



# Sistema automatizado para la **gestión** de **horarios docentes**

## **Autores**

Aneli Valdés Acosta  
María de Dolores Guardia Macías

## **Tutor**

Ing. Michel Mora Lozano

## **Cotutor**

Ing. Yoselyn Castillo Viera

Ciudad de la Habana, mayo 30 | 2007 "Año 49 de la Revolución"



**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN CIENCIAS INFORMÁTICAS**

***“Nunca será tarde para buscar un mundo mejor y más nuevo,  
si en el empeño ponemos coraje y esperanza”.***

*Alfred Tennyson*

## AGRADECIMIENTOS

- *Agradecemos a todas las personas que de una forma u otra han compartido con nosotras alegrías y tristezas en estos cinco años inolvidables, a todos aquellos compañeros de madrugadas de estudio, de travesuras, de fiestas..., a todos y cada uno de ellos gracias por tan lindos recuerdos de la universidad.*
- *Agradecemos a UCITeVe, nuestro eterno grupo, por tantas cosas que hemos vivido juntos.*
- *Agradecemos a Indira y Thompson por su gran ayuda y el apoyo brindado. Mil gracias por todo.*
- *Agradecemos de todo corazón a Murphy. Él sabe por qué.*

De Aneli:

- *Mi mayor agradecimiento es para “Lolita Guardia”. Gracias por aguantar mis locuras y compartir conmigo tantos días de trabajo y sacrificio. No podía haber encontrado una mejor compañera de Tesis.*
- *A las muchachitas por su paciencia conmigo y su apoyo incondicional en todos los momentos. Gracias por esa amistad tan sincera que me han demostrado estos cinco años y que estoy segura no terminará aquí.*
- *A todas mis amistades, gracias por haber compartido conmigo tantos momentos buenos y malos, a los chicos de UCITeVe, ustedes son lo mejor.*

De Maria de Dolores:

- *Mi agradecimiento sincero para Aneli, mi compañera de tesis, gracias por tu paciencia, por aguantar mis peleas y compartir conmigo todos los esfuerzos y días de trabajo para lograr construir este trabajo de diploma.*
- *Agradezco a mi tía Isabel y mi tío Jaimen por apoyarme estos 5 años y estar al lado mío en todo momento.*

- *Agradezco a las niñas por compartir conmigo la estancia en la escuela, gracias por brindarme su amistad.*
- *Agradezco a Carlos Javier por apoyarme y estar ahí en los momentos que más lo necesitaba.*
- *Agradezco a todos mis amigos que hicieron que mis años en la universidad fueran lindos e hicieron que yo fuera mejor cada día, gracias a todos.*
- *Agradezco a todas las personas que me han ayudado y han estado pendiente de mí durante estos años.*

## DEDICATORIA

De María de Dolores:

*Quiero dedicar mi trabajo a mi familia, a ustedes mami y papi por ser mi guía, por ayudarme y apoyarme y por hacer de mí la persona que soy hoy. A ti mi hermanito por ser mi amigo y por estar conmigo para cualquier cosa, todo lo que soy es gracias a ustedes, a los tres los quiero mucho. Gracias por ayudarme y por darme su cariño.*

*También quiero dedicar mi trabajo a una persona muy especial que por cosas de la vida ya no la tengo a mi lado, a ti "Mami Lupe" te dedico este trabajo, aunque no estés siempre te llevo conmigo, te quiero mucho y que Dios te tenga en la Gloria.*

De Aneli:

*Quiero dedicar mi trabajo a las tres personas más importantes de mi vida, a Mami, Papi y Ñaña, deben saber que todo cuanto soy y he podido hacer se los debo a ustedes, a su inmenso cariño y confianza en mí. Gracias por quererme tanto y estar siempre conmigo.*

*Lo dedico además a abuela Orquídea, sé cuanto has soñado con este momento, y cuanta confianza tienes en mí. Gracias por tanto cariño.*

## DECLARACIÓN DE AUTORÍA

Declaramos que somos las únicas autoras de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

Autor(es): María de Dolores Guardia Macías

\_\_\_\_\_

Aneli Valdés Acosta

\_\_\_\_\_

Tutor: Ing. Michel Mora Lozano

\_\_\_\_\_

## **OPINIÓN DEL TUTOR**

## RESUMEN

Los horarios docentes son un elemento fundamental en la organización de cualquier centro educacional. Constituyen el elemento organizativo de las actividades escolares y juegan un papel importante en la labor docente y educativa. Tradicionalmente el proceso de elaboración de horarios docentes se ha considerado una tarea complicada, debido a la gran cantidad de información y condiciones que deben tenerse en cuenta.

Actualmente en la Universidad de las Ciencias Informáticas (UCI), la gestión y confección de los horarios docentes se realiza de forma manual. Deben cumplirse una gran cantidad de restricciones, por lo que la confección semanal de los horarios es una tarea sumamente difícil que consume mucho tiempo y generalmente requiere de varias iteraciones y revisiones para obtener un horario congruente y lo más óptimo posible, que se ajuste a la dinámica y características especiales del centro.

Este trabajo consiste en la creación de un sistema automatizado para la gestión de los horarios docentes en la Facultad 9 de la UCI, en respuesta a la necesidad de mejorar y perfeccionar la manera en que actualmente se lleva a cabo esta tarea. La utilización de un sistema automatizado para generar los horarios docentes significaría una mejora considerable en este proceso en cuanto a eficiencia, tiempo y organización, contribuyendo favorablemente al desarrollo del proceso docente y educativo de la universidad.

En este documento se plasman los resultados del estudio realizado sobre la gestión de los horarios docentes, las técnicas de Inteligencia Artificial aplicables a la solución de este problema y las tecnologías web que dan soporte al sistema que se propone. Finalmente se muestran los resultados del análisis y diseño de la propuesta del sistema, y se exponen algunas recomendaciones con el objetivo de darle continuidad al tema desarrollado.

**PALABRAS CLAVES:** horarios docentes, planificación.

## ABSTRACT

Teaching schedules are a fundamental element in the organization of any educational center. They constitute the organizational element of school activities and play an important role in the teaching and educational work. Traditionally the process of elaboration of teaching schedules has been considered a complex task due to the great quantity of information and conditions to have in account.

At present at the University of Informatics Sciences (UCI), the confection process of teaching schedules comes true manually. There are great quantities of restrictions, for this reason the weekly confection of schedules is a very hard and consuming task that need a lot of time. This process generally requires several repetitions and reviews to obtain a congruent and the most favorable schedule; it must adjust to dynamics and special characteristics of the center.

This work consists in the creation of an automatized system for the process of elaboration of teaching schedules in the Faculty 9 of the UCI, in answer to the need of get better the way that at present it takes effect in this task. The utilization of an automated system to generate the teaching schedules would mean a considerable improvement in this process with efficiency, time and organization, contributing to favorably the development of the teaching and educational process of the university.

This document presents the results of the accomplished study on the elaboration process of teaching schedules, the applicable techniques of Artificial Intelligence to solve this kind of problems, and Web technologies that gives support to the proposed system.

Finally this document shows the analysis and design results of the proposal of the system, and they expose some recommendations for the sake of giving the developed theme continuity.

**KEY WORDS:** Teaching schedules, planning.

# ÍNDICE

<b>Introducción</b> .....	<b>1</b>
<b>Capítulo 1. Fundamentación Teórica</b> .....	<b>3</b>
1.1 Introducción.....	3
1.2 Objeto de Estudio.....	3
1.2.1 Descripción General.....	3
1.2.2 Conceptos asociados al dominio del problema .....	4
1.2.3 Descripción actual del dominio del problema .....	5
1.2.4 Situación Problemática.....	6
1.3 Análisis de otras soluciones existentes .....	7
1.4 Conclusiones.....	9
<b>Capítulo 2. Tendencias y Tecnologías actuales</b> .....	<b>10</b>
2.1 Introducción.....	10
2.2 Inteligencia Artificial.....	11
2.2.1 Concepto de Inteligencia Artificial .....	11
2.2.2 Aplicaciones de la Inteligencia Artificial.....	12
2.2.3 La Inteligencia Artificial y los modelos computacionales.....	13
2.2.4 Características de los problemas de IA.....	14
2.2.5 Problema de Generación de Horarios.....	15
2.2.6 Métodos de Solución de Problemas.....	16
2.2.7 La búsqueda como fundamento de la IA.....	17
2.2.8 Programación Lógica y Programación con Restricciones .....	17
2.2.9 Lenguajes para de la IA: PROLOG .....	19
2.2.1 SWI-Prolog .....	20
2.3 Arquitectura de Software.....	21
2.3.1 Arquitectura Cliente Servidor.....	22
2.3.2 Arquitecturas en Capas .....	23
2.4 Software Libre .....	24
2.5 Aplicaciones Web.....	26
2.5.1 Tecnologías del lado del cliente .....	27
2.5.2 Tecnologías del lado del servidor.....	29
2.6 Sistemas de Gestión de Bases de Datos (SGBD) .....	32
2.6.1 MySQL .....	32
2.7 Servidor Web.....	33
2.7.1 Servidor Web Apache .....	34

2.8 Metodología de Desarrollo de Software .....	35
2.8.1 El Proceso Unificado de Desarrollo. RUP .....	36
2.8.2 El Lenguaje Unificado de Modelado. UML .....	37
2.9 Conclusiones.....	37
<b>Capítulo 3. Presentación de la Solución Propuesta .....</b>	<b>39</b>
3.1 Introducción.....	39
3.2 Modelo del Negocio Actual.....	39
3.2.1 Procesos de negocio.....	39
3.2.2 Reglas del Negocio .....	40
3.2.3 Actores del Negocio .....	41
3.2.4 Trabajadores del Negocio .....	41
3.2.5 3.2.4 Diagrama de Caso de Uso del Negocio .....	42
3.2.6 Casos de Uso del Negocio.....	43
3.2.7 Modelo de Objetos .....	47
3.3 3.3 Requisitos Funcionales .....	47
3.4 Requisitos No Funcionales.....	50
3.4.1 Requerimientos de Software .....	50
3.4.2 Requerimientos de Hardware.....	50
3.4.3 Requerimientos de apariencia o interfaz external.....	51
3.4.4 Requerimientos de Seguridad.....	51
3.4.5 Requerimientos de Usabilidad.....	51
3.4.6 Requerimientos de Soporte.....	51
3.5 Descripción del Sistema Propuesto .....	52
3.5.1 Descripción de los actores .....	52
3.5.2 Diagrama de Casos de Uso del Sistema.....	52
3.5.3 Descripción Detallada de los Casos de Uso del Sistema.....	54
3.6 Conclusiones.....	65
<b>Capítulo 4. Construcción de la Solución Propuesta .....</b>	<b>66</b>
4.1 Introducción.....	66
4.2 Diagramas de Clases Web.....	67
4.2.1 Diagrama de Clases Web del CUS Autenticar Usuario.....	67
4.2.2 Diagrama de Clases Web del CUS Administrar Curso .....	68
4.2.3 Diagrama de Clases Web del CUS Administrar Balance de Carga .....	71
4.2.4 Diagrama de Clases Web del CUS Confección de Horarios.....	74
4.3 Principios de diseño .....	75
4.3.1 Estándares de la interfaz de la aplicación .....	75
4.3.2 Estándares de Codificación.....	77
4.4 Concepción general de la ayuda .....	77

4.5	Diseño de la Base de Datos .....	79
4.5.1	Diagrama de Clases Persistentes .....	79
4.5.2	Modelo de Datos .....	80
4.6	Generalidades de la Implementación .....	85
4.6.1	Aplicación Web.....	85
4.6.2	Algoritmo de Inteligencia Artificial.....	86
4.7	Modelo de Despliegue.....	88
4.8	Modelo de Implementación .....	89
4.9	Conclusiones.....	90
<b>Capítulo 5.</b>	<b>Estudio de Factibilidad .....</b>	<b>91</b>
5.1	Introducción.....	91
5.2	Planificación .....	91
5.2.1	Identificar los Puntos de casos de uso Desajustados .....	91
5.2.2	Ajustar los Puntos de Casos de Uso .....	92
5.2.3	Calcular esfuerzo de FT Implementación.....	95
5.2.4	Calcular esfuerzo de todo el proyecto .....	96
5.3	Costos .....	96
5.4	Beneficios tangibles e intangibles .....	97
5.4.1	Beneficios tangibles .....	97
5.4.2	Beneficios Intangibles .....	97
5.5	Análisis de costos y beneficios.....	98
5.6	Conclusiones.....	98
<b>Conclusiones</b>	.....	<b>99</b>
<b>Recomendaciones</b>	.....	<b>100</b>
<b>Bibliografía</b>	.....	<b>101</b>
<b>Glosario de Términos y Siglas</b>	.....	<b>104</b>
<b>Anexos</b>	.....	<b>106</b>

## ÍNDICE DE TABLAS

Tabla 1. Descripción de los Actores del Negocio.....	41
Tabla 2. Descripción de los trabajadores del negocio .....	41
Tabla 3. Descripción de los actores del sistema.....	52
Tabla 4. Descripción Detallada del CUS Autenticar Usuario .....	54
Tabla 5. Descripción Detallada del CUS Administrar Curso .....	55
Tabla 6. Descripción Detallada del CUS Administrar Balance de Carga .....	58
Tabla 7. Descripción Detallada del CUS Confección de Horarios .....	63
Tabla 8. Factor de peso de los actores sin ajustar .....	92
Tabla 9. Factor de peso de los casos de uso sin ajustar .....	92
Tabla 10. Factor de complejidad técnica .....	93
Tabla 11. Factor de ambiente .....	94
Tabla 12. Esfuerzo del Proyecto.....	96

## ÍNDICE DE FIGURAS

Figura 1. Diagrama de Casos de Uso del Negocio.....	42
Figura 2. Diagrama de Actividades del CUN Entrega del Balance de Carga .....	43
Figura 3. Diagrama de Actividades del CUN Entrega del Informe de Afectaciones de Locales.....	44
Figura 4. Diagrama de Actividades del CUN Entrega de Informes.....	45
Figura 5. Diagrama de Actividades del CUN Confección de Horarios.....	46
Figura 6. Diagrama de Clases del Modelo de Objetos .....	47
Figura 7. Diagrama de Casos de Uso del Sistema.....	53
Figura 8. Diagrama de Clases Web del CUS Autenticar Usuario.....	67
Figura 9. Diagrama de Clases Web del CUS Administrar Curso (Sección Registrar Curso) .....	68
Figura 10. Diagrama de Clases Web del CUS Administrar Curso (Sección Actualizar Curso) .....	69
Figura 11. Diagrama de Clases Web del CUS Administrar Curso (Sección Eliminar Curso).....	70
Figura 12. Diagrama de Clases Web del CUS Administrar Balance de Carga (Sección Registrar Balance de Carga) .....	71
Figura 13. Diagrama de Clases Web del CUS Administrar Balance de Carga (Sección Actualizar Balance de Carga) .....	72
Figura 14. Diagrama de Clases Web del CUS Administrar Balance de Carga (Sección Eliminar Balance de Carga) .....	73
Figura 15. Diagrama de Clases Web del CUS Confección de Horarios .....	74
Figura 16. Diagrama de Clases Web del CUS Confección de Horarios .....	75
Figura 17: Colores empleados en la aplicación .....	<b>¡Error! Marcador no definido.</b>
Figura 18. Diagrama de Clases Persistentes .....	79
Figura 19. Modelo de Datos. Sección 1 (Datos Básicos) .....	80
Figura 20. Modelo de Datos. Sección 2 (Afectaciones).....	81
Figura 21. Modelo de Datos. Sección 3 (Balance de Carga).....	82
Figura 22. Modelo de Datos. Sección 4 (Informe A-P-G) .....	83
Figura 23. Modelo de Datos. Sección 5 (Horario).....	84
Figura 24. Modelo de Despliegue .....	88
Figura 25. Modelo de Implementación.....	89

## INTRODUCCIÓN

Los horarios docentes son un elemento fundamental en la organización de cualquier centro educacional. Constituyen el elemento organizativo de las actividades docentes y permiten una visión general de la distribución cuantitativa y cualitativa de las asignaturas que se imparten. Tradicionalmente el proceso de elaboración de horarios docentes se ha considerado una tarea complicada, realizándose de forma manual. En el mercado actual del software existen algunas soluciones de carácter general, dirigidas principalmente a la gestión académica; pero en la mayoría de las ocasiones, no se ajustan a las condiciones reales de los distintos centros educacionales, siendo esta una de las razones por la cual su uso no se ha generalizado.

Actualmente en la Universidad de las Ciencias Informáticas (UCI), la gestión y confección de los horarios docentes se realiza de forma manual. Los horarios son elaborados por la dirección de cada una de las facultades en estrecha relación con los distintos departamentos docentes, que a nivel central, diseñan y confeccionan los planes de estudio de las asignaturas, así como el cronograma de actividades y exámenes. Deben tenerse en cuenta una gran cantidad de restricciones, por ejemplo: la disponibilidad de los locales, las características de los profesores, las asignaturas, los grupos, entre otras. La confección semanal de los horarios docentes es una tarea sumamente difícil que consume mucho tiempo y generalmente requiere de varias iteraciones y revisiones para obtener un horario congruente y lo más óptimo posible.

Este trabajo consiste en la creación de un sistema automatizado para la gestión de los horarios docentes en la Facultad 9 de la UCI, en respuesta a la necesidad de mejorar y perfeccionar la manera en que actualmente se lleva a cabo la planificación, dándole respuesta al **problema** que da origen a esta investigación: la inexistencia de un sistema automatizado para generar los horarios docentes de la Facultad 9 de la Universidad de las Ciencias Informáticas.

Surge esta necesidad producto de la creación y el dinámico desarrollo de la Universidad de las Ciencias Informáticas, centro de características especiales, basado en un modelo de aprendizaje desde la producción, que potencia el desarrollo de la informática dentro del sistema científico cubano y pretende

convertirse en el centro de la enseñanza superior de mayor fortaleza científica y tecnológica en el campo de la informática.

El **objeto de estudio** de esta investigación es el proceso de elaboración de los horarios docentes y el **campo de acción** es la automatización de la gestión de los horarios docentes en la Facultad 9 de la Universidad de las Ciencias Informáticas.

La utilización de un sistema automatizado para generar los horarios docentes significaría una mejora considerable en este proceso en cuanto a eficiencia, tiempo y organización, contribuyendo favorablemente al desarrollo del proceso docente y educativo de la universidad.

El **objetivo general** de este trabajo es proponer un prototipo funcional del Sistema Automatizado para la Gestión de Horarios Docentes.

Para cumplir este objetivo se plantean las siguientes **tareas**:

1. Estudio del arte del desarrollo de aplicaciones Web, la Inteligencia Artificial y el diseño y desarrollo de Bases de Datos.
2. Confección de la documentación completa UML del sistema propuesto.
3. Desarrollo de la base de datos de los horarios docentes de la Facultad 9.
4. Desarrollo de un algoritmo basado en Inteligencia Artificial capaz de confeccionar los horarios docentes.
5. Desarrollo de un prototipo funcional del Sistema Automatizado para la Gestión de Horarios Docentes.

# CAPÍTULO 1

## Fundamentación Teórica

### **1.1 Introducción**

En el presente capítulo se brinda una visión general de los aspectos relacionados con el proceso de confección de los horarios docentes en la Facultad 9, además se explica la estructura organizacional de la misma, así como los aspectos más importantes de la situación problemática.

### **1.2 Objeto de Estudio**

#### **1.2.1 Descripción General**

La planificación en la condición del socialismo constituye el eslabón central en la dirección de cualquier proceso. La planificación en la esfera de la educación ocupa un lugar importante por cuanto la organización y la planificación de la vida de la escuela no son una acción burocrática, sino un requerimiento educativo, una cuestión pedagógica que contribuye al trabajo formativo que debe realizarse, tanto con los alumnos como con el colectivo de trabajadores.

La planificación del trabajo debe ser objeto de atención primordial por cuanto constituye un instrumento idóneo para organizar y dirigir el desarrollo del proceso docente-educativo, así como las actividades que lo aseguran. (1)

La correcta organización y el buen funcionamiento de los centros docentes constituyen la garantía más inmediata de una eficaz actividad educativa. Por este motivo, desde un razonamiento absolutamente ortodoxo, se han ido decantando un conjunto de normas, que con el paso de los años se han revelado eficaces y pragmáticas a la hora de lograr una buena gestión escolar. Los horarios docentes constituyen

un elemento importante dentro de la gestión escolar, y repercuten directamente sobre el aprendizaje y adecuado desarrollo del estudiantado.

Existen algunas normas importantes que se recomiendan para favorecer el aprendizaje y el bienestar físico y mental de los estudiantes de los distintos niveles escolares, por ejemplo: planificar las asignaturas complejas en las horas de más alta capacidad de trabajo, teniendo en cuenta además, la variación de esta durante el día y la semana docente.

Desde el punto de vista pedagógico resulta fundamental la correcta concepción y organización de las asignaturas y sus distintas actividades docentes (conferencias, clases prácticas, laboratorios, etc.). Una planificación adecuada de las asignaturas garantiza una correcta asimilación de los contenidos.

Además existen otras normas generales como el establecimiento del receso corto entre turnos y un receso de al menos 15 minutos en cada sesión. Estas y otras normas que cada centro educacional establece según sus características y restricciones, deben ser cuidadosamente aplicadas para obtener un horario higiénico y que favorezca la labor docente y educativa del centro.

El proceso de formación en la Universidad de las Ciencias Informáticas es intenso, existen muchas disciplinas y gran parte de ellas poseen un alto nivel de profundidad y una complejidad considerable; por lo tanto, se necesita que las materias y disciplinas estén bien organizadas y dosificadas, de manera tal que la planificación favorezca la asimilación de los contenidos (conocimientos, habilidades y valores).

Debido a la dinámica de la UCI la planificación juega un papel fundamental pues deben conciliarse las actividades docentes, las afectaciones imprevistas, la disponibilidad de los locales, entre otras; de manera que se logren los objetivos del proceso de formación, logrando un balance entre todas las actividades tanto políticas, docentes, investigativas, productivas, culturales o deportivas.

### **1.2.2 Conceptos asociados al dominio del problema**

En la Universidad de las Ciencias Informáticas existe una Vicerrectoría de Formación, dividida en cuatro direcciones fundamentales: La Dirección de Teleformación, La Dirección de Formación del Profesional, La Dirección de Formación Posgraduada y La Dirección de Planificación y Control del

Proceso de Formación; esta última y la Dirección de Formación del Profesional son las encargadas de diseñar a nivel central el plan de estudios y su planificación en cada semestre.

Existen además once departamentos docentes centrales, que responden a las disciplinas de la carrera: Marxismo, Preparación para la Defensa, Matemática, Matemática Aplicada, Ingeniería y Gestión de Software, Técnicas de Programación, Sistemas Digitales, Práctica Profesional, Idioma Extranjero, Física y Ciencias Empresariales.

La Universidad también cuenta con diez Facultades, y en cada una de ellas existen tres departamentos docentes que a su vez contienen distintas disciplinas: Departamento de Ciencias de la Especialidad (Sistemas Digitales, Práctica Profesional, Ingeniería y Gestión de Software y Técnicas de Programación), Departamento de Ciencias Básicas (Física, Matemática, Matemática Aplicada y Ciencias Empresariales) y el Departamento de Humanidades (Preparación para la Defensa, Ciencias Sociales, e Idioma Extranjero). Cada una de estas disciplinas es atendida directamente por el departamento docente central correspondiente.

En cada una de las Facultades existe además un Vicedecanato de Formación, integrado por el Vicedecano de Formación y el planificador. El Vicedecano de Formación es el encargado de dirigir y desarrollar el trabajo metodológico a través de todos los niveles de la Facultad y además está a cargo de la planificación y el control del proceso docente y el trabajo educativo. El planificador es el que atiende directamente la parte organizacional del proceso docente y ejecuta todas las orientaciones y pautas establecidas por el Vicedecano de Formación.

### ***1.2.3 Descripción actual del dominio del problema***

En la Facultad 9 los horarios docentes se confeccionan cada semana, para ello son necesarios una serie de documentos imprescindibles, que son mencionados a continuación: Balance de Carga, Informe de Asignaturas-Profesor-Grupo, Informe de Afectaciones de Profesores e Informe de Afectaciones de Locales.

El Balance de Carga es un documento confeccionado por la Dirección de Planificación y Control del Proceso de Formación al inicio de cada semestre, contiene el balance de carga por asignatura para cada

año, así como el resumen de evaluaciones y actividades (Conferencias, Clases Prácticas, Laboratorios, etc.) de cada asignatura. De este documento, se obtiene la frecuencia semanal de cada asignatura y el tipo de actividad en cada frecuencia.

Para confeccionar el horario es fundamental que cada uno de los jefes de los departamentos docentes de la facultad (Departamento de Ciencias de la Especialidad, Departamento de Ciencias Básicas, y Departamento de Humanidades), elabore y entregue al Vicedecano de Formación el Informe de Asignaturas-Profesor-Grupo. Este informe contiene las asignaturas, los profesores que imparten cada asignatura y los grupos a los que cada profesor imparte clases.

Los jefes de los departamentos también deben entregar al Vicedecano de Formación, el Informe de Afectaciones de Profesores, que contiene las afectaciones de los profesores, es decir aquellos turnos en que no pueden impartir docencia por encontrarse en Cursos de Postgrados, Maestrías o afectados por alguna razón.

Además el Informe de Afectaciones de Locales, es adquirido por la facultad al inicio del semestre, indicando cuáles aulas, salones de conferencias y laboratorios están disponibles para la docencia y en qué sesiones.

Algunos de los documentos anteriormente mencionados prácticamente no sufren modificación durante el semestre como es el caso del Balance de Carga, mientras que otros como el Informe de Afectaciones de Profesores y el Informe de Afectaciones de Locales son renovados cada semana.

Una vez que el Vicedecano de Formación cuenta con toda la información requerida, solicita al planificador la confección del horario docente de la semana. El horario terminado es revisado por el Vicedecano de Formación y si es aprobado se publica para el conocimiento de toda la Facultad.

#### **1.2.4 Situación Problemática**

La elaboración de los horarios docentes resulta un problema para el Vicedecanato de Formación de la Facultad 9 desde varios puntos de vista.

La obtención de la información necesaria para confeccionar los horarios muchas veces es un proceso complicado, sobre todo la información de las afectaciones de los profesores. Sucede que en ocasiones la información con que se cuenta sobre las afectaciones de los profesores no está actualizada, y los distintos departamentos no tienen un formato o modelo estándar en el cual recoger está información, de manera que resulta trabajoso para el planificador utilizar los datos.

Además no existe una fecha establecida para recibir esta información y ocurren retrasos en su entrega. La no existencia de uniformidad y centralización en la entrega de estos listados, trae como consecuencia que en ocasiones los profesores no conocen el modo de informar y canalizar sus afectaciones, dirigiéndose equivocadamente al planificador, siendo en realidad su Jefe de Departamento el encargado de registrar, informar y autorizar debidamente sus afectaciones.

Otro aspecto deficiente dentro del proceso de confección de los horarios viene dado por la propia inexistencia de un sistema automatizado para guardar y gestionar los datos y es el referente a la dificultad para el manejo y uso de la información. Existen datos como el Balance de Carga, el Informe de Asignaturas-Profesor-Grupo, y el Informe de Afectaciones de Locales que prácticamente no sufren cambios a lo largo de un semestre, pero contienen información útil sobre la docencia en la facultad, que podría ser gestionada y consultada de manera rápida y fácil a través de un sistema informático, sin embargo en la práctica la información contenida en estos documentos es muy difícil de utilizar.

Esta dificultad para el manejo de la información provoca el punto clave de la situación problemática, que es explícitamente la confección de los horarios docentes. El planificador se enfrenta a muchísima información y un listado de restricciones que debe cumplir el horario, referidas a los distintos elementos, entiéndase restricciones de locales, profesores, grupos, asignaturas y del propio horario. Todo esto trae como consecuencia que la tarea de confeccionar el horario sea complicada y precise de mucho tiempo, y que además en muchas ocasiones requiera de varias iteraciones para obtener un horario congruente.

### **1.3 Análisis de otras soluciones existentes**

En la Universidad de las Ciencias Informáticas, en junio del 2006, se presentó un trabajo de diploma con el título “Sistema para la Gestión de Horarios Docentes”, que como objetivos planteaba “...el análisis y

diseño de un sistema que permita de forma rápida y eficiente la confección de los horarios docentes, así como facilitar a todas las personas involucradas en el proceso un fácil acceso al mismo y una eficaz actualización de los cambios". (2)

Este trabajo además de no contemplar la implementación de la aplicación y proponer un sistema que dista de lo que realmente necesita la Vicerrectoría de Formación de la Facultad 9; describe un flujo de actividades que tiene contradicciones y no corresponde con el modo en que se confecciona el horario docente en la Facultad 9.

Por las razones antes expuestas, este trabajo de diploma no se considera una solución válida para el problema planteado en esta investigación. Para más información véase el Anexo 1: Análisis de Tesis Precedente.

En el mercado actual del software existen varios Sistemas de Gestión Académica, de gran prestigio y una amplia comunidad de usuarios, que incluyen centros educacionales de distintos niveles. La gran mayoría de estos Sistemas de Gestión Académica incluyen un módulo destinado a los horarios docentes, que tienen funcionalidades para la organización, confección, edición y transferencia de horarios escolares.

Dentro de las aplicaciones más usadas se encuentran el Generador de Horarios para Centros de Enseñanza GHC 6.1, de la compañía Peñalara Software, que cuenta con un potente motor que genera resultados óptimos. Se destacan además el G2C/HD, el HorW, el KRONOWIN – Milenio 6 para Windows 98/ME/XP y NT y el Ormuz, que posee varios módulos que en su conjunto brindan un servicio de gestión académica muy completo.

No obstante de todas las facilidades y opciones de estas aplicaciones, no resultan adecuadas para la Facultad 9 de la Universidad de las Ciencias Informáticas, debido, entre otras razones, a que no se ajustan a la dinámica y las condiciones especiales del centro.

## **1.4 Conclusiones**

En este capítulo se profundizó en el conocimiento del proceso de confección de los horarios docentes en la Facultad 9, explicando su funcionamiento y particularidades. Se abordó además el tema de las soluciones existentes en el mercado actual del software para la gestión de horarios docentes.

## CAPÍTULO 2

# Tendencias y Tecnologías actuales

### 2.1 Introducción

Este capítulo está dedicado a la descripción de las tendencias y tecnologías actuales que se tendrán en cuenta para la confección de la solución. Una gran parte del capítulo se dedica al tema de la Inteligencia Artificial (IA), tratándose aspectos como el concepto, las principales áreas de aplicación, la relación de esta con los modelos computacionales, así como las características básicas de los problemas a los que se enfrenta, describiéndose específicamente el problema de generación de horarios docentes o *TimeTabling*.

Posteriormente se explica de manera breve y sencilla qué es la búsqueda y qué papel juega dentro de la IA, así como cuáles son las principales características de la Programación Lógica y de la Programación con Restricciones, dos de los paradigmas de programación que se aplican en el área de la Inteligencia Artificial. Finalmente se aborda el tema de Prolog, como uno de los lenguajes básicos de la misma, y se dedica un epígrafe a SWI-Prolog, una de las implementaciones más comunes y completas del lenguaje Prolog.

El capítulo también está dedicado de manera general a las aplicaciones web y los sistemas gestores de bases de datos, tratándose varios aspectos relacionados con ellos, tales como su concepto, características, estándares y lenguajes así como la relación entre ellos. Se tratan además aspectos como el uso de software libre y software propietario.

Finalmente el resto del capítulo es dedicado al Proceso Unificado de Desarrollo (RUP) como la metodología de desarrollo de software que guiará todo el proceso de confección del sistema y al Lenguaje Unificado de Modelado, como lenguaje de modelado visual para construir diagramas y documentar todo el proceso de desarrollo.

## 2.2 Inteligencia Artificial

### 2.2.1 Concepto de Inteligencia Artificial

La Inteligencia Artificial (IA) es una de las ramas de la Ciencia de la Computación que evoluciona con mayor rapidez. Es una de las disciplinas más prometedoras y a la que varios científicos están dedicando muchísimos esfuerzos en todo el mundo. En ella se estudian modelos computacionales orientados a dar solución a problemas que no son solubles con las técnicas de la computación llamada procedural. (3)

No hay una definición definitiva y única del término Inteligencia Artificial, G.F y W.A. Stubblefield en su libro *“Artificial Intelligence: structures and strategies for complex problem solving”*, la definen como la rama de la Ciencia de la Computación que se relaciona con la automatización del comportamiento inteligente. (4)

Por su parte T.Manukata en su libro *“Fundamentals of the new artificial intelligence: beyond traditional paradigms”*, la define “como el estudio de cómo hacer a las computadoras capaces de hacer cosas para las que los humanos necesitan inteligencia para hacerlas”. (4)

Más que la existencia de una definición formal y acabada de qué se entiende por IA, lo importante es que la misma es una disciplina que brinda métodos para resolver problemas para los cuales el enfoque algorítmico tradicional de computación no es suficiente. La insuficiencia puede estar dada, entre otras por las siguientes dos razones. Primera: No existe un algoritmo para resolver el problema (por ejemplo diagnosticar una enfermedad, jugar ajedrez, elaborar el plan de acción de un robot, etc.). Segunda, se conoce un algoritmo para resolver el problema pero cuando la dimensión del mismo crece, ya ese algoritmo no es computacionalmente aplicable (por ejemplo el problema del viajero vendedor).

La IA es un campo del saber que ha experimentado un rápido desarrollo y se ha diversificado a partir de la década de 1990. El núcleo tradicional de la Inteligencia Artificial ha sido denominado la IA simbólica tradicional, comprende el desarrollo de diversas formas de representación del conocimiento en forma simbólica y explícita y los métodos de solución de problemas usando este conocimiento basados en técnicas de búsqueda, los sistemas basados en el conocimiento, el procesamiento del lenguaje natural,

entre otras. Nuevos campos se han desarrollado en esta disciplina los cuales han potenciado la aplicabilidad de la misma. Entre ellos las redes neuronales artificiales, los algoritmos genéticos, los sistemas borrosos (fuzzy systems) y la teoría de conjuntos rugosos. Estos nuevos campos son denominados por algunos autores como Computación Blanda.

### **2.2.2 Aplicaciones de la Inteligencia Artificial**

Los primeros trabajos en el campo de la Inteligencia Artificial datan de la década del 50. Los trabajos iniciales estuvieron dirigidos a emular el proceso del pensamiento humano. Desde sus inicios la IA ha evolucionado tanto en la formalización, como en la implementación de sus técnicas. Actualmente ya son innumerables los campos de aplicación de las técnicas de IA. Ejemplos de ellos son los siguientes:

- **Juegos:** Fueron precisamente, las primeras aplicaciones de la IA, pues constituían una excelente demostración de que las computadoras podían realizar actividades inteligentes. Actualmente se han desarrollado múltiples programas para jugar al ajedrez, las damas, resolver el cubo de Rubik, entre otros.
- **Solución general de problemas:** Para la solución de problemas no algorítmicos o intratables computacionalmente la IA ofrece una alternativa eficiente. Se han desarrollado, de esta forma, demostradores de teoremas, simplificadores de fórmulas matemáticas, planificadores, entre otros.
- **Sistemas Expertos:** Constituye hoy en día el área de aplicación de la IA de mayor éxito. Los sistemas expertos permiten almacenar y utilizar el conocimiento de uno o varios expertos humanos en un dominio de aplicación concreto. Su uso incrementa la productividad, mejora la eficiencia en la toma de decisiones o simplemente permite resolver problemas cuando los expertos no están presentes. Muchos son los ejemplos de sistemas expertos desarrollados. Entre ellos MYCIN (para el diagnóstico médico), PROSPECTOR (para la detección de yacimientos minerales), DENDRAL (para el análisis de la estructura de sustancias químicas), etc.
- **Redes Neuronales:** Actualmente existe una intensa actividad alrededor de las redes neuronales. Una red neuronal es un modelo computacional que pretende simular el funcionamiento del cerebro

a partir del desarrollo de una arquitectura que toma rasgos del funcionamiento de este órgano. Las redes neuronales permiten resolver problemas para los cuales hasta ahora no hay algoritmos. (5)

- **Procesamiento de lenguaje natural:** Los programas de procesamiento de lenguaje natural utilizan las técnicas de IA para permitir que las computadoras comprendan y generen interfaces en lenguaje natural, lo cual haría más fácil el uso de estos programas por el usuario. La comprensión del lenguaje natural incluye el análisis de la sintaxis o gramática, la semántica o significado y el contexto del lenguaje. Un ejemplo de programas de procesamiento de lenguaje natural son los traductores de idiomas.
- **Robótica:** La robótica es el campo de la ingeniería dedicada a simular ciertas capacidades físicas del hombre. Los robots no pueden actuar ante situaciones cambiantes y tomar decisiones si no se dotan de software de IA. Un robot dotado de inteligencia puede decidir cambiar el orden de la secuencia de sus operaciones, eliminar algunos pasos o modificar alguno de ellos. Para eso se dotan a los robots de sensores de presión, temperatura, de luz, de cámaras de video, etc., además de software para la toma de decisiones. Al incorporar la IA a los robots aumenta su flexibilidad y utilidad.
- **Educación:** La IA puede ser usada en la educación y el entrenamiento, mediante la creación de programas que actúan como tutores inteligentes. Los tutoriales convencionales son rígidos en el sentido de que sus textos son fijos para todos los estudiantes. Al dotarlos de técnicas de IA, ellos son capaces de variar su contenido en dependencia del conocimiento, habilidad e inteligencia del estudiante. Por otro lado, pueden proveerse de interfaces en lenguaje natural e incluso pueden adquirir nuevo conocimiento automáticamente en interacción con el tutor humano.

### **2.2.3 La Inteligencia Artificial y los modelos computacionales**

El desarrollo de la IA es, realmente, una etapa cualitativamente superior dentro del progreso de la Ciencia de la Computación. Es importante, por tanto, analizar la relación existente entre la IA y los modelos computacionales convencionales. Existen diferentes modelos computacionales, entre los que podemos mencionar:

- Programación procedural: Un programa es una secuencia de instrucciones que transforman los datos para alcanzar la solución de un problema.
- Programación funcional: Un programa es considerado una función matemática que describe una relación entre una entrada y una salida.
- Programación lógica: Un programa es un conjunto de cláusulas de Horn<sup>1</sup> que declaran lógicamente el conocimiento sobre el dominio de un determinado problema.
- Programación orientada a objetos: Un programa es un modelo físico que simula el comportamiento de una parte real o imaginaria del mundo.

Estos paradigmas difieren entre sí, pero tienen en común que los programas están dirigidos a alcanzar la solución del problema dado y precisamente la IA se dedica al estudio de nuevas formas de representación del conocimiento, métodos de solución de problemas, razonamiento con incertidumbre y otras.

#### **2.2.4 Características de los problemas de IA**

Los problemas que se presentan comúnmente en la práctica pueden ser algorítmicos tratables, algorítmicos no tratables y no algorítmicos. En estos últimos no es posible encontrar un algoritmo para su solución. Esto nos indica que se debe reconsiderar la noción determinística desarrollada de algoritmo por una noción más flexible que incluya métodos de aproximación a la solución de un problema, como los que pone en práctica el ser humano cuando se halla ante problemas para los cuales no posee un algoritmo específico de solución.

Estos métodos se caracterizan por los siguientes rasgos fundamentales: no siempre nos conducen al resultado, no siempre el resultado es único y no siempre dan un resultado óptimo. La introducción de tales métodos en el diseño y programación de algoritmos computacionales es tarea de la IA.

Los problemas algorítmicos tratables se pueden resolver con las técnicas tradicionales. Los algorítmicos no tratables, llamados también NP-completos, son aquellos para cuya solución existen algoritmos no deterministas que se ejecutan de manera tratable, pero que hasta el momento se duda de que pueda

---

<sup>1</sup> Cláusulas de Horn: constituyen reglas del tipo "modus ponendo ponens", es decir, "Si es verdad el antecedente, entonces es verdad el consecuente".

existir un algoritmo determinista que los resuelva de manera tratable. Las técnicas de IA nos permiten resolver algunos de estos problemas.

La Inteligencia Artificial aborda problemas poco estructurados, donde no se conoce de antemano cuál es el mejor método para resolverlo. Hay que descubrir, si acaso, alguna solución. Esta es la razón de la palabra heurística cuyo significado se asocia a búsqueda. La esencia de la palabra heurística es contraria a la de algoritmo en el sentido de que ella es un camino para buscar lo nuevo, mientras el algoritmo es un camino para realizar lo ya muy bien conocido. Así se comprende que el paradigma primario para la resolución de problemas en Inteligencia Artificial sea la búsqueda de la solución orientada por la heurística, para tratar de reducir la explosión combinatoria que genera la búsqueda de todos los caminos posibles que se presenta en la mayoría de los problemas reales.

De manera general las características de los problemas de IA son las siguientes:

- Son intratables, no están bien definidos y su información es incompleta o incierta.
- Se resisten por su complejidad computacional a una solución algorítmica clásica.
- Requieren la introducción de estrategias de búsqueda de soluciones con heurísticas asociadas.
- Necesitan gran cantidad de conocimiento.
- Necesitan procesos de solución que incluyan la modelación del razonamiento.
- Tiene en cuenta el proceder de los expertos para el hallazgo no determinista ni siempre óptimo pero al menos satisfactorio de soluciones.

### **2.2.5 Problema de Generación de Horarios**

Dentro de los campos de aplicación de las técnicas de IA, la solución general de problemas es una de las áreas más interesantes. Dentro de los problemas más comunes en esta área se destaca el problema de generación de horarios o *TimeTabling*.

Este problema pertenece a la categoría de problema general de asignación, donde el tiempo es un factor importante, es un problema de gran complejidad computacional, razón por la cual actualmente se tiene gran interés por obtener métodos eficientes para su solución.

El problema de asignación o generación de horarios pertenece a la clase de problemas NP-Completo, para los cuales no se conoce un algoritmo de tiempo polinomial determinístico. Desde el punto de vista de complejidad teórica, es clasificado como NP-Completo al tratarse como un problema de decisión, es decir, determinar si existe o no una solución y es clasificado como NP-Duro al tratarse como problema de solución, es decir para construir una solución. En la actualidad no se conocen líneas generales para construir soluciones válidas a este problema.

### **2.2.6 Métodos de Solución de Problemas**

La Ciencia de la Computación trata cada vez de resolver nuevos problemas que sean difíciles de solucionar mediante las técnicas computacionales existentes. Estos problemas generalmente no tienen solución algorítmica conocida, o esta es tan compleja que no tiene una implementación práctica computacional.

Los problemas de decisión que generalmente se presentan en la vida empresarial pueden caer en esta clase de problemas. Por lo general existe una serie de recursos escasos o bien requisitos mínimos que hay que cumplir que condicionan la elección de la estrategia adecuada. Por lo general, el objetivo al tomar la decisión es llevar a cabo el plan propuesto de manera óptima (bien incurriendo en costos mínimos o bien buscando el beneficio).

No es de extrañar que la resolución de este tipo de problemas haya atraído la atención de los investigadores, dando lugar a la llamada investigación operativa. Un ejemplo de esta es la llamada computación lineal, siendo el algoritmo Simplex, el punto de referencia de la misma. Sin embargo como ocurre en muchos casos, la cantidad de variables que intervienen en el modelo hacen que la aplicación de Simplex no resulte eficiente, dado que el tiempo de cálculo necesario es excesivamente largo, incluso para grandes supercomputadoras. No es que el Simplex no pueda llegar al resultado óptimo buscado, sino que el tiempo de respuesta no es operativo.

La respuesta fue desarrollar nuevas técnicas de solución de problemas, similares a las humanas; una de las más importantes fue la búsqueda.

### **2.2.7 La búsqueda como fundamento de la IA**

La búsqueda de la IA difiere de la búsqueda convencional sobre estructuras de datos, esencialmente en que busca en un espacio problema, no en una pieza de dato particular. Se busca un camino que conecte la descripción inicial del problema con una descripción del estado deseado para el problema, es decir, el problema resuelto. Este camino representa los pasos de solución del problema.

A diferencia de las estructuras de datos que están predefinidas y que ya existen cuando comienza la búsqueda, los espacios problema son generalmente definidos proceduralmente, es decir, el espacio problema es creado a medida que es explorado.

Hay diferentes alternativas para realizar la búsqueda: búsqueda aleatoria, búsqueda a ciegas y búsqueda dirigida. La búsqueda dirigida es la base de la IA. La búsqueda puede ser dirigida por dato (forward) o dirigida por objetivo (backward).

Una búsqueda dirigida por dato comienza a partir de la información (o hechos) y trata de extraer conclusiones. Una búsqueda dirigida por objetivo comienza a partir de las expectativas de lo que es el objetivo o lo que sucederá, entonces busca las evidencias que respalden o contradigan esas expectativas (o hipótesis).

### **2.2.8 Programación Lógica y Programación con Restricciones**

La programación lógica es una nueva concepción de la programación. Difiere de los lenguajes procedural y funcional, no sólo en su fundamentación conceptual sino también en su estilo de programación (o paradigma). Los programadores de este tipo de lenguajes se involucran más en la descripción del problema en forma declarativa que en definir detalles de algoritmos para obtener una solución. De este modo los programas se asemejan a especificaciones en vez de implementaciones en lenguajes de programación convencionales.

Sin embargo los lenguajes de programación lógica demandan más recursos computacionales que los lenguajes convencionales; es decir, en programación lógica un programador simplemente declara las propiedades lógicas del problema que ha de ser resuelto. La descripción del problema es usada por el

sistema para encontrarle una solución (inferir una solución). Por todo esto la programación lógica constituye un paradigma de la programación declarativa.

En programación lógica la descripción de problemas se realiza a través de un formalismo lógico, basado en el cálculo de predicados de primer orden. Las teorías que pueden ser usadas para describir y analizar formalmente un lenguaje lógico tiene sus fundamentos en la lógica matemática.

La Programación con restricciones es un paradigma de la programación, donde las relaciones entre las variables son expresadas en términos de restricciones. Actualmente es usada como una tecnología de software para la descripción y resolución de problemas combinatorios particularmente difíciles, especialmente en las áreas de planificación y programación de tareas (calendarización).

Se trata de un paradigma de programación basado en la especificación de un conjunto de restricciones, las cuales deben ser satisfechas por cualquier solución del problema planteado, en lugar de especificar los pasos para obtener dicha solución.

La programación con restricciones se relaciona mucho con la programación lógica y con la investigación operativa. De hecho cualquier programa lógico puede ser traducido en un programa con restricciones y viceversa. Muchas veces los programas lógicos son traducidos a programas con restricciones debido a que la solución es más eficiente que su contraparte.

La diferencia entre ambos estriba principalmente en sus estilos y enfoques en el modelado del mundo. Para ciertos problemas es más natural (y por ende más simple) escribirlos como programas lógicos, mientras que en otros es más natural escribirlos como programas con restricciones.

El enfoque de la programación con restricciones se basa principalmente en buscar un estado en el cual una gran cantidad de restricciones sean satisfechas simultáneamente.

Los lenguajes de programación con restricciones son típicamente ampliaciones de otro lenguaje. El primer lenguaje utilizado a tal efecto fue Prolog. Por esta razón que este campo fue llamado inicialmente Programación Lógica con Restricciones. Ambos paradigmas comparten características muy similares, tales como las variables lógicas (una vez que una variable es asignada a un valor, no puede ser cambiado), o el backtracking.

Para la resolución del problema de generación de horarios, es común que se implementen soluciones utilizando cualquiera de los dos paradigmas (Programación Lógica o Programación con Restricciones) o en algunos casos se realiza una implementación que combina potencialidades de ambos paradigmas.

### **2.2.9 Lenguajes para de la IA: PROLOG**

Los lenguajes básicos para la implementación de las técnicas de IA son LISP y Prolog por las facilidades que brindan para el tratamiento simbólico, entre otras razones. LISP permite diseñar las estructuras de datos y los algoritmos que las manejan de una manera muy eficiente. Prolog, a su vez, tiene muy buen poder expresivo, permitiendo una formulación bastante natural y directa en la solución de muchos problemas.

El lenguaje Prolog (acrónimo de PROgramación LÓGica) se creó en la década del 70 por Alain Colmerouer y sus colaboradores de la Facultad de Ciencias en Marsella, Francia. La idea inicial del diseño se basó en una teoría del lenguaje y la Lógica Matemática, pero la estrecha relación con la lógica se convertía en algo pesado cuando se implementaba Prolog sobre una computadora, debido a esto se pusieron más restricciones prácticas y el resultado final fue un lenguaje sin trabas, independiente de detalles que sólo se necesitan en lógica. (6)

Prolog utiliza árboles como su estructura básica. En un programa Prolog cada árbol es un hecho o un átomo de conocimiento y la naturaleza bidimensional de los árboles contribuye a la potencia de expresión del lenguaje. La búsqueda en Prolog está dirigida por objetivos (encadenamiento hacia atrás) y el método de solución de problemas que utiliza es un método de búsqueda a ciegas denominado primero en profundidad.

Prolog es esencialmente un lenguaje declarativo, puesto que la mayoría de los enunciados de un programa Prolog típico son aserciones o afirmaciones. Un programa Prolog consta de un conjunto de aserciones, las cuales se considera que constituyen los axiomas propios de una teoría junto con un conjunto de cláusulas, a los que se consideran teoremas que hay que probar.

Dentro de los beneficios de Prolog se encuentran los siguientes:

- Permite crear programas de IA de manera rápida y fácil.
- Es ideal para implementar sistemas expertos y procesamiento del lenguaje natural.
- Los mecanismos de inferencia y los procedimientos son parte de él (built-in).
- El lenguaje Prolog se ha convertido en la esencia de las formas sintácticas existentes del uso de la lógica en la descripción de problemas.

### **2.2.1 SWI-Prolog**

Dado el éxito de Prolog, muchas compañías de software han creado sus propias versiones del mismo, por esta razón, existen varias implementaciones del lenguaje Prolog, tanto comerciales como de libre distribución. La mayoría de ellas se adaptan al estándar ISO<sup>2</sup>, por lo que los programas Prolog que se generen de acuerdo con dicho estándar podrán ejecutarse en cualquiera de estos sistemas. La diferencia es mínima entre versiones ya que su sintaxis y semántica es la misma; la variación que más resalta es el cambio de plataforma para la cual fueron desarrollados.

Dentro de las implementaciones más usadas se destacan Ciao Prolog, GNU Prolog, Visual Prolog, Strawberry Prolog y SWI-Prolog. La mayoría de estas implementaciones están bajo licencia LGPL (Lesser General Public License), implementan el estándar ISO para Prolog, y son soportadas en varios sistemas operativos como Unix, Windows y Mac OSX. No obstante, entre las distintas implementaciones, además de la plataforma y la licencia, se destacan algunas diferencias en cuanto a funcionalidades, como por ejemplo librerías e interfaces que aumentan la utilidad de estos sistemas.

SWI-Prolog, es posiblemente el intérprete de Prolog de mayor comunidad de usuarios y desarrolladores, quienes lo han mantenido en continuo desarrollo desde su creación en 1986.

Su desarrollo ha estado guiado desde un inicio por la intención de construir grandes aplicaciones en Prolog, por esta razón presenta características importantes como la conectividad con el lenguaje C, administración de memoria, procesamiento multihilo, así como un entorno de desarrollo rápido e

---

<sup>2</sup> ISO (Organización Internacional para la Estandarización): Es una organización internacional no gubernamental, compuesta por representantes de los organismos de normalización (ONs) nacionales, que produce normas internacionales industriales y comerciales.

interactivo. Además cuenta con librerías para gráficos, bases de datos, redes, servicios web, entre muchas otras que completan el sistema. (13)

A partir de su versión 5, tanto el sistema como sus extensiones se distribuyen bajo licencia LGPL (Lesser General Public License).

Además cuenta con una extensa documentación y funciona en plataformas Unix, Windows, y Macintosh. Todas estas características hacen a SWI-Prolog un sistema considerablemente flexible, rápido y de alto rendimiento. Actualmente es usado para investigaciones, desarrollo de aplicaciones comerciales y con fines educativos.

## **2.3 Arquitectura de Software**

El concepto de arquitectura incluye los aspectos estáticos y dinámicos más significativos del sistema. La Arquitectura de Software es el conjunto de decisiones significativas sobre la organización de un sistema, la selección de los elementos estructurales y sus interfaces, de los cuales el sistema está compuesto junto con su comportamiento. La Arquitectura describe los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente.

Roger Pressman, uno de los autores más reconocidos en el tema de la Ingeniería de Software a nivel mundial, define a la arquitectura de software como "...descripción de subsistemas y componentes de un sistema informático y la relación entre ellos...". (26)

No es novedad que ninguna definición de la AS (Arquitectura de Software) es respaldada unánimemente por la totalidad de los arquitectos. El número de definiciones circulantes alcanza un orden de tres dígitos, amenazando llegar a cuatro. No obstante la abundancia de definiciones del campo de la AS, existe en general un acuerdo de que ella se refiere a la estructura a grandes rasgos del sistema, estructura consistente en componentes y relaciones entre ellos. (16)

Una de los aspectos fundamentales dentro de la arquitectura de software son los estilos arquitectónicos. Un estilo es un concepto descriptivo que define una forma de articulación u organización arquitectónica. El

conjunto de los estilos cataloga las formas básicas posibles de estructuras de software, mientras que las formas complejas se articulan mediante composición de los estilos fundamentales. (16)

Un estilo arquitectónico o variante arquitectónica define a una familia de sistemas informáticos en términos de su organización estructural. Un estilo arquitectónico describe componentes y las relaciones entre ellos con las restricciones de su aplicación, la composición asociada y el diseño para su construcción. Los sistemas empresariales distribuidos pueden agrupar los siguientes estilos arquitectónicos, entre otros:

- Modelo-Vista-Controlador (MVC).
- Arquitecturas en Capas.
- Arquitecturas Orientadas a Objetos.
- Arquitecturas Basadas en Componentes.
- Arquitecturas Orientadas a Servicios.
- Arquitectura cliente/servidor.

Cada uno de estos estilos, plantea su propia estructura de componentes, teniendo un propósito específico y por lo tanto un área de aplicabilidad bien determinada.

### **2.3.1 Arquitectura Cliente Servidor**

Esta consiste en varios clientes distribuidos en diferentes nodos, conectados en red a uno o varios nodos servidores, donde el servidor puede servir a varios clientes a la vez. En los nodos clientes generalmente encontramos presentación de usuario y en los nodos servidores la lógica del negocio.

La arquitectura C/S es una forma de dividir y especializar programas y equipos de cómputo de forma que la tarea que cada uno de ellos realiza se efectúa con la mayor eficiencia posible y permita simplificar las actualizaciones y mantenimiento del sistema. (17)

Las características más importantes que se distinguen en el modelo cliente/servidor son: (18)

- Orientado a servicios. El servidor los ofrece y el cliente los consume.

- **Compartición de recursos.** Servicios ofrecidos a muchos clientes. Un servidor puede atender muchos clientes que solicitan esos servicios.
- **Transparencia de ubicación.** El servidor es un proceso que puede residir en el mismo ordenador que el cliente o en uno distinto a lo largo de una red. Un programa puede ser un servidor en un momento y convertirse en un cliente posteriormente.
- **Mezcla e igualdad.** Esta es una de las más importantes ventajas de este paradigma. Una aplicación cliente/servidor, idealmente es independiente del hardware y de sistemas operativos; mezclando e igualando estas plataformas.
- **Interacción a través de mensajes,** para envío y respuesta de servicios.
- **Servicios encapsulados,** exponiendo los servicios a través de interfaces, lo que facilita la sustitución de servidores sin afectar los clientes; permitiendo a la vez una fácil escalabilidad.

### **2.3.2 Arquitecturas en Capas**

El estilo de Arquitectura de Capas, define cómo organizar el modelo de diseño en capas, que pueden estar físicamente distribuidas, lo que significa que los componentes de una capa sólo pueden hacer referencia a componentes en capas inmediatamente inferiores. Varios autores definen el estilo en capas como una organización jerárquica tal que cada capa proporciona servicios a la capa inmediatamente superior y se sirve de las prestaciones que le brinda la inmediatamente inferior.

Es común utilizar el estilo capas sobre una arquitectura cliente servidor, lo cual simplifica la comprensión y la organización del desarrollo de sistemas complejos, reduciendo las dependencias de forma que las capas más bajas no son conscientes de ningún detalle o interfaz de las superiores. Además este modelo fomenta la reutilización.

En las aplicaciones distribuidas contemporáneas pueden encontrarse distintas variantes del estilo capas: arquitecturas de dos capas, arquitecturas de tres capas y arquitecturas de n capas. La arquitectura de tres capas es una de las más usadas en las aplicaciones web, tanto para sistemas sencillos o de mediana complejidad, como para sistemas más complejos.

La arquitectura de tres capas, es la arquitectura común de muchos sistemas de información, abarcando una interfaz para el usuario y el almacenamiento de datos persistentes. Una vista clásica de esta arquitectura plantea tres capas verticales: Presentación (Interfaz de usuario), Lógica de Aplicaciones (tareas y reglas que rigen el proceso) y Almacenamiento (Mecanismos de almacenamiento persistente).

La calidad tan especial de esta arquitectura consiste en aislar la lógica de la aplicación y en convertirla en una capa intermedia bien definida y lógica del software. (11) Esta separación entre la lógica de aplicación de la interfaz de usuario añade una enorme flexibilidad al diseño de la aplicación. Pueden construirse y desplegarse múltiples interfaces de usuario sin cambiar en absoluto la lógica de aplicación siempre que esté presente una interfaz claramente definida a la capa de presentación.

## **2.4 Software Libre**

El concepto de “Software Libre” se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software (19). A veces se malinterpreta el término de software libre, entendiéndose como gratis, sin embargo el software libre no tiene ninguna relación con el precio, se refiere a la libertad no al precio. Software Libre significa especialmente a cuatro clases de libertad para los usuarios.

- Libertad para ejecutar el programa en cualquier sitio, con cualquier propósito y para siempre.
- Libertad para estudiarlo y adaptarlo a las necesidades propias. Esto exige el acceso al código fuente.
- Libertad de redistribución, de modo que es permitido colaborar con vecinos y amigos.
- Libertad para mejorar el programa y publicar las mejoras. También exige el código fuente.(20)

Finalmente puede concluirse que Software libre es cualquier programa cuyos usuarios gocen de las libertades anteriormente mencionadas. Estas libertades se garantizan de acuerdo con la legalidad vigente por medio de una licencia.

Equivalente a software libre es el término "Open Source Software" (programas de código fuente abierto). Este término empezó a utilizarse en 1998 por algunos usuarios de la comunidad del software libre, tratando de usarlo como reemplazo al ambiguo nombre original del software libre.

La definición oficial de software de código fuente abierto, tal y como está publicada por la Open Source Initiative<sup>3</sup>, se acerca mucho a la definición de software libre. El significado obvio del término "código abierto" es "se puede mirar el código fuente", lo cual es un criterio más débil y flexible que el del software libre. Un programa de código abierto puede ser software libre, pero también puede serlo un programa semi-libre o incluso uno completamente propietario. El software de código abierto debe obedecer, entre otros los siguientes lineamientos:

- Libre Redistribución.
- Acceso al código fuente.
- Permitir trabajos derivados y modificaciones del código fuente.
- No discriminación contra personas o grupos.
- No discriminación respecto al campo de aplicación del software.

Un programa de código abierto puede ser modificado en sus partes más internas, a diferencia de un programa propietario del que se distribuye un fichero binario sin posibilidades de modificaciones internas.

El software propietario se distribuye bajo condiciones muy diferentes. Una licencia propietaria prohíbe la modificación, copia o redistribución sin el permiso de la compañía. Por el contrario, el software libre se suele desarrollar por comunidades de programadores entusiastas, colaborando vía Internet, sin interés por una compensación material. No obstante existen autoridades, particulares, sociedades u organizaciones sin ánimo de lucro trabajan, para unificar y para mantener las mejoras de la comunidad de desarrolladores acordes a la compatibilidad y el cumplimiento de los estándares.

Teniendo en cuenta lo anterior, resultan evidentes las ventajas de la utilización de soluciones Open Source, a continuación se mencionan algunas ventajas:

---

<sup>3</sup> *Open Source Initiative: Es una organización dedicada a la promoción del software abierto, fundada en febrero de 1998.*

- Gracias a la existencia de muchos colaboradores dispuestos a ayudar, estas soluciones se enriquecen constantemente por la comunidad de desarrolladores.
- Libre distribución: todo el mundo tiene derecho a usarlo, acceder a él, aprender de él e incluso modificarlo.
- Rápida solución de incidencias: las comunidades de software libre reaccionan más rápidamente ante incidencias serias (hasta 5 veces más rápido). (21)
- Bajo costo de adquisición e innovación tecnológica.
- Independencia del proveedor y adaptación del software. (22)

Por supuesto que el uso de estas soluciones también presenta desventajas:

- Su novedad: es algo relativamente nuevo y aún no ofrece confianza suficiente para los desarrolladores que se sienten más cómodos con sus herramientas propietarias de siempre.
- Soluciones difíciles de aprender: su aprendizaje se basa en el autoaprendizaje.
- Ausencia de un canal comercial para su distribución: su difusión se basa en las comunidades de usuarios. (21)

## **2.5 Aplicaciones Web**

De manera sencilla puede definirse que una aplicación web es una aplicación informática que los usuarios utilizan accediendo a un servidor web a través de Internet o de una intranet.

Dentro de las principales ventajas de las aplicaciones web pueden mencionarse la libertad de cambiar de sistema operativo o hardware sin afectar el funcionamiento de las aplicaciones del servidor, además no se requiere complicadas combinaciones de hardware/software para utilizar estas aplicaciones, solo se necesita un ordenador con un navegador web. Otra de las ventajas es que facilita el trabajo a distancia, se puede utilizar la aplicación web desde cualquier PC o computadora portátil con conectividad. Por otra parte la habilidad para actualizar y mantener aplicaciones web sin distribuir e instalar software en miles de potenciales clientes es otra razón de la popularidad de estas aplicaciones.

Desde el punto de vista tecnológico y de desarrollo actualizar o hacer cambios en las aplicaciones web es sencillo y sin riesgos de incompatibilidades. Existe solo una versión en el servidor lo que implica que no hay que distribuirla entre los demás computadores. Por lo tanto la utilización de ésta tecnología conlleva a reducir costos y complicaciones, y proporciona mayor libertad para realizar cualquier tipo de cambios; tanto es así que una de las estrategia que está emergiendo para las empresas proveedoras de software, es proveer acceso vía web al software. Para aplicaciones previamente distribuidas como de escritorio, esto puede requerir el desarrollo de una aplicación totalmente nueva o simplemente adaptar la aplicación para usar una interfaz web. Estos programas permiten al usuario pagar una cuota mensual o anual para usar la aplicación, sin necesidad de instalarla en la computadora del usuario. Las compañías que siguen esta estrategia son llamadas Proveedores de Aplicaciones de Servicio, este modelo de negocios está atrayendo la atención de la industria del software.

La Web es, básicamente, un entorno cliente/servidor con tres componentes: el lado del cliente, el lado del servidor y la red. Para la comunicación entre el cliente y el servidor, a través de la red se usa el protocolo llamado HTTP<sup>4</sup> (Hypertext Transfer Protocol). Una de las diferencias de Internet con los demás medios de comunicación es la interacción y personalización de la información con el usuario; sin embargo el HTML, Hyper Text Markup Language, es un lenguaje para crear documentos HTML, es decir, páginas Web, es estático, lo que condujo al surgimiento de tecnologías interactivas, del lado del cliente y del lado del servidor.

### **2.5.1 Tecnologías del lado del cliente**

Las tecnologías del lado del cliente están representadas por programas interpretados como JavaScript, VBScript que manipulan eventos, etc. y por programas ejecutados como Applets, Plugins y ActiveX, que muestran documentos de diferentes medios como audio, videos, animaciones, entre otros.

Las tecnologías del lado del cliente, como Java Script, a menudo en la forma de Dynamic HTML (DHTML), son muy importantes en las páginas Web modernas. "Dynamic HTML" es un término usado para describir la combinación de HTML, CSS (Hojas de Estilo en Cascada o Cascading Style Sheet) y scripts que permiten la interactividad de los documentos. (14)

---

<sup>4</sup> HTTP (Protocolo de Transferencia de Hipertexto): Es el protocolo usado en cada transacción de la Web (WWW).

HTML es un lenguaje de marcado, que se utiliza para formatear texto, e información de manera general. Se basa en el uso de etiquetas o marcas, para la definición del formato del texto, los distintos elementos que conforman la página, sus propiedades y disposición. Este lenguaje es interpretado por los navegadores, procesado y convertido en el DOM, el cual se muestra como página web a los usuarios.

Las Hojas de Estilo en Cascada o Cascading Style Sheet (CSS), permiten especificar el estilo de los documentos de una página, separado de su estructura, las CSS son muy utilizadas en la actualidad, gracias a algunas de sus ventajas, como es el hecho de que permiten una amplia manejabilidad del estilo de los documentos y realizar cambios y actualizaciones en el diseño gráfico con mucha facilidad. Cada vez es mayor la tendencia de los desarrolladores web de basar sus diseños casi totalmente en el uso de CSS evitando incluir elementos de estilo en los documentos HTML.

El HTML Dinámico o DHTML designa el conjunto de técnicas que permiten crear sitios web interactivos utilizando una combinación de lenguaje HTML estático, un lenguaje interpretado en el lado del cliente (como Java Script), el lenguaje de hojas de estilo en cascada (CSS) y la jerarquía de objetos de un DOM (Document Object Model).

Básicamente el DOM es un API para modificar dinámicamente el contenido de páginas Web, esto trae varias ventajas como son la manipulación interactiva de contenidos, la liberación de carga del servidor, el aprovechamiento de recursos computacionales del cliente, la flexibilidad prácticamente ilimitada y la utilización de estándares lo que aumenta considerablemente la compatibilidad de las aplicaciones. (15)

El Modelo de Objetos del Documento o DOM es una jerarquía de objetos asociados al navegador que proporciona el control sobre la presentación de las páginas, pues permite el acceso a todos los elementos de la misma. Cada uno de los navegadores implementa su propia versión del DOM. Los lenguajes de script como el Java Script permiten manipular los objetos del DOM, creando comportamiento dinámico del lado del cliente, a través de mecanismos para controlar y manipular eventos y objetos.

Las tecnologías del lado del servidor se basan en el servidor web que ejecuta la aplicación; ésta, una vez ejecutada, genera cierto código HTML; el servidor toma este código recién creado y lo envía al cliente por medio del protocolo HTTP.

## **2.5.2 Tecnologías del lado del servidor**

En el dominio de la red, los lenguajes de lado servidor más ampliamente utilizados para el desarrollo de páginas dinámicas son el ASP (Active Server Pages), PHP (Hipertext Preprocesor) y PERL (Practical Extracting and Reporting Lenguaje).

### **2.5.2.1 PHP**

PHP (acrónimo de "PHP: Hypertext Preprocessor") es un lenguaje "open source" interpretado de alto nivel, embebido en páginas HTML y ejecutado en el servidor (7). PHP es un lenguaje de secuencia de comandos de servidor diseñado específicamente para la Web.

PHP fue concebido en 1994 y es fruto del trabajo de un hombre, Rasmus Lerdorf. Ha sido adoptado por otras personas de talento y ha experimentado varias transformaciones importantes hasta convertirse en el producto actual. Hoy en día existe una gran comunidad de desarrolladores y programadores que continuamente implementan mejoras y nuevas funcionalidades y librerías; todo esto ha traído como consecuencia que PHP, sea uno de los lenguajes más usados en el desarrollo de aplicaciones web en todo el mundo.

Entre los competidores principales de PHP se puede citar a Perl, Microsoft Active Server Pages (ASP), Java Server Pages (JSP), entre otros. En comparación con estos productos, PHP cuenta con muchas ventajas, entre las que se encuentran las siguientes:

- PHP es gratuito. Al tratarse de software libre, puede descargarse y utilizarse en cualquier aplicación, personal o profesional, de manera completamente libre.
- PHP es muy eficiente. Mediante el uso de un único servidor, puede responder a millones de acceso al día. (8)
- PHP puede ser utilizado en cualquiera de los principales sistemas operativos del mercado, incluyendo Linux, muchas variantes Unix (incluido HP-UX, Solaris y OpenBSD), Microsoft Windows, Mac OS X, RISC OS, entre otros. (7)

- PHP soporta la mayoría de servidores web de hoy en día, incluyendo Apache, Microsoft Internet Information Server, Personal Web Server, Netscape y iPlanet, Oreilly Website Pro server, Caudium, Xitami, OmniHTTPd y muchos otros. (7)
- PHP tiene soporte para una gran cantidad de bases de datos. Las siguientes bases de datos están soportadas actualmente: Oracle (OCI7 y OCI8), dBase, InterBase, FrontBase, PostgreSQL, MS-SQL, MySQL, Velocis, Informix, entre otras. Adicionalmente, PHP soporta ODBC (The Open Database Connection standard), por lo que se puede conectar a cualquier base de datos que soporte este estándar. (7)
- PHP incorpora además una gran cantidad de funciones integradas para realizar útiles tareas relacionadas con la Web.

#### **2.5.2.2 PHP 5**

Desde el surgimiento de PHP, han existido varias versiones. PHP 3 fue la primera versión que se parecía fielmente al PHP tal y como se conoce hoy, fue liberado oficialmente en junio de 1998 y brindaba a los usuarios finales una sólida infraestructura para muchísimas bases de datos, protocolos y APIs. La gran extensibilidad de PHP3 atrajo a docenas de desarrolladores a unirse y enviar nuevos módulos de extensión.

En mayo de 2000, PHP 4 fue lanzado bajo el poder del motor Zend Engine 1.0., con soporte para la mayoría de los servidores web. Posteriormente el 13 de julio de 2004, fue lanzado PHP 5, utilizando el motor Zend Engine II (o Zend Engine 2). Este nuevo motor aumentó considerablemente las potencialidades de PHP, siendo el soporte sólido para Programación Orientada a Objetos la ventaja más importante de PHP5 respecto a versiones anteriores; pero debe tenerse en cuenta que PHP no es un lenguaje orientado a objetos puro, aunque aporta las suficientes características de los lenguajes orientados a objetos para el diseño y la implementación basados en objetos. En PHP 5 hay un nuevo modelo de Objetos. El manejo de PHP de objetos ha sido reescrito por completo, permitiendo un mejor desempeño y más características. (7)

Aunque el modelo de objetos estaba presente desde PHP3 y PHP4, tenía varias debilidades. Los objetos eran solamente colecciones de datos y métodos, agrupados en clases que soportaban herencia simple.

Dentro las debilidades existentes se encontraban que todos los métodos y atributos solo tenían visibilidad pública (no existía ni visibilidad privada, ni protegida), los objetos se pasaban como parámetros por valor y no por referencia. El nuevo Modelo de Objetos propuesto por PHP5, que se ha visto un poco influenciado por Java, se diferencia en que al crear un objeto, se recibe un apuntador hacia el objeto, y no el objeto en sí mismo. Cuando este apuntador es pasado como parámetro a funciones, asignado o copiado, solo el apuntador es el pasado, asignado o copiado. El objeto nunca es copiado o duplicado, evitando comportamientos indeseados o una mala utilización de los recursos, al tener varias copias de un mismo objeto en memoria todo el tiempo.

PHP5 permite un manejo eficiente de conceptos avanzados de programación como interfaces, herencia y polimorfismo e incluye modificadores de control de acceso para implementar el encapsulamiento, piedra angular en la programación orientada a objetos, de la que adolecía PHP4.

Otras de las características propias de PHP5 son las siguientes:

- Extensión PHP Data Object: La extensión PHP Data Objects (PDO) define una interfaz ligera y consistente para acceder a bases de datos en PHP.
- Soporte de XML<sup>5</sup>.
- Soporte nativo para SOAP a través de una extensión desarrollada con este fin.
- MySQLi, nueva extensión para conexiones a MySQL, la cual aprovecha las nuevas características del motor MySQL 4.1 y posterior.
- Extensión SQLite, el cual consiste en un pequeño motor de bases de datos incluido en PHP.
- Extensión Tidy, la inclusión de esta librería permite recorrer, diagnosticar, reparar y limpiar documentos HTML.
- Extensión de PERL: permite llamar scripts, manipular objetos y ejecutar la funcionalidad nativa de PERL.
- Iteradores de datos.

---

<sup>5</sup> XML (*Extensible Markup Language*): Lenguaje de marcas extensible, es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C).

- Excepciones de errores.
- Nuevo administrador de memoria el cual permite un mejor soporte en el trabajo multihilo.

## **2.6 Sistemas de Gestión de Bases de Datos (SGBD)**

Una base de datos es un conjunto de datos interrelacionados, almacenados con carácter más o menos permanente en la computadora, puede ser considerada una colección de datos variables en el tiempo.

Un Sistema Gestor de Bases de Datos, es el software que permite la utilización y/o actualización de los datos almacenados en una o varias bases de datos, por uno o varios usuarios desde diferentes puntos de vista a la vez.

Un SGBD tiene los siguientes objetivos específicos:

- Independencia de los datos y los programas de aplicación.
- Minimización de la redundancia.
- Integración y Sincronización de las Bases de Datos.
- Integridad de los datos.
- Seguridad y protección de los datos.
- Facilidad de protección de la información.
- Control Centralizado.

Actualmente existen SGBD libres y comerciales. Dentro de los SGDB libres se destacan PostgreSQL, MySQL, SQLite, Sybase ASE Express Edition para Linux entre otras y dentro de los comerciales se encuentran dBase, Fox Pro, IBM Informix, Microsoft Access, Microsoft SQL Server, Oracle, y Paradox.

### **2.6.1 MySQL**

MySQL, el sistema de gestión de bases de datos SQL Open Source más popular. Lo desarrolla, distribuye y soporta MySQL AB, que es una compañía comercial, fundada por los desarrolladores de MySQL. Es una

compañía Open Source de segunda generación que une los valores y metodología Open Source con un exitoso modelo de negocio. (9)

MySQL es un servidor multiusuario y de subprocesamiento múltiple. Utiliza SQL (Structured Query Language, Lenguaje de consulta estructurado), el lenguaje estándar para la consulta de bases de dato utilizado en todo el mundo. MySQL lleva disponible desde 1996 pero su nacimiento se remonta a 1979. Ha obtenido el galardón Choice Award del Linux Journal Readers en varias ocasiones. (8)

Entre los competidores principales de MySQL, se puede citar a PostgreSQL, Microsoft SQL Server y Oracle. MySQL cuenta con muchas ventajas, entre las que se encuentran las siguientes:

MySQL es muy rápido, fiable y fácil de usar. Ofrece hoy en día una gran cantidad de funciones. Su conectividad, velocidad, y seguridad hacen a MySQL Server altamente apropiado para acceder a bases de datos en Internet.

MySQL software es Open Source, por lo que es posible usar y modificar el software. Si se desea puede usarse de manera gratis o comprar la versión comercial.

MySQL se puede utilizar en una gran cantidad de sistemas Unix diferentes, así como bajo Microsoft Windows. Funciona además sobre múltiples plataformas, incluyendo AIX, BSD, FreeBSD, HP-UX, GNU/Linux, Mac OS X, NetBSD, Novell Netware, OpenBSD, OS/2 Warp, QNX, SGI IRIX, Solaris, SunOS, SCO OpenServer, SCO UnixWare, Tru64, Windows 95, Windows 98, Windows NT, Windows 2000, Windows XP, Windows Vista y otras versiones de Windows.

## **2.7 Servidor Web**

Un servidor web es un programa que implementa el protocolo HTTP (Hypertext Transfer Protocol). Este protocolo está diseñado para transferir lo que llamamos hipertextos, páginas web o páginas HTML (Hypertext Markup Language): textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de música.

Un servidor web se encarga de mantenerse a la espera de peticiones HTTP llevada a cabo por un cliente HTTP conocido como navegador. El navegador realiza una petición al servidor y éste le responde con el contenido que el cliente solicita.

En las aplicaciones en el lado del servidor: el servidor web ejecuta la aplicación; ésta, una vez ejecutada, genera cierto código HTML; el servidor toma este código recién creado y lo envía al cliente por medio del protocolo HTTP. Las aplicaciones de servidor suelen ser la opción por la que se opta en la mayoría de las ocasiones para realizar aplicaciones web. La razón es que, al ejecutarse esta en el servidor y no en la máquina del cliente, este no necesita ninguna capacidad adicional, como sí ocurre en el caso de querer ejecutar aplicaciones javascript o java. Así pues, cualquier cliente dotado de un navegador web básico puede utilizar este tipo de aplicaciones.

### **2.7.1 Servidor Web Apache**

El servidor HTTP Apache es un software (libre) servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows , Macintosh y otras, que implementa el protocolo HTTP/1.1. Apache es uno de los servidores Web más potentes del mercado, ofreciendo una perfecta combinación entre estabilidad y sencillez.

Las principales características de Apache son:

- Funcionalidad en múltiples plataformas.
- Elaborado índice de directorios.
- Soporte del último protocolo http 1.1.
- Sencilla administración basada en la configuración de un único archivo.
- Soporte para CGI (Common Gateway Interface) y FastCGI

Entre sus ventajas fundamentales se encuentran:

- Corre en una multitud de Sistemas Operativos, lo que lo hace prácticamente universal.
- Es una tecnología gratuita de código fuente abierto.

- Es un servidor altamente configurable de diseño modular. Es muy sencillo ampliar las capacidades del servidor Web Apache. Actualmente existen muchos módulos para Apache que son adaptables a este.
- Trabaja con gran cantidad de Perl, PHP y otros lenguajes de script. También trabaja con Java y páginas jsp. Teniendo todo el soporte que se necesita para tener páginas dinámicas.
- Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Es posible configurar Apache para que ejecute un determinado script cuando ocurra un error en concreto.
- Tiene una alta posibilidad de configuración en la creación y gestión de logs.

## **2.8 Metodología de Desarrollo de Software**

En un proyecto de desarrollo de software la metodología define Quién debe hacer Qué, Cuándo y Cómo debe hacerlo. Una metodología es un proceso y proceso de desarrollo de software no es más que el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software.

Para el éxito de un proyecto de software resulta fundamental la correcta aplicación de una metodología de desarrollo de software. No existe una metodología de software universal. Las características de cada proyecto (equipo de desarrollo, recursos, etc.) exigen que el proceso sea configurable. En la actualidad existen varias metodologías OO basadas en UML (Unified Modeling Language, Lenguaje Unificado de Modelado): Rational Unified Process (RUP), OPEN y MÉTRICA 3, entre otras.

Las metodologías no ágiles son aquellas que están guiadas por una fuerte planificación durante todo el proceso de desarrollo; llamadas también metodologías tradicionales o clásicas, donde se realiza una intensa etapa de análisis y diseño antes de la construcción del sistema.

Actualmente son muy populares las denominadas Metodologías Ágiles en el Desarrollo de Software. Un proceso es ágil cuando el desarrollo de software es incremental (entregas pequeñas de software, con ciclos rápidos), cooperativo (cliente y desarrolladores trabajan juntos constantemente con una cercana

comunicación), sencillo (el método en sí mismo es fácil de aprender y modificar, bien documentado), y adaptable (permite realizar cambios de último momento)

Las metodologías ágiles plantean valores y principios para permitir a los equipos desarrollar software rápidamente y respondiendo a los cambios que puedan surgir a lo largo del proyecto. Estas metodologías están especialmente orientadas para proyectos pequeños y ofrecer una alternativa a los procesos de desarrollo de software tradicionales, caracterizados por ser más rígidos y dirigidos por la documentación que se genera en cada una de las actividades desarrolladas. (12)

Las principales diferencias de las metodologías ágiles con respecto a las tradicionales, afectan no sólo al proceso en sí, sino también al contexto del equipo así como a su organización. No obstante la gran popularidad de las metodologías ágiles, también son muy criticadas. Por ejemplo las metodologías ágiles, no prestan la atención adecuada a la recolección completa de requisitos, siendo la ingeniería de requisitos, un factor fundamental para el logro de un proyecto de software, además desde el punto de vistas de mantenimiento y uso del sistema la poca documentación que se obtiene siguiendo esta metodología no es una buena práctica. Otro de los aspectos más criticados de las metodologías ágiles es que no se dedica suficiente tiempo ni esfuerzo al análisis y diseño, lo que puede conducir a productos de baja calidad, que deben ser probados y corregidos de manera constante.

RUP resulta un caso particular que puede usarse tanto al estilo ágil como al estilo tradicional. Esto se debe al especial énfasis que presenta en cuanto a su adaptación a las condiciones del proyecto (mediante su configuración previa a aplicarse). Realizando una configuración adecuada, podría considerarse ágil.

### ***2.8.1 El Proceso Unificado de Desarrollo. RUP***

El Rational Unified Process (RUP) es el resultado de varios años de desarrollo y uso práctico en el que se han unificado técnicas de desarrollo y muchas metodologías utilizadas por los clientes. La versión que se ha estandarizado vio la luz en 1998 y se conoció en sus inicios como Proceso Unificado de Rational 5.0; de ahí las siglas con las que se identifica a este proceso de desarrollo.

El Proceso Unificado está basado en componentes y se sostiene sobre tres ideas básicas: casos de uso, arquitectura y desarrollo iterativo e incremental. Para hacer que estas ideas funcionen se necesita un

proceso polifacético, que tenga en cuenta ciclos, fases, flujos de trabajo, gestión de riesgos, control de calidad, gestión de proyecto y control de configuración. El Proceso Unificado ha establecido un marco de trabajo que integra todas estas diferentes facetas. (11)

El objetivo del Proceso Unificado es guiar a los desarrolladores en la implementación y distribución eficiente de sistemas que se ajusten a las necesidades de los clientes. La eficiencia se mide en términos de coste, calidad y tiempo de desarrollo. Sin embargo, el Proceso Unificado es más que un simple proceso, es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas de software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto.

### **2.8.2 El Lenguaje Unificado de Modelado. UML**

El Proceso Unificado utiliza el Lenguaje Unificado de Modelado (Unified Modeling Language, UML) para preparar todos los esquemas de un sistema de software. De hecho UML es una parte esencial del Proceso Unificado, sus desarrollos fueron paralelos.

El UML se define como un lenguaje que permite especificar, visualizar, y construir los artefactos de los sistemas de software. Es un sistema notacional, destinado a los sistemas de modelado que utilizan conceptos orientados a objetos. UML estandariza los artefactos y la notación, pero no define un proceso oficial de desarrollo. (12)

Básicamente UML permite a los desarrolladores visualizar los resultados de su trabajo en esquemas o diagramas estandarizados. (11)

## **2.9 Conclusiones**

En este capítulo se abordaron varios temas que constituyen el fundamento teórico y tecnológico para la realización del sistema. El capítulo, de manera general, está enfocado en la justificación de la elección de las tendencias y tecnologías actuales que se tendrán en cuenta para la construcción del prototipo funcional. Se dedicó especial atención al tema de la Inteligencia Artificial, haciendo énfasis en los lenguajes y herramientas que se emplearán. El resto del capítulo realiza una breve reseña sobre las

tecnologías necesarias para el desarrollo de la aplicación web, caracterizando brevemente a cada una de ellas y mostrando sus ventajas y potencialidades. Además se describen las principales características de la metodología de desarrollo de software que se aplicará durante todo el proceso de desarrollo del sistema.

# CAPÍTULO 3

## Presentación de la Solución Propuesta

### **3.1 Introducción**

En este capítulo se refleja los aspectos centrales del modelamiento del negocio, abordando aspectos como los procesos del negocio, las reglas que lo rigen, así como la descripción de los actores y trabajadores del negocio. Se presenta además el diagrama de casos de uso del negocio, y el diagrama de actividades de cada uno de los casos de uso del negocio. Posteriormente se explican los requerimientos funcionales y no funcionales del sistema y se presenta el diagrama de casos de uso del sistema con la descripción de los casos de uso del sistema que se consideran como críticos.

### **3.2 Modelo del Negocio Actual**

#### **3.2.1 Procesos de negocio**

En la Facultad 9, la confección de los Horarios Docentes es una tarea dirigida personalmente por el Vicedecano de Formación. El proceso de confección de los horarios requiere para llevarse a cabo, tres conjuntos de datos básicos que son recibidos por el Vicedecano de Formación. Se requiere del Balance de Carga, documento entregado por la Dirección de Planificación y Control del Proceso de Formación, de dos informes que deben ser entregados por los jefes de departamentos docentes de la Facultad, el Informe de Asignatura-Profesor-Grupo y el Informe de Afectaciones de los Profesores y además se necesita el Informe de Afectaciones de Locales, el cual es entregado por la dirección de la Facultad. Una vez que se cuenta con toda esta información el Vicedecano de Formación orienta al Planificador que confeccione los horarios docentes semanales de todos los grupos de la Facultad. El planificador procede a confeccionar los horarios, teniendo en cuenta toda la información recibida así como un conjunto de reglas y restricciones establecidas para la confección de los horarios en la facultad y en la universidad.

### **3.2.2 Reglas del Negocio**

Las reglas del negocio describen políticas que deben cumplirse o condiciones que deben satisfacerse, es decir, regulan los distintos aspectos del negocio.

#### **Restricciones de Horario:**

- El horario se planifica semanalmente, de lunes a sábado.
- El horario consta de seis turnos de clases: tres en la sesión de la mañana y tres en la sesión de la tarde.
- A ningún grupo o profesor se les planifica el tercer y cuarto turno de clases, para no afectar el horario de almuerzo.

#### **Restricciones de los Profesores:**

- A los profesores externos y adjuntos no se les planifica el último turno de clases de la tarde.
- A los Alumnos Ayudantes solo se les planifica turnos en la sesión contraria a su horario regular de clases.
- Para la planificación de la docencia de los profesores se tendrán en cuenta el Informe de Afectaciones - Profesores.

#### **Restricciones de Asignaturas:**

- Aquellas asignaturas de mayor grado de dificultad se tratarán de planificar en los primeros turnos de clases.
- Las asignaturas serán planificadas según las especificaciones del Balance de Carga.

#### **Restricciones de Locales:**

- De los locales se tendrá en cuenta su disponibilidad, especificada en el Informe de Afectaciones de Locales.

**Restricciones de los Grupos:**

- De cada grupo se especificará si tiene local fijo.
- De cada grupo se especificará la sesión predominante.

**3.2.3 Actores del Negocio**

El término actor significa el rol que algo o alguien juega cuando interactúa con el negocio. Un actor del negocio es cualquier individuo, grupo, entidad o sistemas externos; con los que el negocio interactúa.

**Tabla 1.** Descripción de los Actores del Negocio

<b>Actores del Negocio</b>	<b>Descripción del Actor</b>
Dirección de Planificación y Control del Proceso de Formación	Inicia el proceso de entrega del Balance de Carga, documento imprescindible para confeccionar los horarios.
Dirección de la Facultad	Inicia el proceso de entrega del Listado de Afectaciones de Locales, documento imprescindible para confeccionar los horarios.
Jefe de Departamento	Inicia el proceso de entrega del Informe de Asignatura-Profesor-Grupo y del Informe de Afectaciones de los Profesores documento imprescindible para confeccionar los horarios.
Vicedecano de Formación	Interviene en todos los procesos del negocio, y es quien inicia y controla el proceso de confección de los horarios docentes.

**3.2.4 Trabajadores del Negocio**

Un trabajador del negocio es una abstracción de una persona o grupo de personas, o un sistema automatizado; que actúa en el negocio realizando una o varias actividades.

**Tabla 2.** Descripción de los trabajadores del negocio

<b>Trabajador del Negocio</b>	<b>Descripción del Trabajador</b>
Planificador	Es el encargado de llevar a cabo el proceso de confección del horario docente.

### 3.2.5 3.2.4 Diagrama de Caso de Uso del Negocio

El diagrama de Casos de Uso del Negocio representa gráficamente a los procesos del negocio y su interacción con los actores del negocio.

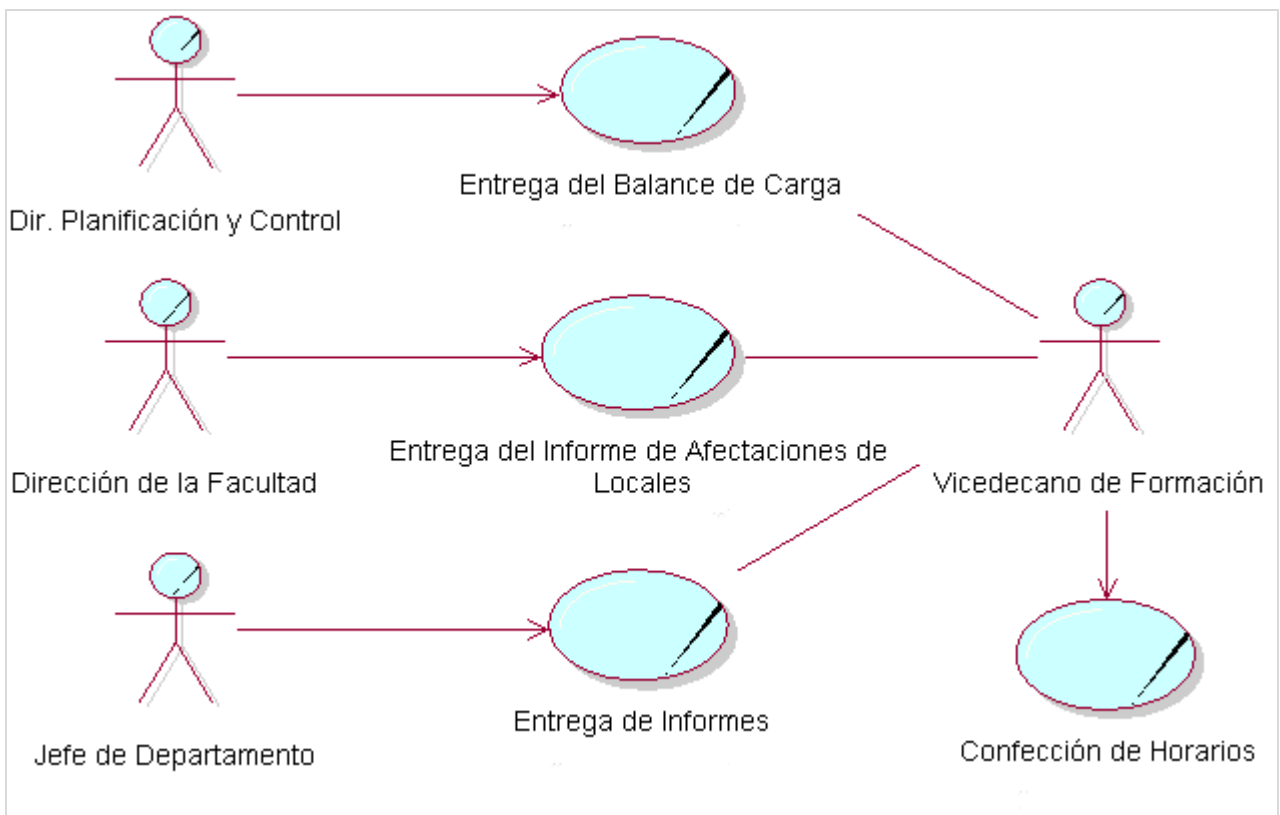


Figura 1. Diagrama de Casos de Uso del Negocio

### 3.2.6 Casos de Uso del Negocio

#### 3.2.5.1 Caso de Uso de Negocio Entrega del Balance de Carga

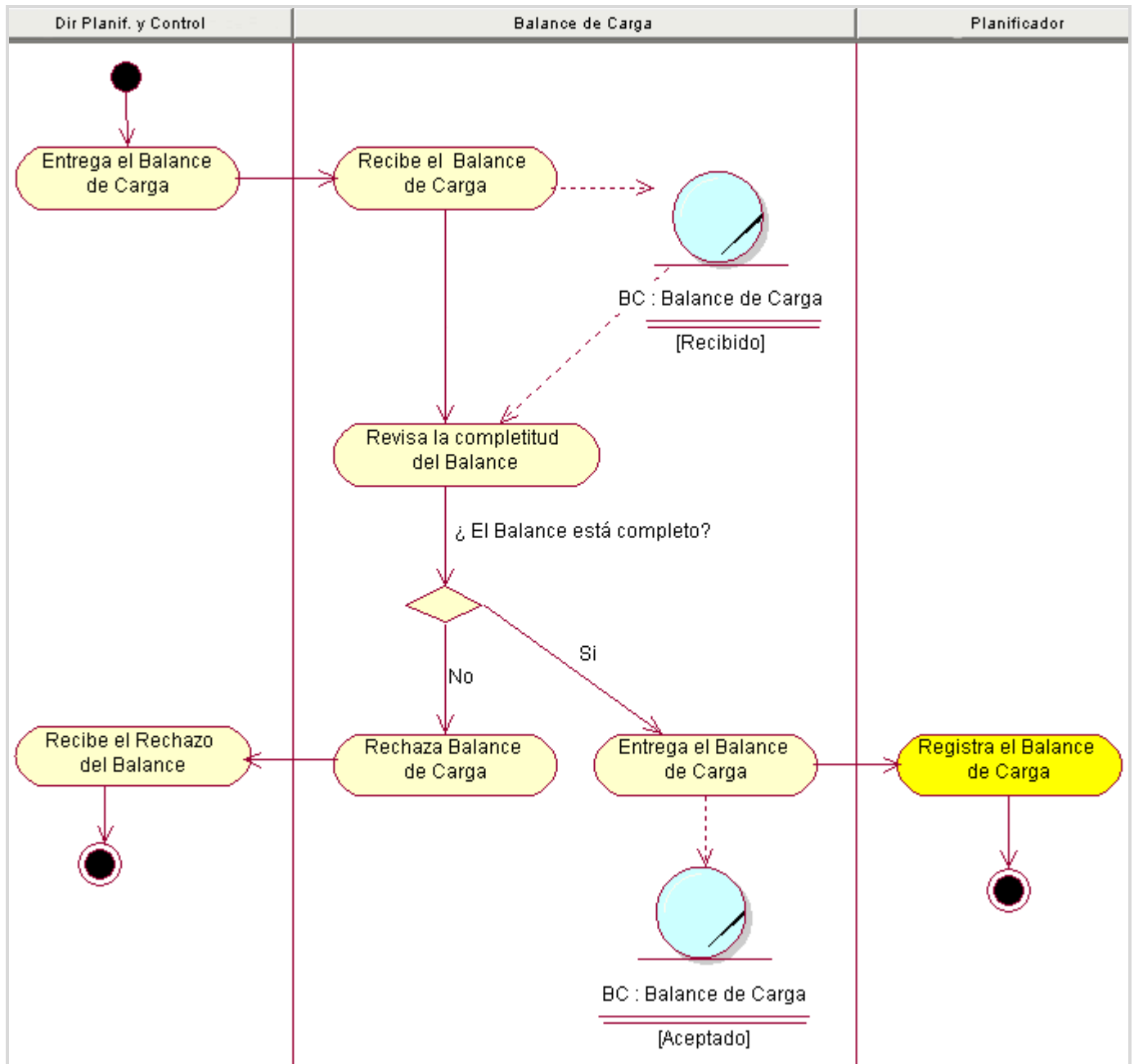


Figura 2. Diagrama de Actividades del CUN Entrega del Balance de Carga

3.2.5.2 Caso de Uso de Negocio Entrega del Informe de Afectaciones de Locales

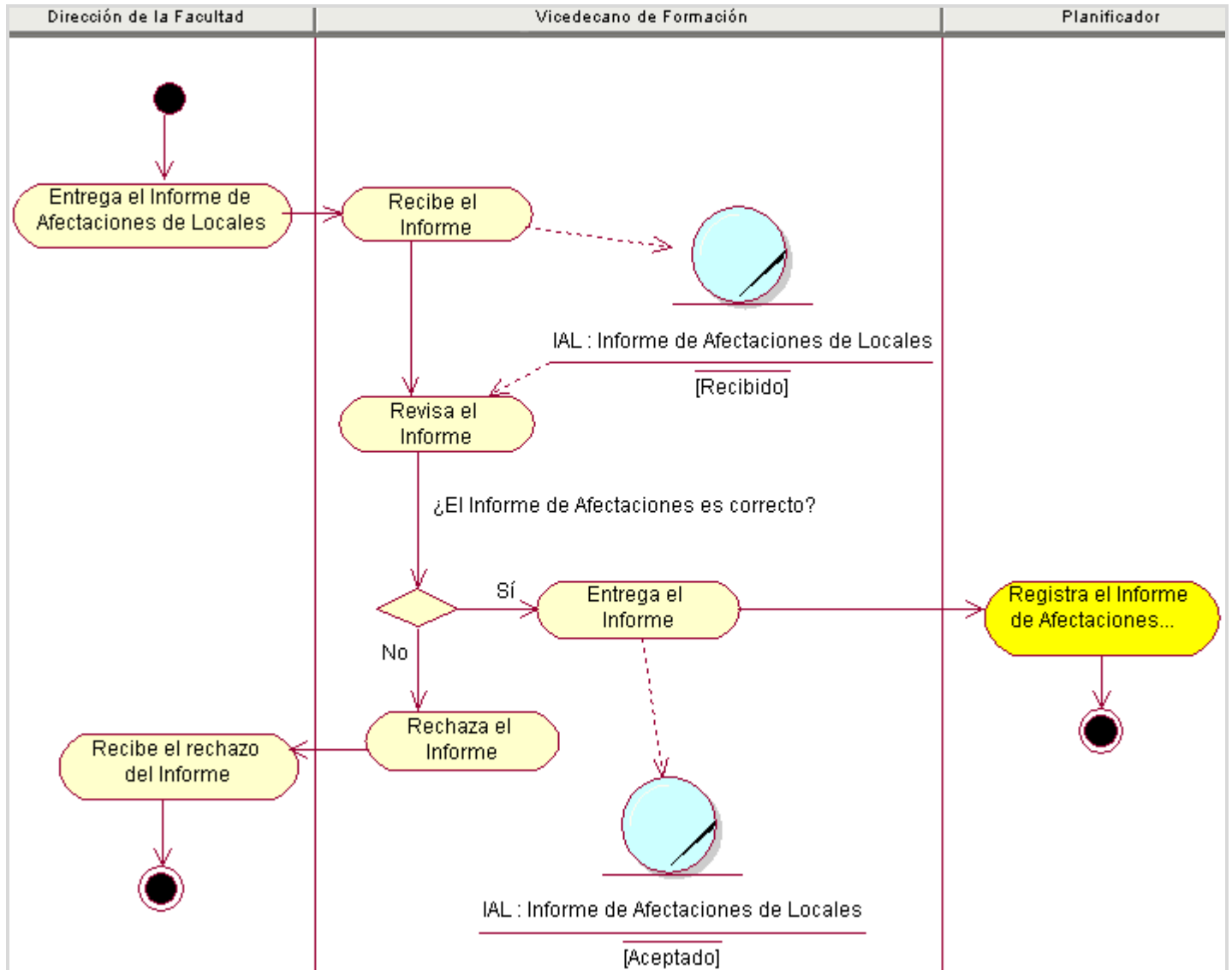


Figura 3. Diagrama de Actividades del CUN Entrega del Informe de Afectaciones de Locales

3.2.6.3 Caso de Uso de Negocio Entrega de Informes

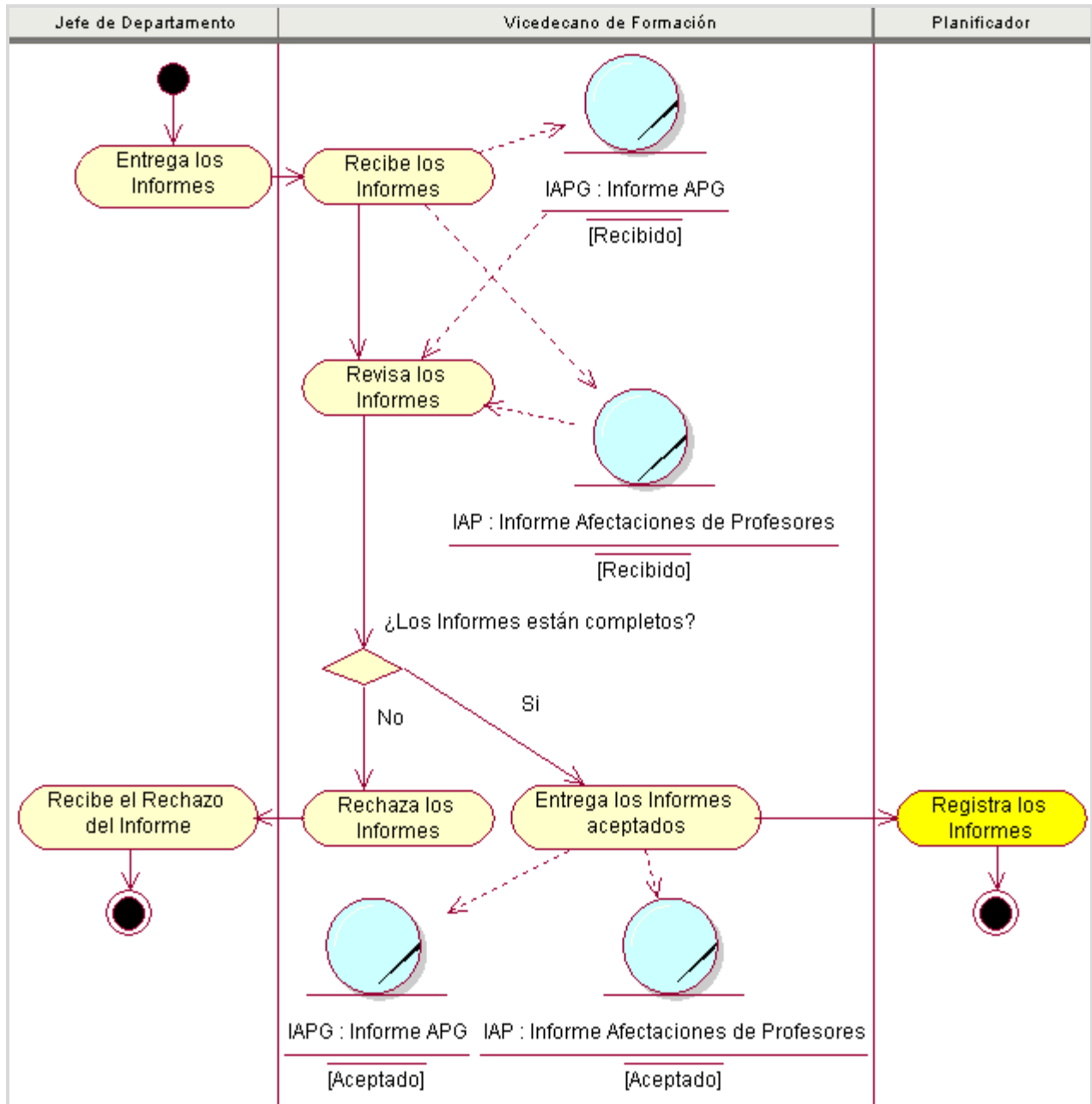


Figura 4. Diagrama de Actividades del CUN Entrega de Informes

3.2.6.4 Caso de Uso de Negocio Confección de Horarios

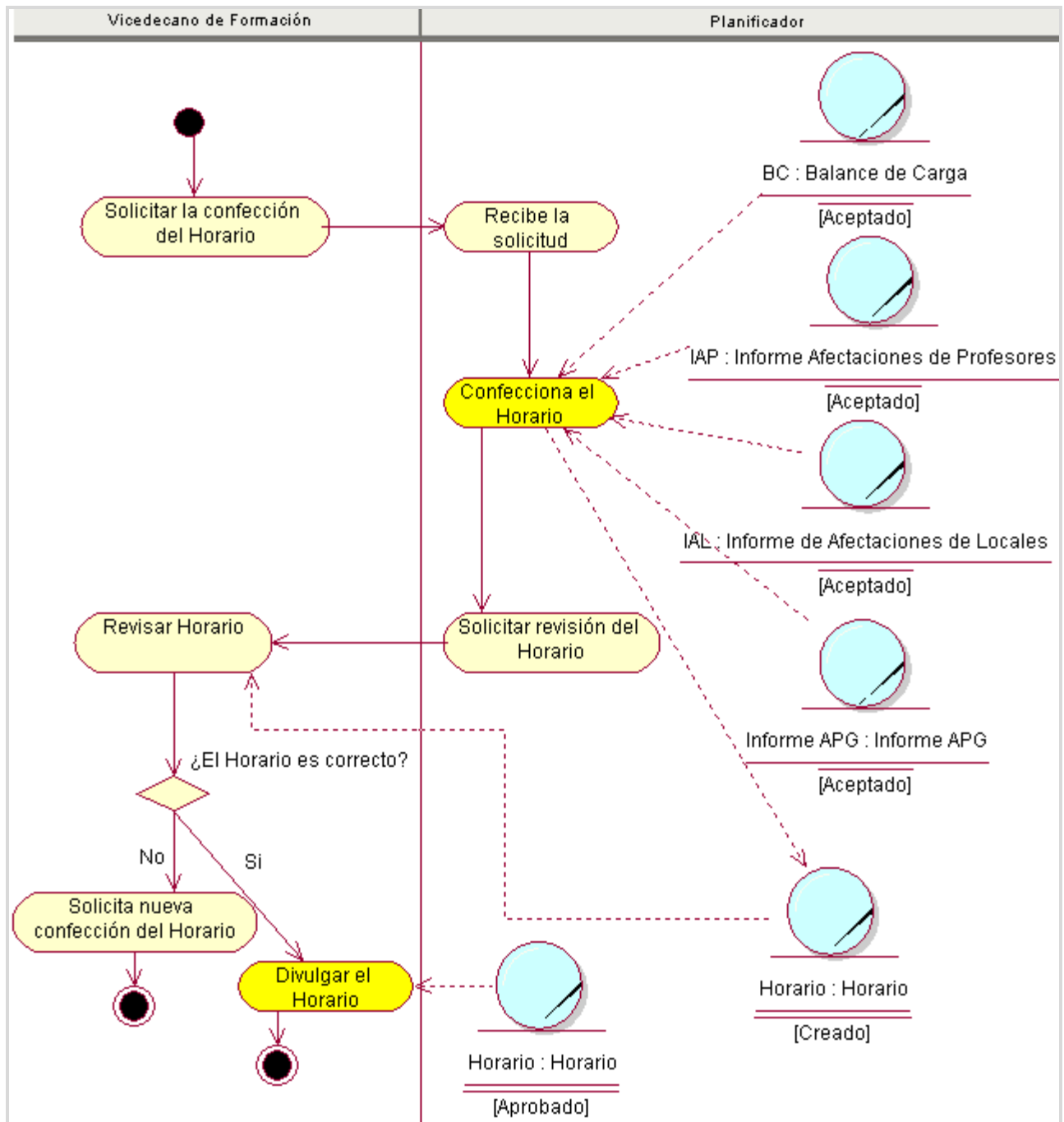


Figura 5. Diagrama de Actividades del CUN Confección de Horarios

### 3.2.7 Modelo de Objetos

El diagrama de clases, como artefacto que se construye para describir el modelo de objetos del negocio, muestra la participación de los trabajadores y entidades del negocio y la relación entre ellos.

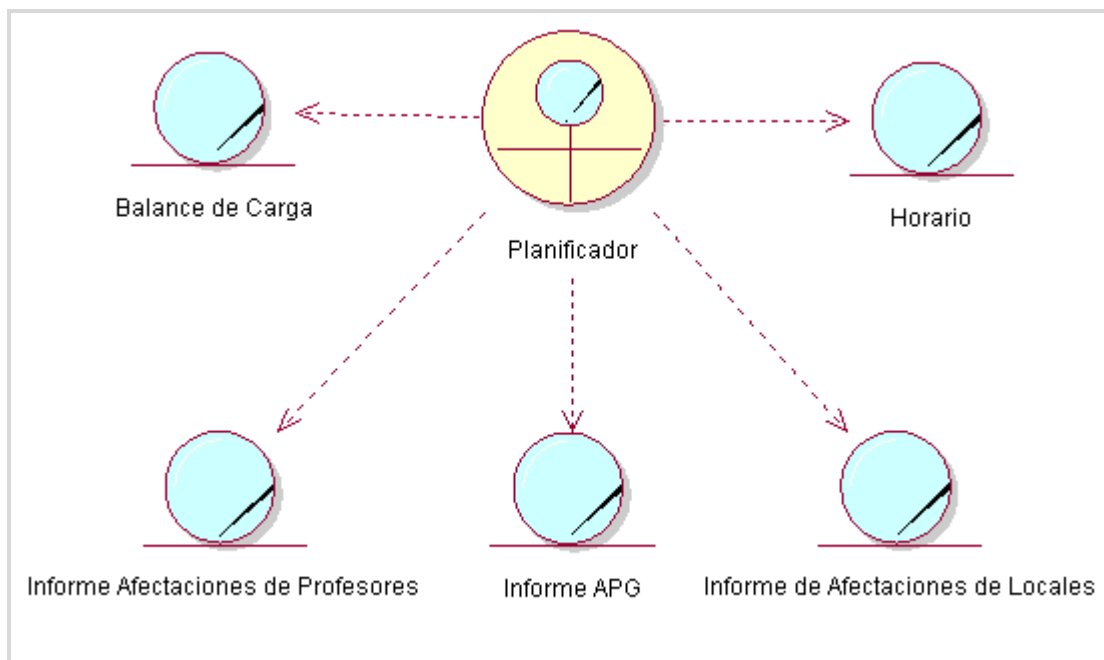


Figura 6. Diagrama de Clases del Modelo de Objetos

### 3.3 3.3 Requisitos Funcionales

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir.

**R 1** - Autenticar Usuario

**R 2** - Administrar Curso

**2.1** - Crear Curso

**2.2** - Modificar Curso

**2.3 - Eliminar Curso**

**R 3 - Administrar Asignatura**

**3.1 - Crear Asignatura**

**3.2 - Modificar Asignatura**

**3.3 - Eliminar Asignatura**

**3.4 - Mostrar Listado de Asignaturas**

**3.5 - Imprimir Listado de Asignaturas**

**R 4 - Administrar Profesor**

**4.1 - Crear Profesor**

**4.2 - Modificar Profesor**

**4.3 - Eliminar Profesor**

**4.4 - Mostrar Listado de Profesores**

**4.5 - Imprimir Listado de Profesores**

**R 5 - Administrar Grupo**

**5.1 - Crear Grupo**

**5.2 - Modificar Grupo**

**5.3 - Eliminar Grupo**

**5.4 - Mostrar Listado de Grupos**

**5.5 - Imprimir Listado de Grupos**

**R 6 - Administrar Locales**

**6.1 - Crear Locales**

**6.2 - Modificar Locales**

**6.3 - Eliminar Locales**

**6.4 - Mostrar Listado de Locales**

**6.5 - Imprimir Listado de Locales**

**R7 - Administrar Informe A-P-G**

**7.1 - Crear Informe A-P-G**

**7.2 - Modificar Informe A-P-G**

**7.3 - Eliminar Informe A-P-G**

**7.4 - Mostrar Informe A-P-G**

**7.5 - Imprimir Informe A-P-G**

**R 8 - Administrar Afectaciones de Profesores**

**8.1 - Crear Informe de Afectaciones de Profesores**

**8.2 - Modificar Informe de Afectaciones de Profesores**

**8.3 - Eliminar Informe de Afectaciones de Profesores**

**8.4 - Mostrar Informe de Afectaciones de Profesores**

**8.5 - Imprimir Informe de Afectaciones de Profesores**

**R 9 - Administrar Afectaciones de Locales**

**9.1 - Crear Informe de Afectaciones de Locales**

**9.2 - Modificar Informe de Afectaciones de Locales**

**9.3 - Eliminar Informe de Afectaciones de Locales**

**9.4 - Mostrar Informe de Afectaciones de Locales**

**9.5 - Imprimir Informe de Afectaciones de Locales**

**R10 - Administrar Balance de Carga**

**10.1 - Crear Balance de Carga**

**10.2 - Modificar Balance de Carga**

**10.3 - Eliminar Balance de Carga**

**10.4 - Mostrar Balance de Carga**

**10.5 - Imprimir Balance de Carga**

**R11 - Generar Horario**

**R12 - Mostrar Horario**

**12.1 - Imprimir Horario**

### **3.4 Requisitos No Funcionales**

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

#### **3.4.1 Requerimientos de Software**

- En el Servidor Web y de Base de Datos se requiere tener instalado el WAMP 5 que incluye PHP 5, Servidor Apache 2 y MySQL 5 y además requiere el SWI-Prolog en su versión 5.6.31. Puede tener cualquiera de los siguientes sistemas operativos: Windows XP, NT, 2000, 2003.
- Las PC clientes requieren de un navegador web, preferiblemente Internet Explorer 6.0 o Firefox 2.0 y conectividad.

#### **3.4.2 Requerimientos de Hardware**

- Las PC clientes requieren al menos de un microprocesador Intel Celeron, Pentium 3 o superior, un mínimo de 128 MB de RAM, y 1.5 GB de espacio disponible en disco duro. Requiere además de una impresora.

- El Servidor Web y de Base de Datos requiere al menos de un microprocesador Pentium 4 o superior, un mínimo de 728 MB de RAM y 10 GB de espacio disponible en disco duro.

### ***3.4.3 Requerimientos de apariencia o interfaz externa***

El sistema debe tener una interfaz sencilla, con una apariencia amigable para el usuario y de fácil navegación.

### ***3.4.4 Requerimientos de Seguridad***

La información que se maneja a través del sistema debe estar protegida de acceso de personas no autorizadas.

### ***3.4.5 Requerimientos de Usabilidad***

El sistema será utilizado por usuarios familiarizados con la planificación docente y las características de la docencia en la universidad. Además deberá poder aplicarse en cualquier facultad de la Universidad.

### ***3.4.6 Requerimientos de Soporte***

En caso de que ocurran cambios importantes en la estructura organizacional del centro desde el punto de vista docente o modificaciones en el modelo actual de planificación docente se debe realizar una actualización del sistema por lo que el sistema debe ser diseñado e implementado de manera que permita extensiones, modificaciones y un mejoramiento progresivo de sus funcionalidades.

### 3.5 Descripción del Sistema Propuesto

El sistema es una aplicación web, destinada a ser usada en el Vicedecanato de Formación, solamente tendrá un usuario: el Planificador, quien será el encargado de toda la interacción con el sistema. Está dividido en varias áreas de trabajo, que facilitan la interacción del planificador con el sistema y permitan dar cumplimiento a los requerimientos funcionales.

#### 3.5.1 Descripción de los actores

Los actores del sistema son terceros que no forman parte del sistema sino interactúan con él.

**Tabla 3.** Descripción de los actores del sistema

Nombre del Actor	Descripción del Actor
Planificador	Es el encargado de hacer todas las entradas de datos al sistema, administrar los profesores, asignaturas, grupos, locales, y además administra todos los informes como el Balance de Carga, el Informe de Asignaturas-Profesor-Grupo, y los Informes de Afectaciones de los Profesores y los Locales. Además es el encargado de confeccionar los horarios docentes.

#### 3.5.2 Diagrama de Casos de Uso del Sistema

Un diagrama de casos de uso del sistema representa gráficamente a los procesos y su interacción con los actores.

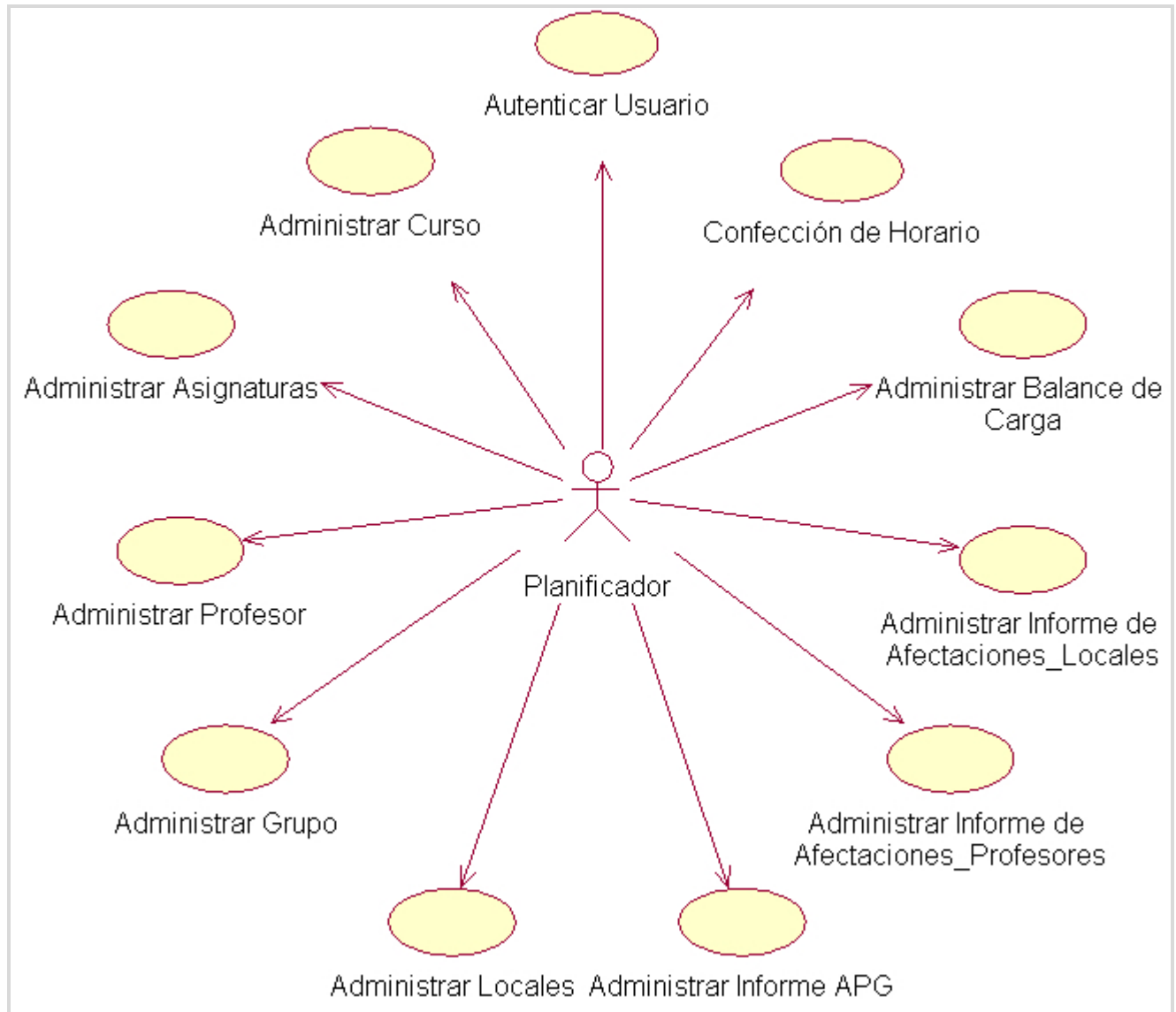


Figura 7. Diagrama de Casos de Uso del Sistema

### 3.5.3 Descripción Detallada de los Casos de Uso del Sistema

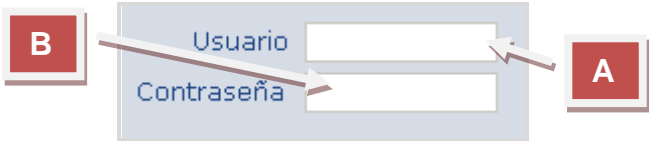
A continuación se muestran las descripciones detalladas de cuatro de los Casos de Uso del Sistema (CUS). Estos Casos de Uso se consideran críticos y resultan representativos, teniendo en cuenta que el resto de los Casos de Uso del Sistema presentan una estructura y comportamiento muy similar. El resto de las descripciones de los Casos de Uso del Sistema pueden ser consultadas en el Anexo 4: Descripción de los Casos de Uso del Sistema.

#### 3.5.3.1 Descripción Detallada del CUS Autenticar Usuario

#### 3.5.3.2

Tabla 4. Descripción Detallada del CUS Autenticar Usuario


<b>Nombre del Caso de Uso</b>	<b>Autenticar Usuario</b>
<b>Actores</b>	Planificador ( Inicia )
<b>Propósito</b>	Permitir la autenticación de los usuarios.
<b>Resumen</b>	El Caso de Uso se inicia cuando el planificador introduce los datos requeridos para acceder a las áreas restringidas de la aplicación web, estos se verifican y el caso de uso finaliza al permitir o denegar el acceso.
<b>Referencias</b>	R1
<b>Precondiciones</b>	-
<b>Poscondiciones</b>	Se habilita el acceso del usuario a las áreas restringidas de la aplicación o se deniega el acceso.
<b>Curso Normal de los Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
1. El usuario entra su nombre de usuario(A) y contraseña (B).	1.1 El sistema busca el nombre de usuario. 1.2 En caso de existir el usuario el sistema compara la contraseña. 1.3 En caso de que la contraseña sea correcta, el sistema permite el acceso a las áreas restringidas.
<b>Curso alternativo</b>	
	1.2 En caso de no existir el usuario se emite un mensaje de usuario

	incorrecto. 1.3 En caso de que la contraseña no sea correcta se emite un mensaje de contraseña incorrecta.
<b>Prototipos de Interfaz:</b>	
 <p>Pantalla 1. Formulario de Autenticación.</p>	
<b>Prioridad:</b>	Crítico

### 3.5.3.3 Descripción Detallada del CUS Administrar Curso

Tabla 5. Descripción Detallada del CUS Administrar Curso

<b>Nombre del Caso de Uso</b>	<b>Administrar Curso</b>
<b>Actores</b>	Planificador ( Inicia )
<b>Propósito</b>	Permitir crear un nuevo curso, modificar los datos de un curso existente o eliminar un curso determinado.
<b>Resumen</b>	El Caso de Uso se inicia cuando el planificador selecciona la opción “Administrar Curso, lo que le permite interactuar en tres escenarios: “Registrar Curso”, “ Actualizar Curso” y” Eliminar Curso”.
<b>Referencias</b>	R2
<b>Precondiciones</b>	Usuario del sistema ya autenticado.
<b>Poscondiciones</b>	Se registra un nuevo curso, se actualizan los datos de un curso determinado o se elimina un curso seleccionado.
<b>Curso Normal de los Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
1. El Planificador elige una de las siguientes opciones: Registrar Curso, Actualizar Curso o Eliminar Curso.	2. El sistema ejecuta alguna de las siguientes acciones: <ul style="list-style-type: none"> <li>a) Si el Planificador selecciona la opción Registrar Curso, el sistema muestra la sección “Registrar Curso”.</li> <li>b) Si el Planificador selecciona la opción Actualizar Curso, el sistema muestra la sección “Actualizar Curso”.</li> <li>c) Si el Planificador selecciona la opción Eliminar Curso, el sistema</li> </ul>

	muestra la sección “Eliminar Curso”.
<b>Sección “Registrar Curso”</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
3. El Planificador introduce los datos del curso y selecciona la opción “Registrar Curso”.	<p><b>3.1</b> El sistema verifica que los datos introducidos por el Planificador estén correctos y que todos los campos requeridos obligatoriamente hayan sido establecidos.</p> <p><b>3.2</b> En caso de que todos los campos estén correctos el nuevo curso es registrado en la base de datos.</p>
<b>Curso alternativo</b>	
	<b>3.2</b> En caso de que todos los datos obligatorios no hayan sido introducidos por el Planificador, el sistema muestra un mensaje informando que es necesario introducir todos los datos obligatorios.
<b>Prototipos de Interfaz:</b>	
 <p style="text-align: center;"><b>Pantalla 2. Formulario de registro de cursos.</b></p>	
<b>Sección “Actualizar Curso”</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
3. El Planificador selecciona el curso que desea modificar y selecciona la opción modificar.	<b>3.1</b> El sistema muestra los datos del curso seleccionado por el Planificador y brinda la posibilidad de introducir nuevos datos, es decir de modificar los datos registrados.
4. El Planificador introduce los nuevos datos y selecciona la opción “Actualizar Curso”	<p><b>4.1</b> El sistema verifica que los datos introducidos por el Planificador estén correctos y que todos los campos requeridos obligatoriamente hayan sido establecidos.</p> <p><b>4.2</b> En caso de que todos los campos estén correctos los datos del curso son actualizados en la base de datos.</p>
<b>Curso alternativo</b>	

4.2 En caso de que todos los datos obligatorios no hayan sido introducidos por el Planificador, el sistema muestra un mensaje informando que es necesario introducir todos los datos obligatorios.

**Prototipos de Interfaz:**

Curso 2007-2008

**Datos Registrados:**

Año de Inicio: 2007      Número de Periodos: 1  
 Año de Fin: 2008

Períodos	Semanas
1	2

**Descripción:** Este es un Curso de ejemplo...

Pantalla 3. Modificar Curso. Datos registrados.

**Datos Nuevos:**

Año de Inicio: 2006 ▼      **Descripción:**  
 Año de Fin: 2007 ▼     

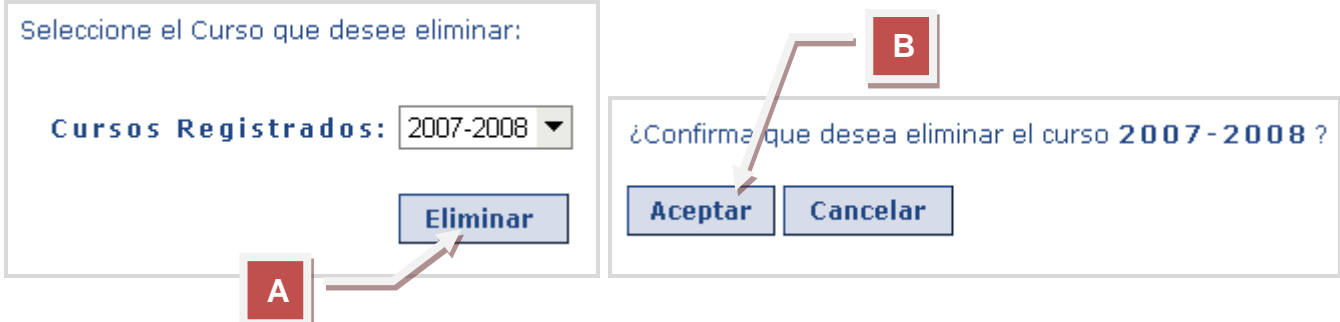
**Actualizar Curso**

Pantalla 4. Modificar Curso. Nuevos Datos.

**Sección “Eliminar Curso”**

Acciones del Actor

Respuesta del Sistema

3. El Planificador selecciona el curso que desea eliminar y selecciona la opción eliminar (A).	3.1 El sistema muestra mensaje para confirmar que el usuario desea eliminar el curso seleccionado.
4. El Planificador selecciona la opción Aceptar. (B)	4.1 El sistema elimina de la base de datos el curso seleccionado.
Prototipos de Interfaz:	
 <p style="text-align: center;">Pantalla 5. Formulario eliminar curso</p>	
Prioridad:	Crítico

### 3.5.3.4 Descripción Detallada del CUS Administrar Balance de Carga

Tabla 6. Descripción Detallada del CUS Administrar Balance de Carga

Nombre del Caso de Uso	Administrar Balance de Carga
Actores	Planificador ( Inicia )
Propósito	Permite registrar, modificar, eliminar o consultar los datos del Balance de Carga.
Resumen	El Caso de Uso se inicia cuando el planificador selecciona la opción "Administrar Balance de Carga", posteriormente el Planificador selecciona el Curso, Periodo y Año y luego puede interactuar en cuatro escenarios relacionados con el Balance de Carga: "Registrar Balance de Carga", "Actualizar Balance de Carga", "Eliminar Balance de Carga" y "Consultar Balance de Carga".
Referencias	R10
Precondiciones	Usuario del sistema ya autenticado y Curso al que se le va a administrar el Balance de Carga ya creado.

<b>Poscondiciones</b>	Se registra, actualiza o elimina el Balance de Carga del Curso-Periodo-Año seleccionado o se muestra el correspondiente Balance de Carga.
<b>Curso Normal de los Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
1. El Planificador selecciona el Curso, el Período y el Año de la cual desea administrar el Informe Balance de Carga.	1.1 El sistema brinda la posibilidad de registrar, modificar, eliminar o consultar el Balance de Carga correspondiente al Curso, el Período, Año y Semana.
2. El Planificador elige una de las siguientes opciones: Registrar, Actualizar, Eliminar o Consultar Balance de Carga.	2.1 El sistema ejecuta alguna de las siguientes acciones: a) Si el Planificador selecciona la opción Registrar Balance de Carga, el sistema muestra la sección "Registrar Balance de Carga". b) Si el Planificador selecciona la opción Actualizar Balance de Carga, el sistema muestra la sección "Actualizar Balance de Carga". c) Si el Planificador selecciona la opción Eliminar Balance de Carga, el sistema muestra la sección "Eliminar Balance de Carga". d) Si el Planificador selecciona la opción Consultar Balance de Carga, el sistema muestra la sección "Balance de Carga".
<b>Sección "Registrar Balance de Carga"</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
3. El Planificador selecciona la opción "Registrar Balance de Carga" y Selecciona una Semana.	3.1 Si no existe ningún Balance de Carga en el Curso, Período, Año, Semana, el sistema brinda la posibilidad de registrar Balance de Carga.
4. El Planificador introduce los datos del Balance de Carga y selecciona la opción "Registrar".	4.1 El sistema verifica que los datos introducidos por el Planificador estén correctos y que todos los campos requeridos obligatoriamente hayan sido establecidos. 4.2 En caso de que todos los campos estén correctos y no existan errores o inconsistencias en el Balance de Carga, este es registrado en la base de datos.
<b>Curso alternativo</b>	

**3.1** Si está registrado un Balance de Carga muestra un mensaje informando que ya existe un Balance de Carga registrado, brindando al Planificador la opción de eliminarlo y crear uno nuevo o modificar el existente.

**4.2** En caso de que todos los datos obligatorios no hayan sido introducidos por el Planificador, el sistema muestra un mensaje informando que es necesario introducir todos los datos obligatorios y además muestra el listado de errores o inconsistencias del Balance de Carga.

**Prototipos de Interfaz:**

**Registrando Balance de Carga**

**Curso 2007-2008 Período # 1 Semana 1**

**Seleccione:**

**Asignaturas**

**Encuentro**

Programación 1

Conferencia

**Añadir**

**Vista Previa del Balance de Carga**

Asignatura	Encuentro	Eliminar
Matemática 1	Conferencia	X
Matemática 1	Clase Práctica	X
Marxismo	Conferencia	X
Marxismo	Conferencia	X
Práctica Profesional	Conferencia	X
Práctica Profesional	Clase Teórico-Práctica	X
Práctica Profesional	Laboratorio	X

**Registrar**

**Cancelar**

**Pantalla 6. Registrar Balance de Carga**

**Sección “Actualizar Balance de Carga”**

Acciones del Actor	Respuesta del Sistema																								
3. El Planificador selecciona la opción "Actualizar Balance de Carga".	3.1 El sistema muestra el Balance de Carga y brinda la posibilidad de introducir nuevos datos, es decir de modificar los datos registrados.																								
4. El Planificador introduce los nuevos datos y selecciona la opción "Actualizar".	4.1 El sistema verifica que los datos introducidos por el Planificador estén correctos y que todos los campos requeridos obligatoriamente hayan sido establecidos. 4.2 En caso de que todos los campos estén correctos y no existan errores o inconsistencias en el Balance de Carga, este es actualizado en la base de datos.																								
<b>Curso alternativo</b>																									
	4.2 En caso de que todos los datos obligatorios no hayan sido introducidos por el Planificador o existan inconsistencias o errores, el sistema muestra un mensaje informando que es necesario introducir todos los datos obligatorios y además muestra el listado de errores o inconsistencias en el Balance de Carga.																								
<b>Prototipos de Interfaz:</b>																									
<p style="text-align: center;"><b>Actualizando Balance de Carga</b></p> <p style="text-align: center;"><b>Curso 2007-2008 Período # 1 Semana 1</b></p> <p style="text-align: center;"><b>Datos Registrados</b></p> <table border="1" data-bbox="347 1192 1268 1543"> <thead> <tr> <th></th> <th><b>Asignatura</b></th> <th><b>Encuentro</b></th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Matemática 1</td> <td>Conferencia</td> </tr> <tr> <td>2</td> <td>Matemática 1</td> <td>Clase Práctica</td> </tr> <tr> <td>3</td> <td>Marxismo</td> <td>Conferencia</td> </tr> <tr> <td>4</td> <td>Marxismo</td> <td>Conferencia</td> </tr> <tr> <td>5</td> <td>Práctica Profesional</td> <td>Conferencia</td> </tr> <tr> <td>6</td> <td>Práctica Profesional</td> <td>Clase Teórico-Práctica</td> </tr> <tr> <td>7</td> <td>Práctica Profesional</td> <td>Laboratorio</td> </tr> </tbody> </table> <p style="text-align: center;"><b>Pantalla 7. Actualizando Balance de Carga.</b></p>			<b>Asignatura</b>	<b>Encuentro</b>	1	Matemática 1	Conferencia	2	Matemática 1	Clase Práctica	3	Marxismo	Conferencia	4	Marxismo	Conferencia	5	Práctica Profesional	Conferencia	6	Práctica Profesional	Clase Teórico-Práctica	7	Práctica Profesional	Laboratorio
	<b>Asignatura</b>	<b>Encuentro</b>																							
1	Matemática 1	Conferencia																							
2	Matemática 1	Clase Práctica																							
3	Marxismo	Conferencia																							
4	Marxismo	Conferencia																							
5	Práctica Profesional	Conferencia																							
6	Práctica Profesional	Clase Teórico-Práctica																							
7	Práctica Profesional	Laboratorio																							

**Nuevos Datos:**

**Seleccione:**

**Asignaturas**      **Encuentro**

Programación 1      Conferencia      **Añadir**

**Vista Previa del Balance de Carga**

Asignatura	Encuentro	Eliminar
Matemática 1	Conferencia	X
Matemática 1	Clase Práctica	X
Marxismo	Conferencia	X
Marxismo	Conferencia	X
Práctica Profesional	Conferencia	X
Práctica Profesional	Clase Teórico-Práctica	X
Práctica Profesional	Laboratorio	X

**Actualizar**      **Cancelar**

**Pantalla 8. Actualizando Balance de Carga.**

**Sección “Eliminar Balance de Carga”**

Acciones del Actor	Respuesta del Sistema
3. El Planificador selecciona la opción Eliminar Balance de Carga.	3.1 Es sistema muestra mensaje para confirmar que el usuario desea eliminar el Balance de Carga.
4. El Planificador selecciona la opción Aceptar.	4.1 El sistema elimina el Balance de Carga y muestra un mensaje informándole al Planificador que ha sido efectuada la eliminación del Balance de Carga.

**Prototipos de Interfaz:**

¿Confirma que desea eliminar el Balance de Carga del Curso 2007-08 , Período # 1 ?

**Aceptar**

**Cancelar**

Pantalla 9. Eliminar Balance de Carga

Sección “Mostar Balance de Carga”	
Acciones del Actor	Respuesta del Sistema
3. El Planificador selecciona mostrar el Balance de Carga.	3.1 En caso de que exista el sistema le muestra el Balance de Carga con todos sus datos.
Curso alternativo	
	3.1 En caso de que no exista balance registrado se emite un mensaje informando que no existe.
Prioridad:	Crítico

### 3.5.3.5 Descripción Detallada del CUS Confección de Horarios

Tabla 7. Descripción Detallada del CUS Confección de Horarios

<b>Nombre del Caso de Uso</b>	Confección de Horarios
<b>Actores</b>	Planificador ( Inicia )
<b>Propósito</b>	Permitir generar y mostrar los horarios generados.
<b>Resumen</b>	El Caso de Uso se inicia cuando el planificador selecciona alguna de las dos opciones disponibles en el Área de Horario, puede ser “Generar Horario” o “Mostrar Horario”. El Caso de Uso va a permitir generar los horarios docentes y posteriormente puede mostrarlos o imprimirlos.
<b>Referencias</b>	R11, R12
<b>Precondiciones</b>	Usuario del sistema ya autenticado y Curso al que se le va a confeccionar el Horario creado.
<b>Poscondiciones</b>	Se genera un nuevo Horario, o se muestra un horario deseado.
<b>Curso Normal de los Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>

<p>1. El Planificador selecciona una de las siguientes opciones: “Generar Horario” o “Mostrar Horario”</p>	<p>1.1 El sistema ejecuta alguna de las siguientes acciones: Si el Planificador selecciona la opción “Generar Horario”, el sistema mostrará la sección “Generar Horario”. Si el Planificador selecciona la opción “Mostrar Horario”, el sistema mostrará la sección “Mostrar Horario”.</p>
<p><b>Sección “Generar Horario”</b></p>	
<p><b>Acciones del Actor</b></p>	<p><b>Respuesta del Sistema</b></p>
<p>2. El Planificador selecciona el Curso, el Período y la Semana de la cual desea generar el horario y selecciona la opción “Generar”</p>	<p>2.1 El sistema genera el horario. 2.2 En caso de que el horario haya sido confeccionado satisfactoriamente el sistema muestra al Planificador un mensaje informándole que el horario ha sido confeccionado.</p>
<p><b>Curso alternativo</b></p>	
	<p>2.2 Si el horario no pudo ser generado, por problemas en los datos, o demasiadas restricciones, entonces el sistema mostrará un listado con todos los errores que ocurrieron durante la generación del horario</p>
<p><b>Sección “Mostrar Horarios”</b></p>	
<p><b>Acciones del Actor</b></p>	<p><b>Respuesta del Sistema</b></p>
<p>2. El Planificador selecciona el Curso, Período, Semana y Grupo del cual desea mostrar el horario y selecciona la opción “Mostrar”.</p>	<p>2.1 Si existe el horario el sistema lo muestra</p>
<p><b>Curso alternativo</b></p>	
	<p>2.1 Si no existe el horario, el sistema muestra al Planificador un mensaje informándole que no existe el horario solicitado.</p>
<p><b>Prototipos de Interfaz:</b></p>	

Grupo 9101 Semana 1							
Hora	T	L	M	Mi	J	V	S
8:00AM	1	M1(A2)	Marx(A2)		M1(A3)		
9:45AM	2	I1(Lab210)	P1(A2)	II(A2)	P1(A3)	HI(A3)	
11:30AM	3			I1(A2)			
1:30PM	4				PP(lab212)		
3:15PM	5	EF	PP(lab212)		PP(Lab212)		
5:00PM	6		PP(lab212)				

Pantalla 10. Horario generado

**Prioridad:** Crítico

### 3.6 Conclusiones

En este capítulo se planteó una visión general del negocio planteado a esta investigación, logrando modelarse los actores, trabajadores, entidades y casos de uso del negocio. Además se enumeran todos los requisitos funcionales con los que cumplirá el sistema así como los requerimientos no funcionales que debe cumplir la aplicación. También se muestra el diagrama de casos de uso del sistema y se describen detalladamente varios de los casos de uso del sistema que resultan críticos para el funcionamiento de la aplicación.

# CAPÍTULO 4

## Construcción de la Solución Propuesta

### **4.1 Introducción**

Este capítulo abordará temas medulares para la construcción del sistema propuesto, como los diagramas de clases web, el diagrama de clases persistentes y el modelo de datos. Se plantean además las principales pautas de diseño que se siguieron en la interfaz gráfica de la aplicación. También son explicados varios aspectos referidos a la implementación del sistema, explicando las características de la aplicación web y las características del algoritmo de IA. Se muestran además el Diagrama de Despliegue y el Modelo de Implementación.

## 4.2 Diagramas de Clases Web

### 4.2.1 Diagrama de Clases Web del CUS Autenticar Usuario

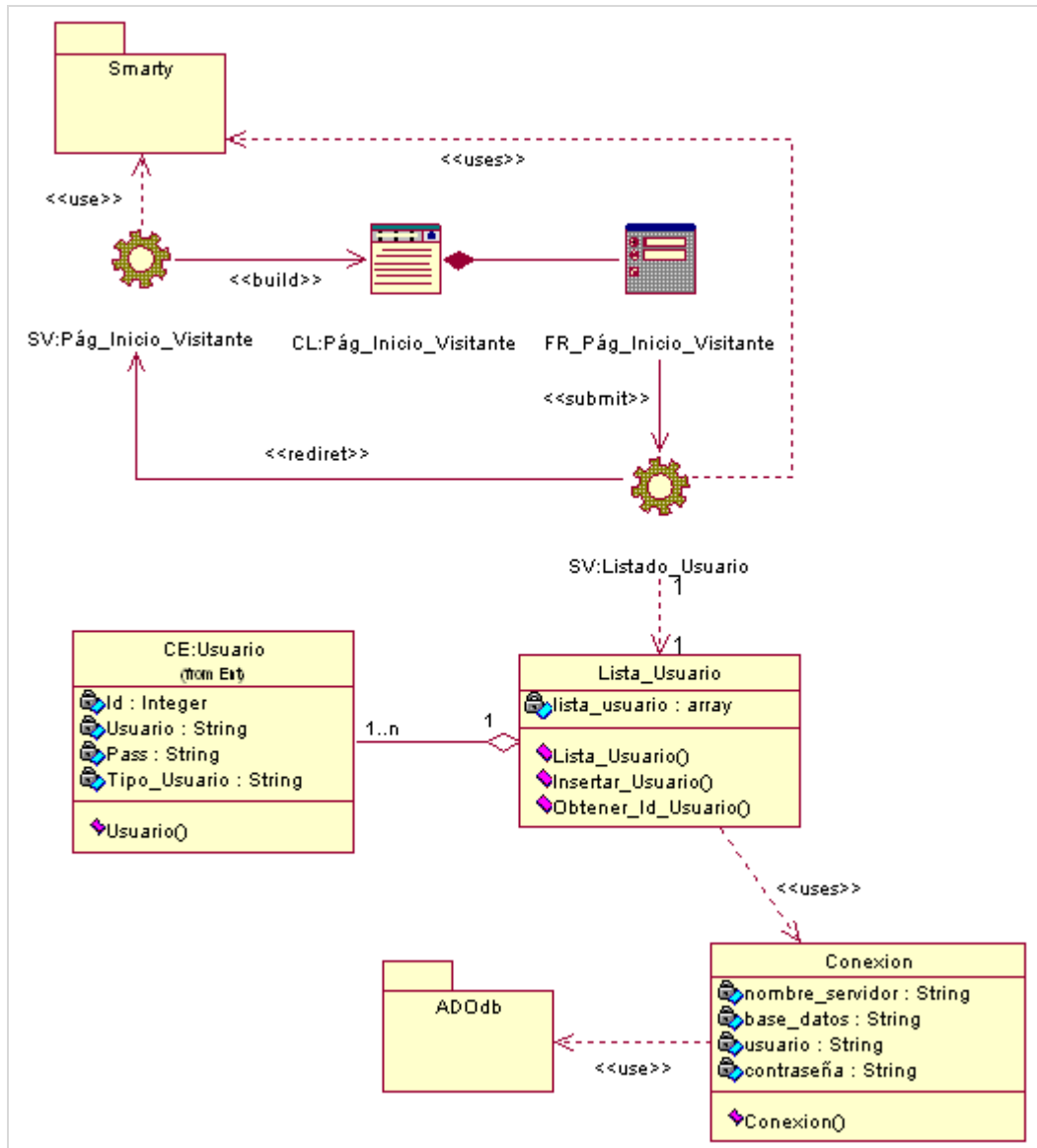


Figura 8. Diagrama de Clases Web del CUS Autenticar Usuario

### 4.2.2 Diagrama de Clases Web del CUS Administrar Curso

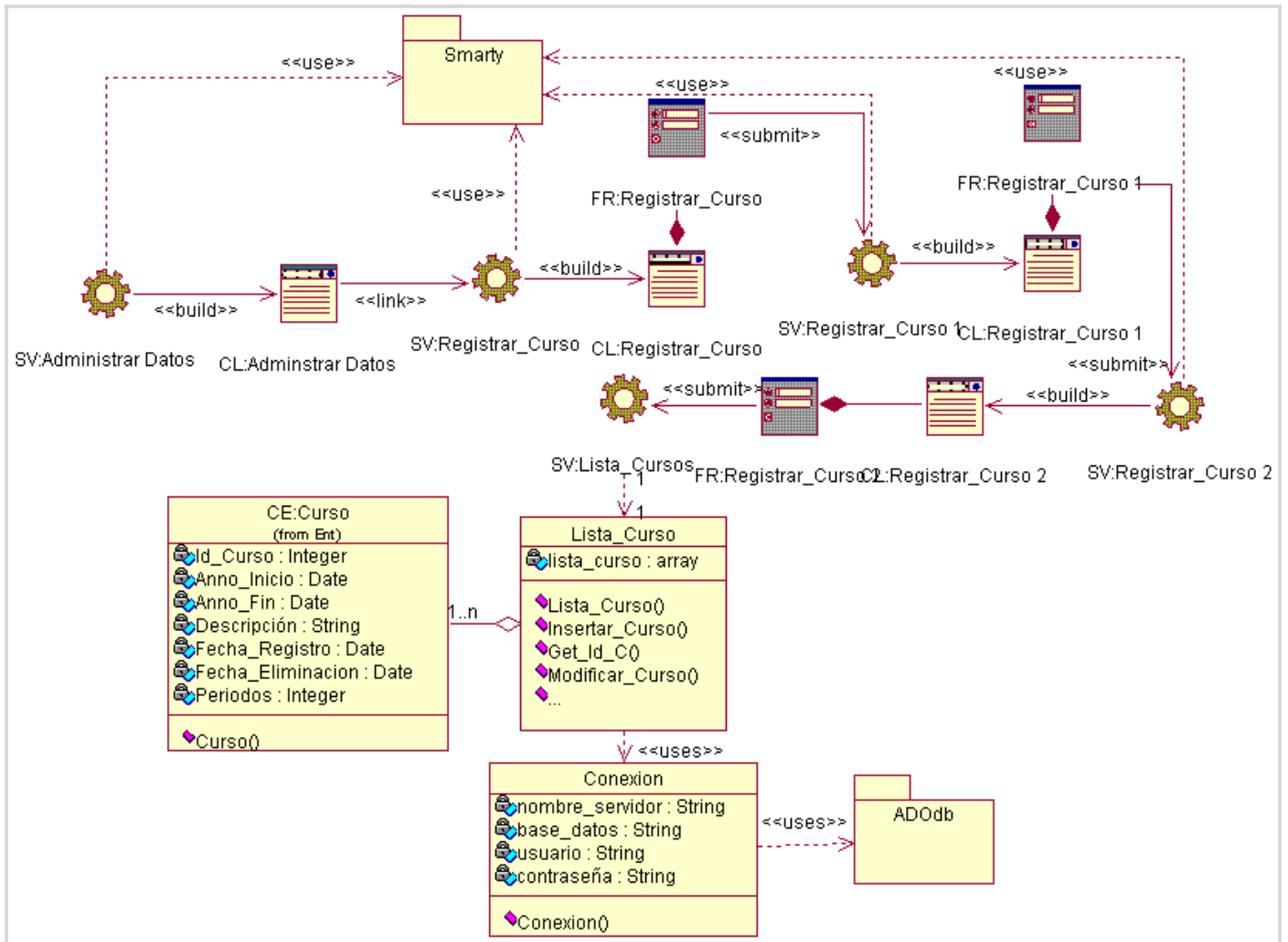


Figura 9. Diagrama de Clases Web del CUS Administrar Curso (Sección Registrar Curso)

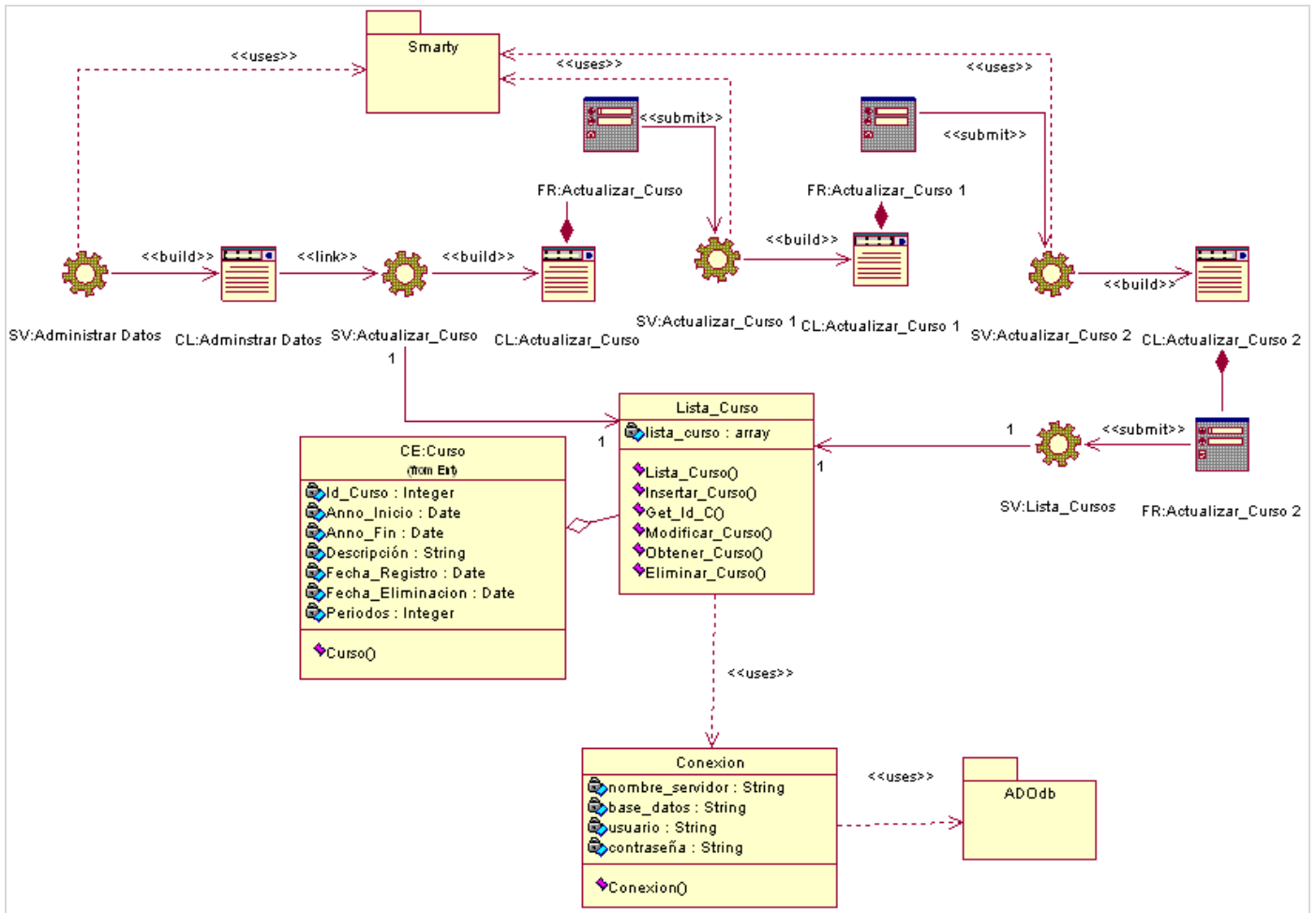


Figura 10. Diagrama de Clases Web del CUS Administrar Curso (Sección Actualizar Curso)

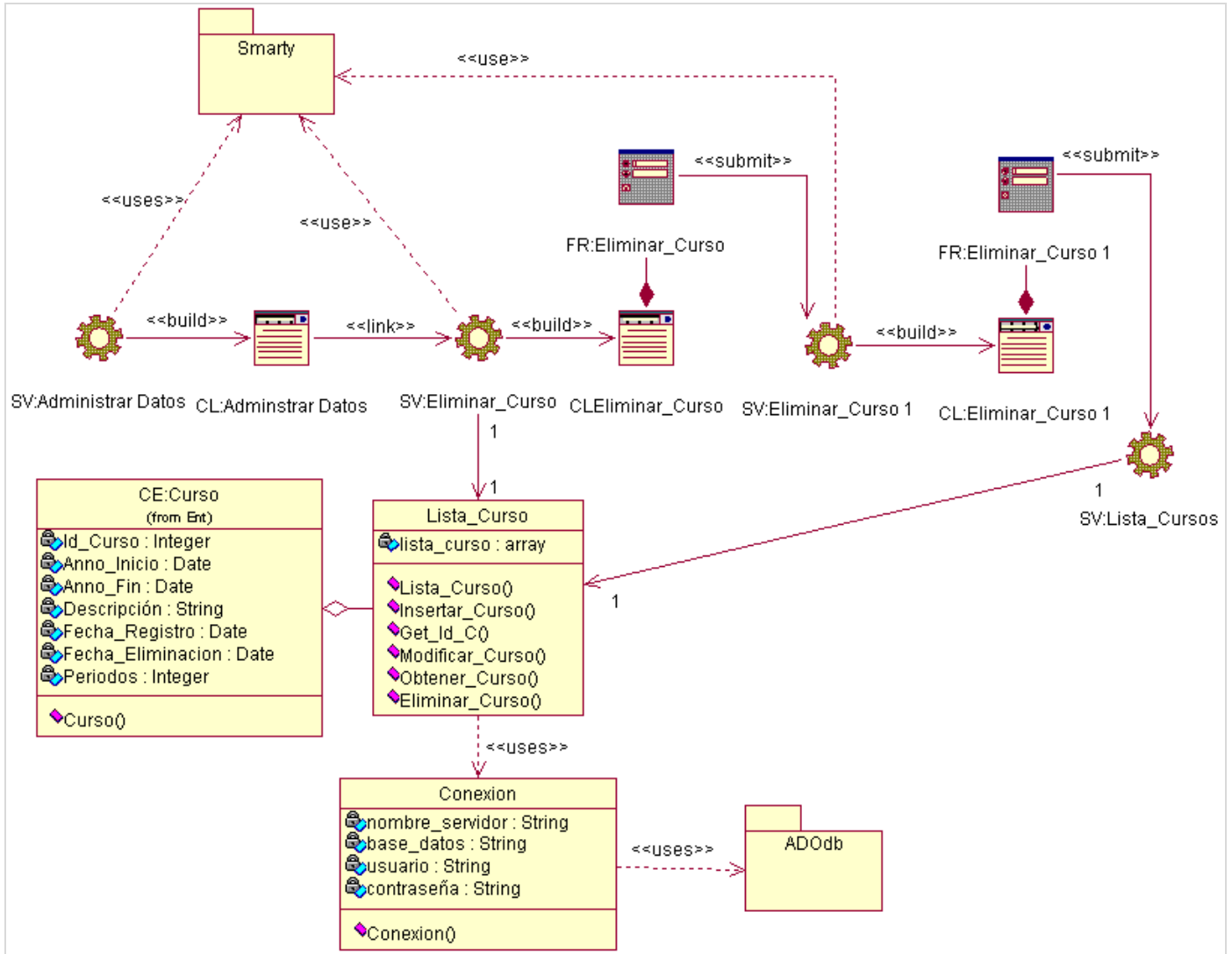


Figura 11. Diagrama de Clases Web del CUS Administrar Curso (Sección Eliminar Curso)

### 4.2.3 Diagrama de Clases Web del CUS Administrar Balance de Carga

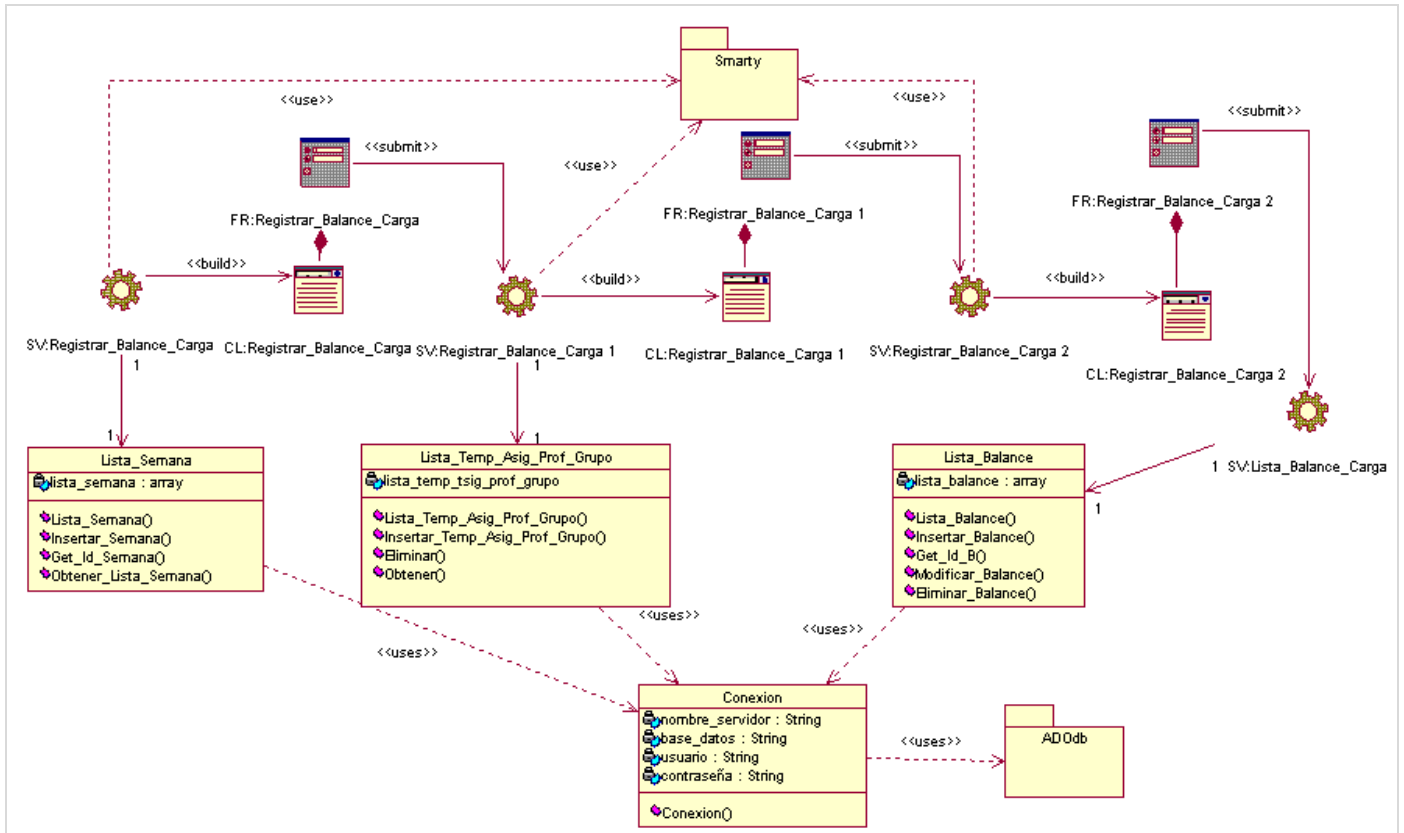


Figura 12. Diagrama de Clases Web del CUS Administrar Balance de Carga (Sección Registrar Balance de Carga)

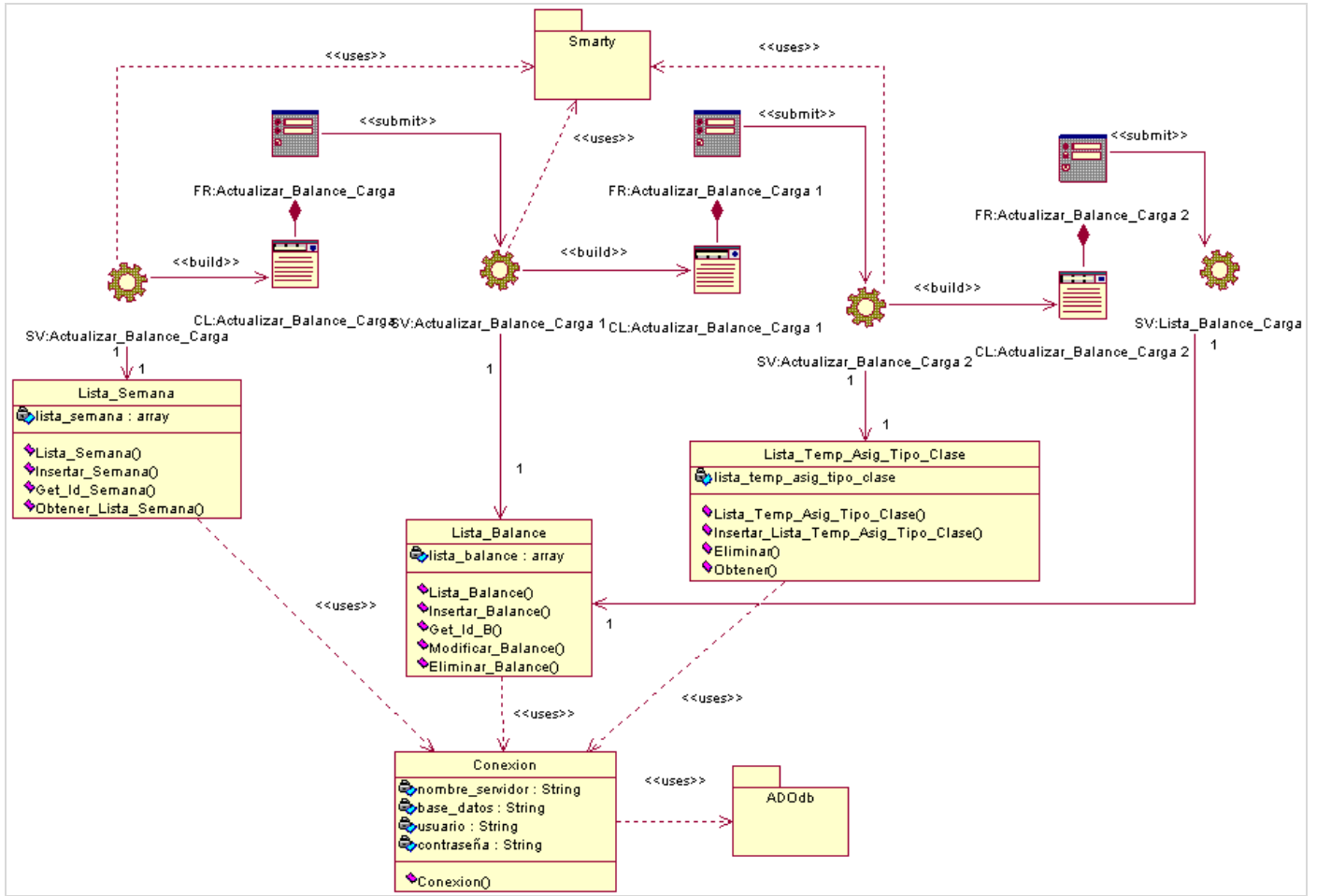


Figura 13. Diagrama de Clases Web del CUS Administrar Balance de Carga (Sección Actualizar Balance de Carga)

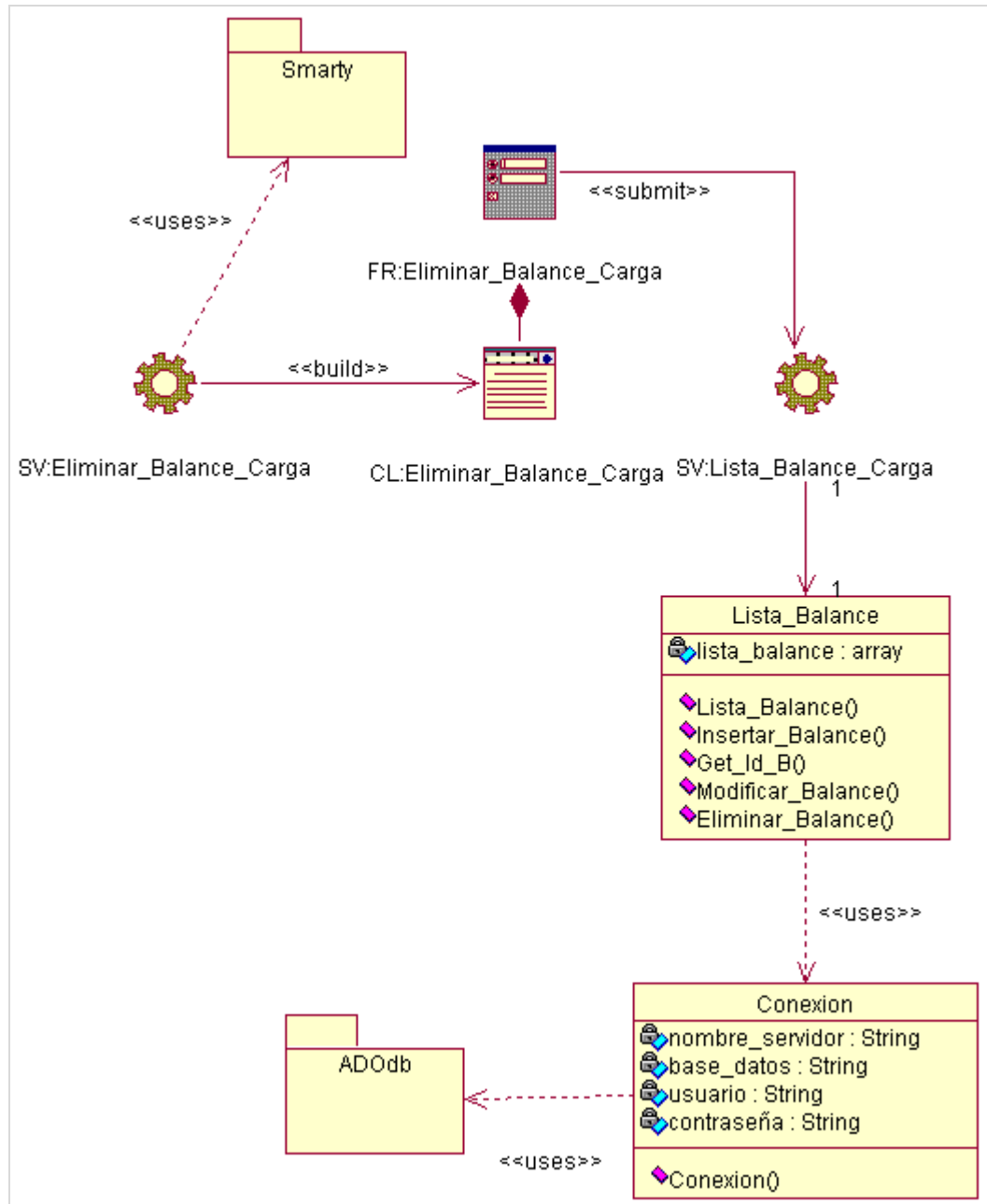


Figura 14. Diagrama de Clases Web del CUS Administrar Balance de Carga (Sección Eliminar Balance de Carga)

### 4.2.4 Diagrama de Clases Web del CUS Confección de Horarios

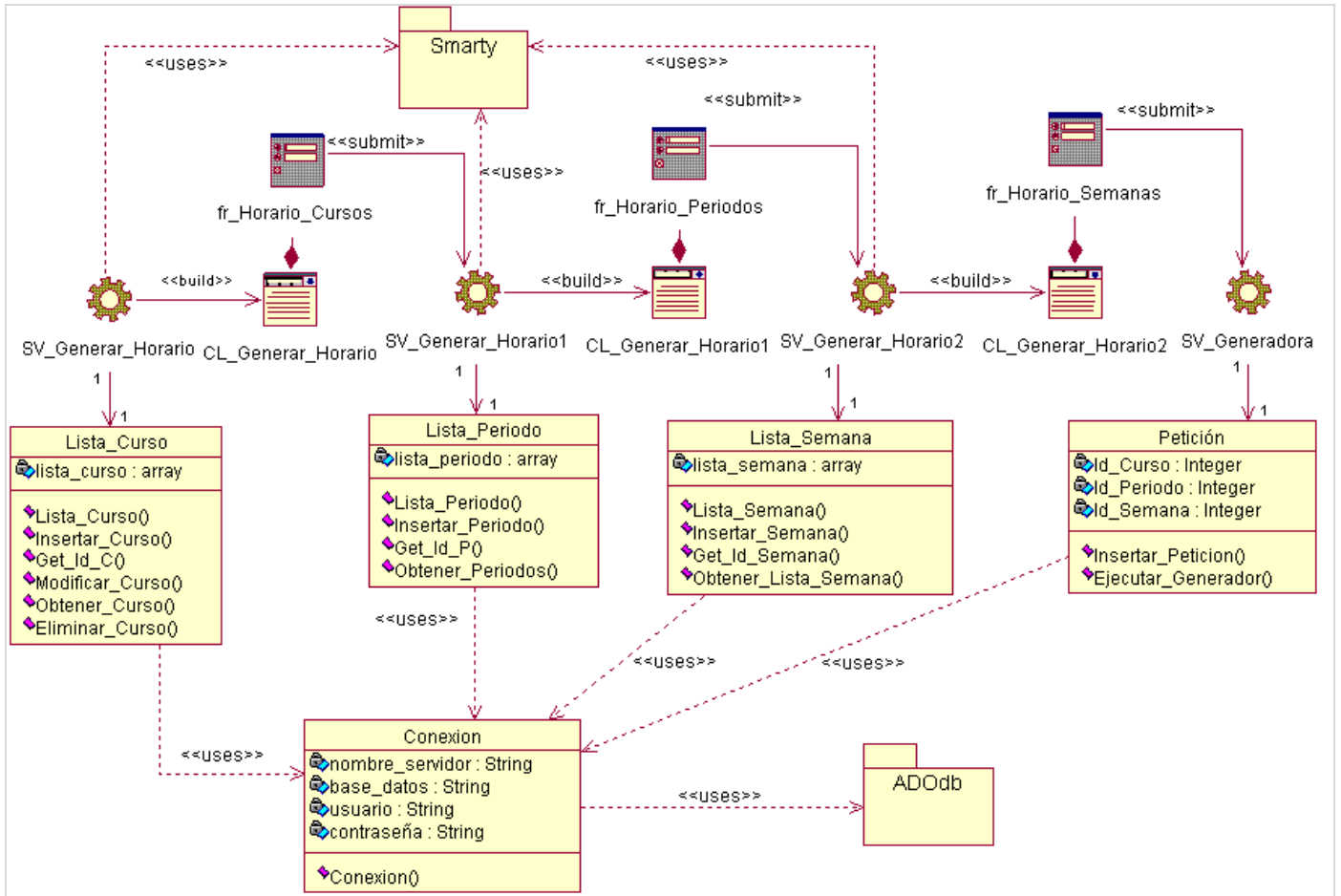


Figura 15. Diagrama de Clases Web del CUS Confección de Horarios

## 4.3 Principios de diseño

### 4.3.1 Estándares de la interfaz de la aplicación

La interfaz gráfica de la aplicación fue concebida con un ambiente sencillo, amigable y de navegación fácil e intuitiva para el usuario. Dada la funcionalidad y el objetivo del sistema, este no presenta demasiadas imágenes o elementos llamativos que puedan distraer la atención, por estas razones se emplean colores sobrios y fuentes legibles y comunes. Se presentan algunas imágenes, pero utilizadas como íconos, para facilitar la navegación y la interacción del usuario con las diferentes áreas dentro de la aplicación. Todas las páginas de la aplicación presentan una estructura básica similar, formada por 5 elementos: “Encabezado”, “Barra de Navegación Horizontal”, “Barra de Navegación Vertical”, “Cuerpo” y “Pie de Página”. A continuación se muestran estos elementos:

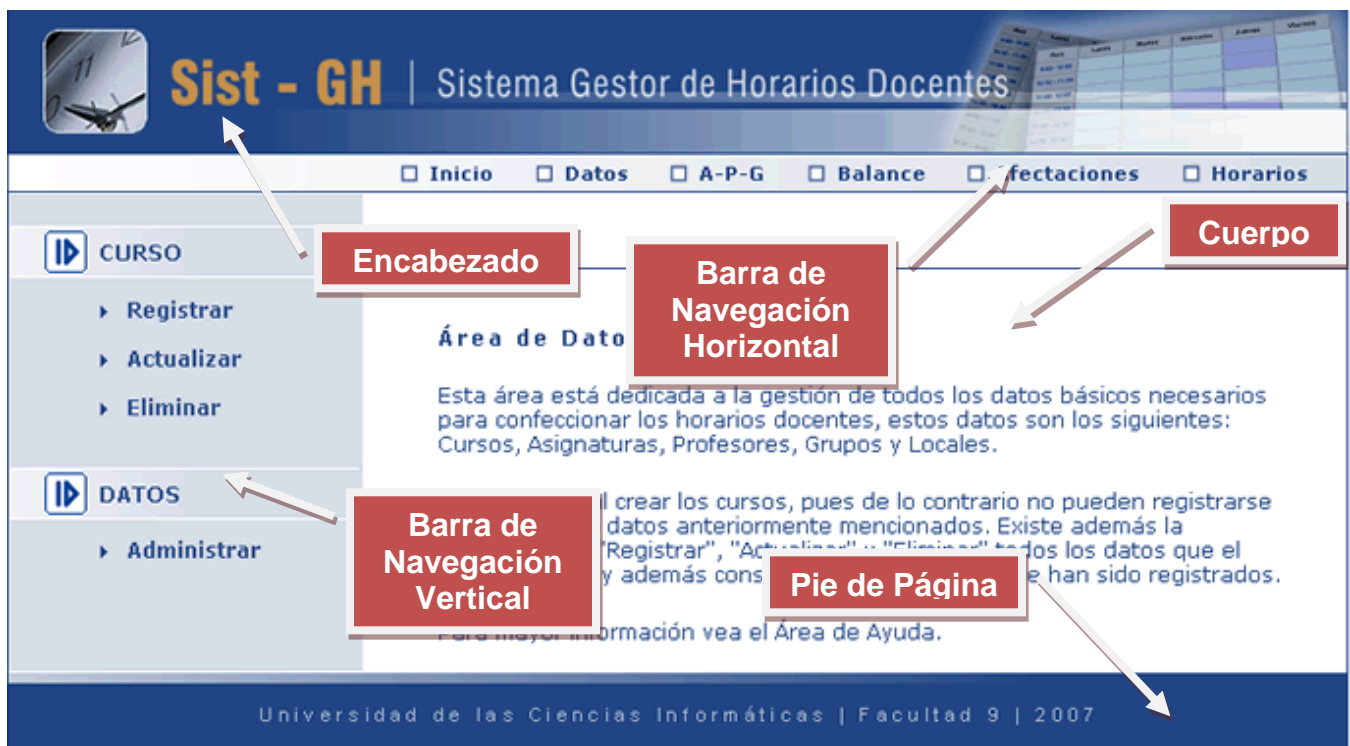


Figura 16. Diagrama de Clases Web del CUS Confección de Horarios

La fuente empleada es **Verdana** de tamaño **12** en las variantes normal y **negrita** y de color **#274C87** (RGB R31 G73 B125).

La gama de colores utilizada en el sitio varía desde el blanco para el fondo de las páginas, naranja para el título, y varias tonalidades de grises y azules para el reto de los elementos. A continuación se muestran los colores empleados:

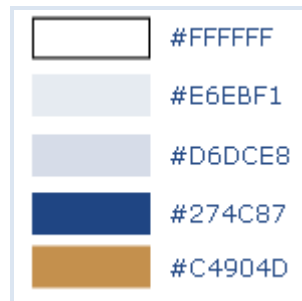


Figura 17. Colores empleados en la aplicación

El diseño de la página fue realizado utilizando CCS, tecnología muy utilizada actualmente para el diseño web, que permite de una manera fácil y muy eficiente dar formato a elementos HTML.

El uso de CSS brinda varias ventajas, dentro de las que se destaca el control del diseño pues el hecho de separar el contenido del diseño, resulta muy útil para modificar aspectos del diseño de un sitio web, ya se puede cambiar dicho aspecto en todas las páginas del sitio modificando únicamente la hoja de estilo, con lo que se logra además de un ahorro de tiempo una mayor uniformidad en el diseño. Además el empleo adecuado de hojas de estilo garantiza la adecuada visualización del sitio en los distintos navegadores al ser las CSS un estándar definido por la W3C <sup>6</sup>(World Wide Web Consortium).

La aplicación fue concebida para una correcta visualización de todos los elementos de la página tanto para resoluciones de 800X600 o mayores.

<sup>6</sup> El World Wide Web Consortium, abreviado W3C, es un consorcio internacional que produce estándares para la World Wide Web.

### **4.3.2 Estándares de Codificación**

El empleo de estándares de codificación resulta de vital importancia para el desarrollo de las aplicaciones pues facilitan el entendimiento del código fuente permitiendo su reutilización y futuro mantenimiento. Además es muy útil el empleo de comentarios que expliquen el objetivo del código y esclarezcan el significado de aquellos aspectos del código fuente que puedan resultar dudosos.

Con el objetivo de facilitar el mantenimiento del software y la adecuada organización y limpieza de su código se emplearon los siguientes estándares de codificación:

- Empleo de comentarios en todas las declaraciones de clases y funciones más complejas.
- Utilización nombres de clases significativos, que expresen total o parcialmente su significado.
- Utilización del carácter “\_”, para los nombres de las clases y las variables, en substitución del carácter espacio para hacer el código más legible.
- Los nombres de las clases se escribieron con mayúsculas.
- Organización del código de forma estructurada, en bloques de código, para una mejor lectura y comprensión del mismo.

## **4.4 Concepción general de la ayuda**

El hecho de que las aplicaciones web incluyan ayuda para guiar al usuario en las tareas que necesite ejecutar resulta fundamental para el éxito de cualquier sistema. En este caso particular el usuario de la aplicación está familiarizado con los términos y elementos del negocio presentes en la aplicación, no obstante es necesario que conozca de forma detallada la manera en que la aplicación da respuesta a las situaciones que él resolvía de manera manual y ahora son automatizadas. Con este objetivo la Ayuda está concebida dentro del diseño del sitio, es accesible desde la página inicial o de bienvenida y puede ser accedida sin necesidad de autenticación. La Ayuda se mostrará en una página HTML en una nueva ventana distinta de la de la aplicación principal, con el objetivo de que el usuario pueda consultarla cuando la necesite mientras navega por la aplicación.

La Ayuda estará estructurada de manera sencilla, contando con un índice temático, dividido por las distintas áreas de la aplicación, y los elementos más significativos y complejos del funcionamiento del sistema. Además contará con abundante información sobre las condiciones y restricciones de datos sobre las que fue concebida la aplicación y el algoritmo de generación de horarios.





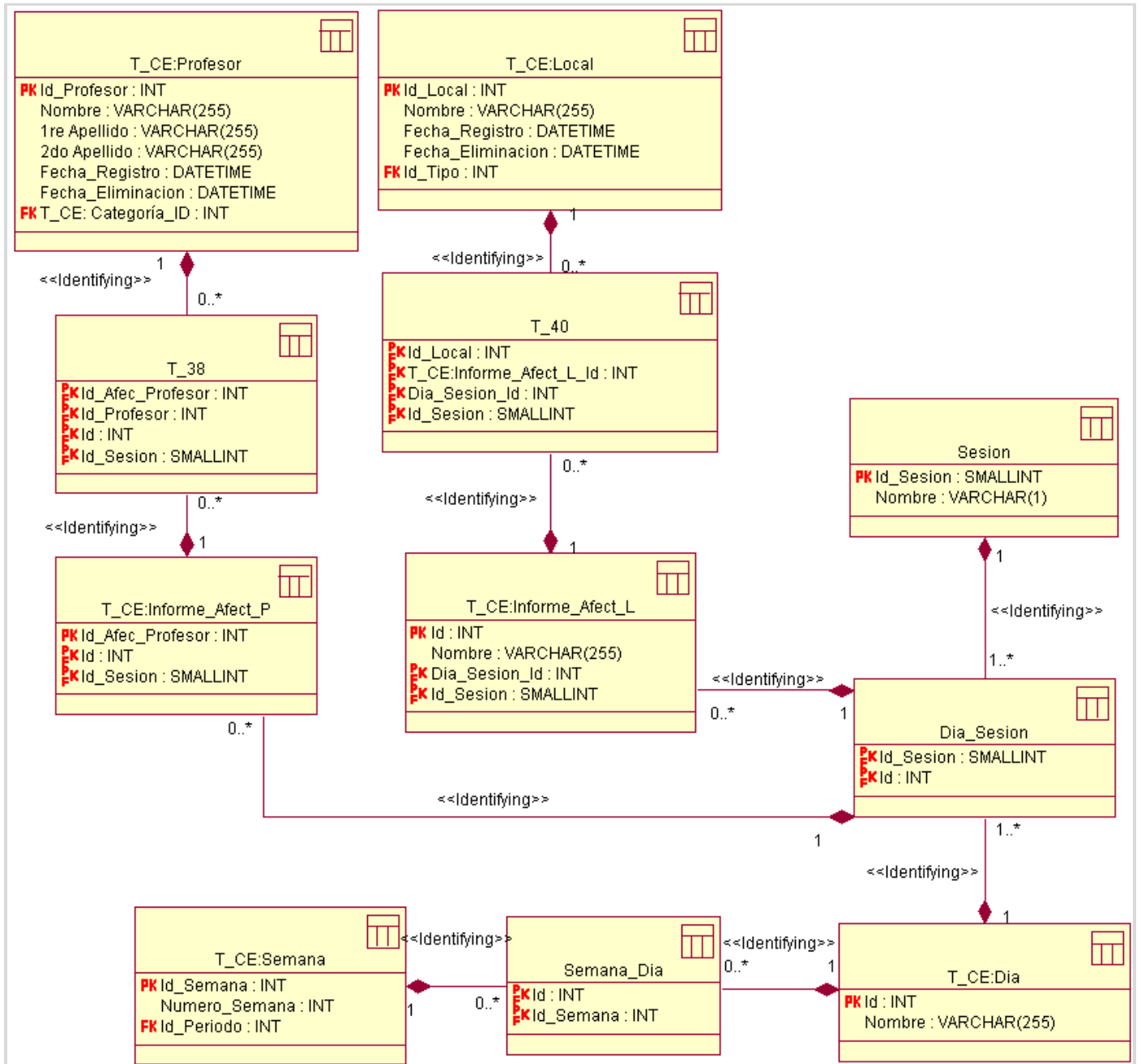


Figura 20. Modelo de Datos. Sección 2 (Afectaciones)

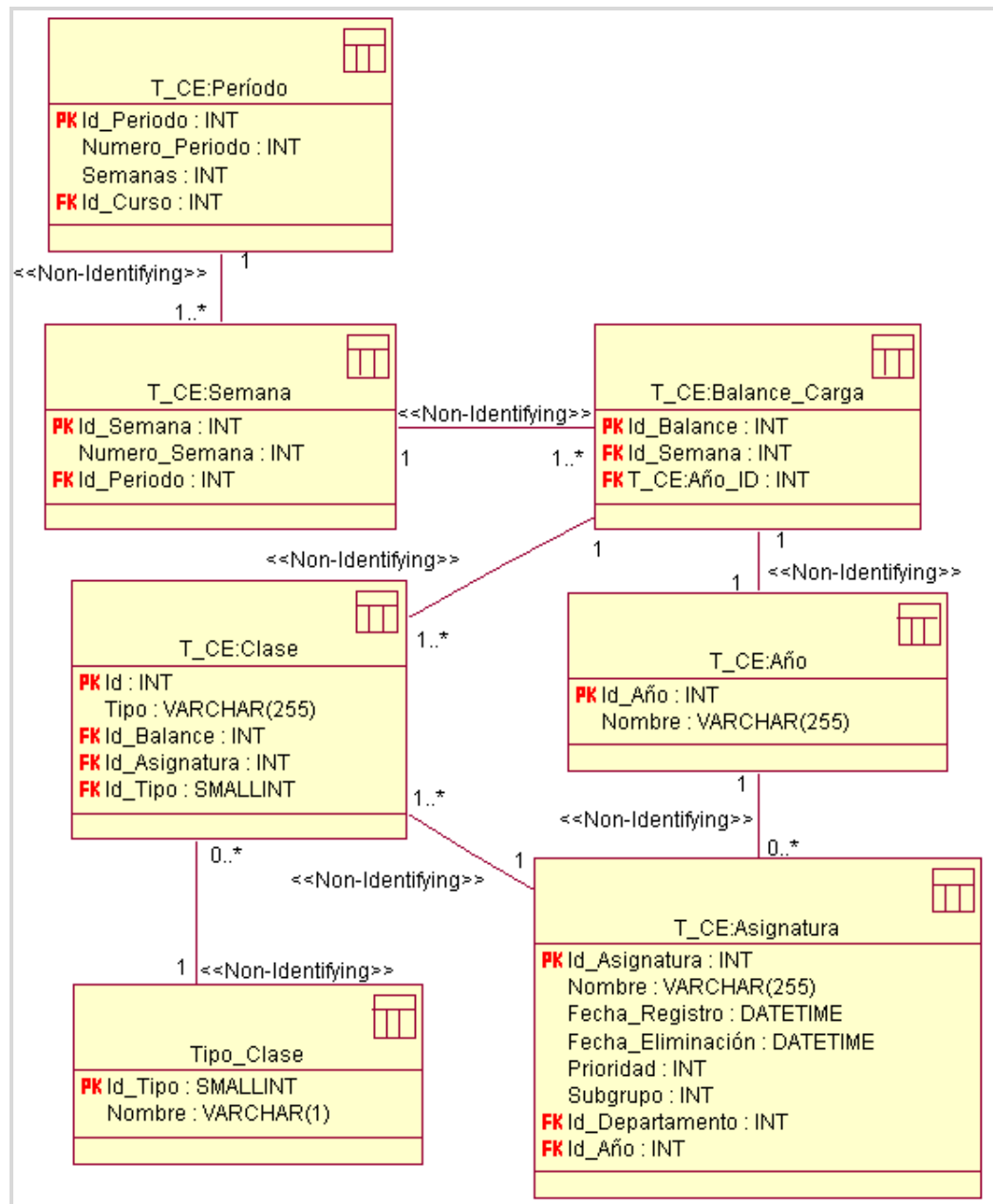


Figura 21. Modelo de Datos. Sección 3 (Balance de Carga)

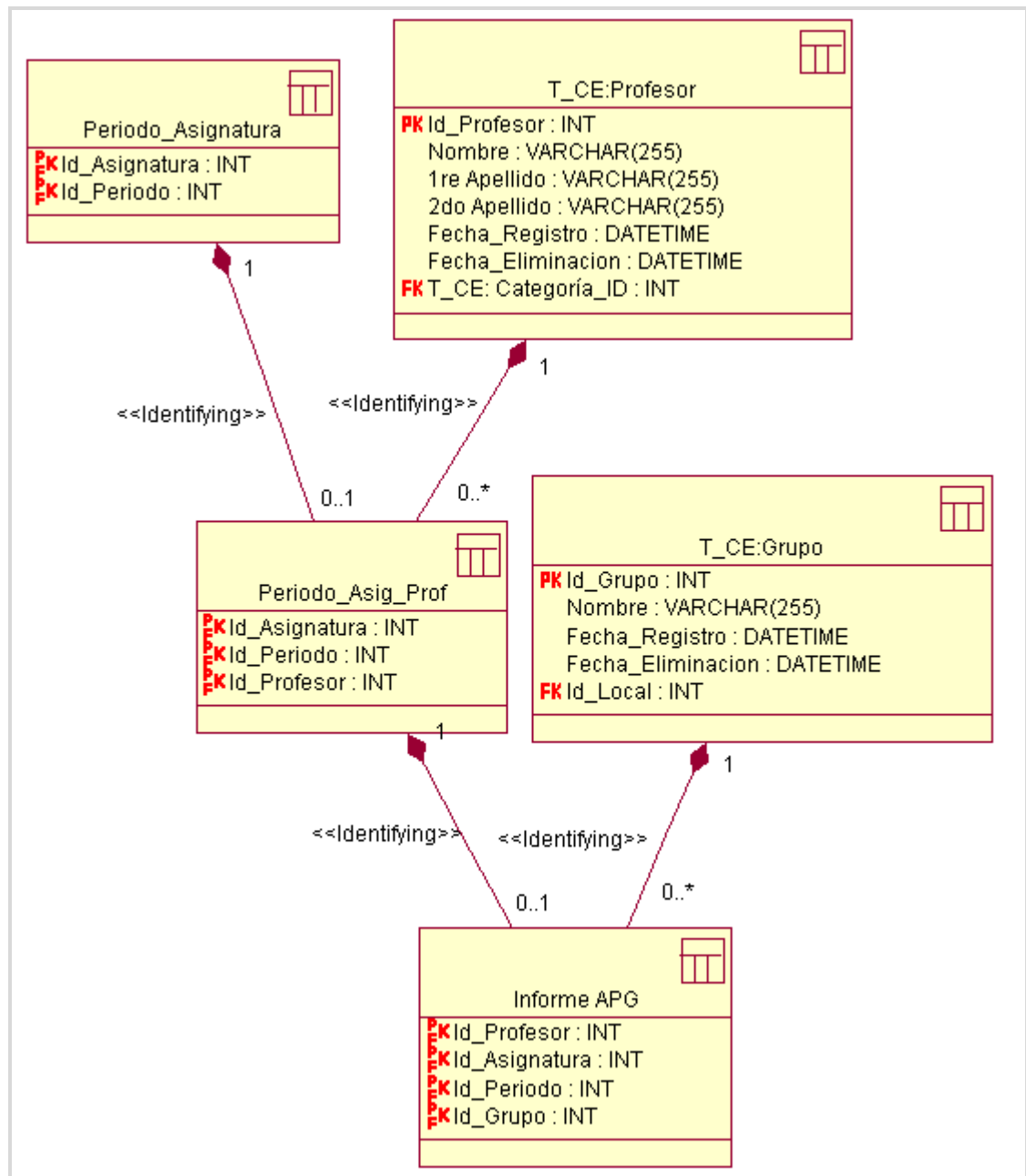


Figura 22. Modelo de Datos. Sección 4 (Informe A-P-G)

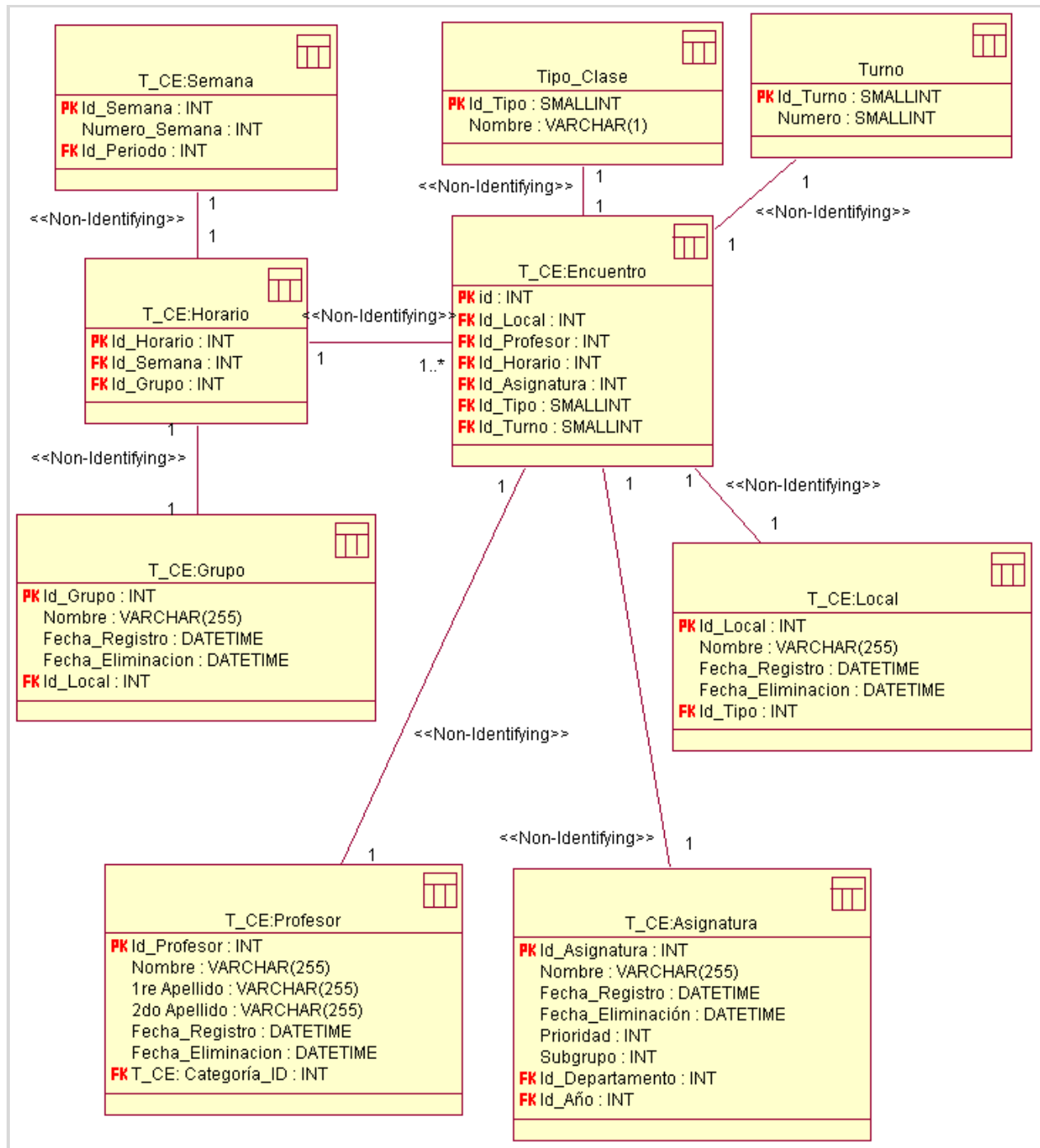


Figura 23. Modelo de Datos. Sección 5 (Horario)

## 4.6 Generalidades de la Implementación

### 4.6.1 Aplicación Web

La aplicación utiliza la librería ADOdb de PHP, como abstracción de la capa de datos. ADOdb es un conjunto de librerías de bases de datos que nos permite conectarse a varias bases de datos de una manera portable. Las funciones de acceso a base de datos en PHP no están estandarizadas. Por lo que se necesita librería que esconda las diferencias entre cada API<sup>7</sup> de base de datos (encapsular las diferencias) permitiendo así cambiar fácilmente de base de datos. Actualmente ADOdb maneja muchísimas bases de datos, por ejemplo: MySQL, Oracle, Microsoft SQL Server, Sybase, Sybase SQL Anywhere, Informix, PostgreSQL, FrontBase, SQLite, Interbase (versiones de Firebird y Borland), Foxpro, Access, ADO, DB2, SAP DB and ODBC. A continuación aparecen algunas de las características especiales y ventajas de la librería ADOdb:

- Resulta fácil para programadores familiarizados con Windows, debido a que muchas de las convenciones son similares al ADO de Microsoft.
- Fue diseñada para una alta velocidad de procesamiento. Es considerada por muchos la más veloz de las librerías de abstracción de bases de datos para PHP. (23)
- Es Open Source, se distribuye bajo la licencia dual BSD y GPL y cuenta con una amplia comunidad de usuarios y desarrolladores.
- Permite el desarrollo de sistemas muy portables y escalables. Actualmente es usada por varios CMS como PostNuke, EasyPublish y aplicaciones comerciales como PhpLens.

La aplicación fue concebida teniendo en cuenta la separación Presentación/Contenido, paradigma con muchísimas ventajas para el desarrollo de aplicaciones web, al separar el trabajo de los diseñadores y los programadores. Con este objetivo se utilizó Smarty. Smarty es un motor de plantillas para PHP, más específicamente, esta herramienta facilita la manera de separar la aplicación lógica y el contenido en la presentación. (24) La programación en PHP tiene la tendencia a combinar la labor del diseñador y el programador, ya que las instrucciones PHP se encuentran embebidas en el código HTML, lo que trae

---

<sup>7</sup> Una API (del inglés *Application Programming Interface - Interfaz de Programación de Aplicaciones*) es el conjunto de funciones y procedimientos que ofrece cierta librería para ser utilizado por otro software como una capa de abstracción.

consigo grandes dificultades para realizar cambios de diseño o de programación, precisamente Smarty da solución a esta problemática. Actualmente existen más sistemas de plantillas para PHP pero éste es uno de los más avanzados y con más frecuencia de desarrollo.

La aplicación fue concebida y construida siguiendo uno de los patrones arquitectónicos más utilizado para las aplicaciones web, el estilo de tres capas sobre una arquitectura cliente servidor. De acuerdo a las características de este estilo, se establece una división lógica en tres capas.

La primera capa es la Capa de Presentación que es la encargada de interactuar con el usuario. La segunda capa denominada Capa de Aplicación, es donde se localiza la lógica del negocio, esta capa recibe las peticiones del usuario y les da respuesta, accediendo normalmente al repositorio de datos, a través de peticiones a la Capa de Datos, que es la encargada de proporcionar el acceso a los datos.

#### **4.6.2 Algoritmo de Inteligencia Artificial**

El algoritmo para la generación automática de los horarios docentes, fue confeccionado en SWI-Prolog 5.6.31, utilizando el paradigma de Programación Lógica, en combinación con algunos elementos de Programación con Restricciones.

Dentro de la aplicación el algoritmo es concebido como un componente (específicamente un ejecutable). Resulta de vital importancia la utilización del SWI-Prolog ODBC Interface, y el MySQL ODBC Driver, empleando estos dos elementos se establece la conexión entre el algoritmo y la base de datos a través de un DSN<sup>8</sup>.

El SWI-Prolog ODBC Interface es la interfaz del SWI-Prolog para la interacción con ODBC<sup>9</sup>, esta interfaz fue ideada formando dos capas, la primera capa es un núcleo de funcionalidades del ODBC que permite ejecutar sentencias SQL y la segunda capa explota la relación entre los predicados de Prolog y las tablas de la base de datos, proporcionando, con algunas limitaciones, una vista natural de los datos en Prolog.

---

<sup>8</sup> DSN (Data Source Name), significa Nombre Fuente de datos y representa todo lo relativo a una fuente de datos configurada por el usuario para conectarse a una Base de datos.

<sup>9</sup> ODBC son las siglas de Open DataBase Connectivity, que es un estándar de acceso a Bases de Datos desarrollado por Microsoft Corporation, el objetivo de ODBC es hacer posible el acceder a cualquier dato de cualquier aplicación, sin importar qué Sistema Gestor de Bases de Datos (DBMS por sus siglas en inglés) almacene los datos, ODBC logra esto al insertar una capa intermedia llamada manejador de Bases de Datos, entre la aplicación y el DBMS,

El algoritmo se compone de tres módulos:

- **Módulo de Datos:** Módulo encargado de la obtención, manipulación, e inserción de todos los datos provenientes de la base de datos, en la base de conocimientos. La obtención de todos los datos será realizada a través de funciones de la interfaz ODBC, accediendo tanto a tablas como a vistas, estas últimas diseñadas específicamente para facilitar la interacción con el algoritmo.
- **Módulo de Procesamiento:** Módulo central del algoritmo, que funciona sobre todos los hechos dinámicos insertados por el Módulo de Datos. A grandes rasgos el algoritmo, llegará hasta la inserción en la base de conocimientos de un conjunto de estructuras clases que en su conjunto conformarán los horarios. Este módulo utiliza muchísimo la recursividad, el tratamiento de listas sobre estructuras y los predicados dinámicos.
- **Módulo de Validación:** Módulo final del algoritmo, que se encarga de validar si las estructuras clases creadas por el Módulo de Procesamiento son correctas. Si la solución resulta válida, los datos generados serán insertados en la base de datos.

Toda la concepción del algoritmo está basado en una de las técnicas básicas de programación en Prolog, el paradigma Generar/Comprobar. Este paradigma es adecuado para problemas sin solución algorítmica, responde al método de prueba y error y está basado en generadores y retroceso. A continuación se explica brevemente en qué consiste este paradigma:

Al tratar de resolver algunos problemas, no se conoce qué alternativa de entre las posibles es la adecuada; es decir, cuál es el siguiente paso que debe dar el algoritmo para obtener una solución. Para resolver estos problemas se suele emplear una estrategia de búsqueda que proceda, mediante prueba y error, a explorar el espacio de posibilidades. La estrategia más habitual es la vuelta atrás (backtracking), pero existen otras posibilidades como la búsqueda en anchura. Implementar estas estrategias en lenguajes imperativos es engorroso y propenso a errores. Por el contrario, Prolog aporta de forma automática toda la maquinaria necesaria para implementar la búsqueda en profundidad. (25)

Los generadores pueden aplicarse para resolver problemas que deben resolverse mediante búsquedas. La idea se basa en utilizar los generadores y el retroceso de Prolog para implementar el mecanismo de resolución de problemas por prueba y error (vuelta atrás). Se emplea un generador para generar un

candidato a solución al problema y después se comprueba si, efectivamente, el candidato es una solución. Esta estrategia queda recogida en el siguiente patrón de predicado:

**resolver (+Problema, -Solución):- generar (+Problema, -Candidato),  
 comprobar (+Problema, +Candidato),  
 Solución = Candidato.**

La clave del éxito de esta estrategia reside, obviamente, en el generador. Lo ideal es encontrar un generador que obtenga un conjunto de candidatos que incluya todas las soluciones y sea lo más pequeño posible. Es común que un problema requiera emplear varios generadores y una comprobación compuesta de varias condiciones. En tales casos, conviene entrelazar las fases de generación y comprobación, comenzando por los generadores y las condiciones más restrictivas.

## 4.7 Modelo de Despliegue

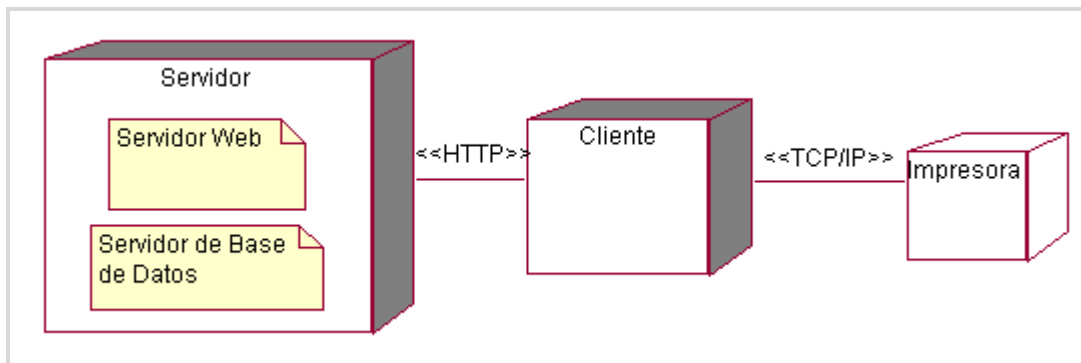


Figura 24. Modelo de Despliegue

## 4.8 Modelo de Implementación

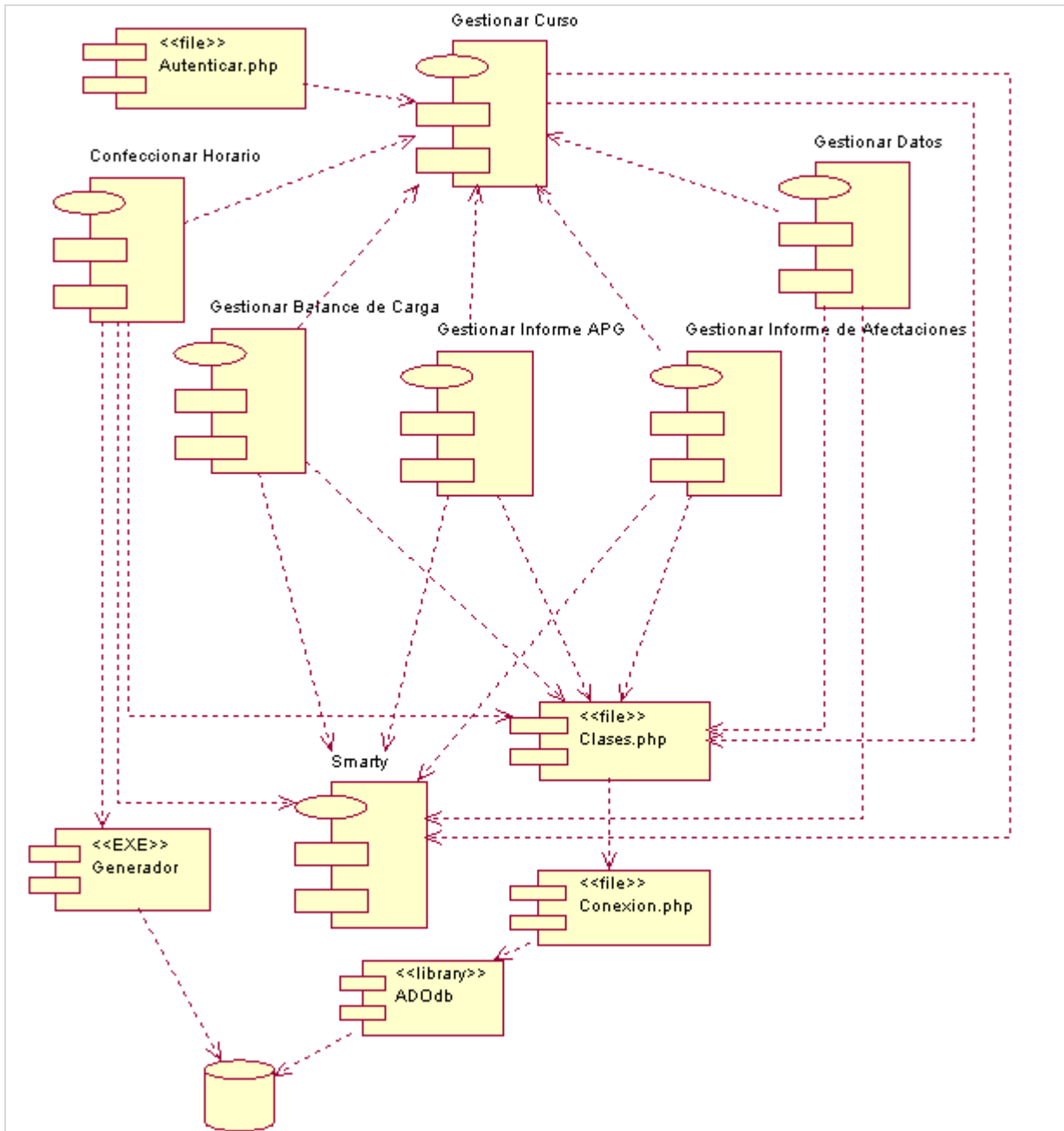


Figura 25. Diagrama de Componentes

## **4.9 Conclusiones**

En este capítulo se han abordado aspectos fundamentales para la construcción del sistema propuesto, como los diagramas de clases web, el diagrama de clases persistentes, el modelo de datos, el modelo de despliegue y el modelo de implementación. Además se describen aspectos del diseño de interfaz, así como algunas generalidades de la implementación de la solución propuesta.

# CAPÍTULO 5

## Estudio de Factibilidad

### 5.1 Introducción

Este capítulo está dedicado al estudio de factibilidad. Se abordan varios aspectos que permiten determinar la viabilidad o factibilidad del desarrollo del sistema propuesto. Se realiza la estimación de costo del proyecto, se plantean los beneficios tangibles e intangibles que reportaría la aplicación y se realiza el análisis de costo y beneficio.

### 5.2 Planificación

Uno de los objetivos de la planificación es lograr estimaciones razonables. Actualmente existen varias técnicas para la estimación de costos, recursos y tiempo relacionados al desarrollo del software. La técnica de estimación utilizada en este caso es la denominada **Análisis de Puntos de Casos de Uso**. Este método se basa en la posibilidad de predecir el tamaño de un sistema a partir de las características de sus requisitos, expresados en los casos de uso. Debe aclararse que esta técnica no concibe los gastos logísticos y por conceptos de compras de licencias, etc., que deben ser sumados para que se logre obtener una estimación más exacta.

#### 5.2.1 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:**

**Tabla 8.** Factor de peso de los actores sin ajustar

Tipo	Descripción	Peso	Cant * peso
Simple	Otro sistema que interactúa con el sistema a desarrollar mediante una interfaz de programación (API, Application Programming Interface)	1	0*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	1*3
<b>Total</b>			<b>3</b>

Para calcular **UUCW**:

**Tabla 9.** Factor de peso de los casos de uso sin ajustar

Tipo	Descripción	Peso	Cant * peso
Simple	El Caso de Uso contiene de 1 a 3 transacciones	5	1*5
Medio	El Caso de Uso contiene de 4 a 7 transacciones	10	8*10
Complejo	El Caso de Uso contiene más de 8 transacciones	15	2*15
<b>Total</b>			<b>115</b>

Luego: **UUCP** = 3 + 115 **UUCP** = 118

### 5.2.2 Ajustar los Puntos de Casos de Uso

$$UCP = UUCP * TCF * EF$$

Donde:

**UCP**: Puntos de Casos de Uso ajustados

**UUCP**: Puntos de Casos de Uso sin ajustar

**TCF**: Factor de complejidad técnica

**EF**: Factor de ambiente

Para Calcular TCF:

$$TCF = 0.6 + 0.01 * \Sigma (\text{Peso}_i * \text{Valor}_i)$$

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

**Tabla 10.** Factor de complejidad técnica

Factor	Descripción	Peso	Valor	$\Sigma(\text{Peso}_i * \text{Valor}_i)$
T1	Sistema distribuido	2	0	0
T2	Objetivos de performance o tiempo de respuesta	1	5	5
T3	Eficiencia del usuario final	1	5	5
T4	Procesamiento interno complejo	1	5	5
T5	El código debe ser reutilizable	1	4	4
T6	Facilidad de instalación	0.5	4	2
T7	Facilidad de uso	0.5	5	2.5
T8	Portabilidad	2	4	8
T9	Facilidad de cambio	1	5	5
T10	Concurrencia	1	1	1
T11	Incluye objetivos especiales de seguridad	1	3	3
T12	Provee acceso directo a terceras partes	1	5	5
T13	Se requieren facilidades especiales de entrenamiento a los usuarios	1	5	5
			Total	50.5

Luego:

$$\text{TCF} = 0.6 + 0.01 * 50.5 \quad \text{TCF} = 1.105$$

Para Calcular EF:

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

El Factor de Ambiente EF (Environmental Factor) trata de medir cuán familiarizado está el equipo de desarrollo con el tipo de problema del proyecto a realizar.

**Tabla 11.** Factor de ambiente

Factor	Descripción	Peso	Valor	$\Sigma (\text{Peso}_i * \text{Valor}_i)$
E1	Familiaridad con el modelo de proyecto utilizado	1.5	3	4.5
E2	Experiencia en la aplicación	0.5	3	1.5
E3	Experiencia en orientación a objetos	1	4	4
E4	Capacidad del analista líder	0.5	5	2.5
E5	Motivación	1	5	5
E6	Estabilidad de los requerimientos	2	3	6
E7	Personal part-time	-1	3	-3
E8	Dificultad del lenguaje de programación	-1	3	-3
Total				17.5

Luego:

$$\text{EF} = 1.4 - 0.03 * 17.5$$

$$\text{EF} = 0.875$$

Luego:

$$\text{UCP} = 118 * 1.105 * 0.875$$

$$\text{UCP} = 114.09125$$

### 5.2.3 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 calcular **CF**:

**CF** = 20 horas-hombre (si Total EF  $\leq$  2)

**CF** = 28 horas-hombre (si Total EF = 3 ó Total EF = 4)

**CF** = abandonar o cambiar proyecto (si Total EF  $\geq$  5)

**Total EF** = **Cant EF** < 3 (entre E1 –E6) + **Cant EF** > 3 (entre E7, E8)

**Total EF** = 0 + 0

Total EF = 0

**CF** = 20 horas-hombre (porque Total EF  $\leq$  2)

Luego:

**E**= 114.09125 \* 20

**E**= 2281.825 horas-hombre

### 5.2.4 Calcular esfuerzo de todo el proyecto

Tabla 12. Esfuerzo del Proyecto

Actividad	% esfuerzo	Valor esfuerzo
Análisis	10%	570 horas-hombre
Diseño	20%	1141 horas-hombre
Implementación	40%	2282 horas-hombre
Prueba	15%	856 horas-hombre
Sobrecarga	15%	856 horas-hombre
Total	100%	5705 horas-hombre

Si  $E_T = 5705$  Horas/ hombre y se estima que cada mes tiene 192 horas laborables, quedaría:

$$E_T = 29.71 \text{ mes/hombre.}$$

## 5.3 Costos

$$CHM = CH * SxH$$

Donde:

Salario mensual por Hombre (**SxH**) = \$289.00

Cantidad de hombres (**CH**) = 2

CHM: Costo por hombres al mes.

Luego:

$$CHM = 578 \text{ \$/mes}$$

$$\text{Costo} = CHM * E_T / CH$$

$$\text{Costo} = 578 * 29.71 / 2$$

$$\text{Costo} = \$ 8586.19$$

Tiempo total del Proyecto:

$$\text{Tiempo} = E_T / CH$$

$$\text{Tiempo} = 29.71 \text{ meses /2 hombres}$$

$$\text{Tiempo} = 14.855 \text{ meses}$$

De lo obtenido se interpreta que con 2 trabajadores la propuesta tiene un tiempo de duración de **14.855** meses y su costo total se estima en **\$ 8586.19**.

## **5.4 Beneficios tangibles e intangibles**

El principal beneficio obtenido de la aplicación del sistema propuesto es que mejora la forma en que actualmente se lleva a cabo el proceso de gestión de los horarios docentes y por ende favorece la labor docente y educativa de la Facultad 9. En este sentido perfecciona el proceso en varios aspectos, reportando beneficios tanto tangibles como intangibles.

### **5.4.1 Beneficios tangibles**

De manera general puede definirse a los beneficios tangibles como aquellos que reportan ventajas económicas cuantificables. En este caso el sistema propuesto brinda los siguientes beneficios tangibles:

- Incremento en la velocidad del proceso.
- Permite la gestión centralizada y sencilla de una gran cantidad de información.
- Aprovechamiento del poder de cálculo y almacenamiento de los ordenadores.
- Reducción considerable del tiempo requerido para confeccionar los horarios.
- Disminución del material de oficina empleado en archivar y procesar una gran cantidad de información.

### **5.4.2 Beneficios Intangibles**

De manera general puede definirse a los beneficios intangibles como aquellos que reportan beneficios organizativos, de funcionamiento o eficiencia. En este caso el sistema propuesto brinda los siguientes beneficios intangibles:

- Incremento de la eficiencia y precisión en el proceso de confección de los horarios.
- Mejora el funcionamiento y la organización de los procesos del Vicedecanato de Formación y de la Facultad.
- Incremento en la satisfacción de los empleados al facilitar tareas tediosas y complejas.

- Contribuye a la mejora y perfeccionamiento de la labor docente y educativa de la Facultad.

## **5.5 Análisis de costos y beneficios**

Un análisis de costos y beneficios, significa una valoración de la inversión económica comparado con los beneficios que se obtendrán en la comercialización y utilidad de cualquier producto o sistema.

El sistema propuesto será desarrollado utilizando tecnologías y herramientas Open Source, lo que disminuye notablemente los gastos de licencias, haciendo menos costosa la realización del sistema.

Además teniendo en cuenta todos los beneficios tanto tangibles como intangibles, mencionados anteriormente y partiendo de la estimación de costo realizada en epígrafes anteriores, se considera que el sistema propuesto es factible. Su implementación y puesta en práctica será de gran utilidad para la Facultad 9 y también para la Universidad en su conjunto, pues la aplicación es apropiada para cualquier Facultad.

## **5.6 Conclusiones**

En este capítulo se realizó el estudio de factibilidad del sistema propuesto, reflejándose aspectos como la estimación de costos, los beneficios tangibles e intangibles ofrecidos por el sistema propuesto así como el análisis de costos y beneficios. Finalmente se concluye que la realización del sistema es factible desde el punto de vista económico y será de gran utilidad para Facultad 9 y también para la Universidad.

## CONCLUSIONES

Con la realización de este trabajo se obtuvieron los siguientes resultados que dan cumplimiento al objetivo propuesto en esta investigación:

- Se modelaron los procesos que tienen lugar durante la confección de los horarios docentes.
- Se desarrolló un prototipo funcional del Sistema Automatizado para la Gestión de Horarios Docentes.
- Se desarrolló un algoritmo basado en Inteligencia Artificial capaz de generar automáticamente los horarios docentes.

Los resultados de este trabajo y la utilización del sistema desarrollado serán de gran utilidad para la Facultad 9, teniendo en cuenta que contribuye a mejorar el funcionamiento y la organización del proceso docente y educativo. Se considera además que su uso puede ser extendido a todas las facultades de la Universidad de las Ciencias Informáticas.

## RECOMENDACIONES

Al concluir el desarrollo de este documento se recomienda:

- Continuar el desarrollo de esta investigación con el objetivo de perfeccionar y aumentar las funcionalidades de la aplicación web.
- Añadir al sistema un Área de Reportes, que permita la generación de reportes y la consulta de datos de forma personalizada y rápida.
- Continuar el desarrollo del algoritmo con vistas a aumentar el número de variantes de generación, y aumentar las vías de solución ante las distintas situaciones.
- Se recomienda el uso del sistema en todas las facultades de la UCI.

## BIBLIOGRAFÍA

1. CORRALES, D., PÉREZ, C. *La Planificación en el trabajo de la escuela*. La Habana: Editorial Pueblo y Educación, 1976.
2. BATISTA HERNÁNDEZ, Marianny. *Sistema para la gestión de horarios docentes*. Tesis de diploma inédita, Instituto Superior Politécnico "José Antonio Echeverría", Universidad de las Ciencias Informáticas. La Habana, 2006.
3. BELLO PÉREZ, Rafael. *Curso de Métodos de Solución de Problemas para la Inteligencia Artificial*. Las Villas: Grupo de Investigaciones en Inteligencia Artificial, Departamento de Ciencias de la Computación, Universidad Central de Las Villas. 1998.
4. GARCÍA, Zoila Z., et al. *Aplicaciones de la Inteligencia Artificial*. México: Centro Universidad de la Ciénaga, Universidad de Guadalajara, 2000.
5. BELLO PÉREZ, Rafael. *Curso Introductorio a las Redes Neuronales Artificiales*. Las Villas: Grupo de Investigaciones en Inteligencia Artificial, Departamento de Ciencias de la Computación, Universidad Central de Las Villas. 1993.
6. BELLO PÉREZ, Rafael., et al. *Modelos Computacionales Avanzados*. Las Villas: Departamento de Ciencias de la Computación, Universidad Central de Las Villas. 1993.
7. ACHOUR, Mehdi. *Manual de PHP* [en línea]. Gabor Hojtsy. <http://www.php.net>: Grupo de documentación de PHP, 2003, 26 de febrero de 2006 [citado 10 de abril de 2007]. Disponible en World Wide Web: <http://www.php.net/manual/es/>
8. WELLING, L., THOMPSON, L., *Desarrollo Web con PHP y MySQL*. Anaya Multimedia.
9. MySQL AB, *MySQL 5.0 Reference Manual* [en línea]. <http://dev.mysql.com>: MySQL AB, 1997, 15 de mayo de 2007 [citado 17 de mayo de 2007]. Disponible en World Wide Web: <http://dev.mysql.com/doc/refman/5.0/es/what-is.html>
10. JACOBSON, I., BOOCH, G., RUMBAUGH, J., *El Proceso Unificado de Desarrollo de Software*. La Habana: Editorial Félix Varela, 2004.

11. LARMAN, C., *UML y Patrones Introducción al análisis y diseño orientado a objetos*. La Habana: Editorial Félix Varela, 2004.
12. CANÓS, José H., LETELIER, P., PENÁDES, M.C. *Metodologías Ágiles en el Desarrollo de Software* [en línea]. <http://www.willydev.net>: DSIC -Universidad Politécnica de Valencia, 2003 [citado 24 de febrero de 2007]. Disponible en World Wide Web: <http://www.willydev.net/descargas/prev/TodoAgil.Pdf>.
13. WIELMAKER, Jan. *SWI-Prolog 5.6.32 Reference Manual*. [en línea]. <http://www.swi-prolog.org/>: Universidad de Ámsterdam, 1990, marzo 2007 [citado 7 de marzo de 2007]. Disponible en World Wide Web: <http://www.swi-prolog.org/>.
14. LE HÉGARET, P., WHITMER, R., WOOD, L., *Document Object Model (DOM)*. [en línea]. <http://www.w3.org>: W3C DOM IG, 19 de enero 2005, 12 de julio de 2006 [citado 4 de febrero de 2007]. Disponible en World Wide Web: <http://www.w3.org/DOM/#what>
15. LABRA GAYO, José E. *Diseño de sitios Web mediante estándares*. [en línea]. <http://www.di.uniovi.es>: Departamento de Informática, Universidad de Oviedo, julio 2004 [citado 4 de noviembre 2006]. Disponible en World Wide Web: <http://www.di.uniovi.es/~labra/cursos/DOM/DOM.pdf>
16. REYNOSO B, Carlos. *Introducción a la Arquitectura de Software*. [en línea]. <http://www.microsoft.com/spanish/msdn/>: Universidad de Buenos Aires, 26 de junio de 2006 [citado 8 de enero de 2007]. Disponible en: [http://www.microsoft.com/spanish/msdn/arquitectura/roadmap\\_arq/intro.mspx#EAG](http://www.microsoft.com/spanish/msdn/arquitectura/roadmap_arq/intro.mspx#EAG)
17. RÍOS GIL, J.J. *Sistemas de Información*. [en línea]. <http://www.it.uc3m.es/>: Departamento de Ingeniería Telemática Escuela Politécnica Superior Universidad Carlos III de Madrid, 13 de octubre de 2005 [citado 23 de febrero de 2007]. Disponible en World Wide Web: [http://www.it.uc3m.es/mcftp/docencia/si/material/1\\_cli-ser\\_mcfp.pdf](http://www.it.uc3m.es/mcftp/docencia/si/material/1_cli-ser_mcfp.pdf)
18. DUQUE MÉNDEZ, Néstor D. *Conceptos de arquitectura Cliente/Servidor*. [en línea]. <http://www.virtual.unal.edu.co>: Universidad Nacional de Colombia, 27 de noviembre de 2006 [citado 25 de abril de 2007]. Disponible en World Wide Web: [http://www.it.uc3m.es/mcftp/docencia/si/material/1\\_cli-ser\\_mcfp.pdf](http://www.it.uc3m.es/mcftp/docencia/si/material/1_cli-ser_mcfp.pdf)

- 19.** STALLMAN, Richard M. *Software libre para una sociedad libre*. [en línea].  
<http://biblioweb.sindominio.net>: Traficantes de Sueños, diciembre 2004 [citado 15 de febrero de 2007].  
Disponible en World Wide Web:  
<http://biblioweb.sindominio.net/pensamiento/softlibre/softlibre005.html#htoc6>
- 20.** BARAHONA G, J., SEONE P. J., ROBLES, G., *Introducción al Software Libre*. [en línea]  
<http://www.uoc.es>: Universidad Politécnica de Madrid, 21 de septiembre de 2003 [citado 23 de marzo de 2007]. Disponible en World Wide Web: [http://cv.uoc.edu/~fcaulas/20041/90.783/portada\\_Into.pdf](http://cv.uoc.edu/~fcaulas/20041/90.783/portada_Into.pdf)
- 21.** CASTILLO VIERA, Yoselyn, DE LA ICERES T., Gema. *Centralización de la Información del Registro del Estado Civil de Cuba*. Tesis de diploma inédita, Instituto Superior Politécnico "José Antonio Echeverría", La Habana, junio de 2005.
- 22.** JUÁREZ, C. *et al. Software libre vs software propietario: Ventajas y desventajas*.  
<http://www.softwarelibre.cl>: México, Mayo de 2006 [citado 12 de mayo de 2007]. Disponible en World Wide Web: <http://www.softwarelibre.cl/drupal/files/32693.pdf>
- 23.** LIM, J. *ADODB Database Abstraction Library for PHP (and Python)*. [en línea].2004 [citado 15 de febrero de 2007]. Disponible en World Wide Web: <http://adodb.sourceforge.net>
- 24.** OHRT, A., ZMIEVSKI, A. *Smarty Manual*. [en línea] New Digital Group, Inc.11 de febrero de 2005 [citado 12 de marzo de 2007] Disponible en World Wide Web: <http://smarty.php.net/download-docs.php>
- 25.** LÓPEZ OLIVAS, Pablo. *Técnicas Básicas de Programación Prolog* [en línea]  
<http://www.lcc.uma.es>: Departamento de Lenguajes y Ciencias de la Computación, Universidad de Málaga, España, 5 de diciembre de 2006[citado 27 de abril de 2007]. Disponible en World Wide Web: [http://www.lcc.uma.es/~lopez/progdec/apuntes/02-basicas/tecnicas\\_basicas.pdf](http://www.lcc.uma.es/~lopez/progdec/apuntes/02-basicas/tecnicas_basicas.pdf)
- 26.** PRESSMAN, Roger S. *Ingeniería de Software, Un enfoque Práctico*. Quinta Edición. La Habana: Editorial Félix Varela, 2005.
- 27.** GILFILLAN, I. *La Biblia de MySQL*. Anaya Multimedia.

## GLOSARIO DE TÉRMINOS Y SIGLAS

**Informe APG:** Documento que recoge la relación entre las asignaturas, los profesores que las imparten y los grupos que cada profesor atiende en un período de un curso.

**Informe Afect - L:** Documento que recoge las afectaciones de los locales en una semana, período y curso determinado. Aparecen reflejadas según los días de la semana y las sesiones (Mañana y Tarde) y muestran el estado del local, que puede ser Disponible (D) o Afectado (A).

**Informe Afect - P:** Documento que recoge las afectaciones de los profesores en una semana, período y curso determinado. Aparecen reflejadas según los días de la semana y las sesiones (Mañana y Tarde) y muestran el estado del local, que puede ser Disponible (D) o Afectado (A).

**Balance de Carga:** Documento base de toda la planificación docente, en el se recoge para cada año docente la dosificación de clases de cada asignatura en cada semana del semestre, especificándose además el tipo de clase.

**Algoritmo:** Un algoritmo es un conjunto ordenado y finito de operaciones que permite hallar la solución de un problema.

**Complejidad Computacional:** Es la parte de la teoría de la computación que estudia, de manera teórica, los recursos requeridos durante el cálculo para resolver un problema.

**Problemas NP Completos:** 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.

**Problemas NP:** Es el conjunto de problemas que pueden ser resueltos en tiempo polinómico por una máquina de Turing no determinista.

**Máquina de Turing:** Es un modelo matemático abstracto que formaliza el concepto de algoritmo.

**Backtracking:** Es una técnica de resolución general de problemas mediante una búsqueda sistemática de soluciones. El procedimiento general se basa en la descomposición del proceso de búsqueda en tareas parciales de tanteo de soluciones (*trial and error*).

**Licencia LGPL:** Es un editor WYSIWYG para HTML de código abierto que funciona completamente en JavaScript y se distribuye gratuitamente.

**SOAP (*Simple Object Access Protocol*):** Protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML, es uno de los protocolos utilizados en los servicios Web.

**ADODB:** Es un conjunto de librerías de bases de datos para PHP y Python. Esta permite a los programadores desarrollar aplicaciones web de una manera portable, rápida y fácil. La ventaja reside en que la base de datos puede cambiar sin necesidad de reescribir cada llamada a la base de datos realizada por la aplicación.

**Smarty:** Es un motor de plantillas para PHP, cuyo objetivo es separar el contenido de la presentación en una página web.

## ANEXOS

Anexo 1. Análisis de Tesis Precedente.....	106
Anexo 2. Plantilla del Informe de Asignaturas-Profesor-Grupo.....	109
Anexo 3. Plantilla del Informe de Afectaciones de Profesores.....	110
Anexo 4. Descripción Textual del CUN Entrega del Balance de Carga.....	111
Anexo 5. Descripción Textual del CUN Entrega del Informe de Afectaciones de Locales.....	112
Anexo 6. Descripción Textual del CUN Entrega de Informes .....	113
Anexo 7. Descripción Textual del CUN Confección de Horario .....	114

### **Anexo 1. Análisis de Tesis Precedente**

En la Universidad de las Ciencias Informáticas conjuntamente con la Facultad de Ingeniería Informática del Instituto Superior Politécnico José Antonio Echeverría, en junio del 2006 fue presentado un trabajo de

diploma con el Título “Sistema de Gestión de Horarios Docentes”, su autor es Marianny Hernández Batista y los tutores Lic. Ansel Y. Rodríguez González y la Ing. Keydi García Lira.

A pesar de que este trabajo aborda de manera general la misma problemática que la actual investigación, es decir la generación automática de horarios docentes, no fue considerada una solución válida, para el problema planteado en el actual trabajo: *la inexistencia de un sistema automatizado para generar los horarios docentes de la Facultad 9 de la Universidad de las Ciencias Informáticas* por varias razones. A continuación se explican algunas de estas razones, estableciendo una comparación entre los que plantea la tesis precedente y lo que sucede actualmente en la Facultad 9 como centro del campo de acción del actual trabajo.

Razones:

El flujo de procesos planteado difiere del modo en que se lleva a cabo la gestión de los horarios docentes en la Facultad 9. Algunas de las diferencias más notables son:

- La Tesis precedente plantea: “Los jefes de departamento a nivel central son los encargados de entregar los P1 de su correspondiente asignatura al planificador.”(10)
- En Facultad 9: Los P1 de cada asignatura no son el documento que recoge la planificación docente, este documento es el Balance de Carga, el cual es confeccionado a nivel Central por la Dirección de Planificación y Control del Proceso de Formación y es entregado al Vicedecano de Formación.
- La Tesis precedente plantea: “Los Jefes de Departamento entregan al planificador los listados de sus profesores con las asignaturas que impartirán. Los Jefes de Departamento entregan al planificador los listados de sus profesores con sus correspondientes grupos.”
- En la Facultad 9: El documento que recoge la relación asignaturas, profesores de cada asignatura, y grupos a los que imparte clases cada profesor es entregado por cada Jefe de Departamento al Vicedecano de Formación no al planificador.
- La Tesis precedente plantea: Los profesores entregan un informe con sus afectaciones.

- En la Facultad 9: El Informe de Afectaciones de Profesores es entregado por cada Jefe de Departamento al Vicedecano de Formación, no son los profesores quienes entregan sus afectaciones. Estas son controladas, autorizadas e informadas de manera central por los Jefes de Departamentos.

Partiendo de la diferencia en los procesos, naturalmente la modelación del negocio resulta distinta y por lo tanto el sistema propuesto también difiere notablemente.

La mayoría de los procesos que serán objeto de automatización en la tesis precedente no lo son para el trabajo actual, pues en la concepción del sistema difieren en cuanto alcance y objetivos.

- La Tesis precedente plantea: El sitio tiene como propósito automatizar cada uno de los procesos que se llevan a cabo hasta nuestros días. Uno de ellos es el hecho mismo de consultar el horario, una vez listo, cada persona podrá revisar el sitio desde su puesto de trabajo sin necesidad de trasladarse. Cada persona que sea autenticada podrá disfrutar de las ventajas que su rol tenga asignadas. Cualquier persona puede tener acceso a un listado de profesores por grupo, profesor por asignatura o profesor por departamento según haya sido la solicitud. El profesor podrá, mediante el sitio, pedir que se tengan en cuenta sus afectaciones enviando una propuesta de horario que para él sea la más cómoda, además de consultar los espacios en blanco y reservar sus consultas de manera que inmediatamente todos sepan de su decisión y que nadie más pueda tomar ese lugar. Se podrán gestionar las asignaturas, los P1, los grupos, los locales, los departamentos y los profesores. El sitio resolverá el gran problema de repartir los turnos en el horario docente, así como modificarlo en caso de cambios donde se volverían a repartir los turnos, mostrando un grupo de propuestas para que el planificador tenga la posibilidad de decidir cuál es la más óptima. Todo esto se hará respetando un buen número de criterios pedagógicos que se deben de tener en cuenta a la hora de confeccionar un horario docente, para garantizar así el máximo rendimiento de los estudiantes.
- El actual trabajo plantea otro enfoque de los procesos a automatizar y de los usuarios y funcionalidades del sistema, acorde a su modelación del negocio.

Además de los aspectos mencionados anteriormente, se tuvo en cuenta que la tesis precedente tenía varios planteamientos que reflejaban contradicciones e imprecisiones en el flujo de proceso y en la concepción del sistema.

Se tuvo en cuenta además la casi total ausencia de información referida a la aplicación de técnicas de Inteligencia Artificial como parte básica y fundamental del sistema que se propone.

Teniendo en cuenta todo lo anteriormente mencionado, la tesis precedente no se considera una solución válida para el problema de la actual investigación.

La tesis precedente se encuentra disponible en: [http://bibliodoc.uci.cu/TD/TD\\_0147\\_06.pdf](http://bibliodoc.uci.cu/TD/TD_0147_06.pdf)

## Anexo 2. Plantilla del Informe de Asignaturas-Profesor-Grupo

Informe de Asignaturas – Profesor –Grupo		
Curso	Período	Departamento



Nombre del Profesor	lunes		martes		miérc.		jueves		viernes		sábado	
	M	T	M	T	M	T	M	T	M	T	M	T

**Leyenda:** A: Afectado D: Disponible

**Fecha de Entrega:** \_\_\_\_\_

\_\_\_\_\_  
**Firma del Jefe de Departamento**

\_\_\_\_\_  
**Firma del Vicedecano de Formación**

**Anexo 4. Descripción Textual del CUN Entrega del Balance de Carga**

**Anexo 5. Descripción Textual del CUN Entrega del Informe de Afectaciones de Locales**

<b>Caso de Uso del Negocio: Entrega del Balance de Carga</b>	
<b>Actores del negocio</b>	La Dirección de Planificación y Control, Vicedecano de Formación.
<b>Propósito</b>	Entregar al Vicedecano de Formación el Balance de Carga, para que sea utilizado para confeccionar el Horario.
<b>Resumen</b>	El Caso de Uso se inicia cuando La Dirección de Planificación y Control comunica al Vicedecano de Formación cual es el Balance de Carga establecido para el semestre, este lo recibe, revisa y posteriormente entrega al planificador.
<b>Casos de uso asociados</b>	-
<b>Curso Normal de eventos</b>	
<b>Acción del actor</b>	<b>Respuesta del Proceso del Negocio</b>
<b>1</b> La Dirección de Planificación y Control, entrega al Vicedecano el Balance de Carga.	<b>2</b> El Vicedecano recibe el Balance de Carga.
	<b>3</b> El Vicedecano revisa la completitud del documento: a-) En caso de que este completo se pasa al paso 5. b-) Si no está completo se informa a la Dirección de Planificación y Control.
<b>4</b> La Dirección de Planificación y Control recibe el rechazo del documento y finaliza el proceso.	<b>5</b> El Vicedecano entrega el documento recibido al Planificador.
	<b>6</b> El Planificador recibe el Balance de Carga.
<b>Prioridad</b>	Crítico
<b>Mejoras</b>	Este CUN no será automatizado en su totalidad, sólo se automatizará el proceso de registro del Balance de Carga una vez que el Planificador lo recibe. Esto mejorará el proceso, permitiendo el almacenamiento permanente, la gestión, consulta y modificación automatizada de este documento.
<b>Cursos alternos</b>	
-	

<b>Caso de Uso del Negocio: Entrega del Informe de Afectaciones de Locales</b>	
<b>Actores del negocio</b>	Dirección de la Facultad, Vicedecano de Formación.
<b>Propósito</b>	Entregar al Vicedecano de Formación el Listado de los Locales, para que sea utilizado para confeccionar el Horario.
<b>Resumen</b>	El caso de uso se inicia cuando la Dirección de la Facultad comunica al Vicedecano de Formación cual es el Informe de Afectaciones de Locales, este lo recibe, revisa y posteriormente entrega al planificador.
<b>Casos de uso asociados</b>	-
<b>Curso Normal de eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Proceso del Negocio</b>
<b>1</b> La Dirección de la Facultad, entrega al Vicedecano el Informe de Afectaciones de Locales.	<b>2</b> El Vicedecano recibe el Informe de Afectaciones de Locales.
	<b>3</b> El Vicedecano revisa la completitud del documento: a-) En caso de que este completo se pasa al paso 5. b-) Si no está completo se informa a la Dirección de la Facultad.
<b>4</b> La Dirección de la Facultad recibe el rechazo del documento y finaliza el proceso.	<b>5</b> El Vicedecano entrega el documento recibido al Planificador.
	<b>6</b> El Planificador recibe el Informe de Afectaciones de Locales.
<b>Prioridad</b>	Crítico
<b>Mejoras</b>	Este CUN no será automatizado en su totalidad, sólo se automatizará el proceso de registro del Informe de Afectaciones de Locales una vez que el Planificador lo recibe. Esto mejorará el proceso, permitiendo el almacenamiento permanente, la gestión, consulta y modificación automatizada de este documento.
<b>Cursos Alternos</b>	
-	

## Anexo 6. Descripción Textual del CUN Entrega de Informes

<b>Caso de Uso del Negocio: Entrega de Informes</b>	
<b>Actores del negocio</b>	Vicedecano de Formación, Jefe de Departamento.
<b>Propósito</b>	Entregar al Vicedecano de Formación el Informe de Asignatura - Profesor - Grupo y el Informe de Afectaciones de Profesores para elaborar el horario docente.
<b>Resumen</b>	El caso de uso se inicia cuando Jefe de Departamento entrega al Vicedecano de Formación el Informe de Asignatura - Profesor - Grupo y el Informe de Afectaciones de Profesores, este los recibe, revisa y posteriormente entrega al Planificador.
<b>Casos de uso asociados</b>	-
<b>Curso Normal de eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Proceso del Negocio</b>
<b>1</b> El Jefe de Departamento, entrega al Vicedecano el Informe de Asignatura - Profesor - Grupo y el Informe de Afectaciones de Profesores.	<b>2</b> El Vicedecano recibe el Informe de Asignatura - Profesor - Grupo y el Informe de Afectaciones de Profesores.
	<b>3</b> El Vicedecano revisa la completitud de los documentos: a-) En caso de que estén completos se pasa al paso 5. b-) Si no están completos se informa al Jefe de Departamento.
<b>4</b> El Jefe de Departamento recibe el rechazo de los documentos y finaliza el proceso.	<b>5</b> El Vicedecano entrega el Informe de Asignatura - Profesor - Grupo y el Informe de Afectaciones de Profesores al Planificador.
	<b>6</b> El Planificador recibe el Informe de Asignatura - Profesor - Grupo y el Informe de Afectaciones de Profesores al Planificador.
<b>Prioridad</b>	<b>Crítico</b>
<b>Mejoras</b>	Este CUN no será automatizado en su totalidad, sólo se automatizará el proceso de registro del Informe de Asignatura - Profesor - Grupo y del Informe de Afectaciones de Profesores una vez que el Planificador los recibe. Esto mejorará el proceso, permitiendo el almacenamiento permanente, la gestión, consulta y modificación automatizada de estos documentos.
<b>Cursos Alternos</b>	

## Anexo 7. Descripción Textual del CUN Confección de Horario

<b>Caso de Uso del Negocio: Confección de Horarios</b>	
<b>Actores del negocio</b>	Vicedecano de Formación.
<b>Propósito</b>	Confeccionar el Horario Docente de la Facultad.
<b>Resumen</b>	El caso de uso se inicia cuando el Vicedecano de Formación le pide al planificar confeccionar el Horario Docente de la Facultad, este procede a su elaboración y posteriormente solicita la aprobación del mismo.
<b>Casos de uso asociados</b>	-
<b>Curso Normal de eventos</b>	
<b>Acción del Actor</b>	
<b>Respuesta del Proceso del Negocio</b>	
<b>1</b>	El Vicedecano solicita la confección del horario docente de la Facultad.
<b>2</b>	El Planificador recibe la solicitud de confeccionar el horario docente de la Facultad.
<b>3</b>	El Planificador procede a la elaboración del horario docente de la Facultad.
<b>4</b>	EL Planificador solicita la revisión del horario docente de la Facultad.
<b>5</b>	El Vicedecano revisa el documento: a-) En caso de que lo apruebe se continúa al paso 7. b-) Si no lo aprueba el Vicedecano solicita al Planificador una nueva confección del horario docente de la Facultad.
<b>6</b>	El Planificador recibe la solicitud de confeccionar nuevamente el horario docente de la Facultad y se vuelve a comenzar todo el proceso.
<b>7</b>	El Vicedecano publica el horario docente.
<b>Prioridad</b>	Crítico
<b>Mejoras</b>	Este proceso será automatizado, disminuyendo considerablemente el

	<p>tiempo empleado por el planificador en la confección del horario.</p> <p>El horario será generado automáticamente, el planificador sólo deberá revisarlo y si lo desea puede hacerle modificaciones y arreglos específicos, facilitando y simplificando su trabajo.</p>
<b>Cursos Alternos</b>	
-	