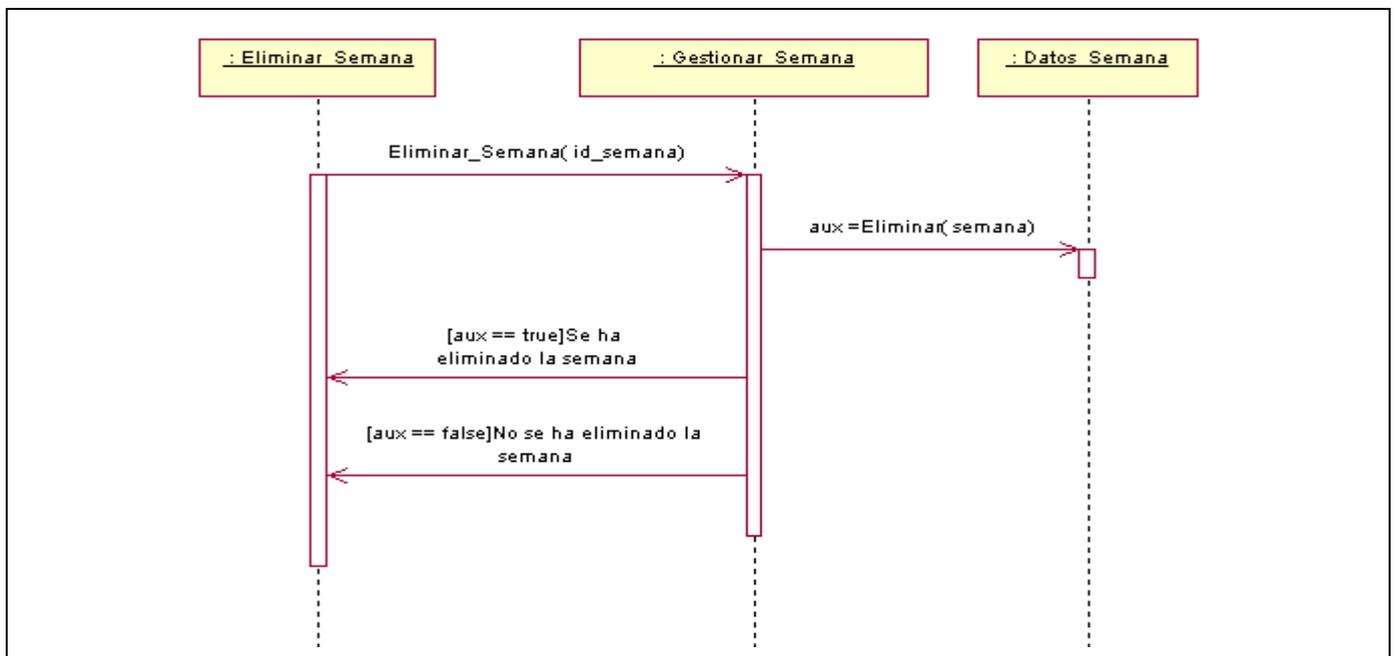


Figura 3.38. Diagrama de secuencia del CU_Gestionar Local: Sección Modificar

Diagrama de secuencia del CU_Gestionar Semana: Sección Insertar**Figura 3.39. Diagrama de secuencia del CU_Gestionar Semana: Sección Insertar****Sección Eliminar****Figura 3.40. Diagrama de secuencia del CU_Gestionar Semana: Sección Eliminar**

Sección Modificar

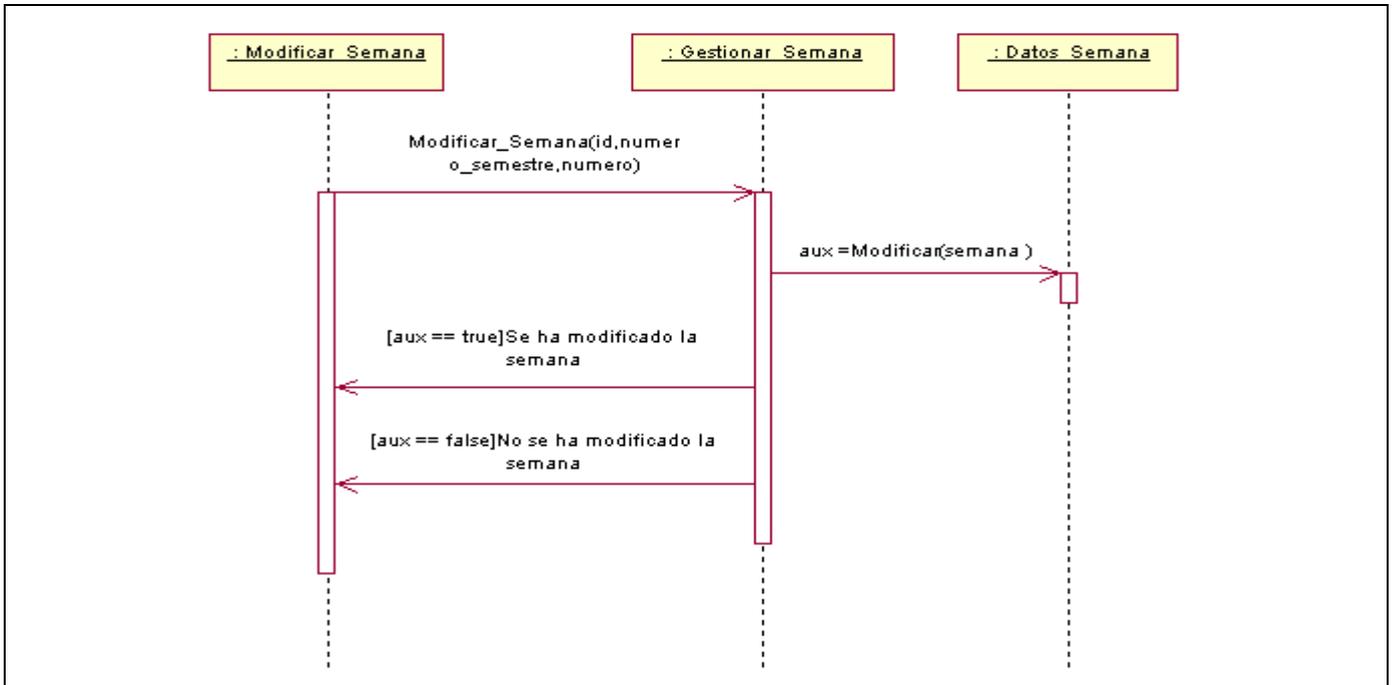


Figura 3.41. Diagrama de secuencia del CU_Gestionar Semana: Sección Modificar

GLOSARIO

XML: (Lenguaje de marcas extensible), es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web. Este permite definir la gramática de lenguajes específicos. Por lo tanto XML no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades. Es una tecnología sencilla que tiene a su alrededor otras que la complementan y la hacen mucho más grande y con unas posibilidades mucho mayores.

Unix: Sistema operativo diseñado para ser usado por mucha gente al mismo tiempo (es multiusuario). Es el sistema operativo más común para servidores en Internet.

J2EE: Es una plataforma que define el estándar para el desarrollo y puesta a punto de aplicaciones empresariales.

CLR: (Common Language Runtime) es el motor de la plataforma. NET, es el encargado de gestionar la ejecución de las aplicaciones, es una aplicación similar a un máquina virtual que se encarga de gestionar la ejecución de las aplicaciones para ella escritas. A estas aplicaciones les ofrece numerosos servicios que facilitan su desarrollo y mantenimiento y favorecen su fiabilidad y seguridad.

API: (Application Programming Interface) es el conjunto de funciones y procedimientos (o métodos si se refiere a programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

.NET Framework: Es un componente integral de Windows que admite la creación y la ejecución de la siguiente generación de aplicaciones y servicios Web XML

ADO.NET: Conjunto de componentes del software que pueden ser usados por los programadores para acceder a datos y a servicios de datos. Es una parte de la biblioteca de clases base que están incluidas en el Microsoft .NET Framework

SDK: (Software Development Kit) es generalmente un conjunto de herramientas de desarrollo que le permite a un programador crear aplicaciones para un sistema bastante concreto.

DLL: (Dynamic Link Library) término con el que se refiere a los archivos con código ejecutable que se cargan bajo demanda del programa por parte del sistema operativo.

IDE: Permitir al programador usar una sola aplicación para diseñar, compilar y depurar aplicaciones.

KDE: (K Desktop Environment) es un entorno de escritorio e infraestructura de desarrollo para sistemas Unix/Linux.

GPL: (General Public License) es la licencia de software libre del proyecto GNU.