



UNIVERSIDAD DE LAS CIENCIAS INFORMATICAS.

Trabajo de diploma para optar por el título de Ingeniería en Informática.

***Asistente de ayuda para la
confección de horarios docentes.***

AUTORES:

Jorge Luis Tamarit Cutiño.

Maykel Santiesteban Díaz.

TUTOR:

Ing: Manuel Garriga Apezteguia.

Ciudad de La Habana, Cuba

Junio, 2007.

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 (firmamos) la presente a los ____ días del mes _____ del año _____.

Firma del Autor
Jorge L Tamarit

Firma del Autor
Maykel Santiesteban

Firma del Tutor.
Ing: Manuel Garriga

OPINIÓN DEL USUARIO DEL TRABAJO DE DIPLOMA.

El Trabajo de Diploma, titulado "Asistente de ayuda para la confección de horarios docentes", fue realizado en la Universidad de las Ciencias Informáticas (UCI) de la provincia de Ciudad Habana. Esta entidad considera que, en correspondencia con los objetivos trazados, el trabajo realizado le satisface:

? Totalmente

? Parcialmente en un ____ %

Los resultados de este Trabajo de Diploma le reportan a esta entidad los beneficios siguientes:

Y para que así conste, se firma la presente a los ____ días del mes de Junio del año 2007.

Representante de la entidad

Cargo

Firma

Cuño

OPINIÓN DEL TUTOR DEL TRABAJO DE DIPLOMA.

Título: Asistente de ayuda para la confección de horarios docentes.

Autores: Jorge L Tamarit Cutiño y Maykel Santiesteban Díaz.

El tutor del presente Trabajo de Diploma considera que durante su ejecución el estudiante mostró las cualidades que a continuación se detallan.

Por todo lo anteriormente expresado considero que el estudiante está apto para ejercer como Ingeniero Informático; y propongo que se le otorgue al Trabajo de Diploma la calificación de: _____.

Nombre completo del tutor

Fecha: _____

Agradecimientos.

Agradecemos a todos aquellos que estuvieron junto a nosotros brindando apoyo y seguridad en nuestros pasos, aquellos que ofrecieron su aporte y preocupación, aquellos que nos acompañaron en el trayecto y que siempre estuvieron física y espiritualmente a nuestro lado así como a la Revolución por la oportunidad que nos ha dado de formarnos en su seno como completos profesionales y por la hermosa obra que realiza.

Gracias a todos.

Dedicatoria.

A quien fuese mi padre durante tantos años y al cual la vida le privo de ver hecho realidad este sueño y de poder compartir en conjunto este momento.

Mi Abuelo.

A quienes han sido en mi vida mis principales guías y supieron darme con amor y cariño, el apoyo, la fuerza, el aliento y la plena confianza que siempre necesité para salir adelante.

Mi Abuela y mi Madre.

A quien considero un verdadero padre y del cual nunca me ha faltado su apoyo y ayuda.

Mi Padre Jorge L Pascual.

A mis Hermanos, Primos, Tías y familia en general y principalmente a aquellos que supieron brindarme una mano segura en la cual confiar.

Mis amigos y mi Novia.

Jorge Luis Tamarit Cutiño.

A mi abuela y mi madre por confiar en mí, y esperar tanto tiempo este momento.

A yuny por confiar en mi, y apoyarme en todo momento.

A mis hermanos Liuska, Hernán por su amor incondicional.

A mis colegas de estudio por brindarme su ayuda desinteresada, compartir los buenos y malos momentos, en fin, hacer estos años inolvidables.

A Ronny, Alberto, Asnioby, Jorgito, Deivis por su ayuda incondicional, y su apoyo en los momentos difíciles.

A la comunidad Open Source por su deseo innato de compartir y estar abierta a todo tipo de sugerencias.

A todos mis profesores que dieron lo mejor de si, en mi formación.

A mis compañeros de tesis por aclarar mis dudas, por la ayuda brindada.

A todos aquellos de una forma u otra hicieron posible la realización de este logro.

A mi abuela y mi madre, sin lugar a duda el fruto de su esfuerzo.

Maykel Santiesteban Díaz.

Resumen.

El presente trabajo lleva como título: Asistente de ayuda para la confección de horarios docentes, fue realizado en la Universidad de las Ciencias Informáticas, centro de estudios universitarios donde la labor docente juega un papel fundamental dentro de sus actividades.

El motivo fundamental de este trabajo es realizar un diseño de posible solución informática, de cierto punto flexible, a las diferentes dificultades presentes en el proceso de elaboración de los horarios docentes en la facultad 2. Estructurado por 5 capítulos donde se aborda sobre la fundamentación teórica del trabajo, el modelo del negocio, los requisitos del sistema, la descripción de la solución propuesta y el estudio de factibilidad.

Se propone como posible solución un prototipo de aplicación de tipo escritorio, haciendo uso para su confección de C# y Prolog como lenguajes de programación y MySQL como gestor de base de datos. Con este trabajo se pretende iniciar en el desarrollo de una solución con el objetivo de mejorar las condiciones de trabajo del personal evitándoles el agotamiento que produce la confección manual de los diferentes horarios docentes.

Índice.

INTRODUCCIÓN.....	1
CAPÍTULO 1 FUNDAMENTACIÓN.....	6
INTRODUCCIÓN.....	6
1.2 OBJETO DE ESTUDIO.....	6
1.2.1 Objetivos estratégicos de la organización.....	6
1.2.2 Flujo actual de los procesos.....	6
1.2.3 Análisis crítico de la ejecución de los procesos.....	7
1.3 PROCESOS OBJETO DE AUTOMATIZACIÓN.....	8
1.4 SISTEMAS AUTOMATIZADOS EXISTENTES VINCULADOS AL CAMPO DE ACCIÓN.....	8
1.5 FUNDAMENTACIÓN DE LOS OBJETIVOS.....	13
1.6 TENDENCIAS Y TECNOLOGÍAS ACTUALES.....	14
1.6.1 Las aplicaciones de escritorio.....	14
1.6.2 Lenguajes de Programación.....	14
1.6.2.1 C#.....	15
1.6.2.2 Prolog.....	16
1.6.2.3 Patrones Diseño.....	18
1.6.2.4 Arquitectura de 3 Capas.....	19
1.6.3 Sistemas de Gestión de Bases de Datos.....	21
1.6.3.1 MySQL.....	22
1.6.4 Proceso de Desarrollo.....	25
1.6.5 Herramientas Utilizadas.....	28
1.6.5.1 Microsoft Visual Studio 2005.....	28
1.6.5.2 Amzi Prolog.....	29
1.6.5.3 EMS MySQL.....	29
1.6.5.4 Rational Rose.....	30
1.7 CONCLUSIONES.....	31
CAPÍTULO 2 MODELO DEL NEGOCIO.....	32
2.1 INTRODUCCIÓN.....	32
2.2 MODELO DEL NEGOCIO ACTUAL.....	32
2.2.1 Proceso de Confeccionar Horario.....	32
2.3 REGLAS DEL NEGOCIO A CONSIDERAR.....	33

2.4 ACTORES DEL NEGOCIO.	34
2.5 DIAGRAMA DE CASOS DE USO DEL NEGOCIO.	34
2.6 TRABAJADORES DEL NEGOCIO.	35
2.7 DESCRIPCIÓN DE LOS CASOS DE USO DEL NEGOCIO.	35
2.7.1 Caso de uso “Gestionar Horario”.....	35
2.7.1.1 Diagramas de Actividades.....	37
2.7.1.2 Diagrama de Actividades referente al CU Gestionar Horario.....	38
2.8 MODELO DE OBJETOS.	39
2.9 CONCLUSIONES.	39
CAPÍTULO 3 REQUISITOS.....	40
3.1 INTRODUCCIÓN.	40
3.2 DEFINICIÓN DE LOS REQUISITOS FUNCIONALES.	40
3.3 DEFINICIÓN DE LOS REQUISITOS NO FUNCIONALES.	42
3.4 ACTORES DEL SISTEMA A AUTOMATIZAR.	44
3.5 PAQUETES Y SUS RELACIONES.....	44
3.6 PAQUETE GESTIONAR CONFIGURACIÓN.....	45
3.6.1 Diagrama de casos de uso del sistema (Paquete Gestionar Configuración).	45
3.6.2 Descripción de los casos de uso del sistema.	46
3.6.2.1 “Caso de uso Gestionar Profesores”.	46
3.6.2.2 “Caso de uso Gestionar Grupos”.	50
3.6.2.3 “Caso de uso Gestionar Asignatura”.....	53
3.6.2.4 “Caso de uso Gestionar Locales”.....	59
3.6.2.5 “Caso de uso Gestionar Especificación Docente”.....	63
3.6.2.6 Caso de Uso Gestionar Distribución Profesores, Grupo, Asignatura.	70
3.7 PAQUETE AUTENTICARSE.....	72
3.7.1 Diagrama de casos de uso del sistema (Paquete Autenticarse).....	72
3.7.2 Descripción de los casos de uso del sistema.	73
3.7.2.1 “Caso de uso Autenticarse”.....	73
3.8 PAQUETE DISTRIBUCIÓN.	75
3.8.1 Diagrama del Caso de uso del sistema (Paquete Distribución).....	75
3.8.2 Descripción de los casos de uso del sistema.	75
3.8.2.1 “Caso de uso Realizar Planificación”.	75
3.9 PAQUETE REPORTE.....	77
3.9.1 Diagrama del Caso de uso del sistema (Paquete Reporte).....	77

3.9.2 Descripción de los casos de uso del sistema.	78
3.9.2.1 “Caso de uso Obtener Locales Disponibles”.	78
3.9.2.2 “Caso de uso Obtener Ubicación de Profesor”.	80
3.9.2.3 “Caso de uso Obtener Estado de Actividad Docente”.	81
3.9.2.4 “Caso de uso Obtener Ubicación de Grupo”.	83
3.9.2.5 “Caso de uso Obtener Planificación Grupo”.	84
3.10 CONCLUSIONES.	85
CAPÍTULO 4 DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA.	86
4.1 INTRODUCCIÓN.	86
4.2 PRINCIPIOS DEL ANÁLISIS.	86
4.2.1 Diagrama de clases del Análisis.	86
4.3 PRINCIPIOS DE DISEÑO.	88
4.3.1 Diagramas de Clases del Diseño.	88
4.3.3 Diagrama de Interacción (Colaboración)	89
4.4 DISEÑO DE LA BASE DE DATOS.	91
4.4.1 Modelo lógico de datos (diagrama de clases persistentes).	91
4.4.2 Modelo físico de datos (modelo de datos).	92
5.5 DIAGRAMA DE DESPLIEGUE.	93
5.6 DIAGRAMA DE COMPONENTES.	93
4.7 CONCLUSIONES	95
CAPÍTULO 5 ESTUDIO DE FACTIBILIDAD.	96
5.1 INTRODUCCIÓN	96
5.2 PLANIFICACIÓN BASADA EN CASO DE USO.	96
5.3 BENEFICIOS TANGIBLES E INTANGIBLES.	101
5.4 ANÁLISIS DE COSTOS Y BENEFICIOS.	102
5.5 CONCLUSIONES.	103
CONCLUSIONES.	104
RECOMENDACIONES.	105
REFERENCIAS BIBLIOGRÁFICAS.	106
BIBLIOGRAFÍA.	108
GLOSARIO DE TÉRMINOS Y SIGLAS.	109

Introducción.

La computación como ciencia desde su surgimiento, se ha enfocado en darle un vital apoyo al hombre en la solución de problemas que han sido en muchos casos imposibles de resolver o que llevan implícito un gran consumo de tiempo el obtener su solución, es precisamente este aspecto el que nos ha impulsado al reto de darle solución a un problema existente hoy en nuestra universidad como lo es la confección de horarios docentes. Dándole a este, un enfoque ó una posible vía de solución desde un punto de vista computacional. Tomando como punto de partida y brindando una leve panorámica sobre el tema, podemos decir que los problemas de horarios consisten en fijar una secuencia de encuentros entre profesores y estudiantes en un período de tiempo predefinido, satisfaciendo un conjunto de restricciones de varios tipos. Resolver manualmente un problema de este tipo, requiere de varias horas, incluso de días de trabajo, dependiendo de los recursos involucrados, sin embargo es muy probable que la solución obtenida, no sea la más adecuada. Por las razones antes mencionadas, los problemas de horarios (PH) merecen una considerable atención, de modo que se les ha dedicado mucho tiempo a buscar una vía de solución para lograr automatizarlos. Durante los últimos años, se ha incrementado la investigación y con ello la bibliografía con respecto al tema. Además muchas aplicaciones de software han sido desarrolladas en este sentido obteniendo éxitos considerables. Podemos ver que las primeras técnicas que se utilizaron para resolver el problema fueron las basadas en: simular la forma tradicional mediante la cual, un ser humano resolvía el problema. Tales técnicas fueron bautizadas con el nombre de heurística directa, las cuales están basadas en argumentación sucesiva. De forma general todos estos enfoques se basan en ubicar la lección más restrictiva primero. Posteriormente comenzaron las investigaciones con el objetivo de aplicar técnicas generales a este tipo de problema. De esta forma comenzaron a utilizarse algoritmos basados en programación entera, flujo de redes, entre otras. Recientemente se registra en la literatura enfoques basados en técnicas de búsquedas las cuales se utilizan en la rama de la inteligencia artificial, entre las cuales podemos citar: simulación de recocido, búsqueda tabú, algoritmos genéticos, satisfacción de restricciones, programación lógica con restricciones, etc. Muchos autores defienden la idea de

que los problemas de horario no pueden ser completamente automatizados debido principalmente a dos razones: es decir podemos encontrar evidencia de que es posible generar horarios mejores que otros, pero no resulta trivial expresarlo en sistemas automáticos y como segundo la aparición determinante de la intervención humana para simplificar los espacios de búsqueda, los cuales son normalmente grandes, guiando el sistema hacia un estado de interés. Debido a lo antes mencionado muchos sistemas permiten al usuario final, ajustar manualmente la salida obtenida. Algunos sistemas sin embargo requieren mayor grado de intervención humana denominándose sistemas interactivos o sistemas de horarios semi-automáticos.

Podemos ver que encontrar una solución óptima computacional para los problemas de horarios se ha convertido en un reto trazado por generaciones y que el tema de asignación de recursos se convierte en un tema importante a tener en cuenta en la totalidad de las instituciones para lograr la organización y correcto funcionamiento de las mismas. Es precisamente la necesidad de organización unida a los diferentes escenarios existentes en el centro los que afectan el proceso de confección manual del horario, por lo que se hace necesario la utilización de alguna herramienta computacional que ayude gradualmente en la formulación de los mismos.

Como primer escenario podemos ver que nuestro centro se encuentra actualmente inmerso en una intensa labor constructiva y a la vez vinculado a las actividades docentes, aspecto que proporciona un grado de complejidad en el proceso de elaboración de los horarios docentes, pues en muchos casos los locales destinados a las labores estudiantiles son insuficientes en comparación con la enorme masa estudiantil. Como segundo escenario podemos ver que la universidad desde su creación ha sido una obra muy atípica desde el punto de vista estructural y de funcionamiento, ya que constantemente se encuentra envuelta en un total dinamismo debido a que es un proyecto creado bajo el calor de la batalla de ideas y en el cual se han depositado enormes expectativas. Justamente la respuesta precisa y positiva de apoyo que brinda nuestro centro a todas las actividades planificadas tanto políticas, culturales y de recreación, es lo que hace que el proceso de confección de los horarios sea propenso a innumerables cambios de

última hora, aspecto que proporciona cierta complejidad a la realización de dicha tarea.

Como tercera causa podemos ver al centro como una institución contenedora de una enorme masa estudiantil, la cual requiere de una gran cantidad de profesores y personal calificado para cubrir todas las materias pertinentes, influyendo en que se tenga que acudir a profesores adjuntos y externos al centro para que presten su servicio en el mismo, provocando que se deba de tener en cuenta dicho aspecto en el momento de elaboración de los horarios. Aspecto que aporta cierto grado de complejidad a la tarea.

El presente trabajo surge con la necesidad de realizar un software que garantice y proporcione cierta ayuda al proceso de creación de los horarios docentes, durante el desarrollo de todo el curso en nuestra facultad, entonces el **problema** lo pudiéramos enfocar de la siguiente manera: ¿Como contribuir con el proceso de confección de horarios docentes?

La necesidad de realizar un software que garantice y proporcione una ayuda en el proceso de creación de los horarios docentes se deriva de que en nuestra facultad este proceso a pesar de ser una actividad que se realiza de forma manual se torna un tanto engorrosa, todo esto en conjunto a un gran consumo de tiempo lo convierte en un proceso lento y tedioso.

Por tanto el **objeto de estudio** de este trabajo son: los diferentes Procesos que se realizan en el local de planificación docente de la facultad, referentes a la correcta elaboración de los horarios.

De ello deriva que el **campo de acción** comprenda los procesos de entrada, procesamiento y salida relacionados con la planificación de horarios docentes en la facultad 2 de la Universidad de las Ciencias Informáticas.

Como posible **Aportes prácticos** podemos citar los siguientes:

1. Disminución del tiempo de elaboración de los horarios docentes.
2. Facilidad de búsqueda de información por parte del personal de uso.
disminuyendo su carga de trabajo.
3. Obtención de diferentes reportes para usos posteriores.

Se ha propuesto como **objetivo general** realizar el análisis y diseño del sistema así como la creación de un prototipo no funcional del mismo.

Para darle cumplimiento a nuestro objetivo general nos planteamos los siguientes

Objetivos específicos:

1. Desarrollar el análisis y el diseño del sistema.
2. Realizar un estudio sobre la interoperabilidad de lenguajes convencionales (C#) con lenguajes descriptivos (Prolog).
3. Implementar un módulo que permita realizar una correcta entrada de datos al sistema.
4. Realizar el diseño de una base de datos relacional que brinde soporte de almacenamiento de información.
5. Implementar un módulo que incluya técnicas de inteligencia artificial, capaz de generar posibles soluciones.
6. Documentar el sistema.

Para la realización de los objetivos anteriores nos hemos trazado las siguientes

Tareas de la investigación:

- ? Analizar y estudiar en detalles el proceso manual de creación de un horario.
- ? Entrevistar a los clientes para identificar los requerimientos necesarios del sistema.
- ? Investigar sobre la existencia y funcionamiento de posibles sistemas de este tipo ya creados tanto en el campo nacional como internacional.
- ? Desarrollar una capa de acceso que gestione la interoperabilidad con el módulo de inteligencia artificial.

Con la presente propuesta se pretende modelar un asistente informático que permita la confección de horarios docentes, con vista a una posterior implementación del mismo.

El presente trabajo esta estructurado en 5 capítulos que resumen la siguiente información:

Capítulo 1. Fundamentación Teórica:

Descripción del objeto de estudio, los procesos que serán objeto de automatización, sistemas automatizados existentes vinculados al campo de acción, tendencias y tecnologías actuales seleccionadas a emplear en el desarrollo de la propuesta y por qué su utilización.

Capítulo 2. Modelo del Negocio:

Descripción detallada de las reglas de negocio, actores, trabajadores y casos de uso así como la representación de los diagramas de casos de uso, actividad y modelo de objetos.

Capítulo 3. Requisitos:

Definición de actores del sistema a automatizar, paquetes y relación entre ellos, diagrama de casos de uso del sistema, definición de los requisitos no funcionales y descripción de los casos de uso del sistema.

Capítulo 4. Descripción de la solución propuesta:

Diagrama de clases del diseño, principios de diseño, diseño de la base de datos, diagramas de colaboración, despliegue y componentes.

Capítulo 5. Estudio de factibilidad:

Estudio de factibilidad económica realizada para este proyecto, en el que se determina si es factible o no el desarrollo del software propuesto, analizando los diferentes criterios que influyen en el cálculo del esfuerzo, tiempo de desarrollo y costo del proyecto.

Capítulo 1 Fundamentación Teórica.

Introducción.

En este capítulo se muestra los principales problemas que fundamentan la propuesta de solución, así como los objetivos generales y específicos que se trazaron como estrategia. Además de brindar una breve panorámica sobre algunos de los sistemas automatizados existentes vinculados a nuestro campo de acción y haciendo un análisis comparativo entre la solución propuesta en este trabajo con respecto a las soluciones ya existentes. Se describen también las tecnologías actuales de desarrollo utilizadas para el análisis, diseño e implementación, abordando sobre el por qué, la importancia y las ventajas de uso que estas brindan en el desarrollo del sistema.

1.2 Objeto de Estudio.

El objeto de estudio de este trabajo son: los diferentes Procesos que se realizan en el local de planificación docente de la facultad, referentes a la correcta elaboración de los horarios.

1.2.1 Objetivos estratégicos de la organización.

Dentro de los principales objetivos estratégicos que cuenta nuestra facultad con la realización de esta vital tarea podemos ver los siguientes: garantizar una distribución organizada y correcta de asignaturas y profesores a los diferentes grupos de estudiantes, para lograr así un eficaz funcionamiento en las distintas actividades docentes que se planifiquen a lo largo del curso y tener un fácil control y previo conocimiento sobre el tiempo de duración de cada actividad docente de los grupos (inicio y fin de cada asignatura).

1.2.2 Flujo actual de los procesos.

Realizar un análisis detallado del flujo de procesos permite conocer como funciona realmente el negocio en específico para así producir un resultado que de solución a la necesidad que se plantea. El resultado puede ser un producto, un servicio, una información o combinaciones de ellos. Hacer un análisis del flujo de los procesos

permite revelar problemas tales como: la circulación doble de la información, la duplicación del trabajo, etc.

Una vez que se da comienzo a la elaboración del horario docente como actividad se activan una secuencia de procesos propios de la misma, mencionando como primero el proceso de Obtención de documentos, cuyo objetivo fundamental se centra en hacer captura o adquisición de todos los documentos de manera previa, con la información necesaria y actualizada, para dar comienzo a la elaboración del horario docente, dentro de estos podemos mencionar documentos como: Balance de Carga, Planificación de los días en que los profesores tienen preparación metodológica, así como documentos propios de los profesores con sus posibles afectaciones.

Concluida esta fase se da continuación al proceso de confección del horario como tal, donde el objetivo primordial se centra en la creación de un horario docente con todas las distribuciones de profesores, aulas y grupos hechas.

Seguido esto podemos ver un último proceso o sea el proceso de verificación o revisión del horario confeccionado, en este proceso la actividad fundamental se centra en realizar una revisión detallada del horario creado en busca de posibles errores de confección.

A continuación se muestra una figura con el flujo de procesos propios de la actividad confeccionar horario.

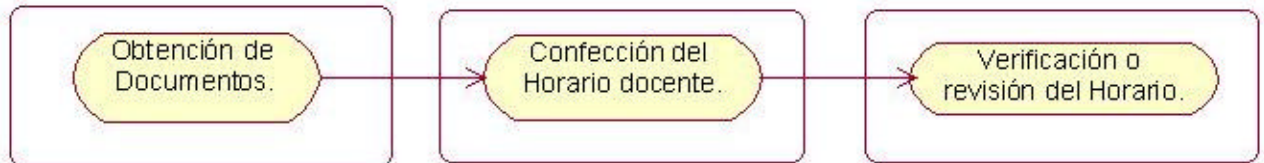


Figura 1.1 Flujo de los Procesos.

1.2.3 Análisis crítico de la ejecución de los procesos.

Teniendo en cuenta que el flujo actual de procesos referentes a la confección del horario ha posibilitado satisfacer las necesidades de la facultad durante todo el período docente de los cursos anteriores cabe mencionar algunos aspectos deficientes que se han detectado y que han generado la situación problemática que se pretende solucionar con la confección del presente trabajo.

Como primero cabe mencionar que en el proceso Obtención de documentos ocurre periódicamente que los documentos que son proporcionados a nivel central como es el Balance de Carga, presenten de manera predeterminada los turnos de clase de la asignatura educación física, aspecto que en ocasiones provoca que la realización del horario se torne un poco más difícil.

En cuanto al proceso propio de realización del horario, podemos ver que esta actividad a pesar de convertirse en una actividad tediosa y estresante para el personal que la realiza, puede quedar con facilidad expuesta a errores en su planificación, desde el punto de entrada de datos debido a la gran cantidad de restricciones y aspectos que se debe tener en cuenta. Una vez culminada esta tarea y entrando en la siguiente actividad referente a la revisión, podemos mencionar que esta tarea se torna también tediosa y agotadora llegando al punto de poder proporcionar nuevos errores a la planificación.

1.3 Procesos Objeto de Automatización.

Con el presente trabajo se pretende modelar el siguiente proceso:

- ? Proceso de confección del horario docente.

Debido a que en este proceso es en el cual se desarrolla de manera concreta el horario docente y es la tarea más expuesta a errores por parte de la planificadora, lo hemos seleccionado como principal actividad a modelar. El cual da comienzo una vez que la planificadora tiene en su poder todos los documentos necesarios que son proporcionados a nivel central.

1.4 Sistemas automatizados existentes vinculados al campo de acción.

Actualmente existen varios sistemas vinculados a nuestro campo de acción, donde cada uno de ellos le da solución y tratamiento al problema siguiendo una estructura propia y diversa desde diferentes puntos, entre los más conocidos podemos citar:

GP-Untis.

Este software generador de horarios es uno de los más usados internacionalmente, el “motor de GP-Untis” con el cual se genera la optimización de horarios esta basado en algoritmos heurísticos genéticos y lineales, lo cual permite que el manejo

de los datos sea regido por un número limitado de restricciones que el usuario determina. El desarrollo sucesivo y continuo del algoritmo ha logrado que por más de 30 años este software garantice la obtención de horarios óptimos, siendo insuperable en la calidad y rapidez de los horarios generados. Aplicable a todo tipo de centros educativos, orientación pedagógica según las premisas que el centro determine, cálculos rapidísimos y resultados óptimos y convincentes, introduce todo tipo de clases y condiciones, manejo intuitivo y estructura clara, combinados con los últimos avances de Windows. Presentando versión "multiusuario" permitiendo que varios usuarios trabajen simultáneamente con el programa, además de ser un software propietario. [1]

ASC Horarios.

Es otro de los generadores de horarios, aSc Horarios posibilita cubrir todos los tipos de áreas especiales. Es posible juntar varios grupos de diferentes clases en una sola, como también agregar a una clase varios profesores. Para asignaturas, clases, aulas y profesores individuales puedes fijar el tiempo para la enseñanza o cuando el aula ó el profesor estén libres. [2] Presenta como características propias el ser un software compatible con el sistema operativo Windows en sus versiones 98/NT/2000/XP/2003/NT/Vista. Admite automáticamente procesadores DualCore. Ordenadores DualCore lo cual posibilita una rápida generación del horario. Sin dejar de mencionar que es un software propietario.

El GHC (Generador de Horarios para Centros Docentes).

Es un planificador, generador y editor de horarios semanales para centros de enseñanza. Esta aplicación se ha desarrollado con el objetivo de racionalizar y automatizar la confección de los horarios escolares. Genera eficazmente un compromiso entre todos los intereses involucrados, equitativo para todos los profesores. Posibilita el intercambio de datos con otras aplicaciones. Mejora proceso de validación de datos. Compatible para sistema operativo: Windows en sus versiones 98/Me/NT/2000/XP. Presenta como desventaja que es una aplicación limitada a 10 profesores además de ser un software propietario. [3]

Timetab.

Es un potente programa de generación de horarios escolares sencillos de manejar y con un asistente para la introducción de datos que permite generar rápidamente un horario para un centro educativo.

- ? Se contempla la utilización de aulas especiales por determinadas asignaturas y se suministra como resultado un horario para cada aula.
- ? Se puede dar preferencia a las asignaturas para que se impartan en las primeras o últimas horas de día. Presenta como limitación que solo es aplicable a un máximo de 4 grupos y 6 profesores. Compatible para sistema operativo: Windows en sus versiones 95/98/Me/NT/2000/XP además de ser un software propietario. [4]

Mimosa.

Software general de planificación de la enseñanza y de programación de horarios. La aplicación usa todas las propiedades de Windows eficientemente, es rápida, tiene una gran capacidad y lo más importante: es tan flexible que puede aplicarse en todo el tipo de escuelas. Entre algunas características podemos citar:

- ? Trazabilidad – Puede rehacer o deshacer hasta las últimas 1000 acciones.
- ? Trabaja en todas las versiones de WINDOWS (3.11, 9x, NT, 200x y XP), necesita 4 Mb de memoria RAM.
- ? Todos los informes pueden convertirse a archivos HTML para Internet.
- ? Configurable – todos los textos imprimibles, recursos y otros parámetros son definidos por el usuario.
- ? Acceso multiusuario – hasta 255 usuarios simultáneos, compatible con cualquier red.

Dentro de las principales desventajas de este software podemos ver que es dirigido principalmente a sistemas operativos de tipo Windows y que es un software propietario. [5]

Podemos ver que estos softwares son herramientas que han combinado diferentes algoritmos tanto matemáticos como informáticos para lograr darle solución a un problema de asignación de recursos tan complejo como lo es el horario.

Mostrando un enfoque general de la connotación y complejidad que contiene dicho problema en el mundo, podemos ver que el Problema de Asignación de Horarios de Clases en una institución universitaria es NP Completo, esto ha motivado la utilización de métodos heurísticos y metaheurísticas para tratar de encontrar “buenas” soluciones en un tiempo computacional razonable. [6]

Los problemas de confección de horarios y asignación de recursos (aulas, profesores) en instituciones educativas han sido estudiados en Investigación operativa desde los años 60. Son problemas en general muy complejos, y cada uno de ellos tiene características propias que hacen que las soluciones de un caso no puedan fácilmente ser adaptadas a otro. Se modelan en general como problemas de programación entera, con un enorme número de variables, y la mayoría de los que aparecen en la práctica pertenecen a la clase de problemas NP-hard [7]

Entre las técnicas utilizadas hasta ahora para resolver este problema se tienen los Algoritmos Genéticos, algoritmos basados en la metaheurística Tabú Search, algoritmo Vértice-Coloración, algoritmo de colonia de hormigas, etc.

Un gran número de variantes del problema de horario ha sido propuesto en la literatura, pero difieren uno de otros en dependencia de la institución donde será desarrollada la aplicación y el tipo de restricción involucradas. Atendiendo a esto se clasifican en tres clases principales.

- ? Horario de exámenes (ETTP): Donde por sus características requiere más profesores y locales, en una misma hora.
- ? Horario de conferencias (LTTP): Para la planificación de conferencias, principalmente en eventos, evitando el solapamiento de las actividades a realizar. Y posibilitar que el participante no este obligado a participar en todas.
- ? Horario en un curso corriente (CTTP): Se refiere a las planificaciones de las clases en un curso universitario corriente.

En algunos casos los problemas de horario consisten en encontrar cualquier solución que satisfaga todas las restricciones, de esta forma el problema es concebido como un problema de búsqueda. En otros casos el problema es concebido como un problema de optimización, es decir, satisfacer todas las restricciones maximizando o minimizando una función objetivo dada, la cual se incluye (está embebida) como restricciones que manipula la aplicación (software), o sea, significa aplicar técnicas de optimización a problemas de búsqueda. En estos casos lo que se esta siendo minimizado es lo que se conoce como lo que falta para lograr la viabilidad del problema (es decir cuanto lo que resta desde una solución

obtenida a la solución real). Incluso cuando se trata de un problema de optimización real, lo que resta para lograr la viabilidad del problema puede ser incluido en la función objetivo. Esto generalmente se realiza para facilitar la navegación (trabajo) en el espacio de búsqueda. En ambos casos (búsqueda u optimización), se define fundamentalmente como un problema de toma de decisión, es decir, en el caso del problema de búsqueda se intenta identificar si existe una solución y en el problema de optimización se intenta identificar si existe una solución, cuyo valor esta contemplado en la función objetivo. Cuando se emplea el término complejidad, este esta estrechamente relacionado al proceso de toma de decisión (como se describe anteriormente). Los problemas de horario son por lo general tratados como NP-completos casi en todas sus variantes, por consiguiente la solución exacta solo es alcanzada solo en casos pequeños (Ej. menos de 10 asignaturas), considerando que en los casos reales están involucrados algunos cientos de asignatura. Eso trae consigo que los métodos heurísticos sean los más factibles para tratarlos, lo cual no garantiza que se alcance la solución óptima. A modo general podemos ver que existen una gran diversidad de metodologías, algoritmos, etc. utilizadas así como softwares contenedores de estos con el fin de obtener soluciones óptimas para el problema de generar horarios y hacer planificaciones, pero el motivo que nos inicia en las primeras faces de elaboración de nuestra propia herramienta generadora de horarios docentes es el siguiente:

Todas estas aplicaciones antes mencionadas son en su totalidad, software propietarios o sea son herramientas donde la adquisición de sus licencias en por ciento monetario se torna muy elevado así como el soporte y mantenimiento de las mismas sería un proceso algo costoso para nuestro centro, en muchos casos son herramientas limitadas a cierta cantidad de profesores y grupos. Son aplicaciones que no se ajustan del todo al proceso y ambiente para el cual se les quiere dar uso en nuestra facultad. Todos estos aspectos nos impulsan al análisis y diseño y a la realización de un prototipo de nuestro propio producto obviando todas esas variantes de software existentes. Teniendo presente una futura elaboración de un software sin restricciones en el proceso de confección del horario, software de fácil adquisición y uso por todos. El sistema propuesto estará enfocado a resolver el problema estructural en el tema de confección de los horarios, el mismo hará uso

de un gestor de bases de datos específico y centrado al propósito que se desea llegar, con el objetivo de disminuir los tiempos de respuesta cuando son altos los volúmenes de información. La nueva propuesta es un sistema único en el cual se van a integrar todas las actividades particulares y propias del proceso de elaboración horarios en el local de planificación de nuestra facultad.

1.5 Fundamentación de los Objetivos.

Para la realización de nuestro trabajo, nos hemos trazados algunos objetivos que giran entorno a nuestro objetivo general el cual es: realizar el análisis y el diseño del sistema así como un prototipo no funcional del mismo que contribuya en la confección de los horarios docentes de la facultad. Estos objetivos son los siguientes:

- ? Desarrollar el análisis y el diseño del sistema

Por medio de este objetivo se pretende obtener una comprensión completa del problema en cuestión. Enfocándolo desde una perspectiva orientada a objetos, utilizando RUP como metodología de desarrollo.

- ? Realizar un estudio sobre la interoperabilidad de lenguajes convencionales (C#) con lenguajes descriptivos (Prolog).

Por medio de este objetivo se logrará vincular los lenguajes que se describen anteriormente. Esto permitirá utilizar las potencialidades de un lenguaje propio de la inteligencia artificial (IA) desde una aplicación de escritorio convencional.

- ? Implementar un módulo que permita realizar una correcta entrada de datos al sistema.

Este objetivo permitirá realizar una entrada de datos con las correspondientes modificaciones de los mismos desde un interfaz sencilla y amigable.

- ? Realizar el diseño de una base de datos relacional que brinde soporte de almacenamiento de información.

Este objetivo permitirá desarrollar una base de datos MySQL capaz de almacenar toda la información necesaria para el correcto funcionamiento del sistema.

- ? Implementar un módulo que incluya técnicas de inteligencia artificial, capaz de generar posibles soluciones.

Por medio de este objetivo podremos cumplimentar todas las posibles restricciones que posibilitarán una correcta distribución y asignación de turnos de clases.

? Documentar el sistema.

Documento que contendrá todo lo relacionado con la documentación técnica propia del sistema, plasmando todas las fases y flujos por las que transitó el desarrollo de la aplicación.

1.6 Tendencias y Tecnologías Actuales.

Teniendo en cuenta las características propias del entorno donde se prevé poner en práctica la posible propuesta de solución, se realizó un estudio sobre las tendencias y tecnologías actuales a ser usadas en el desarrollo del sistema, de las cuales brindamos a continuación una breve descripción:

1.6.1 Las aplicaciones de escritorio.

La aplicación a ventanas es el tipo de aplicación que más a menudo se utiliza cuando trabajamos con el ordenador. Es una aplicación que se pone en marcha de forma local, que utiliza como interfaz del usuario las tecnologías de las ventanas típicas de los sistemas operativos Mac OS (en la que nació), Windows (todas sus variantes).

Casi todos los programas que se utilizan actualmente son aplicaciones de ventanas. Lo son, por ejemplo, Word, Netscape, Explorer, Jbuilder, Visual Studio, WinZip, Paint, XV, etc. (Todos pertenecen a los correspondientes productores). [15]

1.6.2 Lenguajes de Programación.

Un lenguaje de programación es una técnica estándar de comunicación que permite expresar las instrucciones que han de ser ejecutadas en una computadora. Consiste en un conjunto de reglas sintácticas y semánticas que definen un lenguaje informático.

Dentro de los lenguajes propuestos para desarrollar el sistema se encuentra C# como lenguaje convencional y Prolog como lenguaje descriptivo. A continuación una síntesis de los mismos.

1.6.2.1 C#.

C# un poco de Historia.

C# (leído en inglés "C Sharp" y en español "C Almohadilla") es el nuevo lenguaje de propósito general diseñado por Microsoft para su plataforma .NET. Sus principales creadores son Scott Wiltamuth y Anders Hejlsberg, éste último también conocido por haber sido el diseñador del lenguaje Turbo Pascal y la herramienta RAD Delphi. Aunque es posible escribir código para la plataforma .NET en muchos otros lenguajes, C# es el único que ha sido diseñado específicamente para ser utilizado en ella, por esta razón, se suele decir que C# es el lenguaje nativo de .NET.

La sintaxis y estructuración de C# es muy similar a la C++, ya que la intención de Microsoft con C# es facilitar la migración de códigos escritos en estos lenguajes a C# y facilitar su aprendizaje a los desarrolladores habituados a ellos. Sin embargo, su sencillez y el alto nivel de productividad son equiparables a los de Visual Basic.

En resumen, C# es un lenguaje de programación que toma las mejores características de lenguajes preexistentes como Visual Basic, Java o C++ y las combina en uno solo. Microsoft ha escrito la mayor parte de la BCL usándolo, por lo que su compilador es el más depurado y optimizado de los incluidos en el .NET Framework SDK.

Por que usar C#.

C# es un lenguaje orientado a objetos simple, elegante y con seguridad en el tratamiento de tipos, que permite crear cualquier tipo de aplicaciones de escritorio.

C# también proporciona la capacidad de generar componentes de sistema duraderos en virtud de las siguientes características:

- ? Total compatibilidad entre COM y plataforma para integración de código existente.
- ? Gran robustez, gracias a la recolección de elementos no utilizados (liberación de memoria) y a la seguridad en el tratamiento de tipos.
- ? Seguridad implementada por medio de mecanismos de confianza intrínsecos del código.
- ? Plena compatibilidad con conceptos de metadatos extensibles.

Presenta como características generales:

- ? Es una evolución de los lenguajes C y C++. Utiliza muchas de las características de C++ en las áreas de instrucciones, expresiones y operadores.
- ? Presenta considerables mejoras e innovaciones en áreas como seguridad de tipos, control de versiones, eventos y recolección de elementos no utilizados (liberación de memoria).
- ? Proporciona acceso a los tipos de API más comunes: .NET Framework, COM, Automatización y estilo C. Asimismo, admite el modo unsafe, en el que se pueden utilizar punteros para manipular memoria que no se encuentra bajo el control del recolector de elementos no utilizados. [8]

1.6.2.2 Prolog.

Lenguajes declarativos.

Se conoce como lenguajes declarativos en ciencias computacionales a aquellos lenguajes de programación en los cuales se le indica a la computadora que es lo que se desea obtener o que es lo que se está buscando. Tienen como característica ser fiables, elegantes y expresivos. Existen varios tipos de lenguajes declarativos:

- ? Los lenguajes algebraicos, como Maude y SQL.
- ? Los lenguajes funcionales, como Haskell.
- ? Los lenguajes lógicos, como Prolog.

Precisamente este último mencionado cuenta como nuestra segunda propuesta a utilizar para el desarrollo del sistema, del cual se brinda a continuación una síntesis de su funcionalidad y características.

Prolog y la Historia.

El desarrollo de Prolog se inició en 1970 con Alain Coulmeauer y Philippe Roussel, quienes estaban interesados en desarrollar un lenguaje para hacer deducciones a partir de texto. El nombre corresponde a "PROgramming in Logic" (Programación en lógica). Prolog fue desarrollado en Marsella, Francia, en 1972. El principio de resolución de Kowalski, de la Universidad de Edimburgo pareció un modelo apropiado para desarrollar sobre él mecanismo de inferencia. La primera implementación de Prolog se completó en 1972 usando el compilador de ALGOL W

de Wirth y los aspectos básicos del lenguaje actual se concluyeron en 1973. El uso de Prolog se extendió gradualmente entre quienes se dedicaban a la programación lógica principalmente por contacto personal y no a través de una comercialización del producto. Existen varias versiones diferentes, aunque bastante similares. No hay un estándar del Prolog, la versión desarrollada en la Universidad de Edimburgo ha llegado a ser utilizada ampliamente. El uso de este lenguaje no se extendió sino hasta los años ochenta. La falta de desarrollo de aplicaciones eficaces de Prolog inhibió su difusión. Entre las diferentes versiones podemos mencionar algunas como:

- ? LPA Prolog.
- ? Amzi Prolog.
- ? Visual Prolog.
- ? CIAO Prolog.
- ? SWI Prolog.

Programming in LOGiC (PROLOG), es uno de los lenguajes de programación ampliamente utilizados en IA (Inteligencia Artificial), presenta como características principales las siguientes:

- ? Proveniente del francés Programation et Logique, bastante popular en el medio de investigación de la Inteligencia Artificial.
- ? Se enmarca en el paradigma de los lenguajes lógicos, lo que lo diferencia enormemente de otros lenguajes más populares tales como Fortran, Pascal, C, etc.
- ? Basado en lógica y programación declarativa.
- ? Produce estilo de programación orientado a metas.
- ? No se especifica *cómo* debe hacerse sino *qué* debe lograrse (alto nivel).
- ? El programador se concentrará más en el conocimiento, que en los algoritmos.
- ? Puede ser utilizados para escribir programas capaces de examinar a otros programas.
- ? Sistemas expertos e inteligencia artificial.
- ? Validación automática de programas.
- ? Procesamiento de lenguaje natural.
- ? Prototipado rápido de aplicaciones.

? Bases de datos deductivas.

Este lenguaje es útil en la gestión de Juegos, en Inteligencia Artificial y Sistemas Expertos, como lenguaje especialmente pensado para construir bases de conocimientos basados en la lógica que forman parte importante de cualquier agente inteligente, en la construcción de Compiladores e Intérpretes, en el Reconocimiento del Lenguaje Natural, etc.

1.6.2.3 Patrones Diseño.

Se conoce como Patrón: Solución propuesta para un problema. Son abstracciones de soluciones obtenidas en casos concretos. No son soluciones recientes, si no que tienen el respaldo del uso y la experiencia. Se pueden aplicar en distintos contextos. [9]

Los Patrones de Diseño (Design Patterns) son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces. [12]

Un patrón de diseño es una descripción de clases y objetos comunicándose entre sí adaptada para resolver un problema de diseño general en un contexto particular. [17]

Estos se dividen en tres grandes categorías:

? Patrones Creacionales

Solucionan problemas de creación de instancias. Nos ayudan a encapsular y abstraer dicha creación.

? Patrones Estructurales

Solucionan problemas de composición (agregación) de clases y objetos.

? Patrones de Comportamiento

Soluciones respecto a la interacción y responsabilidades entre clases y objetos, así como los algoritmos que encapsulan.

Los patrones de diseño han contribuido a dar flexibilidad y extensibilidad a nuestros diseños. Pero en adición, han demostrado ser una forma muy útil (exitosa) de reutilizar diseño, ya que ellos no sólo nombran, abstraen e identifican aspectos claves de estructuras comunes de diseño, sino que generalmente son descritos en

una forma específica documental, haciendo su comprensión y aplicación fácil para el conjunto de desarrolladores. [12]

En el desarrollo del trabajo se ha hecho uso de Patrones Generales de Asignación de Responsabilidades (GRASP) como son:

1- Patrón Experto.

El patrón experto se centra en asignar una responsabilidad al experto en la información, cuyo experto será la clase que cuenta con la información necesaria para cumplir con la responsabilidad.

2- Patrón Creador.

El patrón creador se centra en asignarle la responsabilidad a una clase de crear instancias de otra, brindando así mejores oportunidades de reutilización.

3- Patrón Controlador.

El patrón controlador asigna la responsabilidad del manejo de un mensaje de los eventos del sistema a una clase. [18]

1.6.2.4 Arquitectura de 3 Capas.

Arquitectura de tres niveles.

Para enfrentarse a estos temas, la comunidad de software desarrollo la noción de una arquitectura de 3 niveles. Una aplicación se divide en tres capas lógicas distintas, cada una de ellas con un grupo de interfaces perfectamente definida. La primera capa se denomina capa de presentación y normalmente consiste en una interfaz gráfica de usuario de algún tipo. La capa intermedia, o capa de empresa, consiste en la aplicación o lógica de empresa, y la tercera capa, la capa de datos, contiene los datos necesarios para la aplicación.

La capa intermedia (lógica de aplicación) es básicamente el código al que recurre el usuario (a través de la capa de presentación) para recuperar los datos deseados. La capa de presentación recibe entonces los datos y los formatea para su presentación. Esta separación entre la lógica de aplicación de la interfaz de usuario añade una enorme flexibilidad al diseño de la aplicación. Puede construirse y desplegarse múltiples interfaces de usuario sin cambiar en absoluto la lógica de aplicación, siempre que la lógica de aplicación presente una interfaz claramente definida a la capa de presentación.

La tercera capa contiene los datos necesarios para la aplicación. Estos datos consisten en cualquier fuente de información, incluido una base de datos de empresa como Oracle o Sybase, un conjunto de documentos XML o incluso un servicio de directorio como el servidor LDAP. Además del tradicional mecanismo de almacenamiento relacional de bases de datos.

La metodología presentada por C. Larman presupone una estructura de tres capas que es típica de los Sistemas de Información. [16]

1. La capa de presentación o interfaz de usuario.

Esta capa reúne todos los aspectos del software que tiene que ver con las interfaces y la interacción con los diferentes tipos de usuarios humanos. Estos aspectos típicamente incluyen el manejo y aspecto de las ventanas, el formato de los reportes, menues, gráficos y elementos multimedia en general.

2. La capa del Dominio de la Aplicación.

Esta capa reúne todos los aspectos del software que tienen que automatizan o apoyan los procesos de negocio que llevan a cabo los usuarios. Estos aspectos típicamente incluyen las tareas que forman parte de los procesos, las reglas y restricciones que aplican. Esta capa también recibe el nombre de capa de la Lógica de la Aplicación.

3. La Capa del Repositorio.

Esta capa reúne todos los aspectos del software que tienen que ver con el manejo de los datos persistentes, por lo que también se le denomina la capa de la Base de Datos

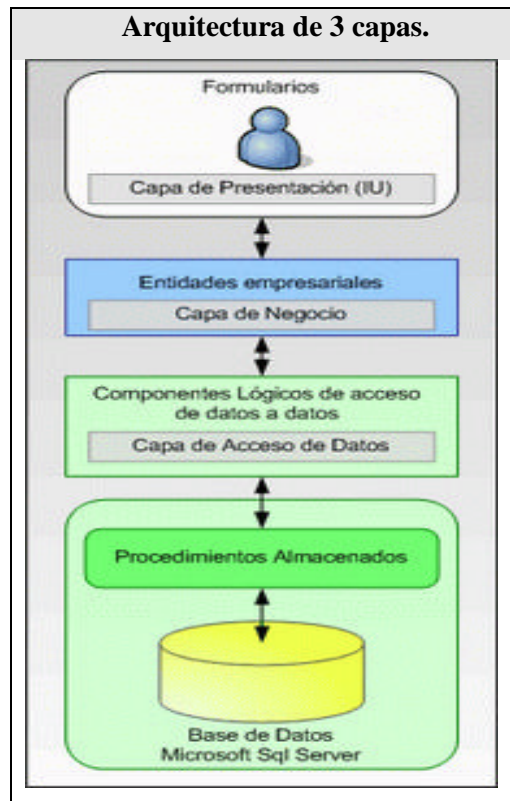


Figura 1.2 Arquitectura de 3 Capas.

1.6.3 Sistemas de Gestión de Bases de Datos.

Un Sistema Gestor de base de datos (SGBD) es un conjunto de programas que permiten crear y mantener una Base de datos, asegurando su integridad, confidencialidad y seguridad. [10]

Por tanto debe permitir:

- Definir una base de datos: especificar tipos, estructuras y restricciones de datos.
- Construir la base de datos: guardar los datos en algún medio controlado por el mismo SGBD.
- Manipular la base de datos: realizar consultas, actualizarla, generar informes.

Algunas de las características deseables en un Sistema Gestor de Base de Datos SGBD son:

- Control de la redundancia.
- Restricción de los accesos no autorizados.

- Cumplimiento de las restricciones de integridad.
- Facilidad de manipulación de la información.
- Control centralizado.

Entre los SGBD comúnmente utilizados en el mundo tenemos Oracle, MySQL, Microsoft SQL Server, PostgreSQL, Interbase, etc. Todos estos presentan un enfoque relacional con un buen basamento matemático centrado en el Álgebra Relacional.

Todos estos gestores presentan características propias que los diferencian ejemplo:

SQL Server: Este sistema incluye una versión reducida, llamada MSDE con el mismo motor de base de datos pero orientado a proyectos más pequeños. No es multiplataforma, ya que sólo está disponible en Sistemas Operativos de Microsoft.

Postgre SQL: es un motor de base de datos, es servidor de base de datos relacional libre, liberado bajo la licencia BSD, además de presentar un sistema denominado MVCC (Acceso concurrente multiversión).

Oracle: requiere de una licencia para poderlo utilizar, es decir, es necesario pagar para poder utilizarlo.

Como SGBD se seleccionó el MySQL por las ventajas que ofrece y por requerimientos propios del sistema.

1.6.3.1 MySQL.

MySQL es un sistema de gestión de bases de datos relacional, licenciado bajo la GPL de la GNU. Su diseño multihilo le permite soportar una gran carga de forma muy eficiente. MySQL fue creado por la empresa sueca MySQL AB, que mantiene el copyright del código fuente del servidor SQL, así como también de la marca.

Aunque MySQL es software libre, MySQL AB distribuye una versión comercial de MySQL, que no se diferencia de la versión libre más que en el soporte técnico que se ofrece, y la posibilidad de integrar este gestor en un software propietario, ya que de no ser así, se vulneraría la licencia GPL.

Este gestor de bases de datos es, probablemente, el gestor más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación es debida, en parte, a que existen infinidad de librerías y otras

herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración.

Historia de MySQL.

MySQL surgió como un intento de conectar el gestor mSQL a las tablas propias de MySQL AB, usando sus propias rutinas a bajo nivel. Tras unas primeras pruebas, vieron que mSQL no era lo bastante flexible para lo que necesitaban, por lo que tuvieron que desarrollar nuevas funciones. Esto resultó en una interfaz SQL a su base de datos, con una interfaz totalmente compatible a mSQL.

MySQL fue fundado por David Axmark, Allan Larsson, y Michael Widenius. No se sabe con certeza de donde proviene su nombre. Por un lado dicen que sus librerías han llevado el prefijo 'my' durante los diez últimos años. Por otro lado, la hija de uno de los desarrolladores se llama My. No saben cuál de estas dos causas (aunque bien podrían tratarse de la misma), han dado lugar al nombre de este conocido gestor de bases de datos. La versión estable de este gestor a días de hoy es la 3.23.49.

Principales características de MySQL.

Interioridades y portabilidad.

- ? Probado con un amplio rango de compiladores diferentes.
- ? Funciona en diferentes plataformas.
- ? APIs disponibles para C, C++, Eiffel, Java, Perl, PHP, Python, Ruby, y Tcl.
- ? Uso completo de multi-threaded mediante threads del kernel.
- ? Múltiples motores de almacenamiento (MyISAM, Merge, InnoDB, BDB, Memory/heap, MySQL Cluster, Federated, Archive, CSV, y Blackhole), permitiendo al usuario escoger la que sea más adecuada para cada tabla de la base de datos.
- ? Proporciona sistemas de almacenamientos transaccionales y no transaccionales.
- ? Usa tablas en disco B-tree (MyISAM) muy rápidas con compresión de índice.
- ? Un sistema de reserva de memoria muy rápido basado en threads.
- ? Joins muy rápidos usando un multi-join de un paso optimizado.
- ? Tablas hash en memoria, que son usadas como tablas temporales.

Tipos de columnas.

- ? Diversos tipos de columnas: enteros con/sin signo de 1, 2, 3, 4, y 8 bytes de longitud, FLOAT, DOUBLE, CHAR, VARCHAR, TEXT, BLOB, DATE, TIME, DATETIME, TIMESTAMP, YEAR, SET, ENUM, y tipos espaciales OpenGIS.

Sentencias y funciones.

- ? Soporte completo para operadores y funciones en las cláusulas de consultas SELECT y WHERE.
- ? Soporte completo para las cláusulas SQL GROUP BY y ORDER BY. Soporte de funciones de agrupación (COUNT (), COUNT (DISTINCT...), AVG (), STD (), SUM (), MAX (), MIN (), y GROUP_CONCAT ()).
- ? Soporte para LEFT OUTER JOIN y RIGHT OUTER JOIN cumpliendo estándares de sintaxis SQL y ODBC.
- ? Soporte para alias en tablas y columnas como lo requiere el estándar SQL.
- ? DELETE, INSERT, REPLACE, y UPDATE devuelven el número de filas que han cambiado (han sido afectadas).
- ? Puede mezclar tablas de distintas bases de datos en la misma consulta (como en MySQL 3.22).

Seguridad.

- ? Un sistema de privilegios y contraseñas que es muy flexible y seguro, y que permite verificación basada en el host.

Escalabilidad y límites.

- ? Soporte a grandes bases de datos.
- ? Se permiten hasta 64 índices por tabla.

Conectividad.

- ? Los clientes pueden conectar con el servidor MySQL usando sockets TCP/IP en cualquier plataforma. En sistemas Windows de la familia NT (NT, 2000, XP, o 2003).

Localización.

Para el desarrollo de este trabajo se contará con el uso de la versión 5.026 de dicho gestor la cual fue salida al consumo de la sociedad de desarrolladores el 3 de octubre de 2006.

Qué hay de nuevo en MySQL 5.0

Las siguientes características se implementan en MySQL 5.0.

- ? Tipo de datos BIT.
- ? Cursores.
- ? Diccionario de datos.
- ? Administrador de instancias.
- ? Matemáticas de precisión.
- ? Procedimientos almacenados.
- ? Modo estricto y tratamiento de errores estándar.
- ? Disparadores.
- ? Tipo de datos VARCHAR.
- ? Vistas.

1.6.4 Proceso de Desarrollo.

A diario la comunidad de desarrolladores a nivel mundial busca adecuarse a las necesidades de los usuarios que cada vez son más exigentes y estrictos en cuanto a la satisfacción de sus necesidades. Lo que implica que los equipos de desarrollo se vean obligados a desarrollar sistemas más complejos y a regirse por estándares, reglas o metodologías para poder crear un software con calidad.

Este trabajo presenta como premisa fundamental: lograr obtener un software con la calidad requerida por lo que nos hemos dado a la tarea de definir una metodología que nos ayude a integrar las múltiples facetas del proceso de desarrollo del mismo. Por lo que se ha escogido como metodología que guiará el proceso de desarrollo de nuestro sistema el Proceso Unificado de Desarrollo de Software (RUP).

El Proceso Racional Unificado o RUP (Rational Unified Process), es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. RUP es en realidad un refinamiento realizado por Rational Software del más genérico Proceso Unificado. Presenta características como:

- ? Forma disciplinada de asignar tareas y responsabilidades (quién hace qué, cuándo y cómo).

? Pretende implementar las mejores prácticas en Ingeniería de Software.

Desarrollo iterativo.

Administración de requisitos.

Uso de arquitectura basada en componentes.

Control de cambios.

Modelado visual del software.

Verificación de la calidad del software.

El RUP es un producto de Rational (IBM). Se caracteriza fundamentalmente por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso. Incluye artefactos (que son los productos tangibles del proceso como por ejemplo, el modelo de casos de uso, el código fuente, etc.) y roles (papel que desempeña una persona en un determinado momento, una persona puede desempeñar distintos roles a lo largo del proceso).

Es una forma disciplinada de asignar tareas, responsabilidades y de asegurar la producción de software de calidad dentro de plazos y presupuestos predecibles.

El RUP divide el proceso de desarrollo en ciclos, teniendo un producto final al final de cada ciclo, cada ciclo se divide en fases que finalizan con un hito donde se debe tomar una decisión importante:

? inicio: se hace un plan de fases, se identifican los principales casos de uso y se identifican los riesgos.

? elaboración: se hace un plan de proyecto, se completan los casos de uso y se eliminan los riesgos.

? construcción: se concentra en la elaboración de un producto totalmente operativo y eficiente y el manual de usuario.

? transición: se implementa el producto en el cliente y se entrena a los usuarios.

RUP está acompañado de una herramienta muy buena que soporta cada uno de los procesos que necesitamos: Rational Rose Enterprise Edition 2003. Además cubre el ciclo de vida de desarrollo de un proyecto y toma en cuenta las mejores prácticas a utilizar en el modelo de desarrollo de software.

UML (Lenguaje Unificado de Modelado).

El Lenguaje Unificado de Modelado preescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software.

UML se puede usar para modelar distintos tipos de sistemas: sistemas de software, sistemas de hardware, y organizaciones del mundo real.

UML ofrece nueve diagramas en los cuales modelar sistemas.

- ? Diagramas de Casos de Uso para modelar los procesos del negocio.
- ? Diagramas de Secuencia para modelar el paso de mensajes entre objetos.
- ? Diagramas de Colaboración para modelar interacciones entre objetos.
- ? Diagramas de Estado para modelar el comportamiento de los objetos en el sistema.
- ? Diagramas de Actividad para modelar el comportamiento de los Casos de Uso, objetos u operaciones.
- ? Diagramas de Clases para modelar la estructura estática de las clases en el sistema.
- ? Diagramas de Objetos para modelar la estructura estática de los objetos en el sistema.
- ? Diagramas de Componentes para modelar componentes.
- ? Diagramas de Implementación para modelar la distribución del sistema.

Como principales Características Presenta:

- ? Lenguaje unificado para la modelación de sistemas.
- ? Tecnología orientada a objetos.
- ? El cliente participa en todas las etapas del proyecto.
- ? Corrección de errores viables en todas las etapas
- ? Aplicable para tratar asuntos de escala inherentes a sistemas complejos de misión crítica, tiempo real, etc.

¿Porque es importante UML?

Hoy en día, UML ("Unified Markup Language") esta considerado como el lenguaje estándar en el análisis y diseño de sistemas de cómputo. Mediante UML es posible establecer la serie de requerimientos y estructuras necesarias para plasmar un sistema de software previo al proceso intensivo de escribir código.

En otros términos, así como en la construcción de un edificio se realizan planos previo a su construcción, en Software se deben realizar diseños en UML previa codificación de un sistema, ahora bien, aunque UML es un lenguaje, éste posee más características visuales que programáticas, mismas que facilitan a integrantes de un equipo multidisciplinario participar e intercomunicarse fácilmente, estos integrantes siendo los analistas, diseñadores, especialistas de área y desde luego los programadores.

Existen varias herramientas CASE (Computer-Aided Systems Engineering), que brindan ayuda a analistas, ingenieros de software y desarrolladores durante el ciclo de vida de desarrollo de un software, pero es Rational Rose líder en el modelado del desarrollo de los proyectos y es esta precisamente, la herramienta que se utiliza en la modelación de este sistema. La herramienta fue desarrollada por los creadores de UML, utilizando la notación estándar en la arquitectura de software. Esta herramienta integra todos los elementos que propone la metodología RUP para cubrir el ciclo de vida de un proyecto.

1.6.5 Herramientas Utilizadas.

1.6.5.1 Microsoft Visual Studio 2005.

Microsoft Visual Studio es una suite comercial de programación para el sistema operativo Microsoft Windows. Conformada por varios lenguajes de programación, entre ellos Visual C++, C# y Visual Basic. Visual Basic es un lenguaje de programación basado en el viejo lenguaje BASIC, brinda un sistema completo para el desarrollo de aplicaciones para el sistema operativo Microsoft Windows.

- ? Incorpora .NET Framework 2.0
- ? Ayuda con refactorización.
- ? Soporte para el nuevo software servidor Team System.

- ? Añadido soporte de tests para todo tipo de aplicaciones.

1.6.5.2 Amzi Prolog.

Amzi Prolog + Logic Server.

Amzi! Prolog + Logic Server hace fácil integrar componentes basados en reglas con Windows, Linux, Sun Solaris, HP/UX y otras aplicaciones. Embeber componentes Prolog basados en reglas en C/C++, C#, Java, Delphi, VB, Web Servers, ASP.NET, etc. Desarrollar bases lógicas en Unicode y/o ASCII, así como integrar componentes lógicos con bases de datos.

Amzi! provee las clásicas herramientas de línea de comandos y un completo entorno de desarrollo interactivo (IDE), basado en el proyecto Open Source Eclipse, que integra un editor de código, listener, depurador, compilador y motor de tiempo de ejecución en un intuitivo entorno gráfico (GUI).

Amzi! Prolog + Logic Server es la herramienta para poder construir componentes inteligentes y embeberlos en cualquier aplicación stand-alone o basada en servidor. Amzi! Prolog utiliza el IDE Eclipse con sofisticadas características de depuración y seguimiento.

1.6.5.3 EMS MySQL.

EMS Mysql Manager dispone de un conjunto de herramientas para llevar la administración de bases de datos MySQL. Dispone de una interfaz gráfica altamente intuitiva y recomendable tanto para principiantes como para usuarios más experimentados, mediante la cual se puede crear y editar parámetros de la base de datos de forma sencilla, además de poder diseñar las bases de datos visualmente. Ofrece también la posibilidad de ejecutar scripts SQL, extraer o imprimir metadata, así como importar y exportar datos. Administrar usuarios y privilegios de usuario, construir consultas SQL visualmente, crear informes de estructuras de bases de datos en formato HTML, etc.

Dentro de sus cambios más recientes podemos ver los siguientes:

- ? Mejorada la interfaz de usuario.
- ? Designador visual de bases de datos.
- ? Soporte de juegos de caracteres de conexión y de campos.

- ? HTTP Tunneling.
- ? Soporte completo de MySQL 5.0.
- ? Servicio nuevo de cortar tablas.
- ? Opciones nuevas de rejillas.
- ? Opciones de Commit de Data Import.

El EMS Mysql manager es el cliente mas completo para Mysql, aunque es un poco lento al iniciar, tiene elementos de gestión y configuración bastante poderosos sin descuidar la parte de construcción de consultas y los servicios de backup y generación de código SQL Dumb. Se puede tener control total sobre los usuarios y sus privilegios sobre la base de datos, aunque en el aspecto de monitoreo de rendimiento no ofrece ninguna opción. [13] Resulta una genial alternativa para la mayoría de las actividades que tienen que ver con el desarrollo, manipulación de datos, modelado, seguridad y acceso a Mysql. [13]

Esta es una herramienta que presenta como requisitos mínimos para su uso tener instalado cualquiera de los siguientes sistemas operativos Windows en sus versiones: 95/98/NT/ME/2000/XP, una PC con procesador a 166 MHz y memoria RAM de 32 MB. Aspectos técnicos que nos hacen posible hacer uso de la misma para el desarrollo de nuestro sistema.

1.6.5.4 Rational Rose.

Existen herramientas CASE de trabajo visuales como el Analise, el Designe, el Rational Rose, que permiten realizar el modelado del desarrollo de los proyectos, en la actualidad la mejor y más utilizada en el mercado mundial es Rational Rose y es la que se utiliza en la modelación de este proyecto. [14]

Rational Rose cubre todo el ciclo de vida de un proyecto: concepción y formalización del modelo, construcción de los componentes, transición a los usuarios y certificación de las distintas fases y entregables.

Es la herramienta CASE que comercializan los desarrolladores de UML y que soporta de forma completa la especificación del UML.

Rose es una herramienta con plataforma independiente que ayuda a la comunicación entre los miembros de equipo, a monitorear el tiempo de desarrollo y

a entender el entorno de los sistemas. Una de las grandes ventajas de Rose es que utiliza la notación estándar en la arquitectura de software(UML), la cual permite a los arquitectos de software y desarrolladores visualizar el sistema completo utilizando un lenguaje común, además los diseñadores pueden modelar sus componentes e interfaces en forma individual y luego unirlos con otros componentes del proyecto. [14]

Se decidió que se utilizaría el Rational Rose Enterprise Edition 2003, para sustentar la documentación, como modelador visual de la notación UML (Unified Modeling Language) para la confección de los diagramas que se ilustran en este documento. Esta herramienta es muy completa y ofrece amplias potencialidades.

1.7 Conclusiones.

En este capítulo se exponen las condiciones y problemas que rodean el objeto de estudio a través de los conceptos y definiciones planteadas. Se evidencia la necesidad de un software que contribuya en la correcta elaboración de los horarios docentes. Para desarrollar el prototipo del sistema se hace uso de la tecnología para los lenguajes de programación: C# y Prolog con soporte de base de datos en MySQL. El proceso de desarrollo es RUP, el cual está basado en la orientación a objetos y el modelado visual usando UML, lo cual permite incorporar al proceso de desarrollo de software un mejor control de los requerimientos y cambios.

Capítulo 2 Modelo del negocio.

2.1 Introducción.

Es importante antes de llevar a cabo el proceso de desarrollo de un software conocer todos los aspectos y particularidades del entorno donde el mismo se desempeñará (ya sea una empresa, una organización etc.), así como todos los posibles procesos que se desempeñan y se llevan a cabo en el ambiente para así tener una clara visión de la problemática y poder buscar soluciones factibles y precisas, además de lograr una comunicación efectiva entre los usuarios y el equipo de proyecto así como también entre los miembros de este último, con el objetivo de llegar a un entendimiento de lo que hay que hacer, ya que esa es precisamente la clave del éxito en la producción de un software. Precisamente debido a todos estos aspectos se hace necesario un modelado del negocio.

El modelado del negocio nos da una visión de qué es necesario hacer para dar respuesta a la solicitud del usuario, por lo que en este capítulo se exponen las condiciones o reglas que deben cumplirse, así como la descripción de los actores y trabajadores del negocio, en conjunto al modelo de objetos.

2.2 Modelo del negocio actual.

En el modelado del negocio se hace indispensable definir, reconocer o identificar los diferentes procesos que abarca el entorno en que se desenvuelve el negocio de estudio. En el capítulo anterior se detallaron y describieron los diferentes procesos que matizan nuestro ambiente de negocio, así como un análisis crítico de la ejecución de los mismos. Teniendo en cuenta las deficiencias detectadas y haciendo un análisis de los problemas potenciales que se manifiestan en cada proceso del negocio hemos llegado a definir una propuesta de negocio que no altera el flujo de procesos pero que influye en la manera en que se ejecutaran los mismos. A continuación se muestra la descripción del negocio propuesto.

2.2.1 Proceso Confeccionar Horario.

Este proceso comienza cuando en la facultad se hace necesario planificar por grupo una correcta distribución de las asignaturas, profesores y locales para llevar

a cabo el funcionamiento de la actividad docente, para esto en el local de planificación de la facultad se procede a la realización de esta tarea a manos de la planificadora que luego de contar con todos los documentos necesarios como son: el Balance de Carga y las Afectaciones de los diferentes profesores procede a la elaboración y confección del mismo.

2.3 Reglas del negocio a considerar.

- ? El horario es el típico, pudiendo identificarse los turnos como:

1ro ----- 8:00 am.	4to --- 1:30 pm
2do ----- 9:45 am.	5to-----3:15 pm
3ro ----- 11:30 am.	6to ---- 5:00 pm

- ? La planificadora en conjunto con el Vicedecano de Formación son las únicas personas autorizadas a la realización del horario.
- ? Para la realización del horario se debe contar con el Balance de Carga y con las diferentes afectaciones de los profesores docentes de manera previa.
- ? El año tiene una cantidad de grupos y cada grupo tiene una planificación diferente.
- ? De cada asignatura se conoce el nombre del profesor que imparte cada grupo, el total de clases semanales a planificar y que tipo de actividad (clase práctica, tele clase ó conferencia).
- ? Un profesor puede dar clase cualquier día y turno excepto en aquellos turnos ya planificado para otros grupos ó en aquellos días en los que tenga preparación metodológica, y a lo sumo a cuatro grupos.
- ? A una asignatura no puede planificársele en un grupo más de una clase en el mismo día.
- ? A los profesores externos no se le puede planificar clases en el último turno de la tarde.
- ? Se debe conocer la cantidad de aulas, locales docentes (salones de conferencia) y laboratorios de manera exacta con que se cuenta.
- ? Se debe conocer los días de producción de los grupos y los Consejos Docentes.

- ? Un profesor puede contar a lo sumo con tres grupos docentes para impartir clases.
- ? Los profesores adjuntos solo pueden dar clases en horarios determinados.

2.4 Actores del negocio.

Un actor del negocio es cualquier individuo, grupo, organización, máquina o sistema de información externo que interactúa con el negocio. El término *actor* significa el rol que algo o alguien juega cuando interactúa con el negocio para beneficiarse de sus resultados. De acuerdo con esta idea un actor del negocio representa un tipo particular de usuario del negocio más que un usuario físico, ya que varios usuarios físicos pueden realizar el mismo papel en relación al negocio o sea, ser instancias de un mismo actor. [14]

A continuación se muestra la tabla 2.1 correspondiente a los actores del negocio y sus respectivas justificaciones.

Actores del Negocio.	Justificación.
Vicedecana de Formación.	Interviene en el proceso confeccionar horario, ya que es el principal responsable de hacerles llegar a los estudiantes de la facultad el horario planificado.

Tabla 2.1 Descripción de los actores del negocio.

2.5 Diagrama de casos de uso del negocio.

El diagrama de casos de uso del negocio representa los procesos de un negocio (casos de uso del negocio) y su interacción con elementos externos (actores), tales como socios y clientes.

A continuación se muestra la figura correspondiente al diagrama de caso de uso del negocio.

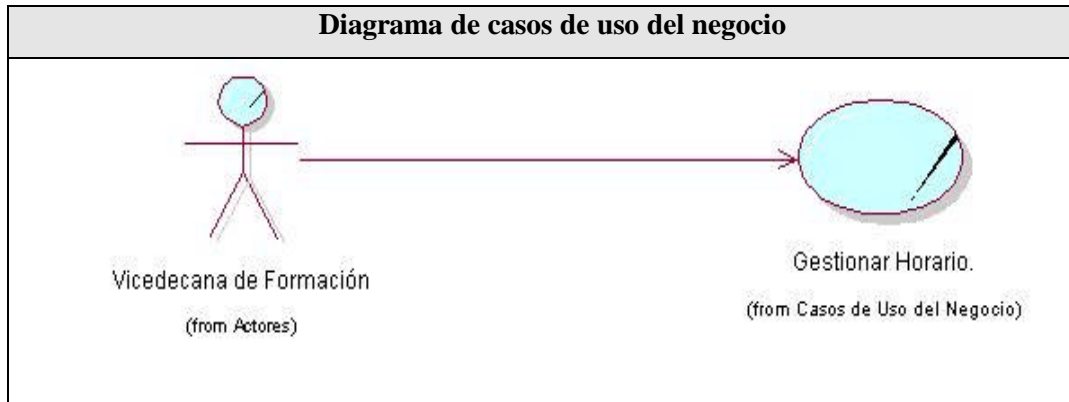


Figura2.1. Diagrama de casos de uso del Negocio.

2.6 Trabajadores del negocio.

Un trabajador define el comportamiento y las responsabilidades de un individuo. Son agentes que participan de acuerdo a un flujo de trabajo determinado en la manipulación y realización de las diferentes tareas ó actividades.

A continuación se muestra la tabla 2.2 correspondiente a los trabajadores del negocio y sus respectivas justificaciones.

Trabajadores del Negocio.	Justificación.
Planificador docente.	Persona cuya responsabilidad es directamente la realización del horario docente así como todo tipo de actividad que se realice en el local de planificación referente a este.

Tabla 1.2 Descripción de los trabajadores del negocio.

2.7 Descripción de los Casos de Uso del Negocio.

2.7.1 Caso de uso “Gestionar Horario”.

Especificación normal del Caso de Uso.

Caso de uso del negocio: Gestionar Horario.	
Actores del negocio: Vicedecana de Formación (inicia).	
Propósito: Obtener una planificación de profesores, asignaturas, locales que satisfaga las condiciones del entorno de estudio.	
Resumen: El caso de uso se inicia cuando la Vicedecana de Formación, entrega la documentación pertinente al planificador, persona encargada de confeccionar el horario docente. Una vez recopilada la información se comienza un proceso de selección, determinando el punto de partida, (es decir, que años se planificara primero, cual asignatura se ubicará primero, etc.). Posteriormente se comienza un proceso de asignación, consultando los documentos pertinentes, lo cual permitirá decidir si es posible realizar la asignación. Finalmente el planificador revisa el horario, intentando identificar posibles inconsistencias. El caso de uso finaliza cuando le es entregado a la Vicedecana de Formación una correcta distribución de asignaturas, profesores y grupos en los días de la semana.	
Curso normal de los eventos:	
Acción del actor	Respuesta del proceso de negocio
1- Solicita confección del horario docente.	2- Recibe la solicitud hecha. 2.1- Solicita la información necesaria para la realización de la tarea.
3- Proporciona la Información y documentación pedida.	4- Recibe la información y documentación necesaria. 4.1- Verifica la disponibilidad de locales docentes. 4.2- Define la cantidad de grupos docentes. 4.3- Asigna los locales a los distintos grupos docentes. 4.4- Selecciona punto de comienzo. 4.5- Selecciona las asignaturas con más frecuencia.

	4.6- Verifica disponibilidad del profesor docente. 4.7- Selecciona el tipo de actividad docente a asignar (conferencia ó clase práctica). 4.8- Realiza las distribuciones de las asignaturas. 4.9- Verifica la existencia de posibles errores. 4.10 Envía distribución de asignaturas.
5- Recibe distribución de asignaturas.	
Flujo alternativo de los eventos:	
Acción del actor:	Respuesta del proceso de negocio:
	4.10- Corrige los errores existentes.
	4.11- Envía distribución de asignaturas.
5- Recibe distribución de asignaturas.	
Prioridad: Alta	
Mejoras: Confeccionar a partir de una fuente de datos referentes a profesores, alumnos y asignaturas una correcta distribución por días, generando un horario útil para el desarrollo docente de la facultad.	
Otras secciones:	

Tabla 2.3 Especificación textual del caso de uso del negocio “Gestionar Horario”.

2.7.1.1 Diagramas de Actividades.

Un diagrama de actividad describe un proceso que explora el orden de las tareas o actividades que logran los objetivos del negocio, es un grafo que contiene estados en que puede hallarse una actividad. Ayudan a describir el detalle de qué es lo que pasa dentro del negocio, esta conformado por calles donde cada una de ellas representa una responsabilidad llevada a cabo por una parte de la organización.

A continuación se muestra en la figura el Diagrama de actividades del caso de uso del negocio “Gestionar Horario”.

2.7.1.2 Diagrama de Actividades referente al CU Gestionar Horario.

Figura 1. Diagrama del caso de uso Gestionar Horario.

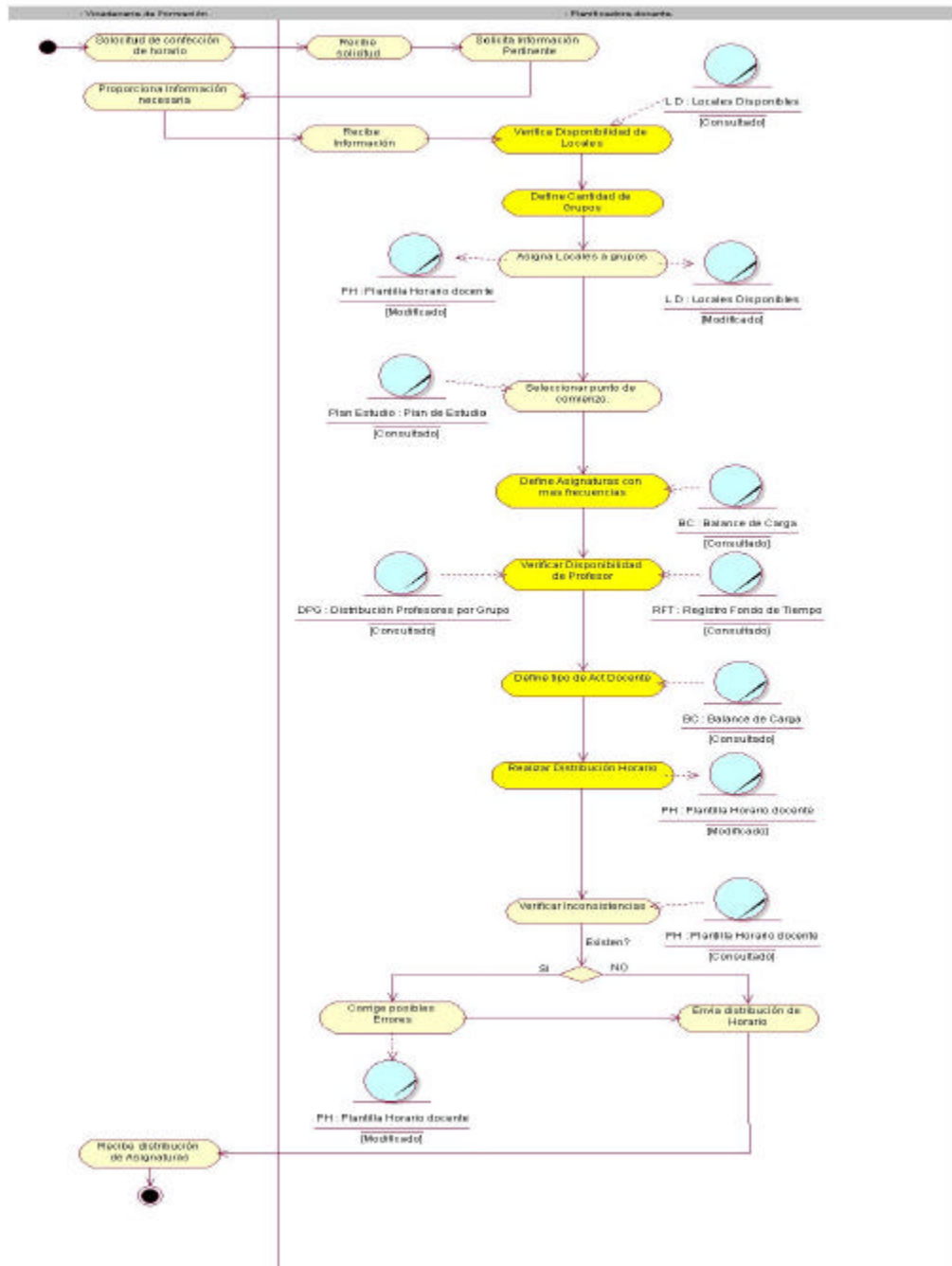


Figura 2.2. Diagrama de Actividades del caso de uso Gestionar Horario.

2.8 Modelo de Objetos.

El modelo de objetos describe cómo colaboran los trabajadores y entidades del negocio dentro del flujo de trabajo del proceso del negocio.

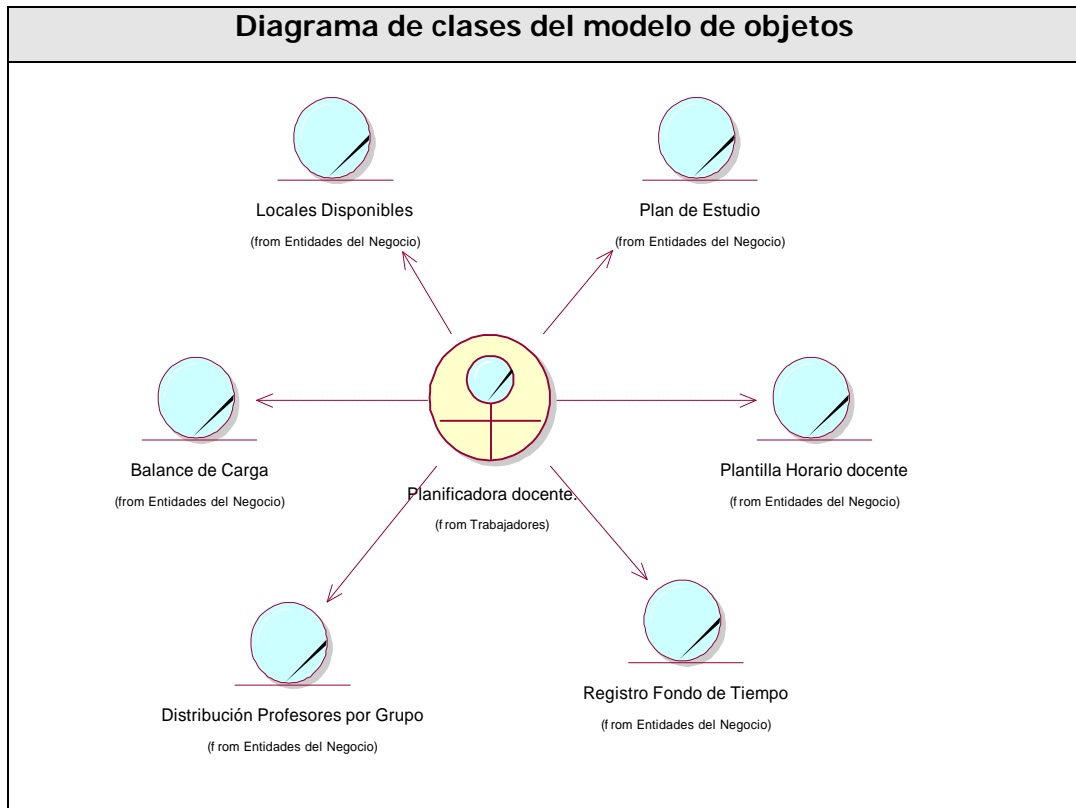


Figura 2.3 Modelo de Objetos.

2.9 Conclusiones.

En este capítulo fue descrito el negocio propuesto, quedando detallado como es que se ejecutan los procesos que se llevan a cabo en la realización del horario docente. Se hizo una breve caracterización de los actores y trabajadores que intervienen así como una descripción de la realización de los casos de uso que encierra el negocio con sus respectivos diagramas de actividades y diagramas de objetos, conformando así nuestro modelo de negocio el cual permite una mejor comprensión del problema que el sistema tiene que resolver.

Capítulo 3 Requisitos.

3.1 Introducción.

En el siguiente capítulo se definen los diferentes requisitos tanto funcionales como los no funcionales que intervienen en el desarrollo del sistema, así como una breve descripción de los actores que interactúan con el sistema y de los distintos casos de usos en los cuales estos intervienen.

Los requerimientos son aquellos requisitos que pueden definirse como:

1. Condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo.
2. 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.
3. Una representación documentada de una condición o capacidad como en 1 o 2.

Estos pueden ser clasificados en funcionales y no funcionales.

3.2 Definición de los requisitos funcionales.

Un requerimiento funcional son capacidades o condiciones que el sistema debe cumplir. A continuación se muestran las siguientes funcionalidades que nuestro sistema debe de tener.

R1. Gestionar Profesores.

- R1.1 Adicionar Profesores.
- R1.2 Eliminar Profesores.
- R1.3 Modificar Profesores.
- R1.4 Definir las afectaciones del Profesor.

R2 Gestionar Grupos.

- R2.1 Adicionar Grupos.
- R2.2 Eliminar Grupos.
- R2.3 Modificar Grupos.

R2.4 Definir distribución de profesores por Grupo.

R3 Gestionar Asignaturas.

R3.1 Adicionar Asignaturas.

R3.2 Eliminar Asignaturas.

R3.3 Modificar Asignaturas.

R3.4 Definir una Especificación.

R3.5 Modificar Especificación

R4 Gestionar Locales.

R4.1 Adicionar Locales.

R4.2 Eliminar Locales.

R4.3 Modificar Locales.

R4.4 Definir Especificación de locales.

R5 Fijar turnos de clases.

R5.1. Adicionar Turnos

R5.2. Eliminar Turnos

R5.3 Realizar Especificación Docente.

R5.4 Asignar turnos a días laborales.

R6 Generar diferentes Reportes.

R6.1 Listar Locales Disponibles.

R6.2 Listar la localización (local), donde se encuentra un determinado profesor en un momento dado.

R6.3 Listar la localización (local), de un determinado grupo, en un momento dado.

R6.4 Listar el tipo de actividad docente (Conferencia, Clase Práctica, Seminario) por grupo, en un momento dado.

R6.5 Generar la planificación en los formatos adecuados. (.xls, .html).

R6.6 Listar la planificación de una determinada asignatura.

R6.6 Listar la planificación de un determinado Grupo.

R6.8 Listar la planificación de un determinado profesor.

R6.9 Imprimir la planificación.

R6.10 Imprimir planificación según un criterio determinado (Asignatura, Grupo, Profesor).

R7 Gestionar Días.

R7.1 Adicionar Días.

R7.2 Eliminar Días.

R8 Gestionar Años.

R8.1 Adicionar Años.

R9 Gestionar Plan de Estudio.

R9.1 Adicionar Plan de Estudio.

R9.2 Modificar Plan de Estudio.

R10 Generar Planificación Docente (Horario Docente).

3.3 Definición de los 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. Normalmente están vinculados a requerimientos funcionales, es decir una vez se conozca lo que el sistema debe hacer podemos determinar cómo ha de comportarse, qué cualidades debe tener o cuán rápido o grande debe ser.

A continuación se muestran los requerimientos no funcionales.

? Apariencia o interfaz externa.

El sistema brindará una apariencia de interfaz sencilla y de fácil uso para aquellos usuarios con pocas habilidades en la computación, no se hará abuso de colores fuertes ni resaltantes brindará también un cómodo manejo en el acceso a las diferentes funcionalidades.

? Usabilidad.

El sistema puede ser usado por cualquier persona, comprendida en edad laboral que posea conocimientos básicos en el manejo de la computadora.

? Soporte.

Se requiere que el producto reciba mantenimiento ante cualquier fallo que ocurra.

? Software.

Para el funcionamiento del sistema es necesario el S.O. Windows 95 o superior.

? Documentación en línea.

El sistema brinda a los usuarios una documentación apropiada para el mejor trabajo con el mismo y mantenimiento del mismo.

? **Hardware.**

Se necesitan como requerimientos mínimos una PC con procesador Pentium II o superior.

? **Portabilidad.**

El producto es usado bajo el S.O. Windows.

El producto corre sobre una aplicación de escritorio, codificada en “.NET” y sus sistemas de bases de datos en MySQL.

? **Disponibilidad.**

Se garantiza a los usuarios del sistema el acceso a la información solicitada en todo momento.

? **Legales.**

El sistema cumple con las normas y leyes establecidas en nuestro país.

? **Seguridad.**

El sistema se encarga de controlar los niveles de acceso y funcionalidad de usuarios, de identificar al usuario antes de que pueda realizar cualquier acción sobre el sistema. Se hacen validaciones de la información, los usuarios acceden de manera rápida y operativa al sistema sin que los requerimientos de seguridad se conviertan en un retardo para ellos.

? **Confiabilidad**

La herramienta de implementación a utilizar tiene soporte para recuperación ante fallos y errores, la información manejada por el sistema está protegida de acceso no autorizado y divulgación.

? **Integridad:**

La información manejada por el sistema será objeto de cuidadosa protección contra la corrupción y estados inconsistentes.

? **Disponibilidad:**

A los usuarios autorizados se les garantizará el acceso a la información y los mecanismos utilizados para lograr la seguridad no ocultarán o retrasará a los usuarios para obtener los datos deseados. Si el usuario no se autentica, no puede acceder a la funcionalidad del sistema.

3.4 Actores del sistema a automatizar.

Los actores del sistema pueden representar el rol que juega una o varias personas, un equipo o un sistema automatizado, son parte del sistema, y pueden intercambiar información con él.

A continuación se definen los actores que interactúan con el sistema.

Actores	Justificación
Planificadora	Persona encargada de hacer la entrada de todos los datos necesarios al sistema para la elaboración del horario, así como de adicionar, modificar, eliminar la información almacenada y de realizar los diferentes reportes.

Tabla 2.1. Definición de actores del sistema a automatizar.

3.5 Paquetes y sus relaciones.

Un Caso de Uso especifica una secuencia de acciones que el sistema puede llevar a cabo interactuando con sus actores. De manera más precisa podemos decir que los Casos de Uso son “fragmentos” de funcionalidad que el sistema ofrece para aportar un resultado de valor para sus actores.

En la modelación de sistemas grandes ó en sistemas complejos, se hace necesario en su mayoría, agrupar los casos de usos y modelar el sistema haciendo uso de paquetes, es decir dividir el sistema en pedazos más pequeños, debido a que estos ayudan en cierto punto a comprender estructuralmente el sistema y hace que se desarrolle el mismo de forma independiente en pequeños módulos.

Para la realización de nuestro sistema se propone que este debe estar conformado por 4 paquetes fundamentalmente: Autenticar, Distribución, Gestionar Configuración y Reporte.

A continuación se muestra en la figura 3.1 el diagrama de paquetes y sus relaciones.

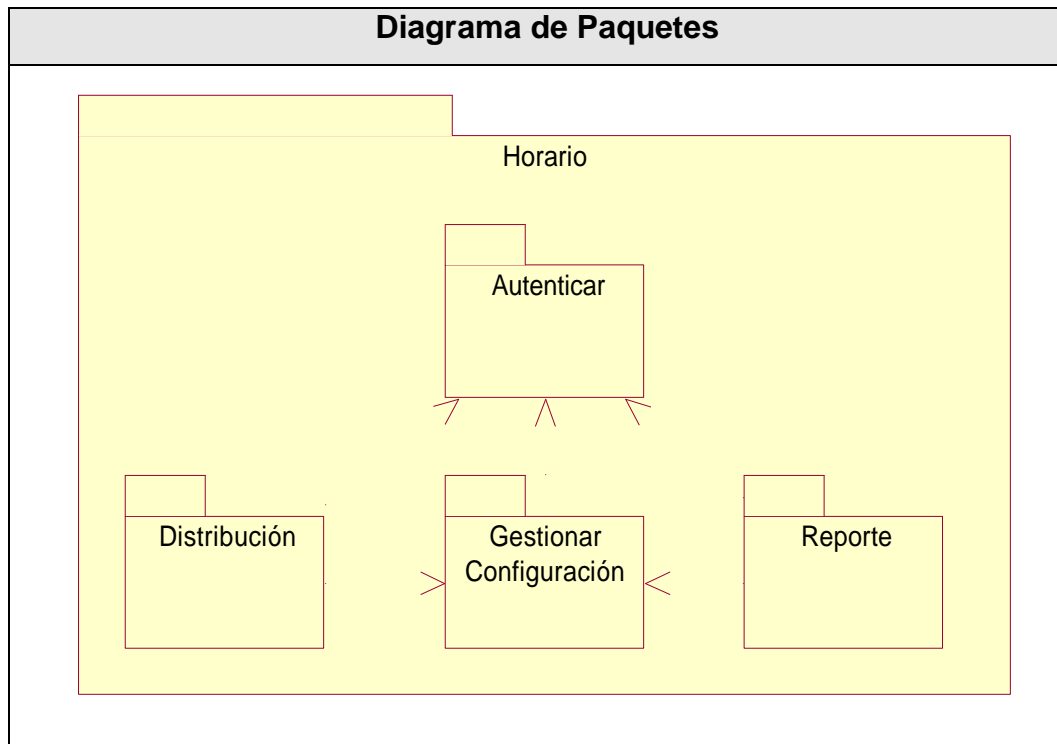


Figura 3.1 Diagrama de paquetes y sus relaciones.

3.6 Paquete Gestionar Configuración.

En este paquete se recogen todas las funcionalidades referentes a las entradas de datos y modificación de las entidades necesarias para la creación del horario.

3.6.1 Diagrama de casos de uso del sistema (Paquete Gestionar Configuración).

A continuación se muestra el diagrama de casos de uso del sistema correspondiente al paquete Gestionar Configuración.

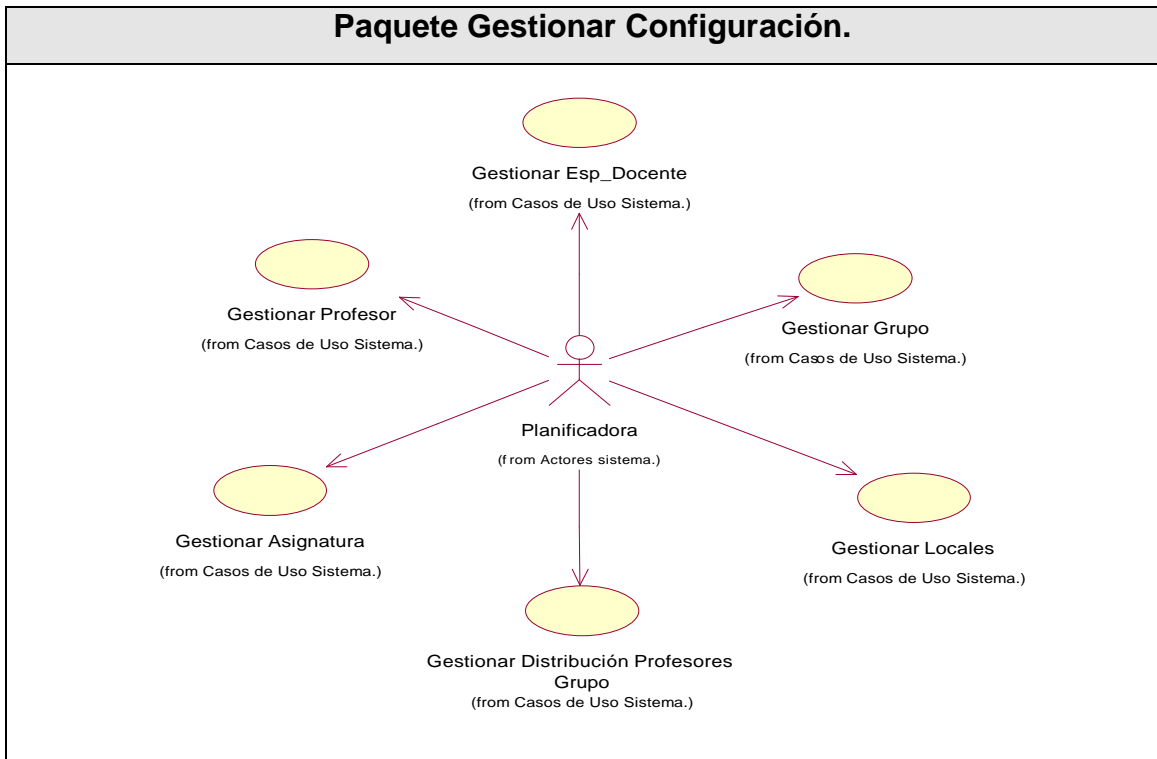


Figura 3.2 Diagramas de casos de usos del paquete Gestionar Configuración.

3.6.2 Descripción de los casos de uso del sistema.

3.6.2.1 “Caso de uso Gestionar Profesores”.

Nombre del Caso de Uso	Gestionar profesores.
Actores	Planificadora.
Propósito	Garantizar que los datos de cada profesor se registren, se modifiquen y se eliminen en el sistema así como que se le definan las afectaciones.
Resumen	El caso de uso se inicia cuando la planificadora una vez en el sistema accede a Profesores, a partir de aquí el sistema brinda la posibilidad de efectuar las operaciones de agregar, modificar,

	eliminar y definir las afectaciones de los diferentes profesores. Finaliza una vez que la planificadora realiza una de las operaciones.
Referencias	R1
Precondiciones	
Poscondiciones	<p>Para los siguientes procesos:</p> <p>Insertar: se crea una instancia de la clase profesor.</p> <p>Modificar: se modifican los atributos de la instancia seleccionada de la clase profesor.</p> <p>Eliminar: se elimina la instancia seleccionada de la clase profesor.</p> <p>Definir afectaciones de profesor: se crea una instancia de la clase afectación para la instancia seleccionada de la clase profesor.</p>
Curso normal de los eventos	
Acciones del actor	Respuestas del Sistema
1. La planificadora decide ejecutar acciones sobre el perfil de un profesor.	1.1 El sistema muestra una interfaz wizard de selección de perfiles.
2. La planificadora selecciona en el perfil profesor una de las operaciones a realizar.	<p>2.1 El sistema ejecuta una de las siguientes acciones:</p> <p>? Si decide agregar un profesor, ir a la sección "Agregar profesor".</p> <p>? Si decide modificar un profesor, ir a la sección "Modificar profesor".</p>

	<p>? Si decide eliminar un profesor, ir a la sección “Eliminar profesor”.</p> <p>? Si decide definir afectación de un profesor ir a la sección “Definir afectación de profesor ”</p>
Sección “Agregar profesor”	
3 La planificadora introduce los datos de Profesor.	<p>3.1 El sistema verifica que los datos introducidos sean válidos y que los requeridos no estén vacíos.</p> <p>3.2 El sistema verifica que el profesor no exista.</p> <p>3.3 El profesor se registra en el sistema.</p> <p>3.4 El sistema muestra un mensaje comunicando que el profesor se agregó satisfactoriamente, dando la posibilidad de agregar otro profesor finalizando así el caso de uso.</p>
Curso alternativo	
	3.2 Se emite un mensaje de error para que se llenen los campos obligatorios o se introduzcan correctamente los datos.
Curso alternativo	
	3.3 Se emite un mensaje informando de la existencia de un profesor.
Sección “Modificar profesor”	
3 La planificadora selecciona dentro del	3.1 El sistema verifica que se indique

listado de profesores el profesor a modificar.	correctamente el profesor a modificar
	3.2 El sistema carga los datos del profesor a modificar.
4 La planificadora realiza las actualizaciones deseadas.	4.1 El sistema verifica que los datos introducidos sean válidos y que los requeridos no estén vacíos. 4.2 El sistema actualiza la información. 4.3 Muestra un mensaje informando sobre la realización satisfactoria de la acción.
Curso alternativo	
	3.2 El sistema muestra un mensaje de no existencia del profesor especificado.
Curso alternativo	
	4.2 Se emite un mensaje de error para que se llenen los campos obligatorios o se introduzcan correctamente los datos.
Sección “Eliminar Profesor”	
3 La planificadora selecciona dentro del listado de profesores el que desea eliminar.	3.1 El sistema verifica que se indique correctamente el profesor a eliminar.
	3.2 El sistema elimina el profesor seleccionado.
	3.3 El sistema muestra un mensaje de acción realizada y finaliza el caso de uso.

Curso alternativo	
	3.2 El sistema muestra un mensaje de no existencia del profesor especificado.
Sección “Definir afectación de profesor”	
3 La planificadora selecciona los diferentes días y turnos en los que el profesor seleccionado no puede prestar servicio.	<p>3.1 El sistema brinda la posibilidad de hacer entrada de las afectaciones por días y turnos.</p> <p>3.2 El sistema hace captura y almacena las afectaciones del profesor seleccionado.</p>
	3.3 El sistema muestra un mensaje de acción realizada satisfactoriamente y termina el caso de uso.
Prioridad	Crítico
Prototipo	Ver Anexo 12.

Tabla 3.2 Descripción del caso de uso “Gestionar profesor”.

3.6.2.2 “Caso de uso Gestionar Grupos”.

Nombre del Caso de Uso	Gestionar grupos.
Actores	Planificadora.
Propósito	Garantizar que los datos de cada grupo se registren, se modifiquen y se eliminen.
Resumen	El caso de uso se inicia cuando la planificadora una vez en el sistema accede a Grupo, a partir de aquí el sistema brinda la posibilidad de efectuar

	las operaciones de agregar, modificar y eliminar. Finaliza una vez que la planificadora realiza una de las operaciones.
Referencias	R2
Precondiciones	
Poscondiciones	<p>Para los siguientes procesos:</p> <p>Insertar: se crea una instancia de la clase grupo.</p> <p>Modificar: se modifican los atributos de la instancia seleccionada de la clase grupo.</p> <p>Eliminar: se elimina la instancia seleccionada de la clase grupo.</p>
Curso normal de los eventos	
Acciones del actor	Respuestas del Sistema
1. La planificadora decide ejecutar acciones sobre el perfil de un grupo.	1.1 El sistema muestra una interfaz wizard de selección de perfiles.
2. La planificadora selecciona en el perfil grupo una de las operaciones a realizar.	<p>2.1 El sistema ejecuta una de las siguientes acciones:</p> <p>? Si decide agregar un grupo, ir a la sección "Agregar grupo".</p> <p>? Si decide modificar un grupo, ir a la sección "Modificar grupo".</p> <p>? Si decide eliminar un grupo, ir a la sección "Eliminar grupo".</p>
Sección "Agregar grupo".	

<p>3 La planificadora introduce los datos del grupo.</p>	<p>3.1 El sistema verifica que los datos introducidos sean válidos y que los requeridos no estén vacíos.</p> <p>3.2 El sistema verifica que el grupo no exista.</p> <p>3.3 El grupo se registra en el sistema.</p> <p>3.4 El sistema muestra un mensaje comunicando que el grupo se agregó satisfactoriamente, dando la posibilidad de agregar otro grupo y finaliza así el caso de uso.</p>
<p>Curso alternativo</p>	
	<p>3.2 Se emite un mensaje de error para que se llenen los campos obligatorios o se introduzcan correctamente los datos.</p>
<p>Curso alternativo</p>	
	<p>3.3 Se emite un mensaje informando de la existencia de un grupo.</p>
<p>Sección “Modificar grupo”.</p>	
<p>3 La planificadora selecciona dentro del listado de grupos el que desea modificar.</p>	<p>3.1 El sistema muestra los datos del grupo seleccionado.</p>
<p>4 La planificadora realiza las actualizaciones deseadas.</p>	<p>4.1 El sistema verifica que los datos introducidos sean válidos y que los requeridos no estén vacíos.</p> <p>4.2 El sistema actualiza la información.</p> <p>4.3 Muestra un mensaje de acción</p>

	realizada satisfactoriamente y finaliza el caso de uso.
Curso alternativo	
	4.2 Se emite un mensaje de error para que se llenen los campos obligatorios o se introduzcan correctamente los datos.
Sección “Eliminar grupo”.	
3 La planificadora selecciona dentro del listado de grupos el que desea eliminar.	3.2 El sistema elimina el grupo seleccionado.
	3.3 El sistema muestra un mensaje de acción realizada satisfactoriamente y finaliza el caso de uso.
Prioridad	Crítico.
Prototipo	Ver Anexo 13.

Tabla 3.3 Descripción del caso de uso “Gestionar Grupo”.

3.6.2.3 “Caso de uso Gestionar Asignatura”.

Nombre del Caso de Uso	Gestionar Asignatura.
Actores	Planificadora.
Propósito	Garantizar que los datos de cada asignatura se registren, se modifiquen y se eliminen, que se registre la especificación de la signatura y se modifique dicha especificación, y que se registre los años en el sistema.

Resumen	El caso de uso se inicia cuando la planificadora una vez en el sistema, accede a Asignaturas, a partir de aquí el sistema brinda la posibilidad de efectuar las operaciones de agregar, modificar, eliminar, definir la especificación para la asignatura, modificar la especificación y agregar los años. Finaliza una vez que la planificadora realiza una de las operaciones.
Referencias	R3,R8, R5.4
Precondiciones	
Poscondiciones	Para los siguientes procesos: Insertar: se crea una instancia de la clase asignatura. Modificar: se modifican los atributos de la instancia seleccionada de la clase asignatura. Eliminar: se elimina la instancia seleccionada de la clase asignatura. Especificación Asignatura: Se crea una instancia de la clase especificación Asignatura. Modificar Especificación Asignatura: Se modifican los atributos de la instancia seleccionada de la clase especificación asignatura. Insertar Año: Se crea una instancia de la clase año.
Curso normal de los eventos	

Acciones del actor	Respuestas del Sistema
1. La planificadora decide ejecutar acciones sobre el perfil de una asignatura.	1.1 El sistema muestra una interfaz wizard de selección de perfiles.
2. La planificadora selecciona en el perfil Asignatura una de las operaciones a realizar.	2.1 El sistema ejecuta una de las siguientes acciones: ? Si decide agregar una asignatura, ir a la sección “Agregar asignatura”. ? Si decide modificar una asignatura, ir a la sección “Modificar asignatura”. ? Si decide eliminar una asignatura, ir a la sección “Eliminar asignatura”. ? Si decide Definir Especificación Asignatura ir a la sección “Definir Especificación Asignatura”. ? Si decide Modificar Especificación Asignatura ir a la sección “Modificar Especificación Asignatura”. ? Si decide Insertar Año ir a la sección “Insertar Año”.
Sección “Agregar Asignatura”.	
3 La planificadora introduce los datos de la signatura.	3.1 El sistema verifica que los datos introducidos sean válidos y que los requeridos no estén vacíos. 3.2 El sistema verifica que la asignatura no exista. 3.3 La asignatura se registra en el

	<p>sistema.</p> <p>3.4 El sistema muestra un mensaje comunicando que la asignatura se agregó satisfactoriamente, dando la posibilidad de agregar otra y finaliza así el caso de uso.</p>
Curso alternativo	
	<p>3.2 Se emite un mensaje de error para que se llenen los campos obligatorios o se introduzcan correctamente los datos.</p>
Curso alternativo	
	<p>3.3 Se emite un mensaje informando de la existencia de la asignatura</p>
Sección “Modificar Asignatura”.	
<p>3 La planificadora selecciona dentro del listado de asignaturas la que desea modificar.</p>	<p>3.1 El sistema muestra los datos de la asignatura seleccionada.</p>
<p>4 La planificadora realiza las actualizaciones deseadas.</p>	<p>4.1 El sistema verifica que los datos introducidos sean válidos y que los requeridos no estén vacíos.</p> <p>4.2 El sistema actualiza la información.</p> <p>4.3 El sistema muestra un mensaje de acción realizada satisfactoriamente y finaliza el caso de uso.</p>
Curso alternativo	
	<p>4.2 Se emite un mensaje de error para que se llenen los campos obligatorios o</p>

	se introduzcan correctamente los datos.
Sección “Eliminar Asignatura”.	
3 La planificadora selecciona dentro del listado de asignaturas la que desea eliminar.	<p>3.1 El sistema elimina la asignatura seleccionada.</p> <p>3.2 El sistema muestra un mensaje de acción realizada y finaliza el caso de uso.</p>
Sección “Definir Especificación Asignatura”.	
3 La planificadora introduce los datos para definir la especificación de la asignatura seleccionada.	<p>3.1 El sistema verifica que los datos introducidos sean válidos y que los requeridos no estén vacíos.</p> <p>3.2 El sistema verifica que la especificación no exista.</p> <p>3.3 La especificación se registra en el sistema.</p> <p>3.4 El sistema muestra un mensaje comunicando que la especificación se registró satisfactoriamente, dando la posibilidad de registrar otra y finaliza así el caso de uso.</p>
Curso alternativo	
	3.2 Se emite un mensaje de error para que se llenen los campos obligatorios o se introduzcan correctamente los datos.
Curso alternativo	
	3.3 Se emite un mensaje informando de la existencia de la especificación.

Sección “Modificar Especificación Asignatura”.	
3 La planificadora selecciona dentro del listado de especificaciones de asignaturas la que desea modificar.	3.1 El sistema muestra los datos de la especificación seleccionada.
4 La planificadora realiza las actualizaciones deseadas.	4.1 El sistema verifica que los datos introducidos sean válidos y que los requeridos no estén vacíos. 4.2 El sistema actualiza la información. 4.3 El sistema muestra un mensaje de acción realizada satisfactoriamente y finaliza el caso de uso.
Curso alternativo	
	4.2 Se emite un mensaje de error para que se llenen los campos obligatorios o se introduzcan correctamente los datos.
Sección “Agregar Año”.	
3 La planificadora selecciona un año de la lista de años.	3.1 El sistema verifica que los datos introducidos sean válidos y que los requeridos no estén vacíos. 3.2 El sistema verifica que el año no exista. 3.3 El año se registra en el sistema. 3.4 El sistema muestra un mensaje comunicando que el año se agregó satisfactoriamente, dando la posibilidad de agregar otro y finaliza así el caso de

	USO.
Curso alternativo	
	3.2 Se emite un mensaje de error para que se llenen los campos obligatorios o se introduzcan correctamente los datos.
Curso alternativo	
	3.3 Se emite un mensaje informando de la existencia de un año
Prioridad	Crítico.
Prototipo	Ver Anexo 14.

Tabla 3.4 Descripción del caso de uso "Gestionar Asignatura".

3.6.2.4 "Caso de uso Gestionar Locales".

Nombre del Caso de Uso	Gestionar Locales.
Actores	Planificadora.
Propósito	Garantizar que los datos de cada local se registren, se modifiquen, se eliminen, se le asigne una especificación de local en el sistema.
Resumen	El caso de uso se inicia cuando la planificadora una vez en el sistema, accede a locales, a partir de aquí el sistema brinda la posibilidad de efectuar las operaciones de agregar, modificar, eliminar, y definir especificación de cada uno de los locales. Finaliza una vez que

	la planificadora realiza una de las operaciones.
Referencias	R4
Precondiciones	
Poscondiciones	<p>Para los siguientes procesos:</p> <p>Insertar: se crea una instancia de la clase local.</p> <p>Modificar: se modifican los atributos de la instancia seleccionada de la clase local.</p> <p>Eliminar: se elimina la instancia seleccionada de la clase local.</p> <p>Definir especificación: se crea una instancia de la clase especificación local.</p>
Curso normal de los eventos	
Acciones del actor	Respuestas del Sistema
1. La planificadora decide ejecutar acciones sobre el perfil de un local.	1.1 El sistema muestra una interfaz wizard de selección de perfiles.
2. La planificadora selecciona en el perfil Locales una de las operaciones a realizar.	<p>2.1 El sistema ejecuta una de las siguientes acciones:</p> <p>? Si decide agregar un local, ir a la sección "Agregar local".</p> <p>? Si decide modificar un local, ir a la sección "Modificar local".</p> <p>? Si decide eliminar un local, ir a la sección "Eliminar local".</p> <p>? Si decide Definir especificación de un local ir a la sección "Definir</p>

	Especificación local”.
Sección “Agregar local”.	
3 La planificadora introduce los datos del local.	<p>3.1 El sistema verifica que los datos introducidos sean válidos y que los requeridos no estén vacíos.</p> <p>3.2 El sistema verifica que el local no exista.</p> <p>3.3 El local se registra en el sistema.</p> <p>3.4 El sistema muestra un mensaje comunicando que el local se agregó satisfactoriamente, dando la posibilidad de agregar otro y finaliza así el caso de uso.</p>
Curso alternativo	
	3.2 Se emite un mensaje de error para que se llenen los campos obligatorios o se introduzcan correctamente los datos.
Curso alternativo	
	3.3 Se emite un mensaje informando de la existencia de un local.
Sección “Modificar local”.	
3 La planificadora selecciona dentro del listado de locales el que desea modificar.	3.1 El sistema muestra los datos actuales del local seleccionado.
4 La planificadora realiza las actualizaciones deseadas.	4.1 El sistema verifica que los datos introducidos sean válidos y que los

	<p>requeridos no estén vacíos.</p> <p>4.2 El sistema actualiza la información.</p> <p>4.3 Muestra un mensaje de acción realizada satisfactoriamente y finaliza el caso de uso.</p>
Curso alternativo	
	<p>4.2 Se emite un mensaje de error para que se llenen los campos obligatorios o se introduzcan correctamente los datos.</p>
Sección “Eliminar local”.	
<p>3 La planificadora selecciona dentro del listado de locales el que desea eliminar.</p>	<p>3.1 El sistema elimina el local seleccionado.</p> <p>3.2 El sistema muestra un mensaje de acción realizada satisfactoriamente y finaliza el caso de uso.</p>
Sección “Definir Especificación local”.	
<p>3 La planificadora introduce los datos necesarios para la configuración.</p>	<p>3.1 El sistema verifica que los datos introducidos sean válidos y que los requeridos no estén vacíos.</p> <p>3.2 El sistema verifica que la configuración no exista.</p> <p>3.3 La configuración se registra en el sistema.</p> <p>3.4 El sistema muestra un mensaje comunicando que la configuración se agregó satisfactoriamente, dando la posibilidad de agregar otra y finaliza así</p>

	el caso de uso.
Curso alternativo	
	3.2 Se emite un mensaje de error para que se llenen los campos obligatorios o se introduzcan correctamente los datos.
Curso alternativo	
	3.3 Se emite un mensaje informando de la existencia de una configuración de ese tipo.
Prioridad	Crítico.
Prototipo	Ver Anexo 15.

Tabla 3.5 Descripción del caso de uso "Gestionar Locales".

3.6.2.5 "Caso de uso Gestionar Especificación Docente".

Nombre del Caso de Uso	Gestionar Especificación Docente.
Actores	Planificadora.
Propósito	Garantizar que se registre y modifique un plan de estudio, que se registre y se eliminen los turnos de clase, que se registren y se eliminen los días laborales, así como que se registre una especificación docente.
Resumen	El caso de uso se inicia cuando la planificadora una vez en el sistema, accede a Datos Generales, a partir de aquí el sistema brinda la posibilidad de

	efectuar las operaciones de agregar, modificar un plan de estudio, agregar ó eliminar los turnos de clases, agregar ó eliminar los diferentes días de la semana y agregar una especificación docente. Finaliza una vez que la planificadora realiza una de las operaciones.
Referencias	R5,R7,R9
Precondiciones	
Poscondiciones	<p>Para los siguientes procesos:</p> <p>Insertar Plan de Estudio: se crea una instancia de la clase Plan de estudio.</p> <p>Modificar: se modifican los atributos de la instancia seleccionada de la clase Plan de estudio.</p> <p>Insertar Turno: se crea una instancia de la clase Turno.</p> <p>Eliminar Turno: se elimina la instancia seleccionada de la clase Turno.</p> <p>Insertar Día: se crea una instancia de la clase Día.</p> <p>Eliminar Día: se elimina la instancia seleccionada de la clase Día.</p> <p>Insertar Especificación Docente: se crea una instancia de la clase Especificación Docente.</p>
Curso normal de los eventos	
Acciones del actor	Respuestas del Sistema
1. La planificadora decide ejecutar acciones	1.1 El sistema muestra una interfaz

sobre el perfil de Datos Generales.	wizard de selección de perfiles.
<p>2. La planificadora selecciona en el perfil Datos Generales una de las operaciones a realizar.</p>	<p>2.1 El sistema ejecuta una de las siguientes acciones:</p> <ul style="list-style-type: none"> ? Si decide agregar un Plan de Estudios, ir a la sección “Agregar Plan de Estudios”. ? Si decide modificar un Plan de Estudios, ir a la sección “Modificar Plan de Estudios”. ? Si decide agregar un Turno de Clases, ir a la sección “Agregar Turno”. ? Si decide eliminar un Turno de Clases, ir a la sección “Eliminar Turno”. ? Si decide agregar un Día, ir a la sección “Agregar Día”. ? Si decide eliminar un Día, ir a la sección “Eliminar Día”. ? Si decide Definir especificación Docente ir a la sección “Definir Especificación Docente”.
Sección “Agregar Plan de Estudios”.	
<p>3 La planificadora introduce los datos referentes al plan de estudio.</p>	<p>3.1 El sistema verifica que los datos introducidos sean válidos y que los requeridos no estén vacíos.</p> <p>3.2 El sistema verifica que el plan de</p>

	<p>estudio no este definido antes.</p> <p>3.3 El plan de estudios se registra en el sistema.</p> <p>3.4 El sistema muestra un mensaje comunicando que el plan de estudio se agregó satisfactoriamente, dando la posibilidad de agregar otro y finaliza así el caso de uso.</p>
Curso alternativo	
	<p>3.2 Se emite un mensaje de error para que se llenen los campos obligatorios o se introduzcan correctamente los datos.</p>
Curso alternativo	
	<p>3.3 Se emite un mensaje informando de la existencia de un plan de estudios ya definido.</p>
Sección “Modificar Plan de Estudios”.	
<p>3 La planificadora selecciona dentro del listado de planes de estudios definidos el que desea modificar.</p>	<p>3.1 El sistema muestra los datos actuales del plan de estudio seleccionado.</p>
<p>4 La planificadora realiza las actualizaciones deseadas.</p>	<p>4.1 El sistema verifica que los datos introducidos sean válidos y que los requeridos no estén vacíos.</p> <p>4.2 El sistema actualiza la información.</p> <p>4.3 Muestra un mensaje de acción realizada satisfactoriamente y finaliza el</p>

	caso de uso.
Curso alternativo	
	4.2 Se emite un mensaje de error para que se llenen los campos obligatorios o se introduzcan correctamente los datos.
Sección “Agregar Turno”.	
3 La planificadora introduce los datos referentes al un Turno de Clases.	<p>3.1 El sistema verifica que los datos introducidos sean válidos y que los requeridos no estén vacíos.</p> <p>3.2 El sistema verifica que el turno de clase no este definido antes.</p> <p>3.3 El turno de clase se registra en el sistema.</p> <p>3.4 El sistema muestra un mensaje comunicando que el turno de clase se agregó satisfactoriamente, dando la posibilidad de agregar otro y finaliza así el caso de uso.</p>
Curso alternativo	
	3.2 Se emite un mensaje de error para que se llenen los campos obligatorios o se introduzcan correctamente los datos.
Curso alternativo	
	3.3 Se emite un mensaje informando de la existencia de un turno de clases ya definido.

Sección “Eliminar Turno”.	
3 La planificadora selecciona dentro del listado de turnos de clases el que desea eliminar.	3.1 El sistema elimina el turno de clases seleccionado. 3.2 El sistema muestra un mensaje de de acción realizada satisfactoriamente y finaliza el caso de uso.
Sección “Agregar Día”.	
3 La planificadora selecciona los Día de la semana a utilizar.	3.1 El sistema verifica que los datos introducidos sean válidos y que los requeridos no estén vacíos. 3.2 El sistema verifica que el Día de la semana no este definido antes. 3.3 El Día de la semana se registra en el sistema. 3.4 El sistema muestra un mensaje comunicando que el Día de la semana se agregó satisfactoriamente, dando la posibilidad de agregar otro y finaliza así el caso de uso.
Curso alternativo	
	3.2 Se emite un mensaje de error para que se llenen los campos obligatorios o se seleccione en la lista.
Curso alternativo	
	3.3 Se emite un mensaje informando de la existencia de un Día de la semana ya

	definido.
Sección “Eliminar Días”.	
3 La planificadora selecciona dentro del listado de Días de la semana el que desea eliminar.	<p>3.1 El sistema elimina el Día de la semana seleccionado.</p> <p>3.2 El sistema muestra un mensaje de de acción realizada satisfactoriamente y finaliza el caso de uso.</p>
Sección “Definir Especificación Docente”.	
3 La planificadora introduce los datos necesarios para la configuración.	<p>3.1 El sistema verifica que los datos introducidos sean válidos y que los requeridos no estén vacíos.</p> <p>3.2 El sistema verifica que la configuración docente no exista.</p> <p>3.3 La configuración docente se registra en el sistema.</p> <p>3.4 El sistema muestra un mensaje comunicando que la configuración docente se agregó satisfactoriamente, dando la posibilidad de agregar otra y finaliza así el caso de uso.</p>
Curso alternativo	
	3.2 Se emite un mensaje de error para que se llenen los campos obligatorios o se introduzcan correctamente los datos.
Curso alternativo	
	3.3 Se emite un mensaje informando de

	la existencia de una configuración docente ya definida.
Prioridad	Crítico.
Prototipo	Ver Anexo 16.

Tabla 3.6 Descripción del caso de uso “Gestionar Especificación Docente”.

3.6.2.6 Caso de Uso Gestionar Distribución Profesores, Grupo, Asignatura.

Nombre del Caso de Uso	Gestionar Distribución Profesores Grupo Asignatura.
Actores	Planificadora.
Propósito	Garantizar que quede registrada la asignación de los profesores con sus correspondientes grupos y con sus correspondientes asignaturas.
Resumen	El caso de uso se inicia cuando la planificadora una vez en el sistema accede a Grupo, a partir de aquí el sistema brinda la posibilidad de efectuar las operaciones de agregar una distribución profesores grupo, asignatura. Finaliza una vez que la planificadora realiza la operación.
Referencias	R2.4
Precondiciones	Que exista un profesor, un grupo y una asignatura previamente en el sistema.
Poscondiciones	Para los siguientes procesos: Insertar: se asignan valores a los

	atributos grupo y asignatura de la instancia profesor.
Curso normal de los eventos	
Acciones del actor	Respuestas del Sistema
1. La planificadora decide ejecutar acciones sobre el perfil de un grupo.	1.1 El sistema muestra una interfaz wizard de selección de perfiles.
2. La planificadora selecciona en el perfil grupo la opción Distribución profesor grupo asignatura (PGA).	2.1 El sistema brinda la posibilidad de seleccionar el profesor, el grupo y la asignatura para realizar la asignación.
3 La planificadora selecciona el profesor, el grupo y la asignatura.	<p>3.1 El sistema verifica que los datos seleccionados sean válidos y que los requeridos no estén vacíos.</p> <p>3.2 El sistema verifica que la asignación no exista y que cumpla con las restricciones lógicas.</p> <p>3.3 La asignación se registra en el sistema.</p> <p>3.4 El sistema muestra un mensaje comunicando que la asignación se agregó satisfactoriamente, dando la posibilidad de agregar otra y finaliza así el caso de uso.</p>
Curso alternativo	
	3.2 Se emite un mensaje de error para que se llenen los campos obligatorios o se introduzcan correctamente los datos.

Curso alternativo	
	3.3 Se emite un mensaje informando de la existencia de la asignación.
Prioridad	Crítico.
Prototipo	Ver Anexo 13.

Tabla 3.7 Descripción del caso de uso “Gestionar Distribución Profesores Grupo”.

3.7 Paquete Autenticarse.

En este paquete se recogen todas las funcionalidades referentes al proceso de autenticarse en el sistema.

3.7.1 Diagrama de casos de uso del sistema (Paquete Autenticarse).

A continuación se muestra el diagrama de casos de uso del sistema correspondiente al paquete Autenticarse.

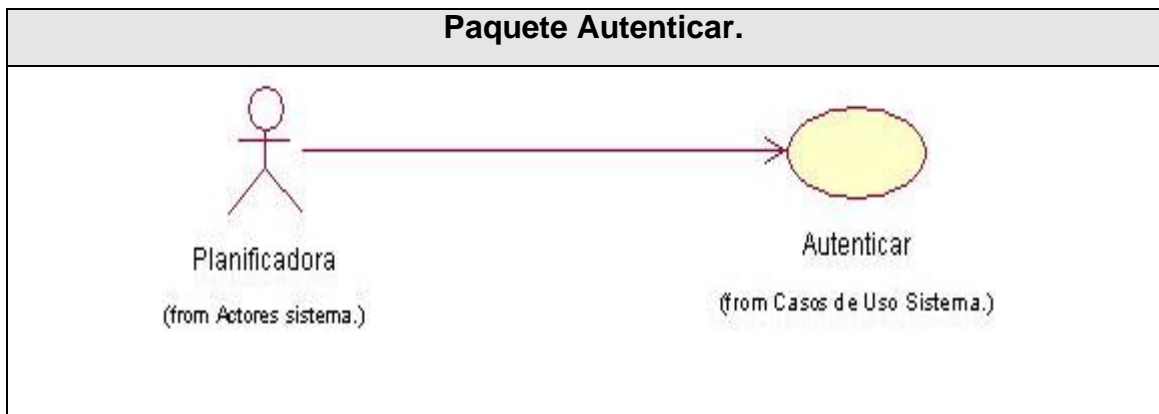


Figura 3.3 Diagrama de casos de uso del sistema del paquete Autenticarse.

3.7.2 Descripción de los casos de uso del sistema.

3.7.2.1 “Caso de uso Autenticarse”.

Nombre del Caso de Uso	Autenticar.
Actores	Planificadora.
Propósito	Mantener una política de seguridad en el sistema y garantizar que los usuarios del sistema (Planificadora) queden registrados antes de acceder al mismo.
Resumen	El caso de uso se inicia cuando la planificadora accede al sistema para registrarse, inmediatamente se visualiza una pantalla donde se les informa la entrada del usuario y de la contraseña, a partir de aquí el sistema brinda la posibilidad de efectuar las operaciones correspondientes para las confección del horario así como las entradas, salidas de datos y modificar Contraseña y Bloqueo de sesión. Finaliza una vez que la planificadora realiza una de las operaciones.
Referencias	RNF.
Precondiciones	Que sea introducido nombre usuario y contraseña.
Poscondiciones	Para los siguientes procesos: Autenticarse: Se crea una instancia de la clase control de usuario.
Curso normal de los eventos	

Acciones del actor	Respuestas del Sistema
1. La planificadora decide ejecutar acciones en el sistema.	1.1 El sistema muestra la interfaz de autenticarse.
2. La planificadora procede al llenado de las casillas con el nombre de usuario y su contraseña correspondiente.	2.1 El sistema valida los datos entrados y ejecuta una de las siguientes acciones: ? Si los datos son correctos, ir a la sección "Correctos". ? Si los datos son incorrectos, ir a la sección "Incorrectos".
Sección "Correctos"	
	2.2 El sistema permite que el usuario acceda a sus funcionalidades y finaliza el Caso de Uso.
Sección "Incorrectos"	
	2.2 El sistema muestra un mensaje de fallo en la autenticación. 2.3 El sistema procede nuevamente a mostrar la interfaz de autenticarse y finaliza el caso de uso.
Prioridad	Crítico.
Prototipo	Ver Anexo 17.

Tabla 3.8 Descripción del caso de uso "Autenticar".

3.8 Paquete Distribución.

En este paquete se recogen todas las funcionalidades referentes a la creación del horario docente.

3.8.1 Diagrama del Caso de uso del sistema (Paquete Distribución).

A continuación se muestra el diagrama de casos de uso del sistema correspondiente al paquete Distribución.

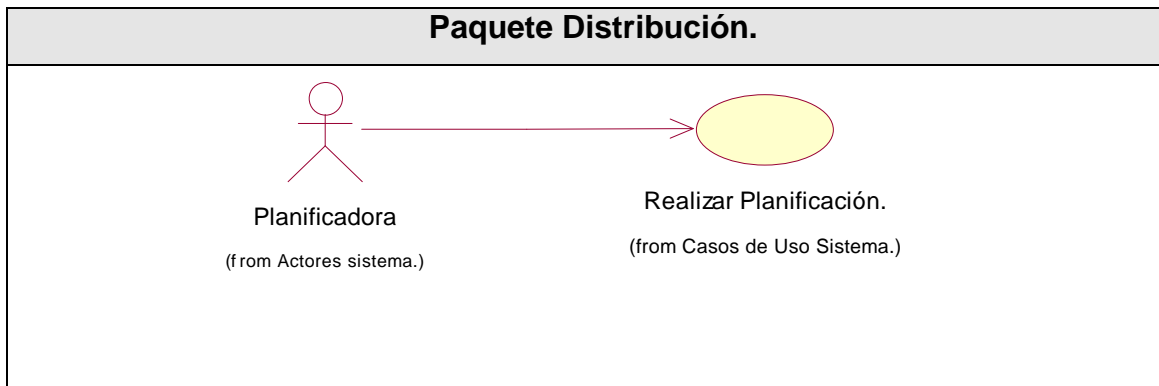


Figura 3.4 Diagrama de casos de uso del sistema del paquete Distribución.

3.8.2 Descripción de los casos de uso del sistema.

3.8.2.1 “Caso de uso Realizar Planificación”.

Nombre del Caso de Uso	Realizar Planificación.
Actores	Planificadora.
Propósito	Garantizar que las distintas asignaturas sean correctamente distribuidas por el sistema en los diferentes días de la semana, teniendo en cuenta los datos y restricciones de entrada necesarios para la confección del horario.

Resumen	El caso de uso se inicia cuando la planificadora accede al sistema y luego de haber definido todas las entradas correspondientes por mediación del wizard de configuración accede a la opción de generar horario.
Referencias	R10
Precondiciones	Deben existir todas las entradas correspondientes para realizar el proceso.
Poscondiciones	Para el siguiente proceso generar horario: Se crea una instancia horario de la clase Planificación.
Curso normal de los eventos	
Acciones del actor	Respuestas del Sistema
1. La planificadora hace entrada de todos los datos correspondientes hasta acceder al perfil Generar Horario.	1.1 Brinda la posibilidad de definir el tipo de planificación (Típica, Bloque).
2. La planificadora selecciona el tipo de planificación a generar: Típica.	2.1 Muestra el estado de la operación.
	2.2 Muestra el resultado de la planificación. Finalizando de este modo el Caso de Uso.
Curso alternativo	
2. La planificadora selecciona el tipo de planificación a generar: Bloque.	2.1 Muestra el estado de la operación.

	2.2 Muestra el resultado de la planificación. Finalizando de este modo el Caso de Uso.
Prioridad	Crítico.
Prototipo	Ver Anexo 18.

Tabla 3.9 Descripción del caso de uso "Realizar Planificación".

3.9 Paquete Reporte.

En este paquete se recogen todas las funcionalidades referentes a la obtención de los posibles reportes que desprenderá nuestra aplicación.

3.9.1 Diagrama del Caso de uso del sistema (Paquete Reporte).

A continuación se muestra el diagrama de casos de uso del sistema correspondiente al paquete Reporte.

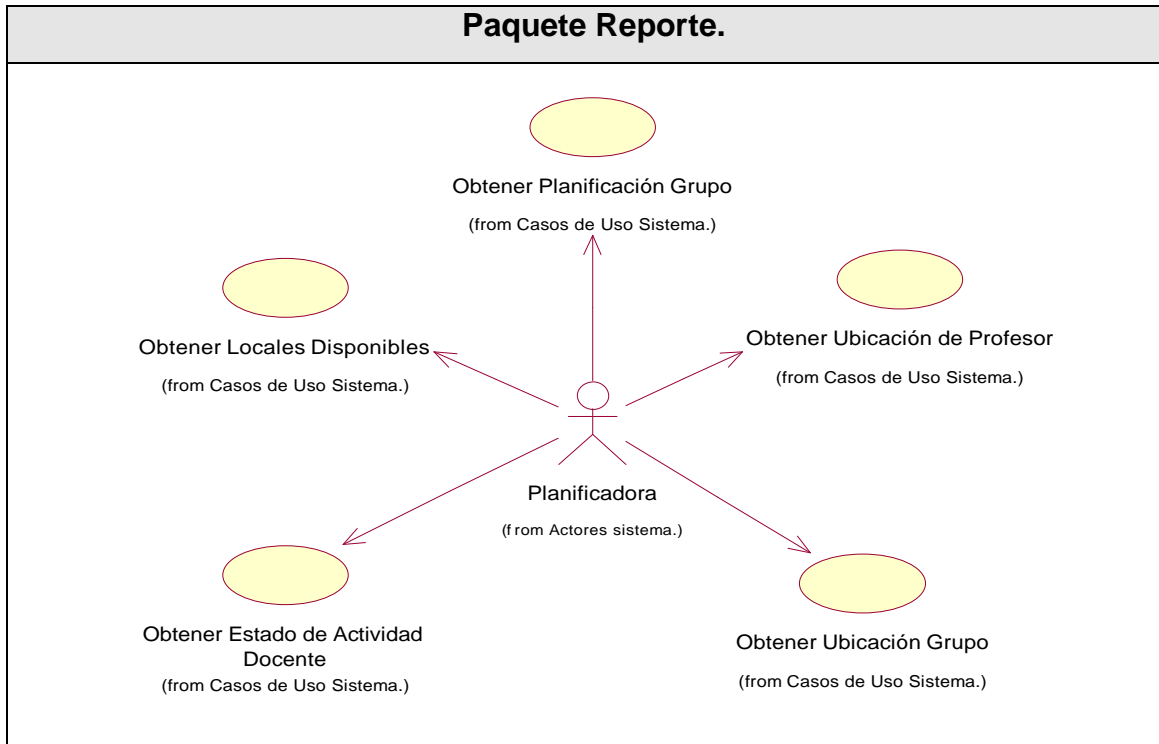


Figura 3.5 Diagrama de casos de uso del sistema del paquete Reporte.

3.9.2 Descripción de los casos de uso del sistema.

3.9.2.1 “Caso de uso Obtener Locales Disponibles”.

Nombre del Caso de Uso	Obtener locales Disponibles.
Actores	Planificadora.
Propósito	Tener conocimiento de los locales disponibles ó libres una vez comenzada la actividad docente.
Resumen	El caso de uso se inicia cuando la planificadora accede al sistema y solicita obtener locales disponibles, inmediatamente se visualiza una pantalla donde se muestra una lista de los posibles locales fuera de actividad

	docente.
Referencias	R6.1
Precondiciones	Debe existir una planificación realizada con anterioridad.
Poscondiciones	Para los siguientes procesos: Obtener relación de locales disponibles. Se retorna una instancia de la clase local que no haya sido asociada a ninguna instancia de la clase grupo.
Curso normal de los eventos	
Acciones del actor	Respuestas del Sistema
1. La planificadora decide obtener el listado de los locales disponibles.	1.1 Despliega una interfaz con los locales disponibles.
2. La planificadora selecciona el local deseado.	2.1 Muestra los detalles del local seleccionado. (Tipo de local, descripción, disponibilidad, etc.) Y finaliza el caso de uso.
Curso alternativo	
1. La planificadora decide obtener el listado de los locales disponibles.	1.1 Se despliega una advertencia, que especifica que no existen locales disponibles en ese momento y finaliza el caso de uso.

Tabla 3.10 Descripción del caso de uso “Obtener Locales Disponibles”.

3.9.2.2 “Caso de uso Obtener Ubicación de Profesor”.

Nombre del Caso de Uso	Obtener Ubicación de Profesores.
Actores	Planificadora.
Propósito	Determinar el local donde se encuentra un determinado profesor impartiendo clases.
Resumen	El caso de uso se inicia cuando la planificadora accede al sistema y solicita obtener ubicación de un profesor, inmediatamente se visualiza un formulario mostrando el local donde se encuentra el profesor impartiendo clases.
Referencias	R7
Precondiciones	Que se haya realizado una planificación con anterioridad.
Poscondiciones	Para los siguientes procesos: Obtener ubicación de profesor: Se retorna el valor correspondiente al atributo local de la clase planificación, correspondiente a la clase seleccionada de la instancia profesor.
Curso normal de los eventos	
Acciones del actor	Respuestas del Sistema
1. La planificadora selecciona la opción localizar profesor.	1.1 Brinda la posibilidad de seleccionar el profesor que se desea localizar.

2 La planificadora selecciona el profesor deseado.	2.1 Muestra el local de clases correspondiente al profesor seleccionado, las actividades docentes a desarrollar por el profesor en cuestión. Así como los días, y los locales, donde se desarrollaran y finaliza el caso de uso.
Prioridad	Secundario.

Tabla 3.11 Descripción del caso de uso “Obtener Ubicación de Profesor”.

3.9.2.3 “Caso de uso Obtener Estado de Actividad Docente”.

Nombre del Caso de Uso	Obtener Estado de Actividad Docente.
Actores	Planificadora.
Propósito	Tener conocimiento del estado (conferencia, clase práctica, tele clase) de las actividades docentes que se imparten en un momento dado.
Resumen	El caso de uso se inicia cuando la planificadora accede al sistema y solicita obtener el estado actual que se encuentra una actividad docente, inmediatamente se visualiza una pantalla donde se muestra una lista de las diferentes actividades docentes por grupo y su estado actual (C, CP, TC).
Referencias	R6.5
Precondiciones	Que se haya realizado una Especificación de Asignaturas con anterioridad.

Poscondiciones	Para los siguientes procesos: Obtener estado de la actividad docente: Se obtiene los datos actuales referentes a los atributos de la instancia especificación asignatura correspondiente a la instancia grupo del grupo seleccionado.
Curso normal de los eventos	
Acciones del actor	Respuestas del Sistema
1. La planificadora selecciona la opción grupo y asignatura, en el menú.	1.1 Muestra una interfaz, solicitando el ingreso, del día, y la sección en el cual se desea obtener el informe de la actividad docente.
2. Ingresar el día correspondiente, la sección y hace clic en el botón aceptar.	2.1 Genera una interfaz, con el estado de la actividad docente, notificando el turno, el aula, correspondiente al grupo seleccionado. Terminando así el caso de uso
Prioridad	Secundario.

Tabla 3.12 Descripción del caso de uso “Obtener Estado de Actividad Docente”.

3.9.2.4 “Caso de uso Obtener Ubicación de Grupo”.

Nombre del Caso de Uso	Obtener Ubicación de grupo.
Actores	Planificadora.
Propósito	Tener conocimiento de la ubicación (local) de cada grupo en un momento dado.
Resumen	El caso de uso se inicia cuando la planificadora accede al sistema y solicita obtener la ubicación real de los grupos docentes, inmediatamente se visualiza una pantalla donde se muestra una lista de las diferentes aulas con sus respectivos grupos docentes.
Referencias	R6.3 y R6.4
Precondiciones	Que exista una asignación de locales a grupos.
Poscondiciones	Para los siguientes procesos: Obtener ubicación de grupo docente: Se retornan los valores referentes al atributo local de la instancia grupo.
Curso normal de los eventos	
Acciones del actor	Respuestas del Sistema
1. La planificadora decide obtener la ubicación de un determinado grupo.	1.1 El sistema muestra una interfaz en la cual se solicita la selección del grupo específico a localizar.
2. La planificadora selecciona el grupo y	2.1 El sistema genera una interfaz con

presiona el botón aceptar.	la localización exacta (local) en que se encuentra dicho grupo y finaliza así el caso de uso.
Curso alternativo	
Prioridad	Secundario.

Tabla 3.13 Descripción del caso de uso “Obtener ubicación Grupo”.

3.9.2.5 “Caso de uso Obtener Planificación Grupo”.

Nombre del Caso de Uso	Obtener Planificación grupo.
Actores	Planificadora.
Propósito	Tener conocimiento de la planificación (horario) de cada grupo.
Resumen	El caso de uso se inicia cuando la planificadora accede al sistema y solicita obtener la planificación de un grupo docente, inmediatamente se visualiza una pantalla donde se muestra la planificación correspondiente a un grupo docente.
Referencias	R10.
Precondiciones	Existan planificaciones hechas con anterioridad.
Poscondiciones	Para los siguientes procesos: Obtener planificación grupo docente: Se retornan los valores referentes a los atributos de la clase planificación para la instancia grupo seleccionada.

Curso normal de los eventos	
Acciones del actor	Respuestas del Sistema
1. La planificadora decide obtener la planificación de un determinado grupo.	1.1 El sistema muestra una interfaz en la cual se solicita la selección del grupo específico.
2. La planificadora selecciona el grupo y presiona el botón aceptar.	2.1 El sistema genera una interfaz con la planificación correspondiente al grupo seleccionado y finaliza así el caso de uso.
Curso alternativo	
Prioridad	Secundario.

Tabla 3.14 Descripción del caso de uso “Obtener Planificación Grupo”.

3.10 Conclusiones.

En este capítulo se comenzó a desarrollar la propuesta de solución, obteniéndose a partir del análisis de los procesos del negocio, un listado con las principales funcionalidades que debe tener el sistema y los requisitos adicionales, se representaron los diferentes paquetes que encierran los diagramas de casos de uso del sistema en conjunto a sus relaciones, y finalmente se describieron las acciones de los actores del sistema con los casos de uso con los que interactúan. Gracias a esto ahora se puede empezar a construir el sistema utilizando como guía todos los requerimientos y las funciones que han sido consideradas como necesarias en este capítulo.

Capítulo 4 Descripción de la solución propuesta.

4.1 Introducción.

Terminada la definición y descripción, en el anterior capítulo, de las funcionalidades deseadas y necesarias del sistema propuesto; se hace necesario definir cómo se desarrollará.

Este capítulo tiene el objetivo de plantear la concepción general del análisis y diseño del sistema propuesto y cómo se implementa éste. Así, se presentan los diagramas de clases, los diferentes diagramas de interacción (colaboración) del diseño que detallan la relación estructural entre los objetos que interactúan; se estructura la información que se desea que persista a través del diseño de la base de datos; se describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. Son también descritos los estándares de diseño y programación.

4.2 Principios del Análisis.

En el análisis podemos estructurar los requisitos de manera que nos facilite su comprensión, su preparación, su modificación y en general su mantenimiento.

En la construcción del modelo de análisis se tienen que identificar las clases que describen la realización de los casos de uso, los atributos y las relaciones entre ellas. Con esta información se construye el Diagrama de clases del análisis, que por lo general se descompone para agrupar las clases en paquetes. Esta descomposición tiene impacto por lo general en el diseño e implementación de la solución. [19]

4.2.1 Diagrama de clases del Análisis.

Un Diagrama de clases del análisis es un artefacto en el que se representan los conceptos en un dominio del problema. Representa las cosas del mundo real, no de la implementación automatizada de estas cosas [19].

A continuación se muestran los diferentes diagramas de clases del análisis correspondiente a los diferentes CU del sistema.

1. Diagrama de clases del Análisis correspondiente al CU Planificación.

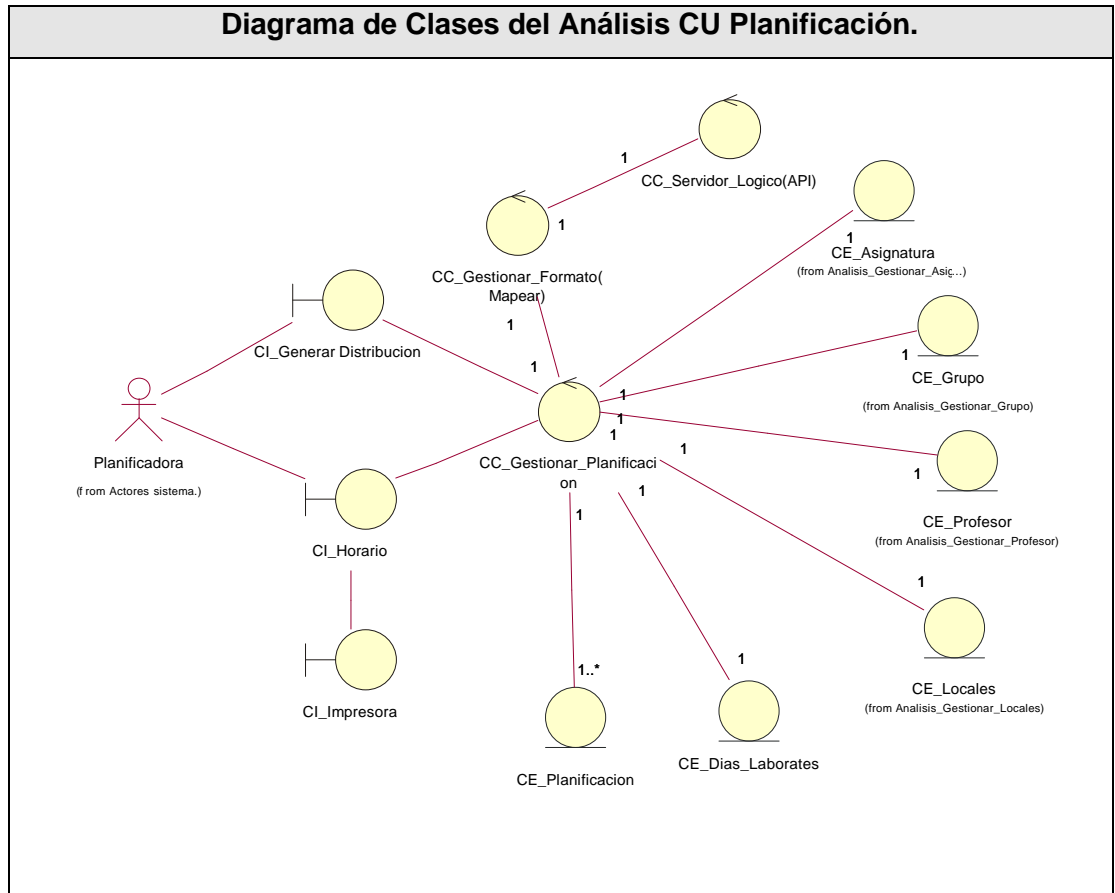


Tabla 4.1 Diagrama de clases del Análisis CU Planificación.

2. DCA correspondiente al CU Autenticar.....Anexo 1.
3. DCA correspondiente al CU Gestionar Asignatura.....Anexo 2.
4. DCA correspondiente al CU Gestionar Grupo.....Anexo 3.
5. DCA correspondiente al CU Gestionar Locales.....Anexo 4.
6. DCA correspondiente al CU Gestionar Profesor.....Anexo 5.
7. DCA correspondiente al CU Gestionar Especificación Docente.....Anexo 6.
8. DCA correspondiente al CU Gestionar Distribución Profesores grupo Asignatura..... Anexo 7
9. DCA correspondiente al CU Obtener Estado de Actividad.....Anexo 8.
10. DCA correspondiente al CU Locales Disponibles.....Anexo 9.
11. DCA correspondiente al CU Obtener Ubicación Profesor.....Anexo 10.

- 12. DCA correspondiente al CU Obtener Ubicación Grupo.....Anexo 11
- 13. DCA correspondiente al CU Obtener Planificación Grupo.....Anexo 12

4.3 Principios de diseño.

El modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso centrándose en como los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar.

Sirve de abstracción de la implementación y es utilizada como entrada fundamental de las actividades de implementación. [20]

4.3.1 Diagramas de Clases del Diseño.

Un diagrama de clases del diseño es una representación más concreta que el diagrama de clases del análisis. Representa la parte estática del sistema así como las clases y sus relaciones.

A continuación se muestran los diferentes diagramas de clases del Diseño correspondiente a los diferentes CU del sistema.

1. Diagrama de clases del Diseño correspondiente al CU Planificación.

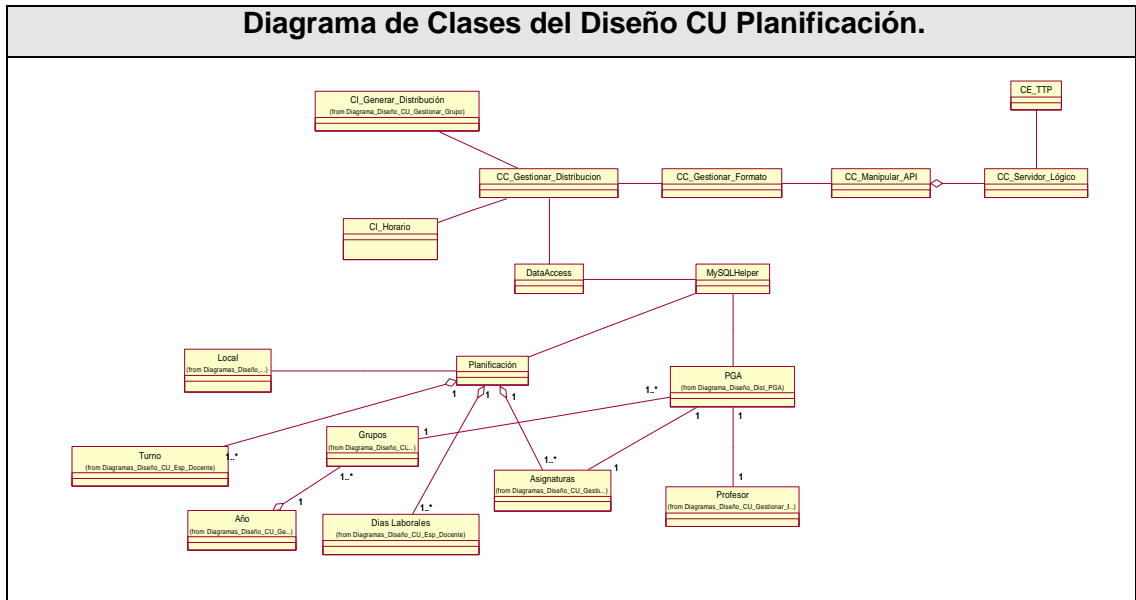


Tabla 4.2 Diagrama de clases del Diseño CU Planificación.

2. DCD correspondiente al CU Autenticar.....	Anexo 1.
3. DCD correspondiente al CU Gestionar Asignatura.....	Anexo 2.
4. DCD correspondiente al CU Gestionar Grupo.....	Anexo 3.
5. DCD correspondiente al CU Gestionar Locales.....	Anexo 4.
6. DCD correspondiente al CU Gestionar Profesor.....	Anexo 5.
7. DCD correspondiente al CU Gestionar Especificación Docente.....	Anexo 6.
8. DCD correspondiente al CU Gestionar Distribución Profesores grupo Asignatura.....	Anexo 7
9. DCD correspondiente al CU Obtener Estado de Actividad.....	Anexo 8.
10. DCD correspondiente al CU Locales Disponibles.....	Anexo 9.
11. DCD correspondiente al CU Obtener Ubicación Profesor.....	Anexo 10.
12. DCD correspondiente al CU Obtener Ubicación Grupo.....	Anexo 11
13. DCA correspondiente al CU Obtener Planificación Grupo.....	Anexo 12

4.3.3 Diagrama de Interacción (Colaboración).

Los diagramas de interacción son diagramas que modelan aspectos dinámicos del sistema.

Un diagrama de Colaboración es una forma alternativa al diagrama de secuencia de mostrar un escenario. Este tipo de diagrama muestra las interacciones entre objetos organizadas entorno a los objetos y los enlaces entre ellos. [21]

A continuación se muestran los diferentes diagramas de Colaboración correspondiente a los diferentes CU del sistema.

1. Diagrama de Colaboración correspondiente al CU Planificación.

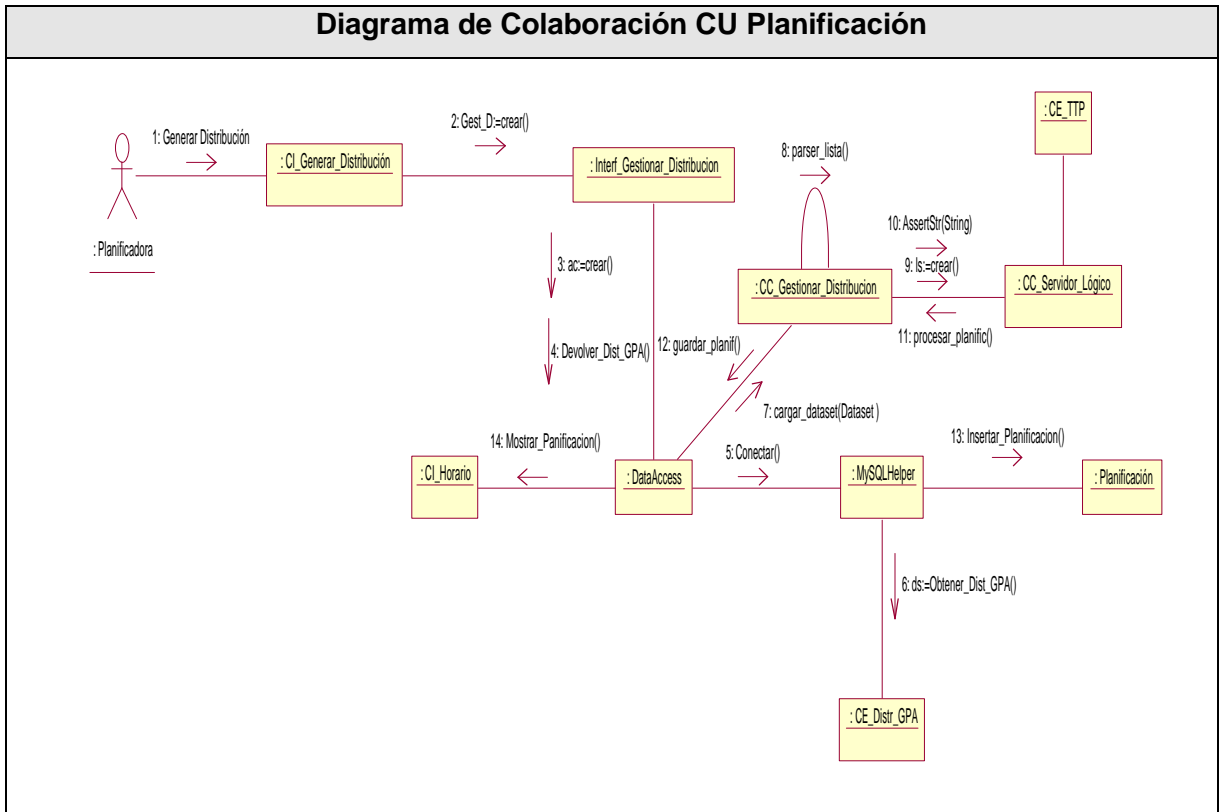


Tabla 4.3 Diagrama de Colaboración correspondiente al CU Planificación.

- 2. DCO correspondiente al CU Autenticar.....Anexo 1.
- 3. DCO correspondiente al CU Gestionar Asignatura.....Anexo 2.
- 4. DCO correspondiente al CU Gestionar Grupo.....Anexo 3.
- 5. DCO correspondiente al CU Gestionar Locales.....Anexo 4.
- 6. DCO correspondiente al CU Gestionar Profesor.....Anexo 5.
- 7. DCO correspondiente al CU G Especificación Docente.....Anexo 6.
- 8. DCO correspondiente CU G Distribución Profesor Grupo Asignatura
.....Anexo7.
- 9. DCO correspondiente al CU Obtener Estado de Actividad.....Anexo8.
- 10. DCO correspondiente al CU Locales Disponibles.....Anexo 9.
- 11. DCO correspondiente CU Obtener Ubicación Profesor.....Anexo 10.
- 12. DCO correspondiente CU Obtener Ubicación Grupo.....Anexo 11.
- 13. DCO correspondiente CU Obtener Planificación Grupo.....Anexo 12.

4.4 Diseño de la Base de Datos.

A continuación se muestran las tablas referentes al modelo lógico de datos y al modelo físico.

4.4.1 Modelo lógico de datos (diagrama de clases persistentes).

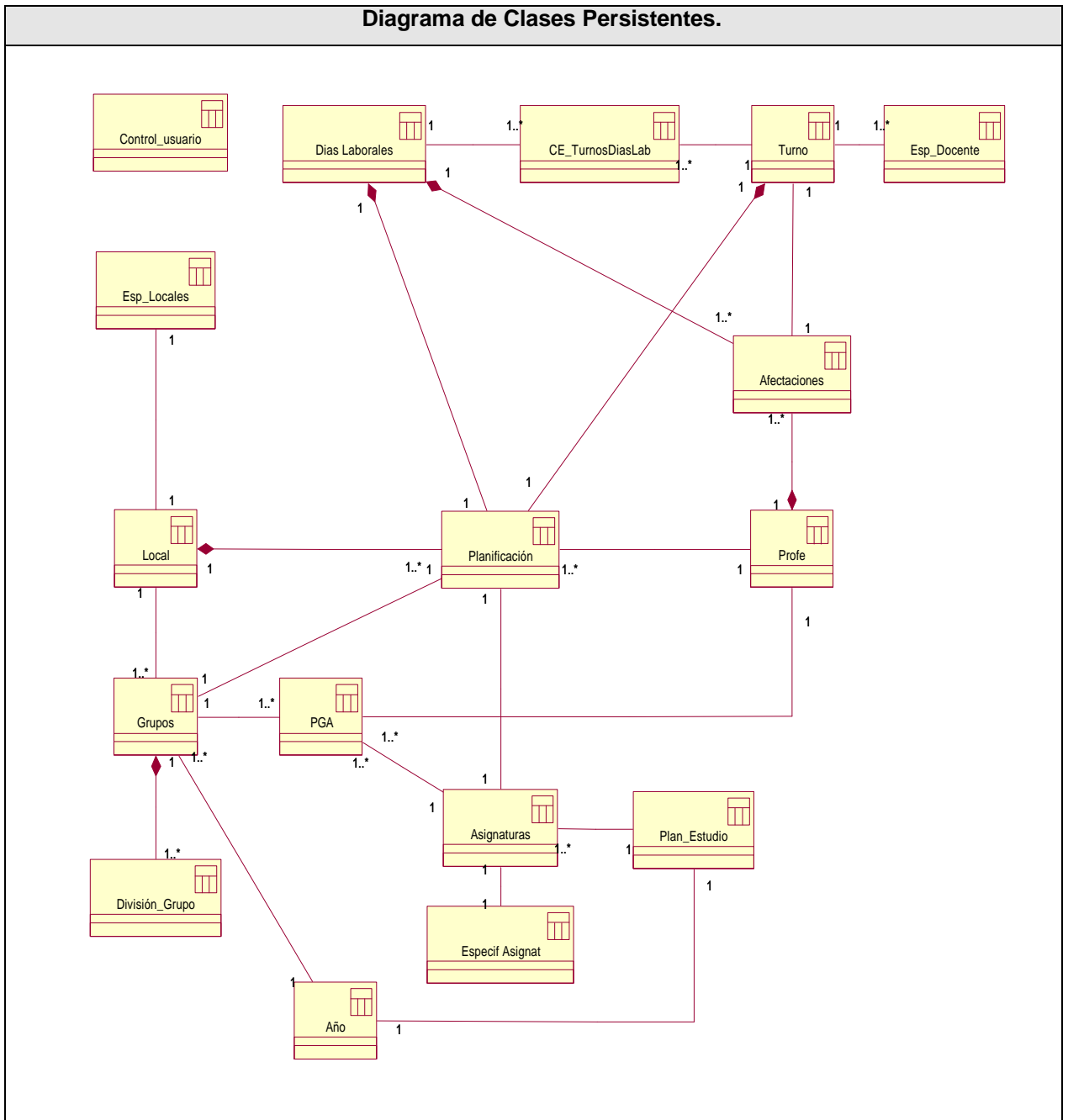


Tabla 4.4 Diagrama de Clases Persistentes.

4.4.2 Modelo físico de datos (modelo de datos).

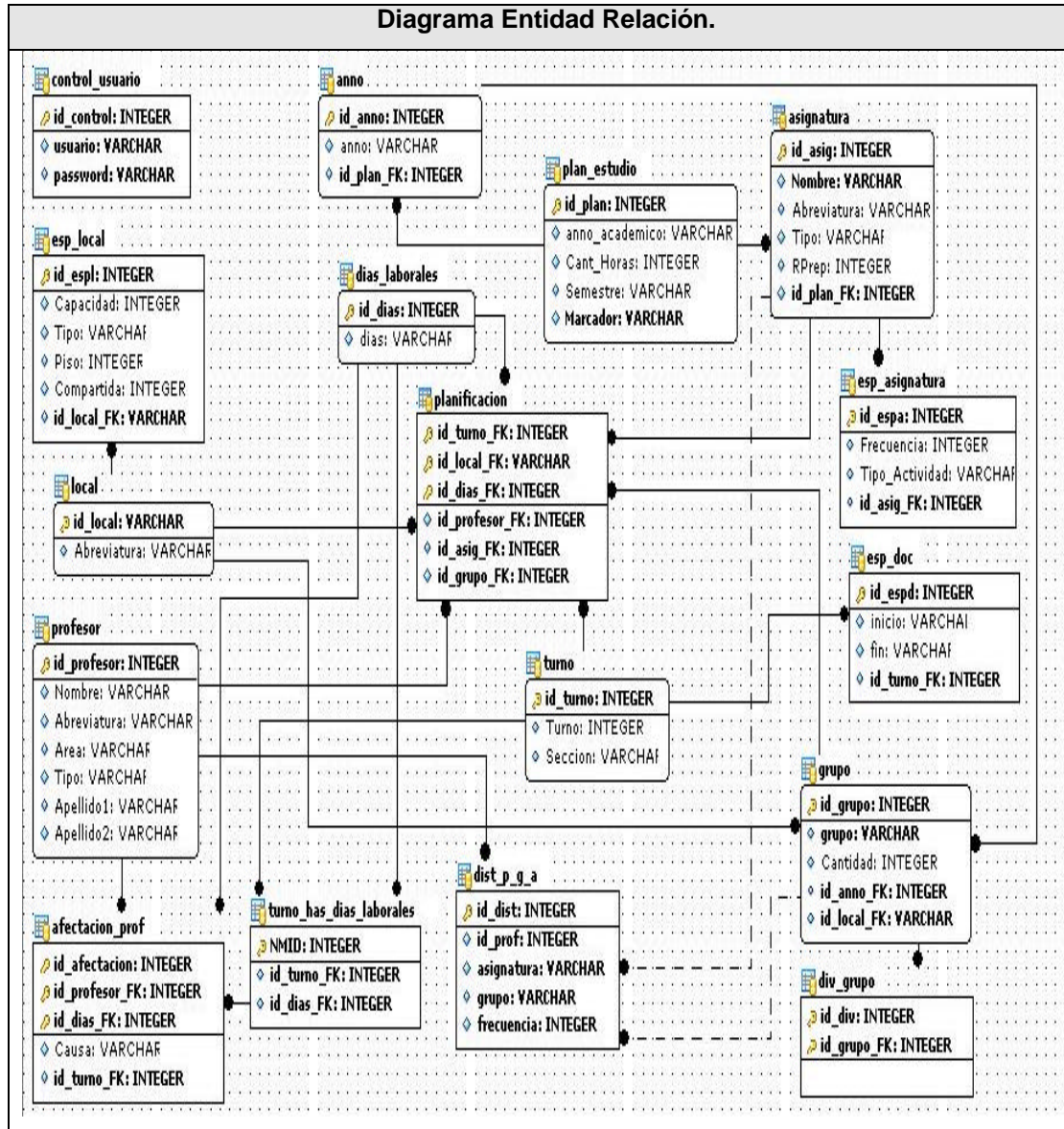


Tabla 4.5 Diagrama Entidad Relación.

6.5 Diagrama de Despliegue.

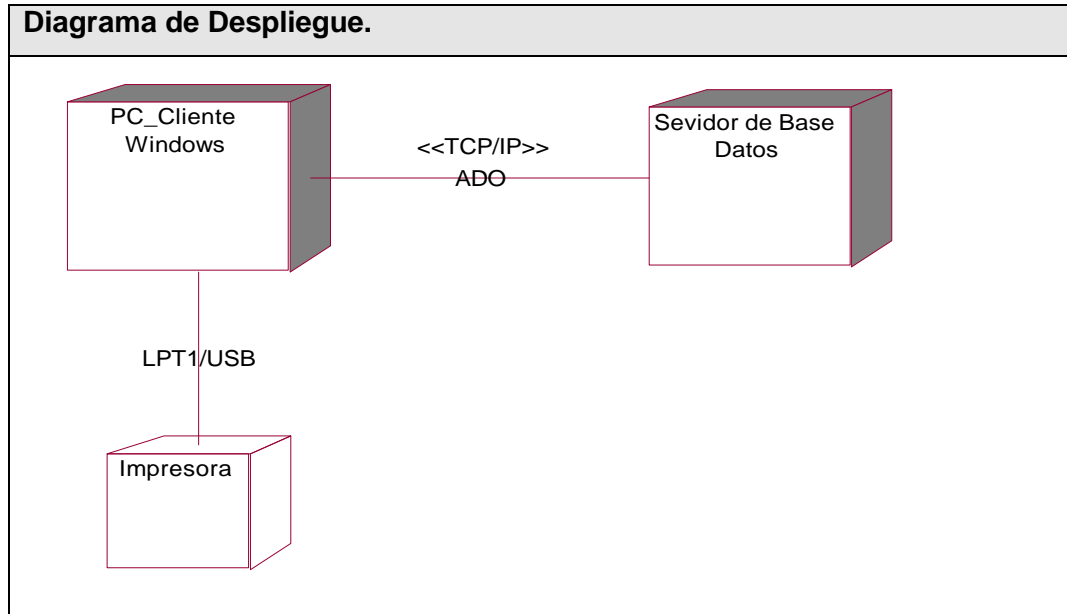


Tabla 4.6 Diagrama de Despliegue.

6.6 Diagrama de Componentes.

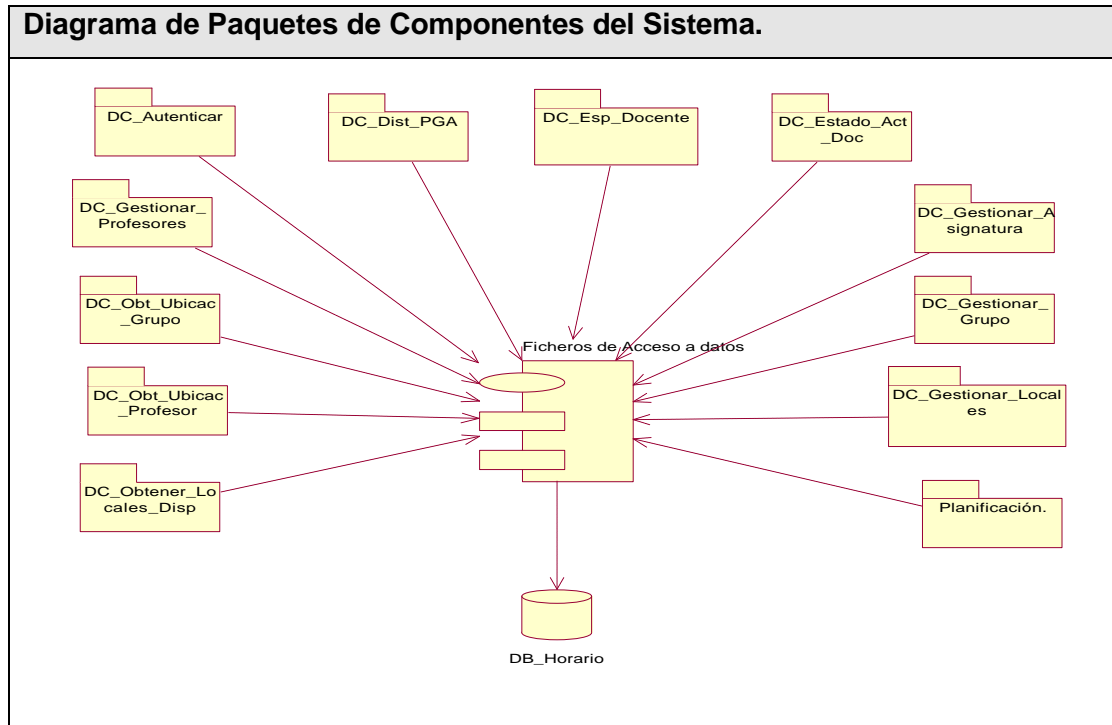


Tabla 4.7 Diagrama de Paquetes de Componentes del Sistema.

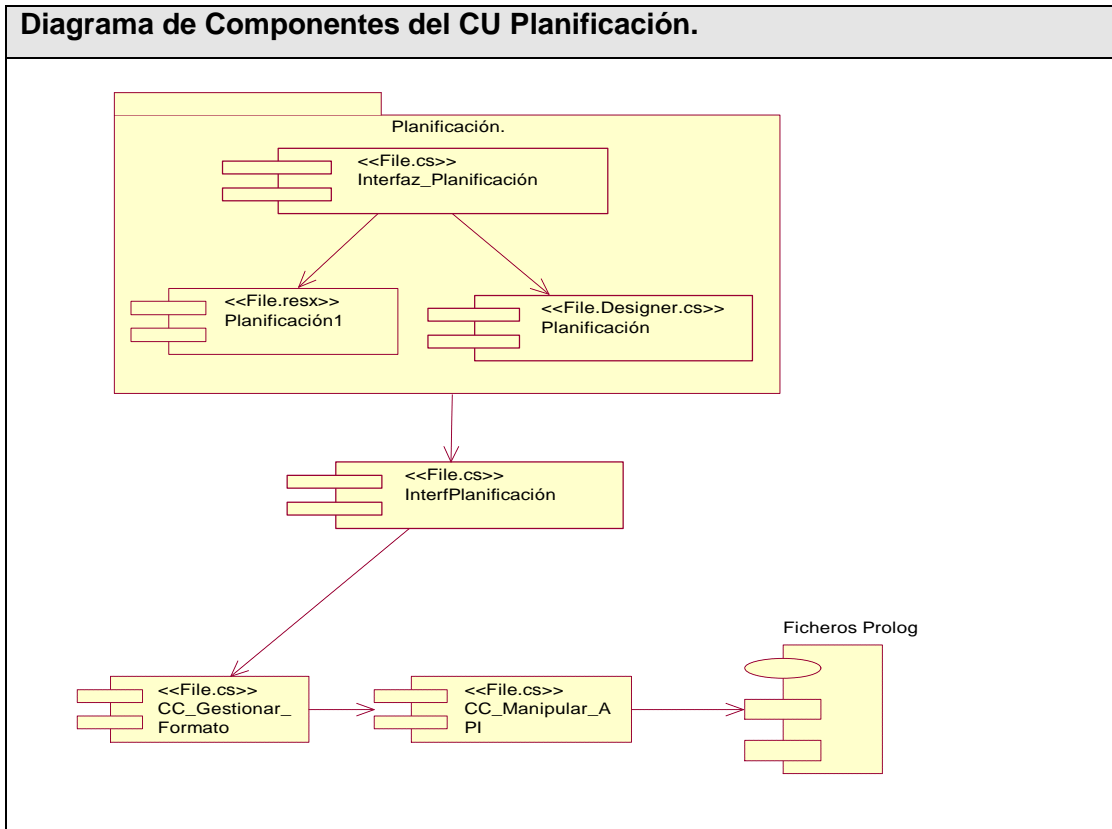


Tabla 4.8 Diagrama de Componentes CU Planificación.

2. DC correspondiente al CU Autenticar.....Anexo 1.
3. DC correspondiente al CU Gestionar Asignatura.....Anexo 2.
4. DC correspondiente al CU Gestionar Grupo.....Anexo 3.
5. DC correspondiente al CU Gestionar Locales.....Anexo 4.
6. DC correspondiente al CU Gestionar Profesor.....Anexo 5.
7. DC correspondiente al CU G Especificación Docente.....Anexo 6.
8. DC correspondiente CU G Distribución Profesor Grupo Asignatura
.....Anexo 7.
9. DC correspondiente al CU Obtener Estado de Actividad....Anexo 8.
10. DC correspondiente al CU Locales Disponibles.....Anexo 9.
11. DC correspondiente CU Obtener Ubicación Profesor.....Anexo 10.
12. DC correspondiente CU Obtener Ubicación Grupo.....Anexo 11.
13. DC correspondiente CU Obtener Planificación Grupo.....Anexo 12.

4.7 Conclusiones

En el presente capítulo se desarrollaron los diferentes diagramas de clases del análisis y del diseño, incluyendo los principios de diseño seguidos, específicamente, los temas de estándares de la interfaz, también se muestran los diferentes diagramas de interacción correspondientes a los distintos Casos de Uso del Sistema que conforman la aplicación. Se desarrolló un diseño de la base de datos del sistema así como los diferentes diagramas de componentes propios del sistema.

Capítulo 5 Estudio de factibilidad

5.1 Introducción

Para un proyecto es de vital importancia realizar un análisis de su viabilidad desde el punto de vista de costos y beneficios. Obteniéndole variables que posibiliten inferir si el proyecto es viable, los recursos humanos que serán necesarios, así como el tiempo de desarrollo requerido.

El capítulo que se describe a continuación describe la estimación de costos del sistema propuesto y sus beneficios.

5.2 Planificación basada en Caso de Uso.

Paso 1. Identificar los Puntos de casos de uso Desajustados

$$UUCP = UAW + UUCW.$$

Donde:

UUCP: Puntos de Casos de Uso sin ajustar

UAW: Factor de Peso de los Actores sin ajustar

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

- Para calcular UAW

Tipo	Descripción	Peso	Cant *peso
Simple	Otro sistema que interactúa con el sistema a desarrollar mediante una interfaz de programación (API, Application Programming Interface AMZI).	1	1*1
Medio	Otro sistema que interactúa con el sistema a desarrollar mediante un protocolo o una interfaz basada en texto	2	0*2
Complejo	Una persona que interactúa con el sistema mediante una interfaz gráfica.	3	8*3
Total			25

- Para calcular UUCW

Tipo	Descripción	Peso	Cant * peso
Simple	El Caso de Uso contiene de 1 a 3 transacciones	5	7*5
Medio	El Caso de Uso contiene de 4 a 7 transacciones	10	4*10
Complejo	El Caso de Uso contiene más de 8 transacciones	15	2*15
Total			105

Luego:

$$\text{UUCP} = 25 + 105 = 130.$$

Paso 2. Ajustar los puntos de caso de uso.

Ajustar los puntos de casos de uso.

$$\text{UCP} = \text{UUCP} * \text{TCF} * \text{EF}.$$

Donde:

UCP: Puntos de Casos de Uso ajustados.

UUCP: Puntos de Casos de Uso sin ajustar.

TCF: Factor de complejidad técnica.

EF: Factor de ambiente.

El factor de complejidad técnica (TCF) se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema. Cada factor se cuantifica en un valor desde 0 (aporte irrelevante) hasta 5 (aporte muy relevante).

Para Calcular TCF

$$\text{TCF} = 0.6 + 0.01 * S (\text{Peso}_i * \text{Valor}_i) \text{ (Donde Valor es un número del 0 al 5)}$$

Significado de los valores

0: No presente o sin influencia.

1: Influencia incidental o presencia incidental.

2: Influencia moderada o presencia moderada.

3: Influencia media o presencia media.

4: Influencia significativa o presencia significativa.

5: Fuerte influencia o fuerte presencia.

Factor	Descripción	Peso	Valor	Comentario	S (Peso_i * Valor_i)
T1	Sistema distribuido.	2	0	El sistema es centralizado.	0
T2	Objetivos de performance o tiempo de respuesta.	1	4	La velocidad es limitada por las entradas provistas por el usuario	4
T3	Eficiencia del usuario final.	1	4	Escasas restricciones de Eficiencia.	4
T4	Procesamiento interno Complejo.	1	5	No hay cálculos Complejos.	5
T5	El código debe ser reutilizable	1	2	No se requiere que el código sea reutilizable	2
T6	Facilidad de instalación.	0.5	3	Escasos requerimientos de facilidad de instalación.	1.5
T7	Facilidad de uso.	0.5	4	Normal.	2
T8	Portabilidad.	2	2	No se requiere que el sistema sea portable.	4
T9	Facilidad de cambio.	1	2	Se requiere un costo moderado de mantenimiento.	2
T10	Concurrencia.	1	0	No hay Concurrencia.	0
T11	Incluye objetivos especiales de seguridad.	1	2	Seguridad normal.	2
T12	Provee acceso directo a terceras partes.	1	4	Los usuarios web tienen acceso directo.	4
T13	Se requieren facilidades	1	2	Pocos usuarios	2

	especiales de entrenamiento a los usuarios.			internos, sistema fácil de usar.	
Total					32.5

$$\text{TCF} = 0.6 + 0.01 * 32.5 = 0.925$$

Para Calcular EF

El factor de ambiente (EF) está relacionado con las habilidades y entrenamiento del grupo de desarrollo que realiza el sistema. Cada factor se cuantifica con un valor desde 0 (aporte irrelevante) hasta 5 (aporte muy relevante).

$$\text{EF} = 1.4 - 0.03 * S (\text{Peso}_i * \text{Valor}_i) \text{ (Donde Valor es un número del 0 al 5)}$$

Factor	Descripción	Peso	Valor	Comentario	S (Peso_i * Valor_i)
E1	Familiaridad con el modelo de proyecto utilizado.	1.5	3	El grupo está bastante familiarizado con el modelo.	4.5
E2	Experiencia en la aplicación.	0.5	3	La mayoría del grupo ha trabajado mucho tiempo en ésta aplicación.	1.5
E3	Experiencia en orientación a objetos.	1	4	La mayoría del grupo programa en Objetos.	4
E4	Capacidad del analista líder.	0.5	2	Se contrató a un Especialista.	1
E5	Motivación.	1	5	El grupo está altamente motivado.	5
E6	Estabilidad de los requerimientos.	2	4	Se esperan cambios.	8
E7	Personal part-time.	-1	0	Todo el grupo es full-time.	0
E8	Dificultad del lenguaje de	-1	1	Se usará lenguaje.	-1

	programación.			C#	
Total					23

$$EF = 1.4 - 0.03 * 23 = 0.71$$

Luego:

$$UCP = 130 * 0.925 * 0.71 = 85.37.$$

Paso 3. Calcular esfuerzo de FT Implementación

$$E = UCP * CF.$$

Donde

E: esfuerzo estimado en horas-hombre.

UCP: Puntos de Casos de Uso ajustados.

CF: Factor de Conversión.

Para obtener el factor de conversión (CF) se cuentan cuantos valores de los que afectan el factor ambiente (E1...E6) están por debajo de la media (3), y los que están por arriba de la media para los restantes (E7, E8). Si el total es 2 o menos se utiliza el factor de conversión 20 Horas-Hombre / Punto de Casos de uso. Si el total es 3 o 4 se utiliza el factor de conversión 28 Horas-Hombre / Punto de Casos de uso. Si el total es mayor o igual que 5 se recomienda efectuar cambios en el proyecto ya que se considera que el riesgo de fracaso del mismo es demasiado alto.

- Para calcular CF.

CF = 20 horas-hombre (si Total_{EF} = 2)

CF = 28 horas-hombre (si Total_{EF} = 3 ó Total_{EF} = 4)

CF = abandonar o cambiar proyecto (si Total_{EF} = 5)

$$\text{Total}_{EF} = \text{Cant } EF < 3 \text{ (entre E1 -E6)} + \text{Cant } EF > 3 \text{ (entre E7, E8)}$$

Como:

$$\text{Total}_{EF} = 2 + 0 = 2$$

Entonces:

CF = 20 horas-hombre (porque Total_{EF} = 2)

Luego:

$$E = 85.37 * 20 = 1708 \text{ horas-hombre.}$$

Paso 4. Calcular esfuerzo de todo el proyecto

Actividad	% esfuerzo	Valor esfuerzo
Análisis.	10%	427 horas-hombre.
Diseño.	20%	854 horas-hombre.
Implementación.	40%	1708 horas-hombre.
Prueba.	15%	640 horas-hombre.
Sobrecarga.	15%	640 horas-hombre.
Total	100%	4269 horas-hombre.

Suponiendo que una persona trabaje 8 horas por día, y un mes tiene como promedio 30 días; la cantidad de horas que puede trabajar una persona en 1 mes es 240 horas.

Si $E_T = 4269$ horas-hombre y por cada 240 horas yo tengo 1 mes eso daría un $E_T = 18$ mes-hombre.

Esto quiere decir que 1 persona puede realizar el problema analizado en aproximadamente 1 año y medio (18 meses).

Si en el equipo hay 2 personas y todas realizan el mismo esfuerzo entonces el problema analizado puede terminarse en aproximadamente 9 meses.

Costo del Proyecto.

Se asume como salario promedio mensual: \$ 50.00.

Cantidad de Hombres (CH): 2 personas.

$CHM = 2 * \text{Salario Promedio.}$

$CHM = 100.00$ \$/Mes.

$\text{Costo} = CHM * E_T = 100 * 18 = \$1800.$

5.3 Beneficios tangibles e intangibles.

El sistema propuesto Asistente de ayuda para la confección de horarios docentes no es un producto comercial, su principal objetivo es solventar los problemas relacionados al proceso de planificación, existente en la secretaria docente.

El beneficio fundamental del sistema es proveer una herramienta interactiva (semi-automática), que simplifique la labor del planificador. Utilizando una aplicación de escritorio flexible, de fácil uso y de interfaz amigable que permita registrar, actualizar toda la información necesaria en el proceso de planificación, obtener información de interés del estado del proceso docente, y que de cierta forma brinde cierta flexibilidad al proceso de planificación minimizando el impacto ante cambios de última hora.

Por tanto los, beneficios inmediatos son generalmente intangibles.

- ? Disminuye el tiempo y los recursos invertidos para cumplimentar la tarea, que se realiza de forma manual.
- ? Fácil y rápido acceso y publicación de la planificación obtenida.
- ? Fácil procesamiento de la información y obtención dinámica de reportes, de la situación docente en un momento dado.

5.4 Análisis de costos y beneficios.

El proceso de desarrollo de software lleva implícito un costo asociado, de aquí se deriva que es vital justificar su desarrollo analizando principalmente los beneficios que reportarían su implementación y utilización. Los beneficios pueden ser económicos y de orden social, estos últimos tienen tanta importancia como los primeros. El sistema que se propone esta dirigido principalmente al sector educacional, por lo que podemos concluir que su beneficio es de orden social.

Una vez implantado el sistema contribuirá a aumentar la eficiencia del proceso de planificación, disminuyendo el tiempo invertido en el proceso. Además posibilita gestionar la compleja información que esta involucrada en el proceso de planificación generando informes del estado de la actividad docente en un momento dado. La tecnología utilizado en el desarrollo del sistema principalmente el IDE utilizado no es libre, esta sujeto a restricciones comerciales, por lo que lleva implícito una inversión adicional por concepto de patentes, aunque esta solo se limita al uso del entorno de desarrollo integrado (IDE), pero la especificación del framework sobre la cual se desarrolló, no esta sujeta a restricciones comerciales, por lo que se puede migrar a otras plataformas libres, minimizando el impacto el cambio en la lógica de la aplicación. Esto esta condicionado por la estructuración

en capas de los procesos de negocio que se diseñaron, solo haciéndose extensible a la interfaz gráfica de usuario (GUI).

Analizando el costo del proyecto, los numerosos beneficios que reporta, detallados con anterioridad, se puede concluir que su implementación es realmente factible.

5.5 Conclusiones.

En este capítulo se describió el estudio de la factibilidad con relación al sistema propuesto, tomando en consideración el costo estimado y los beneficios que reportará al ser implementado.

La herramienta propuesta reportará beneficios significativos e importantes en el proceso de planificación en la Universidad de Ciencias Informáticas, al contribuir a mejorar el proceso de gestionar el horario docente, y de controlar su estado en un momento dado, lo cual indica que es factible implementar la herramienta propuesta.

Conclusiones.

A modo de conclusiones se espera que el documento haya servido para la comprensión teórica de la situación problemática existente y su solución, así como el desarrollo de las diferentes etapas de la aplicación usando la metodología RUP.

Se alcanzó, el objetivo propuesto: (Realizar el análisis y el diseño del sistema así como un prototipo no funcional del mismo.) reafirmando la utilidad y validez de emplear las tecnologías informáticas para apoyar las labores que se desarrollan en cualquier tipo de esfera. Se obtuvieron además los siguientes resultados:

- ? Se demostró la eficacia de los lenguajes y tecnologías utilizadas para el desarrollo del prototipo del sistema.
- ? Se realizó una base de datos, donde queda almacenada toda la información necesaria posible a ser manipulada por el sistema para su correcto funcionamiento.
- ? Se realizó la implementación del módulo de especificación y entrada de datos del sistema así como el módulo de inteligencia artificial.
- ? Se han seguido los principios básicos de diseño descritos para el desarrollo del sistema.

Recomendaciones.

Se recomienda:

1. Cumplimentar la integración del módulo de inteligencia artificial con el módulo de gestión de configuración ó entrada de datos.
2. Continuar en el estudio de este tipo de aplicación con el fin de añadir mejoras funcionales enfocadas al beneficio del usuario.
3. Profundizar en el estudio, de algoritmos clásicos utilizados en la resolución de problemas de horario, con el objetivo de añadir a la solución propuesta una mayor flexibilidad en la manipulación de las restricciones involucradas.
4. Implementar futuras versiones del sistema utilizando software libre, eliminado cualquier tipo de restricciones de licencia involucradas a las herramientas de desarrollo.
5. Implementar el sistema utilizando un lenguaje de alto nivel preferiblemente multiplataforma, independiente de un gestor de base de datos, posibilitando su portabilidad a otras plataformas, y facilitando el proceso instalación.
6. Hacer un estudio comparativo sobre los lenguajes utilizados y las formas de enfocar el problema.

Referencias Bibliográficas.

- [1]. Gestión de Horarios. Volume. Disponible en
< <http://www.algoritmica.com.mx/Docs/folletohorariov1.pdf> >.
- [2]. ASC Horarios. Volume, Disponible en
< <http://www.abcdatos.com/programas/programa/l8252.html> >.
- [3]. Generador de Horarios para centros docentes. Volume, Disponible en
<<http://www.abcdatos.com/programas/programa/l4228.html>>.
- [4]. Timetab. Volume. Disponible en
<<http://www.abcdatos.com/programas/programa/l1382.html>>.
- [5]. Mimosa for Windows. Volume, Disponible en
<<http://www.nielsoft.com/software/mimosa.html>>.
- [6]. Marcos Gil Tavallo, A.I.A.M.I. Algoritmo Basado en TABU SEARCH para el Problema de Asignación de Horarios de Clases. Volume, Disponible en
<<http://servicio.cid.uc.edu.ve/facyt/v1n1/1-1-8.pdf>>.
- [7]. Problemas de Horarios. Volume, Disponible en
< <http://incentivos-spu.me.gov.ar/proy1997/01/proy/h01x210.html>>.
- [8]. Resumen de las características de C#. Volume, Disponible en
<http://msdn.microsoft.com/library/spa/default.asp?url=/library/SPA/cscon/html/vclrfAQuickSurveyOfCSharpFeatures_PG.asp >.
- [9]. López, Á. Patrones de Arquitectura de aplicaciones. Patrón. Volume, Disponible en
en <<http://www.inta.gov.ar/pro/radar/actividad/transformacion/p/patron.htm>>.
- [10]. (2004) Sistema Gestor de base de datos SGBD. Volume, Disponible en
<http://www.error500.net/garbagecollector/bases_de_datos/sistema_gestor_de_

base_de_dato.html >.

- [11]. Las principales características de MySQL. Volume, Disponible en < <http://dev.mysql.com/doc/refman/5.0/es/features.html> >.
- [12]. Patrones de Diseño. Conferencia 7 de Ingeniería del Software I, curso 2005-2006, UCI.
- [13]. Sea ud su propio experto Mysql para sus sistemas de Base de Datos. Volume, Disponible en <<http://www.blog.delimce.com/index.php?itemid=39>>.
- [14]. González, C. Módulo Alojamiento del Sistema Automatizado para la Gestión de Información de la Misión Milagro. Trabajo de Diploma para optar por el título de Ingeniero Informático, Instituto Superior Politécnico “José Antonio Echeverría”, Ciudad de la Habana, junio 2006.
- [15] ¿Qué es una aplicación a ventanas? Volume, Disponible en <http://www.htmlpoint.com/guidajava/java_17.htm>.
- [16]. Arquitectura de Capas. Volume, Disponible en <<http://www ldc.usb.ve/~teruel/ci3715/clases/arqCapas.html>>.
- [17]. Larman, Craig. “UML y Patrones”. 1999. Prentice Hall Hispanoamericana.
- [18]. Aplicando UML y Patrones. Prentice-Hall, 2003, 2da Edición, Craig Larman.
- [19]. Conferencia 6 Flujo de Análisis. Modelo de Análisis. RUMBAUGH, James, JACOBSON, Ivar; BOOCH, Grady, “El lenguaje unificado de modelado. Manual de referencia”.2000. Addison Wesley. Capítulos 4.
- [20]. JACOBSON, Ivar; RUMBAUGH, James; BOOCH, Grady, “El proceso unificado de desarrollo”.2000. Addison Wesley. Capítulos 9 Páginas 205-254.
- [21]. “El Proceso Unificado de Software”. Jacobson, Booch, Rumbaugh, Addison – Wesley / Object Technology Series. 2000. Capítulo 9.

Bibliografía.

Algoritmos genéticos. Disponible en

<http://es.wikipedia.org/wiki/Algoritmo_gen%C3%A9tico>.

Algoritmos genéticos y computación evolutiva. Adam Marczyk.2004

<<http://the-geek.org/docs/algen/>>.

Colonia de Hormigas. Disponible en

<<http://delta.cs.cinvestav.mx/~ccoello/tesis/tesis-tellez.pdf.gz>>.

Fundamentos de la Teoría de Grafos. Disponible en

<<http://www.sol.com/algoritmos/pagina03.asp>>.

Introducción a los Algoritmos Genéticos. Disponible en

<<http://www.redcientifica.com/doc/doc199904260011.html><28/3/2007>.>

MySQL. Disponible en

<<http://dev.mysql.com/doc/refman/5.0/es/history.html>>

<<http://dev.mysql.com/doc/refman/5.0/es/features.html>>

Postgres SQL. Disponible en

<<http://es.wikipedia.org/wiki/PostgreSQL>>

Rup. Disponible en

<http://es.wikipedia.org/wiki/Rational_Unified_Process>.

SQL Server. Disponible en

<http://es.wikipedia.org/wiki/SQL_Server>.

Tabú Search. Disponible en

<<http://servicio.cid.uc.edu.ve/facyt/v1n1/1-1-8.pdf>>.

Un estudio para automatizar Horarios. Toma de A Survey of Automated Timetabling. Andrea Schaerf, pág. 35

Uml. Disponible en

<<http://es.tldp.org/Tutoriales/doc-modelado-sistemas-UML/multiple-html/c12.html>>.

<<http://www.osmosislatina.com/lenguajes/uml/basico.htm>>.

Vértice-Coloración. Disponible en

<<http://www.sol.com/algoritmos/pagina02.asp>>.

GLOSARIO DE TÉRMINOS Y SIGLAS.

1. COCOMO: Modelo para la estimación de costos de productos informáticos.
2. CUN: Caso de uso del negocio.
3. CUS: Caso de uso del sistema.
4. MYSQL: Es un sistema de gestión de bases de datos relacional.
5. RUP: (Rational Unified Process) Proceso Unificado de desarrollo. Metodología para el desarrollo de Software.
6. UML: (Unified Modeling Language). Es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema de software.
7. NP COMPLETO: Es el subconjunto de los problemas de decisión en NP tal que todo problema en NP se puede reducir en cada uno de los problemas de NP-completo.
8. METAHEURISTICAS: Pueden concebirse como estrategias generales de diseño de procedimientos heurísticos para la resolución de problemas con un alto rendimiento.
9. ALGORITMOS GENETICOS: El algoritmo genético es una técnica de búsqueda basada en la teoría de la evolución de Darwin.
10. TABU SEARCH: Es una metaheurística diseñada para resolver problemas de optimización combinatoria, que está basada en principios generales de Inteligencia Artificial.
11. VERTICE-COLORACION: Algoritmo usado en la resolución de problemas de horarios.
12. COLONIA DE HORMIGAS: Una técnica relativamente reciente que ha demostrado tener ventajas frente a otras técnicas del área en la resolución de problemas de optimización combinatoria.
13. DCA: Diagrama de Clases del Análisis.
14. DCD: Diagrama de Clases del Diseño.
15. DCO: Diagrama de Colaboración.
16. DC: Diagrama de Componentes.

17. IA: Se denomina inteligencia artificial a la ciencia que intenta la creación de programas para máquinas que imiten el comportamiento y la comprensión humana.