

Universidad de las Ciencias Informáticas
Facultad 7



Título: Aplicación web para la confección del horario docente en la Facultad 7

Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas

Autores: Yaneisy López Marrero

Iliana Almira Morffe

Tutores: Ing. Yaciel Edelio Téllez Toledo

Ing. Ariel Pons Alonso

Ciudad de La Habana, Junio del 2009

“Año del 50 aniversario del triunfo de la Revolución”



*"Lo que hace crecer el mundo no es el descubrir cómo está
hecho, sino el esfuerzo de cada uno para descubrirlo".*

José Martí

DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los 26 días del mes de junio del año 2009.

Yaneisy López Marrero

Ilinana Almira Morffe

Ing. Ariel Pons Alonso.

Ing. Yaciel Edelio Téllez Toledo

DATOS DE CONTACTO

Ing.Yaciel Edelio Téllez Toledo (yetellez@uci.cu): es recién graduado de Ingeniero en Ciencias Informáticas en la Universidad de las Ciencias Informáticas (UCI). Actualmente es profesor de la asignatura de Idioma Extranjero (Inglés) perteneciente al departamento de Ciencias Sociales de la Facultad 7, es jefe del colectivo pedagógico de primer año (coordinador) y se desempeña como responsable del grupo de documentación del Área Temática Sistemas de Apoyo a la Salud (SAS) con el rol de revisor técnico.

Ing.Ariel Pons Alonso (apons@uci.cu): es recién graduado de Ingeniero en Ciencias Informáticas en la Universidad de las Ciencias Informáticas. Imparte las asignaturas Máquinas Computadoras I y II en la Facultad 7. Fue líder de desarrollo del proyecto Registro de Actividades Diarias. Presentó trabajo en el Evento Uciencia 2008 y en la XIII Convención y Feria Internacional Informática 2009, posee publicaciones en las memorias de dichos eventos. Actualmente es líder del grupo de arquitectura de información y diseño del proyecto Atención Primaria para la Salud.

AGRADECIMIENTOS

Agradecemos a todos aquellos que estuvieron junto a nosotros brindando apoyo y seguridad en nuestros pasos, aquellos que ofrecieron su aporte y preocupación.

A quienes nos acompañaron en el trayecto y siempre estuvieron física y espiritualmente a nuestro lado.

A nuestros tutores, nuestros más sagrados agradecimientos por su calidez, sugerencias y confianza.

A la Revolución por la oportunidad que nos ha dado de formarnos en su seno como completos profesionales y por la hermosa obra que realiza.

A todos, nuestro mayor reconocimiento y gratitud.....

DEDICATORIA

Yaneisy

Especialmente a quienes me enseñaron a vivir, soñar y caminar por el camino correcto, mis amados padres, Juan Carlos López Aliaga y Odalis Marrero Puebla, a ellos les regalo mi esfuerzo, mis alegrías y penas, a ellos porque con sólo su presencia me llenan de vida, alegrías y ganas de luchar cada día más, por ellos, por su bienestar y por mí.

A mi hermanito Yurizán por su amor incondicional.

A mi novio Yamil por su comprensión, apoyarme en todo y estar siempre a mi lado.

A mis tutores Yaciel y Ariel, por su ayuda y dedicación.

A mis amigos que han sido mi sostén durante tanto tiempo aquí donde todos estamos separados de la familia dándome apoyo y fuerzas para seguir adelante. Gracias por permitirme ser parte de ustedes.

A mis colegas de estudio por brindarme su ayuda desinteresada, compartir los buenos y malos momentos, en fin, hacer estos años inolvidables Zaira, Roberto, Yoelkis, Ketty, Yamisleidys, Oscar, Rubén, Alberto, Bolmey, Idalmis, en fin a todos muchas gracias.

A mis más allegados Leonor, Emir, el Bello, mi cuñado y mi cuñada, a mi suegra, Elvita, por sus palabras de aliento y por su apoyo en cada momento.

A mi compañera de tesis Iliana, por aguantarme las peleaderas, por su amistad, comprensión y compañía.

Gracias a todos!!

Gracias por ayudarme a lograrlo.....

DEDICATORIA

Iliana

Dedico mi tesis principalmente a lo más grande que tengo en este mundo, mis queridos padres, Maribel Morffe Calzadilla y Emilio Almira Betancourth a ellos le debo todo lo que soy hoy en día, a ellos porque con sólo su presencia me llenan de vida y de ganas de luchar a pesar de las dificultades. Sin su ayuda, comprensión, y la fuerza que siempre me han transmitido de luchar por lo que queremos sin dar marcha atrás.

A toda mi familia, a mis tías, a Carli mi lindo hermanito, a mis abuelitos y a mis tíos que siempre me han dado toda la fuerza para seguir.

A mis amigas Diana, Elizabeth, Yimnay y Dalia que siempre me brindaron su apoyo incondicional y manifestaron siempre su preocupación a pesar de estar lejos.

A mi compañera de tesis, que siempre supimos entendernos a pesar de las dificultades, a mi amigo Bolmey, a Dayami, Yailín, Yunaisy a mis compañeras de cuarto, a mis tutores Yaciel y Ariel, por su ayuda y dedicación.

Muchas Gracias a todos!!

RESUMEN

El presente trabajo tiene como objetivo implementar una aplicación web para el proceso de confección del horario docente de la Facultad 7 de la Universidad de las Ciencias Informáticas.

El sistema se centra en ayudar a la planificadora a confeccionar un horario eficiente y de forma más cómoda disminuyendo o evitando los errores, con el fin de que todos los usuarios puedan visualizar una mejor planificación del Horario Docente.

Para ello se ha realizado una investigación de todos aquellos requisitos que deben tenerse en cuenta para la confección del Horario Docente, en vista a ofrecer una mejor organización de las actividades de la Facultad 7.

Para llevar a cabo este sistema se siguieron los pasos que propone el Proceso Unificado de Desarrollo de Software RUP, además de la utilización de las tecnologías definidas en el documento de arquitectura de la Facultad 7, mencionando, del lado del servidor PHP 5.2.5; para el almacenamiento de datos se puso en marcha un servidor de PostgreSQL 8.2, conjuntamente con la arquitectura de tres capas para el uso de la modelación del tipo Modelo-Vista Controlador, entre otras.

Este Software tiene como propósito mejorar o ayudar a la digitalización de la información manejada por la planificadora a la hora de confeccionar el horario docente de la Facultad 7, brindándole al personal ventajas desde visualización hasta introducción de datos necesarios para confeccionar el horario. Está concebido para funcionar en un ambiente de integración de sistemas informáticos brindando y consumiendo Servicios Web.

CONTENIDO

Introducción.....	1
Capítulo 1 Fundamentación Teórica.....	7
1.1 Introducción.....	7
1.2 Sistemas informatizados existentes a nivel internacional vinculados al campo de acción.....	7
1.2.1 GHC 2008 (Generador de Horarios para Centros Docentes).....	8
1.2.2 KRONOWIN.....	9
1.2.3 ASC Horarios (Applied Software Consultants).....	10
1.2.4 Mimosa.....	11
1.3 Sistemas informatizados existentes a nivel nacional vinculados al campo de acción.....	13
1.4 Tendencias y tecnologías actuales.....	17
1.4.1 Metodología de desarrollo: RUP.....	17
1.4.2 Herramienta CASE: Enterprise Architect (EA) 7.1.....	19
1.4.3 Lenguajes de Programación.....	21
1.4.3.1 - Lenguaje de modelado: UML 2.0.....	21
1.4.4 Lenguajes de Programación y tecnologías del lado del Servidor.....	23
1.4.4.1 - Lenguaje de programación: PHP 5.2.5.....	23
1.4.5 Sistema Gestor de Base de Datos (SGBD): PostgreSQL 8.2.....	25
1.4.6 Patrón de arquitectura: Modelo Vista Controlador (MVC).....	27
1.4.7 Herramienta de diseño: Macromedia Dreamweaver 8.0.....	29
1.4.8 Framework: Symfony 1.1.....	30
1.5 Conclusiones.....	32
Capitulo 2 Modelación del negocio.....	34
2.1 Introducción.....	34
2.2 Modelo del negocio actual.....	34
2.3 Actores del negocio.....	35
2.4 Modelado del Negocio.....	37
2.4.1 Diagrama: Modelo de Proceso del Negocio.....	37
2.4.2 Descripción de los procesos del negocio.....	38
2.4.2.1 - Proceso “Confeccionar Horario”.....	38
2.4.2.2 - Proceso “Evaluar Horario”.....	39
2.4.2.3 - Proceso “Manejo de la afectaciones del profesor”.....	40
2.5 Diagrama de actividades de los procesos del negocio.....	40
2.5.1 Diagrama de actividad del proceso “Confeccionar Horario”.....	41
2.5.2 Diagrama de actividad del proceso “Evaluar Horario”.....	41
2.5.3 Diagrama de actividad del proceso “Manejo de Afectaciones de los profesores”.....	42
2.6 Requerimientos del sistema.....	43
2.6.1 Captura e identificación de los Requerimientos.....	43
2.6.1.1 - Definición de los requisitos funcionales.....	43
2.6.1.2 - Definición de los requisitos no funcionales.....	45
2.7 Actores del sistema.....	48
2.7.1 Diagrama de Casos de Usos del Sistema a Informatizar.....	49
2.7.2 Descripción de los Casos de Uso.....	50

2.7.2.1 - Descripción de Casos de Uso Gestionar Afectación.....	50
2.7.2.2 - Descripción de Casos de Uso Gestionar Asignatura.	52
2.7.2.3 - Descripción de Casos de Uso Gestionar Grupos.....	54
2.7.2.4 - Descripción de Casos de Uso Gestionar Locales.	56
2.7.2.5 - Descripción de Casos de Uso Gestionar Profesores.	58
2.7.2.6 - Descripción de Casos de Uso Buscar Horario.	61
2.7.2.6 - Descripción de Casos de Uso Gestionar Turnos.	62
2.8 Conclusiones.	64
Capítulo 3. Análisis y Diseño del sistema.....	65
3.1 Introducción.	65
3.2 Principios de diseño.....	65
3.2.1 Diagrama de clases del diseño	65
3.2.1.1 Modelo de clases del diseño del CU Buscar Horario.	66
3.2.1.2 Modelo de clases del diseño del CU Modificar afectaciones.....	66
3.2.1.3 Modelo de clases del diseño del CU Buscar afectaciones	67
3.3 Descripción de las clases del Diseño.	68
3.3.1 Descripción de las clases Interfaz para Gestionar Horario:.....	68
3.3.2 Descripción de las clases Interfaz para Gestionar Afectaciones:	69
3.3.3 Descripción de las clases Interfaz para Gestionar Local:	70
3.3.4 Descripción de las clases Interfaz para Gestionar Turno:	70
3.3.5 Descripción de las clases Interfaz para Gestionar Asignatura:	71
3.3.7 Descripción de las clases Interfaz para Gestionar Semestre:	72
3.3.8 Descripción de las clases Interfaz para Gestionar Tipo de Actividad: ...	73
3.3.9 Descripción de las clases controladoras de Gestión.	73
3.4 Principios de diseño empleados para el formato de la aplicación.....	76
3.4.1 Estándares en la interfaz de la aplicación.	76
3.4.2 Tratamiento de errores.....	77
3.4.3 Seguridad.....	78
3.5 Diseño de la Base de Datos.	78
3.5.1 Diagrama del Modelo de Datos.....	79
3.5.2 Descripción de las tablas del Modelo de Datos.....	79
3.6 Diagrama de secuencia del diseño.....	83
3.7 Patrones de diseños utilizados.	84
3.7.1 Alta Cohesión.....	84
3.7.2 Bajo Acoplamiento.	85
3.8 Conclusiones.	85
Capítulo 4 Implementación.....	86
4.1 Introducción	86
4.2 Diagrama de despliegue.....	86
4.3 Diagrama de Componentes	87
4.4 Conclusiones	90
Conclusiones.....	91
Recomendaciones.....	92
Referencias Bibliográficas	93
Bibliografía	96
Glosario de términos	101
Anexos	105

INTRODUCCIÓN

La sociedad cubana ha experimentado un gran desarrollo con la influencia de las nuevas tecnologías informáticas, siendo esta una de las ciencias que ha permitido la manipulación de grandes capacidades de datos y una ejecución rápida de cálculos complejos. El progreso tecnológico, que se ha propiciado por los avances científicos, pone en manos de la comunidad, utensilios y servicios para mejorar su bienestar. La informática está presente en la mayoría de los ámbitos de la sociedad como: ingeniería, industria, administraciones, públicas, medicina, diseño de vehículos, arquitectura, investigación y desarrollo, administración de empresas, restauración y arte.

Cuba ha identificado la conveniencia y necesidad de dominar e introducir en la práctica social las Tecnologías de la Información y las Comunicaciones; la cultura digital es una de las características imprescindibles del hombre nuevo, llevando a cabo la informatización de la sociedad, facilitándole a esta acercarse hacia el objetivo de un desarrollo sostenible. (1)

El impacto de esta ciencia es muy amplio, produciendo efectos positivos en la economía y la sociedad cubana. Muchas personas en el país se han ocupado de la informática desde su surgimiento con éxitos, fracasos y avances rápidos en aras de mejorar la calidad de la educación y la vida en general. Este desarrollo ha permitido que se tracen planes de modernización de las actividades que han quedado notablemente retrasadas en su accionar, como la bancaria, la hostelería y las ventas al detalle para el turismo internacional, entre otras.

"El país debe encaminarse a la modernización informática mediante un programa integral que involucre a las organizaciones que deben proveer los recursos materiales financieros, intelectuales, entidades económicas, políticas y sociales. La industria de los servicios informáticos debe asegurar la modernidad de su base técnica y organizativa, así como elevar el constante nivel científico-técnico de sus especialistas con vistas a garantizar sus propósitos."¹

¹

Resolución Económica V Congreso del PCC, (1997) Editora Política. La Habana, p. 30.

Existe un gran número de empresas e instituciones que se encuentran envueltas en un gran flujo de datos procesados de forma manual, se han elaborado herramientas informáticas para la gestión de la información permitiendo agilizar el trabajo de aquellos que estén vinculados a esta tarea, pudiendo administrar gran cantidad de datos, logrando que la toma de decisiones sea de forma rápida y concisa.

La Universidad de las Ciencias Informáticas (UCI) como proyecto de la Revolución Cubana, denominada en sus inicios "Proyecto Futuro", tiene como objetivo principal desarrollar la industria del software para contribuir al desarrollo económico del país. La UCI cuenta con tecnología informática avanzada y constituye uno de los principales centros telemáticos del país. Impulsando estrategias para informatizar las diferentes áreas administrativas y de servicios que aportan a la evolución de la economía.

El centro está formado por 10 facultades, integradas por un gran número de estudiantes y profesores tanto internos como externos. Cada facultad desarrolla un perfil productivo diferente de acuerdo a los eslabones económicos de la sociedad, donde se encuentra la medicina, economía, educación, software libre. Se le ha dado la tarea de informatizar el mayor número de procesos posibles de acuerdo a estos perfiles, prestando especial atención en aquellos que constituyan mejoras para el funcionamiento de la universidad con el único fin de convertirse en el centro de la enseñanza superior de mayor fortaleza científica del país en el área de la informática y tener un lugar importante dentro del sistema científico cubano. (2)

Uno de estos procesos de informatización que se lleva a cabo en la UCI es el de gestionar la confección de los horarios docentes para tener una buena organización de las actividades docentes, productivas entre otras. Un horario docente constituye el instrumento organizativo de las actividades docentes, que permite brindar una visión general de la distribución cuantitativa de las asignaturas contenidas en el plan de estudio, según el número de horas clases cada año.

Para confeccionar el horario docente se deben tener en cuenta varios aspectos: la asignación de locales destinados a la docencia, el número de profesores por año, la cantidad de asignaturas por año, las horas destinadas a la docencia y a la producción, la cantidad de grupos por año, la planificación de las asignaturas (P1) entre otras. Se han

desarrollado software a nivel mundial que ayudan a la elaboración de los horarios docentes en centros educacionales, pero ninguno cumple con las exigencias de la universidad porque la mayoría han sido realizados para centros pequeños y de poca matrícula.

En la Facultad 7 existe una aplicación informática que facilita la elaboración del horario docente, pero no cuenta con todas las funcionalidades necesarias para lograr un óptimo resultado. Lo que provoca que el trabajo del Vicedecano de Formación y la planificadora se vea limitado en cuanto a la eficiencia, ocasionando insatisfacción en aquellos que puedan estar involucrados, molestias innecesarias y pérdida de tiempo en ocasiones.

Por los pocos locales destinados a las labores estudiantiles que tiene la Facultad 7 y el elevado número de estudiantes, resulta difícil lograr una buena organización teniendo en cuenta que más de dos grupos no coincidan en la misma aula, así como un mismo profesor impartiendo clases en un mismo turno a varios grupos, provocando que el trabajo sea engorroso y de forma semiautomática por parte de los planificadores docentes, pues una vez elaborado el horario este tiene que corregirse de forma manual debido a que la aplicación no cumple con algunos de los requisitos que especifica el Vicedecano de Formación.

Estos errores están dados por problemas como: dos grupos pueden coincidir con el mismo profesor, la numeración de los locales docentes y laboratorios se muestra de forma incorrecta, no permite asignar otro local que no esté asignado a las labores docentes de la Facultad 7. Para los profesores esta aplicación resulta un poco incómoda, debido a que estos no tienen acceso para ingresar sus afectaciones, las cuales pueden estar dadas por distintas razones como pueden ser: enfermedades, problemas personales, posgrados, maestrías, doctorados, pases inter-semestrales, entre otros.

En la versión existente no se encuentra implementada tal funcionalidad lo que provoca que los profesores tengan que enviar sus afectaciones mediante el correo electrónico o personalización del afectado. Si la afectación es aprobada por el Vicedecano de Formación, trae como consecuencia que el horario deba reelaborarse para ese profesor u otro que pueda ser afectado para cubrir dicho turno. Otra problemática de este sistema es que para el profesor que se le ingresa una afectación determinada, no permite que se le

asigne ninguna otra tarea, tomando el día completo como afectado. No permite visualizar el horario desde cualquier lugar de la universidad, porque es una aplicación de escritorio y solamente está instalada en las máquinas de los planificadores docentes teniendo estos que enviar la planificación vía correo electrónico o mostrarla en la pizarra informativa de la Facultad 7.

Luego del análisis realizado, se enuncia como **Problema a Resolver** el ineficiente proceso de confección del horario docente en la Facultad 7 de la UCI. Se define como **objeto de estudio** el proceso de gestión de la información docente en la Facultad 7 de la UCI. El **campo de acción** se basa en el proceso de confección del horario docente en la Facultad 7 de la UCI.

El **Objetivo General** de la investigación es implementar una aplicación web que permita la gestión de la información para la confección del horario docente en la Facultad 7.

Para dar solución al objetivo general, se plantean las siguientes **Tareas de la Investigación**:

1. Realizar una valoración de los Sistemas de Gestión Académica (SGA) en el ámbito nacional e internacional, así como las tendencias tecnológicas actuales.
2. Analizar los artefactos modelados de la aplicación precedente para obtener una visión general de los procesos a informatizar.
3. Realizar un estudio de la versión actual del sistema para detectar los problemas de funcionamiento.
4. Modelar los artefactos del proceso de ingeniería del software usando la herramienta Enterprise Architect 7.1.
5. Diseñar las interfaces de comunicación con el usuario de la aplicación web.
6. Implementar las funcionalidades relacionadas con el proceso de planificación del horario docente en la Facultad 7.
7. Validar la aplicación, partiendo de las pruebas de aceptación del cliente.

Actualidad y necesidad del trabajo:

La versión con la que se cuenta está desarrollada en una aplicación de escritorio, dificultando la visualización del horario docente para las demás personas que necesitan trabajar con este horario. Aunque resuelve parte del trabajo, presenta limitaciones como las antes mencionadas en la situación problemática que hacen necesario el desarrollo de una solución que resuelva todas las dificultades.

Por las pautas definidas en el documento de arquitectura de la Facultad 7, se realizará la nueva versión en formato web, debido a las ventajas que presentan estas aplicaciones como la rápida visualización, estructura clara en cuanto a diseño, actualización constante de los datos que se usan para la elaboración de las páginas.

Este software se encargará del proceso de planificación del horario docente y reportará ventajas como:

- Optimización del funcionamiento al facilitar un acceso más rápido y sencillo de la información sobre la organización de las actividades docentes.
- Mejorar la calidad en los cambios frecuentes del horario.
- Perfeccionamiento de la gestión de la confección del horario docente en cuanto a información, la planificación de los turnos de clases, asignación de locales y profesores a los grupos docentes.
- Visualización de la planificación realizada.
- Información de las afectaciones de los profesores.
- Mejorar el tiempo estimado de la elaboración para los planificadores mediante un tratamiento más eficaz y rápido de la información.

Estas mejoras fomentarán la consecución de saltos de calidad en el proceso de gestión de los horarios docente no solo en la Facultad 7 sino en las demás facultades de la universidad.

Aportes prácticos esperados:

- Brindar a los profesores una aplicación web que les permita desde cualquier lugar de la universidad registrar sus afectaciones al sistema, para que se tengan en cuenta a la hora de realizar el horario y puedan tener una mejor planificación.
- Permitir a los estudiantes de la Facultad 7 visualizar desde cualquier lugar de la universidad el horario docente.

- Mejorar las condiciones de trabajo del personal que se encarga de la elaboración del horario docente de la Facultad 7.

El trabajo está estructurado en cuatro capítulos que se describen a continuación:

El Capítulo 1 aborda la fundamentación teórica, se estudia el estado del arte, los sistemas existentes vinculados al campo de acción, las técnicas, tecnologías y metodologías usadas en las que se apoya la solución del problema. Se mencionan las herramientas que se utilizan para realizar el diseño e implementación de este sistema.

El Capítulo 2 se refiere a las características del sistema y objeto de informatización. Se modela el negocio. Se describen los procesos del negocio. Se especifican los diagramas de actividades de los procesos del negocio así como los requisitos funcionales y no funcionales del sistema.

En el Capítulo 3 se realiza la descripción expandida de los casos de uso del sistema, obteniéndose los diagramas de interacción, diagrama de clases persistentes y despliegue, se presentan los diagramas de clases del diseño del sistema. Se representa la integración del componente con otros sistemas.

El Capítulo 4 aborda la implementación, donde se mencionarán aquellos aspectos relacionados con la construcción de la solución propuesta, se modelan los diagramas de componentes y despliegue; se describen los estándares de diseño, codificación y el tratamiento de errores para darle solución al sistema propuesto.

CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA

Introducción

Para encontrar una solución óptima computacional a los problemas de confección de horarios docentes para centros educacionales con un elevado número de estudiantes y profesores, se ha convertido en un reto trazado por generaciones. El tema de asignación de recursos se convierte en un factor importante a tener en cuenta en la totalidad de las instituciones para lograr la organización y correcto funcionamiento de las mismas.

El presente capítulo contiene la fundamentación teórica relacionada con la confección del horario docente en la Facultad 7, se estudia el estado del arte, se realiza una descripción del objeto de estudio de la investigación. Se brinda una breve panorámica sobre algunos de los sistemas informatizados existentes vinculados al campo de acción y haciendo análisis comparativo entre la solución propuesta en este trabajo con respecto a las soluciones ya existentes.

Se describen también las tecnologías actuales de desarrollo utilizadas para el diseño e implementación, abordando sobre el por qué la importancia y las ventajas de uso que brindan en el desarrollo del sistema.

1.1 Sistemas informatizados existentes a nivel internacional vinculados al campo de acción

Con el avance que están teniendo las tecnologías informáticas el hombre ha ido creando y perfeccionando herramientas para facilitar su trabajo, ayudándolo a tener una buena organización, control y planificación de las actividades que realiza. Han surgido sistemas que se puedan integrar a las diferentes ramas de la sociedad como la educación, donde existen los nombrados Sistemas de Gestión Académica.

La mayoría de estos sistemas incluyen un módulo que se encarga de la elaboración de horarios docentes pero hasta ahora resulta ser un problema porque no se ha creado una herramienta que sea cien por ciento eficiente. Los problemas de programación de horarios son un tipo de problema perteneciente al área de la optimización combinatoria que ha

atraído el interés de muchos investigadores por la alta complejidad de los mismos y la cantidad de recursos consumidos en la búsqueda de buenas soluciones. (3)

Estos problemas han sido abordados desde la década de los sesenta a través de métodos exactos y métodos aproximados. Sin embargo, los métodos exactos han demostrado poca ventaja ante problemas de tamaños muy grandes, en los cuales, realizar un análisis exhaustivo resultaría computacionalmente prohibitivo.

El hombre ha creado herramientas para tratar de darle solución al problema de confección de horarios, cada una con diferentes características adaptadas al ambiente donde se realizan. Ejemplo de ello son los Sistemas de Gestión Académica que se vinculan al campo de acción que se define, las cuales sirven como guía para darle solución a la problemática existente en la UCI, entre los más conocidos se puede citar:

1.1.1 GHC 2008 (Generador de Horarios para Centros Docentes)

El GHC es un planificador, generador y editor de horarios semanales para centros de enseñanza, creado por la empresa española Peñalará para facilitar mecanismos de interfaces de intercambio de datos y horarios con las principales aplicaciones de gestión académica. Esta aplicación tiene como objetivo racionalizar e informatizar la confección de los horarios escolares.

El software cuenta con varias versiones, la GHC 2008 es la última versión de este experimentado programa informático de generación de horarios académicos para centros de enseñanza, esta última incluye un gran número de ventajas entre las que se pueden mencionar:

- Posee una estructura de datos formada por listas relacionadas que garantiza la integridad de la información.
- Dispone de un generador de grupos de alumnos que ayuda a organizar las asignaturas partiendo de las peticiones de matrícula para las combinaciones de optativas que se oferten en el curso académico.
- Puede validar la configuración del horario previamente a la generación de resultados detectando determinadas incompatibilidades.

Capítulo 1: Fundamentación Teórica

- Durante cada proceso de generación se facilitan estadísticas de las sesiones de clases, grupos o profesores cuyos datos han dificultado la obtención de resultados con el fin de analizar las condiciones más conflictivas.
- Tiene una guía para usuarios de acuerdo a la versión en la que se trabaja.

Muchas de las funcionalidades de este sistema servirán como guía de estudio para el desarrollo de la herramienta que se quiere implementar entre las que se encuentran:

- Mantener una interfaz actualizada
- Ayuda contextual HTML
- Adaptación a Windows Vista
- Nuevas vistas de las tablas de tareas y guardias desde el editor de horarios.
- Posibilidad de configurar los datos de los alumnos y profesores haciendo distintas materias en la misma aula.(4)

Algunas desventajas que presenta la herramienta.

- Sin aportar los datos de su licencia en la instalación del programa, la lista de profesores queda limitada a 10 profesores.
- Las claves son distintas para cada versión.
- Solo corre bajo plataforma Windows.
- Es un software propietario por lo que su licencia es de un costo enorme.
- Sus versiones cada vez cuestan más en el mercado.

1.1.2 KRONOWIN

KRONOWIN es un software creado por la empresa ADOSSIS, S.A. creadora de sistemas informáticos, desde su fundación ha dirigido su actividad hacia áreas determinadas dentro de los sectores de informática y electrónica.

KRONOWIN es un completo Generador de Horarios que permite obtener de forma sencilla los horarios de un centro escolar.

Muchas son las ventajas que nos brinda este software y nos podrían servir como ayuda para la confección del sistema que se desea entre ellas se puede mencionar:

- Se puede confeccionar horarios individuales de cada grupo, profesor y aula, así como los horarios generales .Admite fijar previamente sesiones (clases), en el

horario, antes de su generación. Acepta múltiples condiciones de ubicación de clases, tanto para grupos, profesores y aulas, como independientes para cada asignatura.

- Su uso es muy intuitivo debido a que tiene una ayuda que indica los pasos a seguir. Si el usuario lo desea, el programa toma decisiones, (cambios lógicos) del horario. Tablero electrónico asistido, para retocar horario. Selección del Profesor idóneo.
- Amplias posibilidades de configuración en todo el programa.
- Entorno gráfico amigable, con gama cromática asociativa sobre datos relacionados.
- Exportación e importación a distintos formatos y aplicaciones. Formatos oficiales configurables.
- Válido para horarios escolares de cualquier tipo y niveles de estudios.
- Tiene una vista gráfica previa de las modificaciones necesarias al reubicar horas de clases.

Algunas de las desventajas que podemos mencionar de este software son:

- Necesita un microprocesador 1 Ghz así como una memoria RAM de 512 MB por lo que en muchos de nuestros centros de enseñanzas si las computadoras no cumplen con este requisito no se puede utilizar.
- Como la mayoría de los software que se producen en el mundo, este es propietario, y el costo de su licencia y actualizaciones es elevado.

Una funcionalidad importante de este software es la asistencia de entrada de datos, combinación de horarios y su generación. (5)

1.1.3 ASC Horarios (Applied Software Consultants)

ASC Horarios es otro de los generadores de horarios, posibilita cubrir todos los tipos de áreas especiales.

El programa de horarios de clases les ofrece una definición fácil y rápida de asignaturas, clases, aulas, profesores y las horas a la semana enseñadas por cada profesor.

Capítulo 1: Fundamentación Teórica

- Es posible juntar varios grupos de diferentes clases para una lección y también agregar varios profesores a una clase.
- El programa verifica la especificación del horario.
- Verifica en el horario ya generado si cumple todas las exigencias del centro para el cual se realizó.
- Se pueden hacer cambios manuales en el horario, al equivocarse el programa le avisa.
- El programa es capaz de generar informes mensuales y anuales de los estudiantes ausentes y de las suplencias de los profesores.

Algunas de las desventajas que podemos mencionar de este software son:

- Es software propietario.
- Solo corre bajo plataformas Windows.
- Es solo aplicable para centros pequeños, con capacidad para cuatro grupos y seis profesores.

Funcionalidades de ASC Horarios.

- Trasladando las fichas con el mouse consigue cambiar rápidamente partes del horario.
- El programa tiene funcionalidad de no colocar los turnos de clases en posiciones no permitidas y enseña si hay profesores libres.
- En caso de cambios inapropiados, puede deshacer las últimas 100 operaciones. Pulsando las teclas "atrás"(Undo), y con "adelante"(Redo) puede navegar por las últimas 100 operaciones.
- Pulsando la tecla derecha del ratón puede activar rápidamente las funciones necesarias y también hacer que se muestren en la pantalla lecciones que puede colocar a la posición elegida. (6)

1.1.4 Mimosa

Capítulo 1: Fundamentación Teórica

Mimosa es una herramienta de planificación de cursos y horarios adecuada para todo tipo de instituciones de enseñanza, centros educativos, organización de recursos y otras organizaciones.

Esta herramienta ofrece varias ventajas a los clientes que la usan para la elaboración de horarios y que pueden servir además de estudio para la que se quiere desarrollar en la universidad, entre estas ventajas podemos mencionar las siguientes:

- Para cualquier tipo de institución académica se puede preparar horarios individuales ya sea por grupos o asignaturas.
- Trazabilidad – Puede rehacer o deshacer hasta las últimas 1000 acciones.
- Configurable – todos los textos imprimibles, recursos y otros parámetros son definidos por el usuario.
- Independiente de sus redes y bases de datos.
- Trabaja perfectamente en escuelas de cualquier tipo y tamaño.
- Tiene acceso multiusuario, hasta 255 usuarios simultáneos, compatible con cualquier red.
- Software rápido y robusto; corre estable en cualquier plataforma Windows.
- Fácil elaboración de horarios compactos con carga de trabajo balanceada para profesores y alumnos.
- Fácil creación de una solución de horario inicial y herramientas eficientes de optimización para institutos educativos de todo tipo, empresas de adiestramiento, congresos y asambleas, eventos especiales, etc.
- Cualquier tipo de institución académica, sin importar su tamaño, puede preparar sus horarios fácilmente tan solo siguiendo los pasos de introducción de datos sobre profesores o instructores, recursos disponibles, cursos o asignaturas, alumnos, secciones, etc.

- La aplicación es extremadamente flexible y robusta

Desventajas

- Solamente soporta los sistemas operativos: Win98, WinNT, Win2000, WinME, WinXP, Win2003.
- Es un software propietario.
- Necesita 4 Mb de memoria RAM.

Funcionalidades de Mimosa.

- Opera sin importar las dificultades de programación. (7)

1.2 Sistemas informatizados existentes a nivel nacional vinculados al campo de acción.

La informatización de la sociedad cubana es un proceso que busca lograr una mejor eficiencia utilizando las Tecnologías de la Información y las Comunicaciones (TIC) para permitir un mayor aumento de la calidad de vida de los ciudadanos, logrando satisfacer muchas de las necesidades en cuanto a la obtención de información y conocimiento, además de permitir la creación de sistemas que puedan integrarse a muchas de las esferas de la sociedad.

Una de las ramas donde la informatización ha tenido un gran avance es en la educación. A través de los años las instituciones educacionales se han especializado y aumentado su capacidad educacional y científica. Hoy no solo se habla de las escuelas rurales con un maestro y un único estudiante, sino de centros educacionales nacionales con aulas, laboratorios docentes, profesores y estudiantes. Existe una fuerte vinculación de los preceptos martianos de la combinación del estudio y el trabajo como vínculo fundamental para la formación integral del hombre nuevo. Uno de los mayores problemas que aún no se han podido resolver en esta esfera es contar con una herramienta que permita la confección de los horarios docentes (8).

Capítulo 1: Fundamentación Teórica

Debido a que la administración y organización de grandes centros educacionales es lo suficientemente compleja y determinante para la calidad formativa de sus educandos, es una tarea fundamental y esencial en este tipo de institución mantener una organización y conformación de los horarios docentes. Muchas de las universidades cubanas y fundamentalmente la UCI marchan al frente del proceso de informatización de toda la sociedad significando la creación de una nueva visión con respecto al manejo estratégico de los recursos informacionales en función del cumplimiento de su misión , su fomento, su disponibilidad y acceso.

La UCI fundada en septiembre del 2002 por el Comandante en Jefe de la Revolución compañero Fidel Castro Ruz, dentro de sus objetivos recoge informatizar los procesos que le ayudan a su funcionamiento como un centro estudiantil de excelencia donde uno de estos procesos es el de confección de los horarios docentes. La universidad cuenta con diez facultades, en cada una de ellas existe un vicedecano de formación y un planificador docente.

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 establecidas por el vicedecano de formación. Estas personas son las encargadas de la confección del horario docente en cada una de las facultades.

Para lograr una buena organización es necesario llevar una adecuada planificación de las actividades docentes, deportivas, culturales, investigativas, políticas y productivas, siendo el horario docente un método eficaz para alcanzar un buen balance entre estas actividades.

Muchos de los procesos que se llevan a cabo en la UCI están informatizados, incluyendo los de confección del horario docente; actualmente para el desarrollo del trabajo se realizó una investigación donde se recogieron por todas las facultades datos de como se elabora el horario docente y cuales documentos o aplicaciones se cuenta para realizar dicha tarea. Se investigó lo siguiente:

Capítulo 1: Fundamentación Teórica

En la Facultad 7 se realiza el horario docente utilizando como apoyo la herramienta **HDF7**, la cual se utiliza para la confección de horarios docentes.

Tiene un gran número de ventajas como:

- Generar un horario automáticamente.
- Ayuda a optimizar el tiempo de elaboración del horario docente.
- Disminuye los errores y permite agilizar el trabajo del personal encargado de la confección del horario docente.

Tiene excepciones que son muy ventajosas para las personas que trabajan en la confección del horario docente, entre las que se pueden mencionar que un turno de clases no coincida con otro, que pueda permitir especificar el turno en conferencia, laboratorio o clase práctica, para un profesor externo no se le ponga tercer turno, entre otras. (9)

Como no es un software 100% eficiente presenta desventajas como:

- No cuenta con todas las funcionalidades que puedan responder a las necesidades de la facultad.
- Describe un flujo de actividades que tiene contradicciones.
- No permite la visualización del horario desde cualquier lugar de la universidad.

El módulo de Horarios Docentes es una herramienta de mucha utilidad para confeccionar no sólo los horarios del personal docente de cualquier nivel, sino para llevar un control de los lugares que se comparten por distintos grupos de alumnos, pero es una actividad engorrosa dicha elaboración debido a la gran cantidad de recursos que se asignan como las aulas y laboratorios. Las herramientas que antes se mencionan tanto a nivel internacional como nacional, son software que han combinado diferentes algoritmos tanto matemáticos como de programación para lograr darle solución al problema de confección del horario docente.

Los problemas que acarrea confeccionar un horario y la asignación de recursos (aulas, profesores) en instituciones educativas han sido estudiados en diferentes investigaciones educacionales desde los años 60. En general son complejos, y cada uno de ellos tiene características propias que hacen que las soluciones de un caso no puedan fácilmente ser

Capítulo 1: Fundamentación Teórica

adaptadas a otro. En su mayoría se modelan como problemas de programación entera, con un enorme número de variable. (10)

Entre las técnicas utilizadas hasta ahora para resolver este problema de los horarios se tienen los Algoritmos Genéticos (AGs), los que se utilizan para máquinas de aprendizaje y para problemas con una gran cantidad de datos, se basan en la metaheurística Tabú Search, algoritmo Vértice-Coloración el cual relaciona las asignaturas con los alumnos en un grafo.

Dicho algoritmo lo que hace es que cada alumno tiene una lista de asignaturas a las que está matriculado, y hace un recorrido de todos los alumnos y las asignaturas en las que este matriculado un mismo alumno, este le pone una arista entre ese par de asignaturas, indicando que esas dos asignaturas no pueden darse en una misma hora. De esta manera se obtiene otro grafo, pero con tan solo las relaciones existentes entre todas las asignaturas, luego colorea ese grafo, y hace un vértice de coloración. (11)

Otro algoritmo que se puede mencionar es el de Colonia de Hormigas que no es más que una técnica probabilística utilizada para solucionar problemas de cómputo; este algoritmo está inspirado en el comportamiento que presentan las hormigas para encontrar las trayectorias desde la colonia hasta el alimento. El algoritmo de la colonia de la hormiga puede funcionar continuamente y adaptarse a los cambios en tiempo real. Este algoritmo resulta ser una buena opción para resolver problemas de programación de horarios, siendo todos estos algoritmos complejos.

Los sistemas existentes para confeccionar horarios docentes, son en su mayoría software propietario por lo que es necesario comprar las licencias, siendo muy costosas obtener cada versión que se les hace, provocando muchos gastos a la universidad. Por lo general todos soportan el sistema operativo Windows, y como la UCI está enmarcada en migrar todas sus aplicaciones para software libre, no cumplirían con las funcionalidades necesarias. Por la extensa matrícula del centro y el elevado número de profesores, no permite que sean usados debido a que están confeccionados para centros de poca matrícula.

Por las características que se mencionan es necesaria la realización de un producto que se enfoque en el proceso y ambiente para el cual se le quiere dar uso en la Facultad 7 obviando las aplicaciones existentes.

El sistema que se propone permitirá obtener un horario docente que cumplan con los requerimientos de la facultad, la cual presenta características particulares, disminuyendo a su vez el tiempo de elaboración así como mejorar el tratamiento de la información. En la nueva propuesta del sistema se van a integrar todas aquellas funcionalidades con las que se pueda obtener un óptimo resultado.

1.3 Tendencias y tecnologías actuales

Teniendo en cuenta las necesidades evidenciadas en la situación problemática y las características que presenta el entorno en el que se pondrá en práctica la solución propuesta, se realizó un estudio sobre las tendencias y tecnologías actuales a emplear definidas en el documento de arquitectura de la Facultad 7, las cuales se describen a continuación.

Cada día la producción de software busca adecuarse más a las necesidades del usuario. Para lograr la productividad del software se necesita un proceso que integre las múltiples facetas del desarrollo del mismo. Se hace necesario definir la metodología de ingeniería del software que guiará el proceso de informatización donde se ha definido a RUP

1.3.1 Metodología de desarrollo: RUP

El RUP (Rational Unified Process) en español (Proceso Unificado de desarrollo) es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language (Lenguaje Unificado de Modelado), constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. RUP es en realidad un refinamiento realizado por Rational Software del más genérico Proceso Unificado.

Los orígenes de RUP se remontan al modelo espiral original de Barry Boehm. Ken Hartman, uno de los contribuidores claves de RUP colaboró con Boehm en la investigación. En 1995 Rational Software es comprada por una compañía sueca llamada

Capítulo 1: Fundamentación Teórica

Objectory AB. El RUP fue el resultado de una convergencia de Rational Approach y Objectory, proceso desarrollado por el fundador de Objectory Ivan Jacobson. El primer resultado de esta fusión fue el Rational Objectory Process, la primera versión de RUP, fue puesta en el mercado en 1998, siendo el arquitecto en jefe Philippe Kruchten. RUP posee como características principales:

- Forma disciplinada de asignar tareas y responsabilidades (quién hace qué, cuándo y cómo).
- Pretende implementar las mejores prácticas en Ingeniería de Software.
- Desarrollo iterativo.
- Administración de requisitos.
- Uso de arquitectura basada en componentes.
- Control de cambios.
- Modelado visual del software.
- Verificación de la calidad del software.

RUP se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso. Incluye artefactos (que son los productos tangibles del proceso como por ejemplo, el modelo de casos de uso, el código fuente, etc.) y roles (papel que desempeña una persona en un determinado momento, una persona puede desempeñar distintos roles a lo largo del proceso). (12)

RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización; se basa en los siguientes seis principios:

1. Adaptar el proceso: El proceso deberá adaptarse a las características propias del proyecto u organización. El tamaño del mismo, así como su tipo o las regulaciones que lo condicionen, influirán en su diseño específico.
2. Balancear prioridades: Los requerimientos de los diversos inversores pueden ser diferentes, contradictorios o disputarse recursos limitados. Debe encontrarse un balance que satisfaga los deseos de todos.

3. Colaboración entre equipos: El desarrollo de software no lo hace una única persona sino múltiples equipos. Debe haber una comunicación fluida para coordinar requerimientos, desarrollo, evaluaciones, planes, resultados, etc.
4. Demostrar valor iterativamente: Los proyectos se entregan, aunque sea de un modo interno, en etapas iteradas. En cada iteración se analiza la opinión de los inversores, la estabilidad y calidad del producto, y se refina la dirección del proyecto.
5. Elevar el nivel de abstracción: Este principio dominante motiva el uso de conceptos reutilizables tales como patrón del software, lenguajes 4GL2 o esquemas (frameworks3) por nombrar algunos.

Esto previene a los ingenieros de software ir directamente de los requisitos a la codificación de software a la medida del cliente. Un nivel alto de abstracción también permite discusiones sobre diversos niveles arquitectónicos. Éstos se pueden acompañar por las representaciones visuales de la arquitectura, por ejemplo con UML.

6. Enfocarse en la calidad: El control de calidad no debe realizarse al final de cada iteración, sino en todos los aspectos de la producción. (13)

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

- *Concepción*: se hace un plan de fases, se identifican los principales casos de uso y se identifican los riesgos.
- *Elaboración*: se hace un plan de proyecto, se completan los casos de uso y se eliminan los riesgos.
- *Construcción*: se concentra en la elaboración de un producto totalmente operativo y eficiente y el manual de usuario.
- *Transición*: se implementa el producto en el cliente y se entrena a los usuarios. Como consecuencia de esto suelen surgir nuevos requisitos a ser analizados. (15)

1.3.2 Herramienta CASE: Enterprise Architect (EA) 7.1

EA es una herramienta de análisis y diseño que utiliza el Lenguaje Unificado de Modelado (UML), comprensivo, cubriendo el desarrollo de software desde la obtención de los

Capítulo 1: Fundamentación Teórica

requisitos, diseño del modelo, pruebas, cambio de control y mantenimiento para la implementación, con completa trazabilidad.

EA combina el poder de la última especificación UML con alta performance, una interfaz intuitiva, para brindar un modelado avanzado a todo el equipo de desarrollo. Es una herramienta visual con multi-usuario con un gran establecimiento de características, ayudando a los analistas, testers, administradores de proyectos, personal del control de calidad y desarrolladores alrededor del mundo a construir y documentar, software sostenible. (14)

El software de Sparx Systems es usado en el desarrollo de muchos tipos de aplicaciones y sistemas en un amplio rango de industrias, incluyendo: aeroespaciales, bancarias, desarrollo de web, ingeniería, finanzas, medicina, militar, investigación, academia, transporte, reventa, servicios e ingeniería eléctrica. Es también usada efectivamente para UML y el entrenamiento de la arquitectura de la empresa en muchas universidades prominentes, entrenando compañías y universidades en todo el mundo.

Enterprise Architect es renombrado por su enriquecida lista de características. Algunas de ellas están resaltadas en la siguiente lista:

- Modelado Extendido para requisitos, diseño de la interfaz de usuario, mapeo de ideas, modelado de datos y más.
- Velocidad: EA es rápido para cargar y un actor espectacularmente rápido, inclusive con grandes modelos.
- Escalabilidad: EA soporta usuarios simples y el desarrollo de pequeños modelos, o muchos usuarios concurrentes desarrollando modelos extremadamente grandes, con igual facilidad.
- Usabilidad: EA deja iniciar y producir rápidamente, una interfaz de usuario enriquecida, y da la habilidad de crear patrones, plantillas, vistas de modelo y colecciones “favoritas” de los elementos y diagramas más comúnmente.
- La Administración de Requisitos integrada le permite especificar, trazar y verificar requisitos directamente contra el diseño.
- Extensivo soporte para la administración del proyecto, incluyendo recursos, métricas, definición de pruebas, seguimiento y administración.

- Auditoria de proyecto, líneas base y utilidades de comparación que le permiten registrar y monitorear cambios y (con comparación) volver los cambios seleccionados a un estado anterior.
- Control de versión usando una herramienta de compilación de cualquier Control de Código Fuente Común (SCC).
- Seguridad de usuario y grupo integrada, y administración del control de acceso.
- Soporte de pruebas para casos de prueba, JUnit y NUnit.
- Genera reportes detallados con la información que necesita en el formato que su compañía o cliente demanda.
- Permite navegar y explorar su modelo de código fuente en el mismo ambiente.(15)

Soporta la generación directa e ingeniería reversa de código fuente para muchos lenguajes populares incluyendo C++, C#, Java, Delphi, VB.Net, Visual Basic, ActionScript, Python y PHP. Con un editor de código fuente “resaltador de sintaxis” integrado. Permite rápidamente navegar y explorar su código fuente del modelo en el mismo ambiente. El manual de usuario está disponible en línea. Por ser intuitivo y simple de usar hacen posible que esta sea la herramienta destinada para modelar el negocio. (16)

1.3.3 Lenguajes de Programación

Un lenguaje de programación es un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. Es utilizado para controlar el comportamiento físico y lógico de una máquina. (17)

1.3.3.1 - Lenguaje de modelado: UML 2.0

UML es el lenguaje de modelado más conocido y utilizado en la actualidad; aún cuando todavía no es un estándar oficial, está apoyado en gran manera por el Grupo de Administración de Objetos (OMG). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. Ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como: procesos de negocios, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables.

Capítulo 1: Fundamentación Teórica

Es importante destacar que UML es un "lenguaje" para especificar y no un método o un proceso, se utiliza para definir un sistema de software, para detallar los artefactos en el sistema, documentar y construir. La versión que usa este trabajo es la 2.0 debido a las ventajas que ofrece con respecto a su anterior (1.0). En UML 2.0 se definen una serie de diagramas adicionales a los establecidos en su versión anterior. (20) Compárese las figuras que se muestran a continuación:

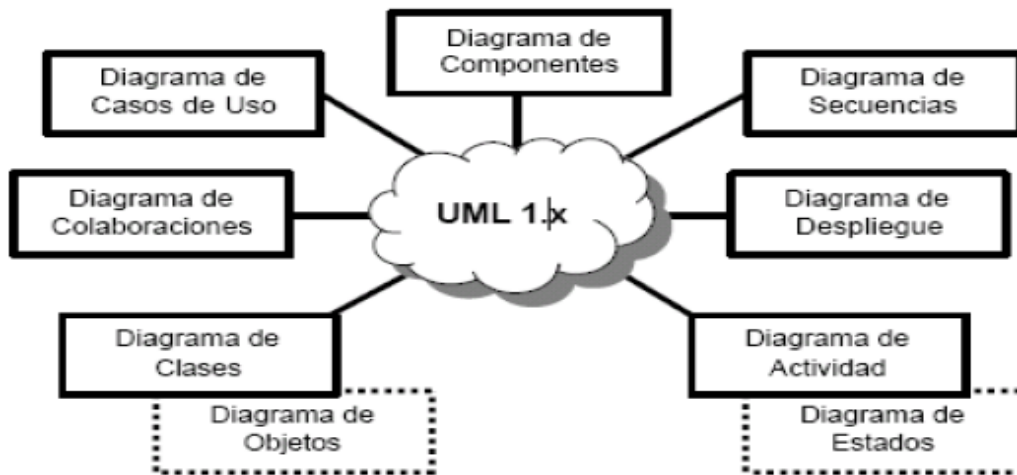


Fig.1 Diagramas de UML 1.0

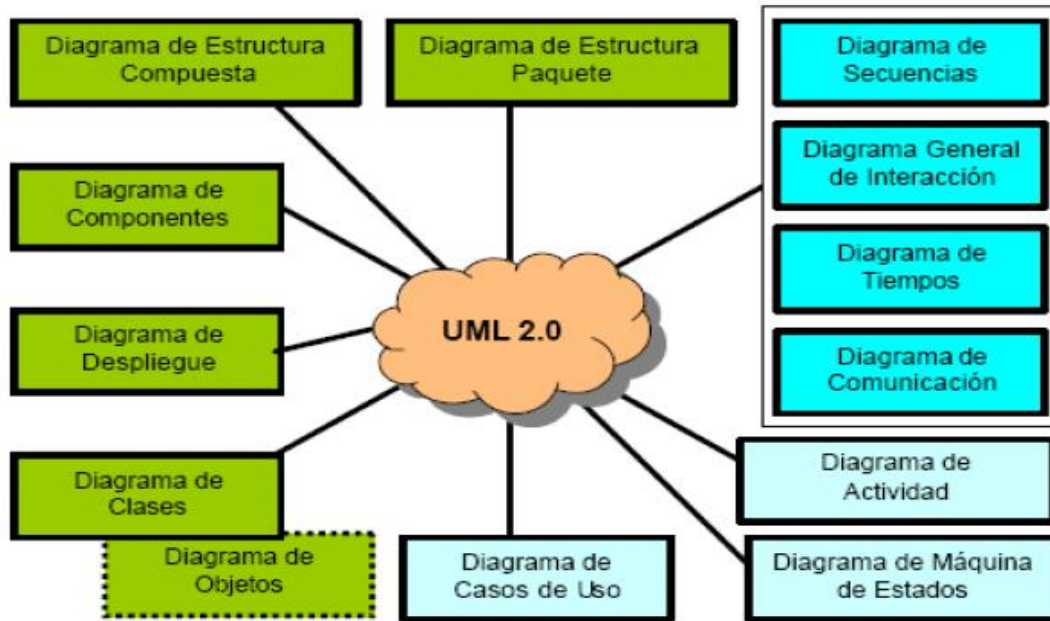


Fig.2 Diagramas de UML 2.0

UML 2.0 es la mayor revisión que se le ha hecho a UML desde la versión 1.0. El modelo conceptual ha sido reestructurado completamente y nuevos diagramas han sido incorporados. Los diagramas tradicionales también han sido mejorados. La nueva versión permitirá a los fabricantes de herramientas CASE, proporcionar a los analistas, arquitectos y desarrolladores, herramientas cada vez más potentes; que les permitan aprovechar mejor los modelos y como consecuencia generar una mayor cantidad de código; reduciendo significativamente el ciclo de desarrollo de sus aplicaciones. (18)

1.3.4 Lenguajes de Programación y tecnologías del lado del Servidor

1.3.4.1 - Lenguaje de programación: PHP 5.2.5

Existe una multitud de lenguajes concebidos o no para Internet. Cada uno de ellos explota más a fondo ciertas características que lo hacen más o menos útiles para desarrollar distintas aplicaciones y además están en dependencia de lo que quiera hacer el cliente con él. En el dominio de la red, los lenguajes del lado del servidor más ampliamente utilizados para el desarrollo de páginas dinámicas son el ASP, PHP, PERL y JAVA.

Capítulo 1: Fundamentación Teórica

PHP: (Hypertext Preprocessor), es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor.

Una de sus características más potentes es el soporte para gran cantidad de bases de datos. Entre las que se pueden mencionar InterBase, mSQL, MySQL, Oracle, Informix, PostgreSQL, entre otras. PHP también ofrece la integración con varias bibliotecas externas, que dan al desarrollador la posibilidad de realizar cualquier tarea, desde generar documentos en pdf (Portable Document Format) hasta analizar código XML (extensible Markup Language) y también para la creación de otro tipo de programas incluyendo aplicaciones con interfaz gráfica usando la Librería Gráfica GIMP Tool Kit (GTK+). (19)

Es software libre, lo que implica menos costos y servidores más baratos que otras alternativas. Es muy rápido y su integración con la base de datos MySQL y el servidor Apache, le permite constituirse como una de las alternativas más atractivas del mercado. Es multiplataforma, funciona tanto para Unix (con Apache) como para Windows (con Microsoft Internet Information Server) de forma que el código que se haya creado para una de ellas no tiene por qué modificarse al pasar a la otra.

Su sintaxis está inspirada en C, ligeramente modificada para adaptarlo al entorno en el que trabaja, de modo que si se está familiarizado con esta sintaxis, le resultará muy fácil aprender PHP. Su librería estándar es realmente amplia, lo que permite reducir los llamados "costes ocultos", uno de los principales defectos de ASP (Active Server Pages). PHP tiene una de las comunidades más grandes en Internet, con lo que no es complicado encontrar ayuda, documentación, artículos, noticias, y recursos. Ofrece una solución simple y universal para las paginaciones dinámicas como la web de fácil programación. Su diseño elegante lo hace perceptiblemente, más fácil de mantener y ponerse al día, a diferencia con el código de otros lenguajes.

Debido a su amplia distribución PHP está perfectamente soportado por una gran comunidad de desarrolladores. Como producto de código abierto, PHP goza de la ayuda de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y se reparan rápidamente. El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar las capacidades de PHP.

El equipo de desarrollo de PHP ha anunciado la disponibilidad del nuevo PHP 5.2.5. Esta revisión viene a solucionar algunos problemas de estabilidad en la serie 5.2.x solucionando más de 60 errores, algunos de ellos que afectaban a la seguridad. Es recomendable realizar la actualización a esta nueva versión. (20)

1.3.5 Sistema Gestor de Base de Datos (SGBD): PostgreSQL 8.2

El software de bases de datos ha experimentado un auge extraordinario a raíz de la progresiva informatización de casi la totalidad de las empresas de hoy día, rapidez, efectividad en los procesos y los grandes flujos de información son la necesidad más apremiante a la hora de optimizar servicios y productos.

Ante esta notable demanda de soluciones informáticas han surgido multitud de gestores de bases de datos, estos son programas que permiten manejar la información de modo sencillo y que prestan servicios para el desarrollo y el manejo de bases de datos.

Objetivos de un SGBD

En un ambiente multiusuario el SGBD ofrece un control centralizado de la información. Los objetivos que plantean estos sistemas están relacionados con la intención de evitar los problemas que existían en los sistemas de información orientados a los procesos. (21)

Los principales objetivos son:

- Evitar la redundancia de los datos, eliminando así la inconsistencia de los mismos.
- Mejorar los mecanismos de seguridad de los datos y la privacidad. Se pueden distinguir cuatro tipos de contextos para usar mecanismos de seguridad: seguridad contra accesos indebidos a los datos, seguridad contra accesos no autorizados a la base de datos, seguridad contra destrucción causada por el entorno (fuego, inundación, robo, etc.), seguridad contra fallos del propio sistema (fallos del hardware, del software, etc.).

- Asegurar la independencia de los programas y los datos, es decir, la posibilidad de modificar la estructura de la base de datos (esquema) sin necesidad de modificar los programas de las aplicaciones que manejan esos datos.
- Mantener la integridad de los datos realizando las validaciones necesarias cuando se realicen modificaciones en la base de datos.
- Mejorar la eficacia de acceso a los datos, en especial en el caso de consultas imprevistas. (22)

Las principales funciones que debe realizar un SGBD son:

- La definición de los datos.
- La manipulación de los datos.
- Garantizar la seguridad e integridad de los datos.
- La gestión de las transacciones y el acceso concurrente.

Rendimiento

Existen monitores del SGBD que van encaminados a averiguar los recursos de la máquina que se están consumiendo, proporcionan tiempos específicos del sistema gestor. Hay la posibilidad de realizar programas específicos que midan exactamente el conjunto de características deseado. Estos programas se basan en la anotación de la hora del sistema antes y después de realizar las operaciones cuyo rendimiento se quiere medir.

Por todos los aspectos vistos anteriormente vimos que con respecto a nuestras necesidades se escogió como sistema de gestión de bases de datos:

PostgreSQL 8.2: es un sistema de gestión de base de datos relacional orientada a objetos de software libre, publicado bajo la licencia Berkeley Software Distribution (BSD). Posee una serie de características positivas respecto a otros. Por ejemplo:

- Posee una gran escalabilidad. Es capaz de ajustarse al número de procesadores y a la cantidad de memoria que posee el sistema de forma óptima, haciéndole capaz de soportar una mayor cantidad de peticiones simultáneas de manera correcta se dice que ha llegado a soportar el triple de carga de lo que soporta MySQL.

- Implementa el uso de rollbacks, subconsultas y transacciones, haciendo su funcionamiento mucho más eficaz.
- Tiene la capacidad de comprobar la integridad referencial, así como también la de almacenar procedimientos en la propia base de datos.
- Multiusuario, multiprogramado, con arquitectura cliente-servidor y control de privilegios de acceso. Además, como dice un Internauta, “Yo creo que los logos que han elegido para cada uno son bastante ilustrativos. PostgreSQL es un elefante: muy robusto y que aguanta una tremenda carga. MySQL es el delfín: bonito, rápido, estilizado, pero con sus limitaciones”. (23)

Las características fundamentales de PostgreSQL son:

- DBMS Objeto-Relacional: PostgreSQL aproxima los datos a un modelo objeto-relacional, y es capaz de manejar complejas rutinas y reglas. Ejemplos de su avanzada funcionalidad son consultas SQL declarativas, control de concurrencia multi-versión, soporte multi-usuario, transacciones, optimización de consultas, herencia, y arreglos.
- Altamente extensible: PostgreSQL soporta operadores, funcionales métodos de acceso y tipos de datos definidos por el usuario.
- Integridad Referencial: PostgreSQL soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos.
- Lenguajes Procedurales: PostgreSQL tiene soporte para lenguajes procedurales internos, incluyendo un lenguaje nativo denominado PL/pgSQL. Este lenguaje es comparable al lenguaje procedural de Oracle, PL/SQL. Otra ventaja de PostgreSQL es su habilidad para usar Perl, Python, o TCL como lenguaje procedural embebido.

1.3.6 Patrón de arquitectura: Modelo Vista Controlador (MVC)

Los patrones de software son soluciones reutilizables a los problemas que ocurren durante el desarrollo de un sistema de software. Éstos proporcionan un proceso consistente o diseño que uno o más desarrolladores pueden utilizar para alcanzar sus

Capítulo 1: Fundamentación Teórica

objetivos. También proporcionan una arquitectura uniforme que permite una fácil expansión, mantenimiento y modificación de una aplicación.

Las aplicaciones Web pueden desarrollarse utilizando cualquier arquitectura. La arquitectura del patrón MVC es un paradigma de programación bien conocido para el desarrollo de aplicaciones con interfaz gráfica. El patrón MVC es un patrón de arquitectura de software en el cual todo el proceso está dividido en tres capas, típicamente estas capas son el Modelo, la Vista y el Controlador.

El Modelo incorpora la capa del dominio y persistencia, es el encargado de guardar los datos en un medio persistente (ya sea una base de datos, un archivo de texto, XML3, registro, etc.). Se refiere a la lógica del negocio o servicio y los datos asociados con la aplicación. La Vista se encarga de presentar la interfaz al usuario, en sistemas Web, esto es típicamente HTML, aunque pueden existir otros tipos de vistas. En la vista, sólo se deben realizar operaciones simples. Es la encargada de la presentación de los datos.

El Controlador es el que escucha los cambios en la vista y los envía al modelo, este último retorna los datos a la vista. Es el que atiende las peticiones y componentes para la toma de decisiones de la aplicación. El propósito del MVC es aislar los cambios. Es una arquitectura preparada para los cambios, que separa los datos y lógica de negocio de la lógica de presentación. (24) A continuación se muestra un esquema de este modelo:

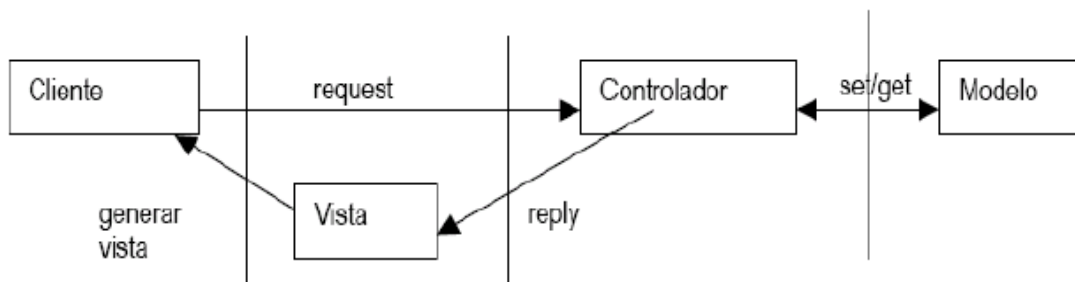


Fig.3 Esquema del Patrón MVC.

1.3.7 Herramienta de diseño: Macromedia Dreamweaver 8.0

Dreamweaver es la herramienta de diseño de páginas web más avanzada, tal como se ha afirmado en muchos medios. Aunque sea un experto programador de HTML el usuario que lo maneje, siempre se encontrarán en este programa razones para utilizarlo, sobre todo en lo que a productividad se refiere.

Cumple perfectamente el objetivo de diseñar páginas con aspecto profesional, y soporta gran cantidad de tecnologías, además muy fáciles de usar:

- Hojas de estilo y capas
- Javascript para crear efectos e interactividades
- Inserción de archivos multimedia.

Además es un programa que se puede actualizar con componentes, que fabrica tanto Macromedia como otras compañías, para realizar otras acciones más avanzadas. (25) Dreamweaver aunque también se puede adquirir por separado, forma parte de la suite de Macromedia, en la que este editor HTML resulta una pieza fundamental. Continúa siendo fiel a mantener un nivel alto en sus herramientas, dirigiéndose a usuarios con conocimientos previos.

Al iniciar Macromedia Dreamweaver aparece, como primera novedad, la elección de una modalidad de programación, lista formada por ASP.Net, PHP, ColdFusion, y HTML. Luego se presenta otra selección, el ambiente de trabajo, en donde encontramos las opciones, ya conocidas de anteriores versiones, WYSIWYG (que consiste en diseñar una página Web sin necesidad de escribir ningún código), la opción de trabajar con el Código, y por último la posibilidad de ver ambas ventanas de desarrollo a la vez.

Los lenguajes de programación que domina Dreamweaver son ASP, CSS, PHP, SQL, JSP, y XML. El potencial del software en cuanto a la capacidad de programar bajo los lenguajes que acabamos de citar es de lo más amplio, permitiendo la creación de aplicaciones y diseños web complejos.

Uno de los puntos de mayor énfasis en esta edición de Dreamweaver son el soporte y las características de desarrollo en CSS(Cascading Style Sheet,) haciendo posible creaciones con más facilidad y precisión, aplicando herramientas capaces de inspeccionar el código escrito. Otro aspecto capaz de ser analizado es la compatibilidad del sitio con los diversos navegadores, para que todos puedan visualizar la página correctamente.

La posibilidad de crear botones flash, formularios, JavaScripts, y más, es de gran ayuda. Además de poder insertar elementos web, tiene una gran precisión en la importación de información de Word y Excel, con las funciones de copiar y pegar.

Otras características:

- Dreamweaver es conocido por ofrecer herramientas avanzadas en el desarrollo de sitios web.
- Un administrador de sitios, para agrupar los archivos según el proyecto al que pertenezcan.
- Un cliente FTP integrado, que permite subir los archivos editados inmediatamente al sitio en Internet.
- Función de autocompletar y resaltado de la sintaxis para instrucciones en HTML y lenguajes de programación como PHP, JSP o ASP. (26)

1.3.8 Framework: Symfony 1.1

Un framework simplifica el desarrollo de una aplicación mediante la automatización de algunos patrones utilizados para resolver tareas comunes. Proporciona estructura al código fuente, forzando al desarrollador a crear código más legible y fácil de mantener. Un framework facilita la programación de aplicaciones, ya que encapsula operaciones complejas en instrucciones sencillas.

Symfony es un framework diseñado para optimizar el desarrollo de las aplicaciones Web creado con PHP 5. Separa las capas del modelo cliente servidor como la lógica de negocio, la lógica de servidor y la presentación de la aplicación Web. Proporciona varias

Capítulo 1: Fundamentación Teórica

herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación Web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. Symfony se diseñó para que se ajustara a los siguientes requisitos:

- Fácil de instalar y configurar en la mayoría de las plataformas (y con la garantía que funciona correctamente en los sistemas Windows).
- Independiente del sistema gestor de bases de datos.
- Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.
- Sigue la mayoría de las mejores prácticas y patrones de diseño para la Web.
- Preparado para aplicaciones empresariales y es adaptable a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo.

Este framework toma lo mejor de la arquitectura MVC y la implementa de forma tal que el desarrollo de la aplicación sea rápido y sencillo. El flujo de trabajo de Symfony es el que se muestra en la siguiente figura:

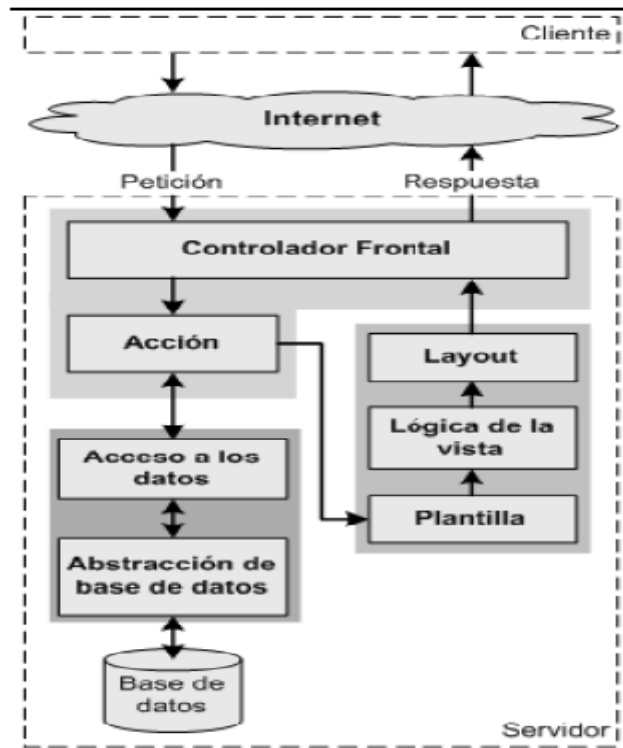


Fig.5 Flujo de trabajo de Symfony.

El uso de un framework que utiliza MVC obliga a dividir y organizar el código de acuerdo a las convenciones establecidas por el framework. El código de la presentación se guarda en la vista, el código de manipulación de datos se guarda en el modelo y la lógica de procesamiento de las peticiones constituye el controlador. Aplicar el patrón MVC a una aplicación resulta bastante útil además de restrictivo.

Symfony facilita el mantenimiento más sencillo de las aplicaciones, la encapsulación, y mayor seguridad en el acceso a la base de datos. Por tales razones, se propone este framework como una de las herramientas a usar para simplificar el desarrollo de este componente. (27)

Conclusiones

La manera en que hoy en día se desarrolla el proceso de confección de horarios docentes no es la más factible. La inexistencia de un software que tenga exactamente las funcionalidades que se requieren para llevar a cabo este proceso, hacen necesaria la implementación de una aplicación que sí cumpla con estos objetivos.

Una vez analizado la situación problemática, definidos los objetivos y el campo de acción, se podrá desarrollar la modelación del negocio y el diseño. Este sistema permitirá cumplir todas las funcionalidades que se requieren para obtener un horario que cumpla con todas la especificaciones pedidas por el cliente.

El proceso se llevará a cabo empleando las tecnologías definidas en el documento arquitectura de la Facultad 7. Se resolverá uno de los mayores problemas organizacionales que presenta no solo la universidad sino los centros educacionales del país.

CAPÍTULO 2 MODELACIÓN DEL NEGOCIO

Introducción

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

El modelo del negocio, posibilita obtener una visión más clara del proceso para dar respuesta a la solicitud del usuario por lo que en este capítulo se abordan temas relacionados con el negocio, así como las condiciones o reglas que deben cumplirse, la descripción de los actores y trabajadores del negocio. (32)

1.2 Modelo del negocio actual

Como paso inicial para el desarrollo del software se realizó un estudio detallado del área de planificación de la Facultad 7 donde se utiliza una herramienta que ayuda a generar el horario, con el objetivo de identificar los procesos y actividades que no cumplan con las funcionalidades requeridas en dicha tarea, introducir la información necesaria y las condiciones que resultan imprescindibles para su elaboración y publicación.

Estas actividades ayudarán a obtener los requerimientos necesarios, posibilitando identificar los casos concretos que debe tener el sistema, la relación que debe existir entre la ingeniería de software y el negocio, con el fin de aclarar el enfoque que quiere tener el cliente con el software, más un conjunto de artefactos para iniciar la modelación de dicha propuesta de solución al problema presentado.

Se identificaron las reglas del negocio que describen políticas que deben cumplirse o condiciones que deben satisfacerse, es decir, reglas que regulan los distintos aspectos del negocio. Se hicieron varias entrevistas a los clientes de este producto, que en este caso son, el Vicedecano de Docencia y la Planificadora Docente, con el objetivo de recoger y visualizar como es que se está confeccionando actualmente el horario docente en la Facultad 7.

En estas investigaciones se identificaron algunas reglas generales como:

- La planificadora y el Vicedecano de Formación son las únicas personas autorizadas a la realización del horario.
- Los jefes de departamentos de cada asignatura brindan la relación de profesores por asignaturas.
- El vicedecano de formación es la persona capacitada para asignarle a cada profesor los grupos a los que impartirá clases.
- Se debe respetar la planificación del programa de las asignaturas (P1) en caso de ocurrir algún imprevisto.
- Se debe tener en cuenta en el momento de la confección del horario la planificación de los profesores de acuerdo a sus afectaciones, como maestría o cursos de superación, que los profesores presenten.
- Los profesores externos deben tener sus turnos entre el primero y el quinto preferentemente.
- Tener en cuenta el peso de las asignaturas para que no coincidan en un mismo día varias asignaturas de mayor peso.
- No se debe planificar, para profesores ni estudiantes, el tercero y cuarto turno de clases consecutivamente, para respetar el horario de almuerzo.

1.3 Actores del negocio

Un actor del negocio es cualquier individuo, grupo, organización, máquina o sistema de información externo que interactúa con el negocio. El término actor significa el rol que algo o alguien juega cuando interactúa con el negocio para beneficiarse de sus resultados. De acuerdo con esta idea un actor del negocio representa un tipo particular de usuario del negocio más que un usuario físico, ya que varios usuarios físicos pueden realizar el mismo papel en relación al negocio o sea, ser instancias de un mismo actor. (28)

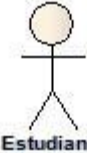
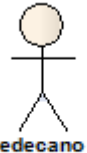
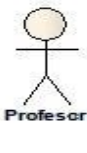
Capítulo 2: Modelación del Negocio


En esta fase inicial del proceso de desarrollo se identificaron 4 actores del negocio:

1. Estudiante
2. Vicedecano de Formación
3. Profesor
4. Usuario_Facultad.

Los tres primeros actores conforman una generalización, obteniéndose como resultado un nuevo actor Usuario_Facultad, teniendo interacción de una forma u otra con el negocio. Este sería entonces un actor del negocio abstracto.

A continuación se muestran los actores con sus descripciones:

Actores	Justificación
 <p>Estudiante</p>	Estudiante de la facultad interactúa con el horario una vez terminado, debido a que es el que lo consulta para ver sus turnos de clases diarios.
 <p>Vicedecano de Formación</p>	Vicedecano de Formación, es responsable de solicitar un cambio en el horario e interactúa con este proceso para evitar contradicciones en el horario evaluándolo para que se pueda visualizar, puede hacer modificaciones en el horario, además de que puede ser un profesor.
 <p>Profesor</p>	Informa sus afectaciones personales (ej. maestría, curso de superación) para que no coincidan con sus turnos de clases.

 <p>Usuario_Facultad</p>	<p>Es la generalización de los cuatro primeros actores. Interviene en todos los procesos. Se beneficia directamente de todo el resultado de cada proceso.</p>
---	---

1.4 Modelado del Negocio

Un proceso es un ordenamiento específico de actividades de trabajo a lo largo del tiempo y lugar, con un comienzo, fin, entradas y salidas claramente definidas.

Un proceso de negocio es una colección de actividades diseñadas para producir una salida específica para un cliente en particular o mercado. El modelado de los procesos de negocio es una parte esencial de cualquier proceso de desarrollo de software. Permite que el analista capture el esquema general y los procedimientos que gobiernan lo que hace el negocio.

Este modelo provee una vista general de donde el sistema de software propuesto se introducirá en la estructura de la organización y sus actividades cotidianas. Puede también proporcionar la justificación para la construcción del sistema capturando los procedimientos manuales y automáticos actuales que serán llevados a cabo dentro del sistema, y la relación costo beneficio. (29)

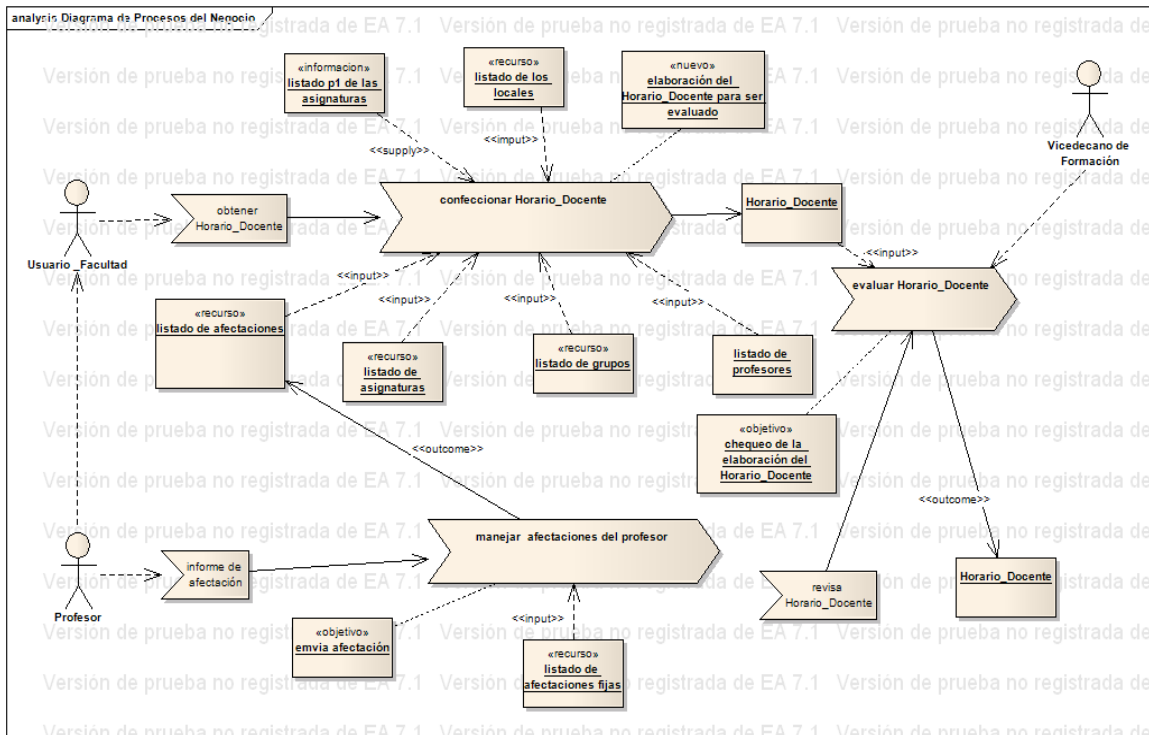
El modelo de procesos del negocio define los siguientes elementos:

- El objetivo o la razón del proceso
- Entradas específicas
- Salidas específicas
- Recursos consumidos
- Actividades que se realizan en cierto orden
- Eventos que dirigen el proceso.

1.4.1 Diagrama: Modelo de Proceso del Negocio.

Capítulo 2: Modelación del Negocio

En este diagrama se muestran todos los procesos del negocio, así como su objetivo y las metas que se propone alcanzar con el flujo de estos.



1.4.2 Descripción de los procesos del negocio.

1.4.2.1 - Proceso “Confeccionar Horario”.

Proceso	Confeccionar Horario
Actor	Usuario_Facultad
Resumen	<p>El proceso se inicia cuando se solicita la creación del horario docente por parte del Vicedecano de Formación a la planificadora.</p> <p>La Planificadora solicita el ingreso a tiempo de las afectaciones del profesor, siendo estas un dato importante para la confección, además de consultar los datos referentes a los locales destinados a la</p>

Capítulo 2: Modelación del Negocio

	<p>docencia de la facultad 7, los grupos docentes, los profesores y las asignaturas que estos imparten, utilizando toda la información procede a confeccionar el horario. Ella revisa el horario, intentando identificar posibles inconsistencias para mandárselo al Vicedecano de Formación quien lo evaluará y determinará si el horario debe ser reelaborado o si se encuentra correcto para publicarlo en la web.</p> <p>Precondiciones: La información que se debe tener son:</p> <ul style="list-style-type: none"> • los P1. • las afectaciones de los profesores. • el listado de los grupos. • los locales, listado de las asignaturas.
--	---

1.4.2.2 - Proceso “Evaluar Horario”.

Proceso	Evaluar Horario
Actor	Vicedecano de Formación
Resumen	<p>El proceso se inicia cuando el Vicedecano recibe el horario confeccionado por la Planificadora para evaluarlo, si el mismo cumple los requisitos necesarios de la planificación definida se publica, si se le encuentra errores, es enviado nuevamente a la planificadora para que haga las modificaciones pertinentes.</p> <p>Precondiciones: Se debe tener en cuenta que todos los requisitos estén bien definidos debido a</p>

Capitulo 2: Modelación del Negocio

	que este es el último proceso y necesita como precondition el horario terminado.
--	--

1.4.2.3 - Proceso “Manejo de la afectaciones del profesor”.

Proceso	Manejo de la afectaciones del profesor
Actor	Planificadora
Resumen	<p>El proceso inicia cuando la planificadora solicita a los profesores ingresar sus afectaciones y estos crean un listado en el sistema con esta información para que la planificadora las vea, los profesores deben poner todas sus afectaciones prestando atención aquellas que son fijas en un tiempo determinado (maestrías, postgrados), ella revisa el listado y verifica si las afectaciones fijas del profesor coinciden con el listado de afectaciones fijas que ella tiene, de tener error el profe debe ser notificado, aunque ella tiene el permiso para arreglarla, pero solo el vicedecano de formación es el autorizado a violar dicha afectación.</p> <p>Pos-condición:</p> <p>Tener el listado de afectaciones, este incluye aquellas afectaciones que son fijas para algunos días de la semana, dígame cursos de maestría, posgrados, doctorados, etc.</p>

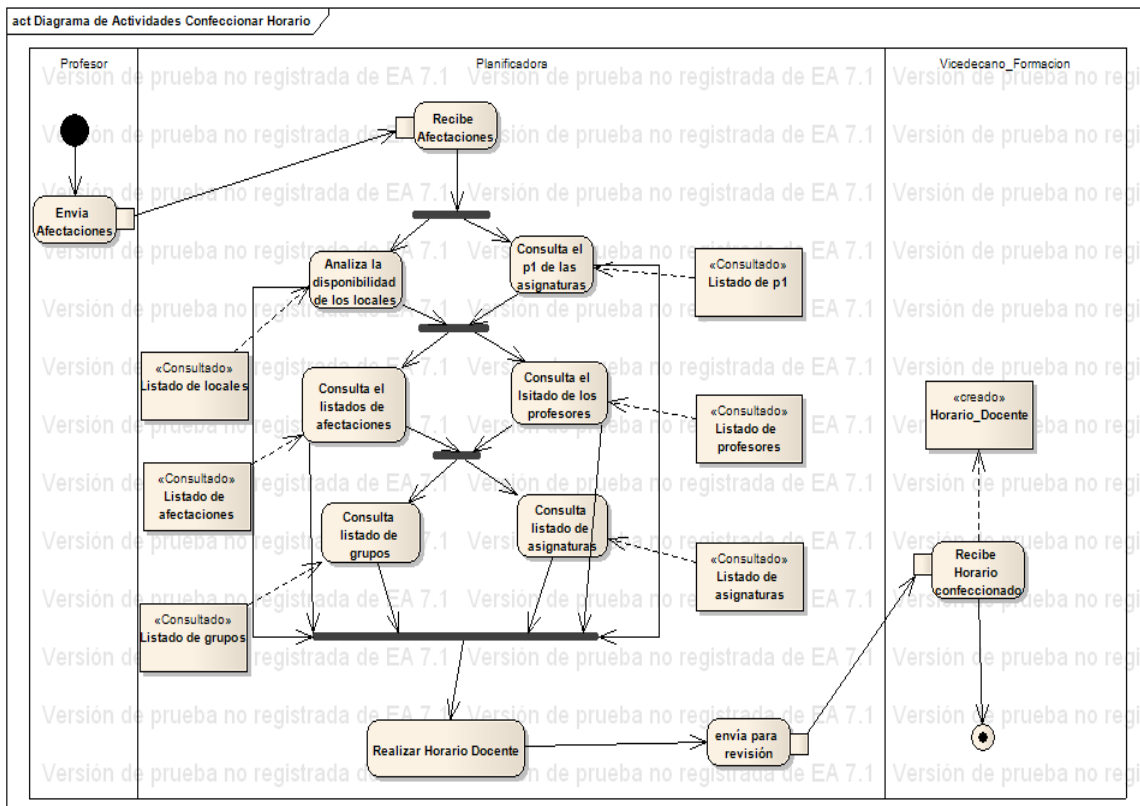
1.5 Diagrama de actividades de los procesos del negocio.

Capítulo 2: Modelación del Negocio

Un diagrama de actividad describe un proceso que explora el orden de las tareas o actividades que logran los objetivos del negocio, es un grafo que contiene los estados en que puede hallarse una actividad. Ayudan a describir en detalle, qué es lo que pasa dentro del negocio, está conformado por calles donde cada una de ellas representa una responsabilidad llevada a cabo por una parte de la organización. (30)

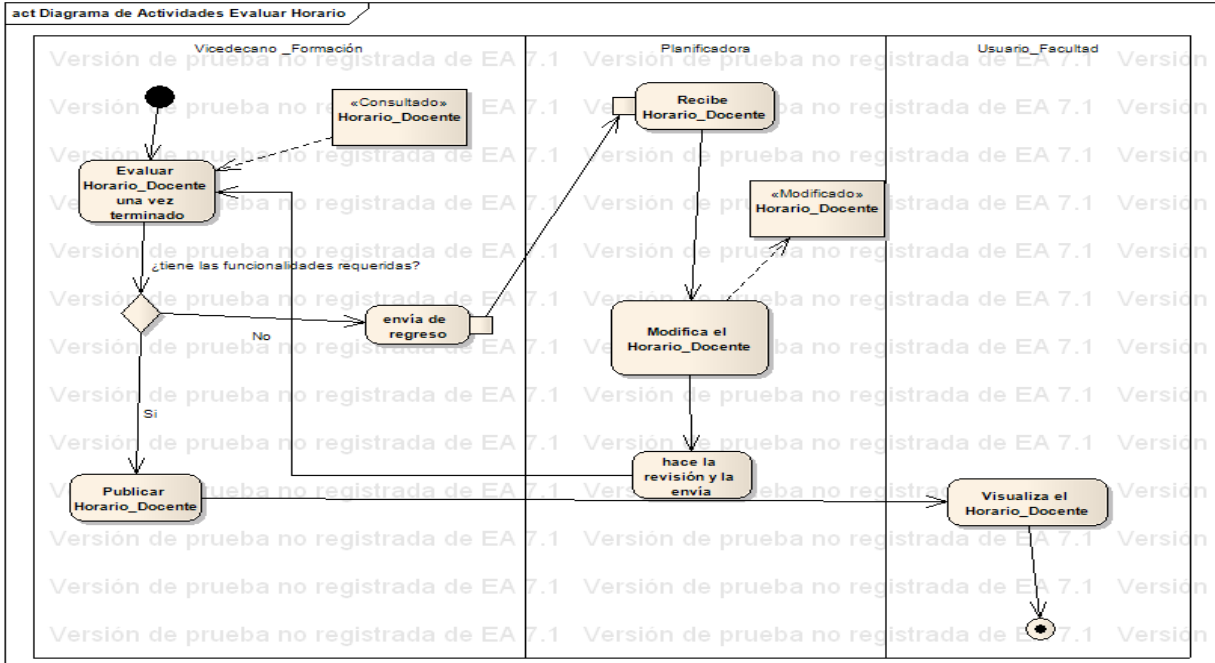
A continuación se muestran los diagramas de actividades de los procesos del negocio.

1.5.1 Diagrama de actividad del proceso “Confeccionar Horario”

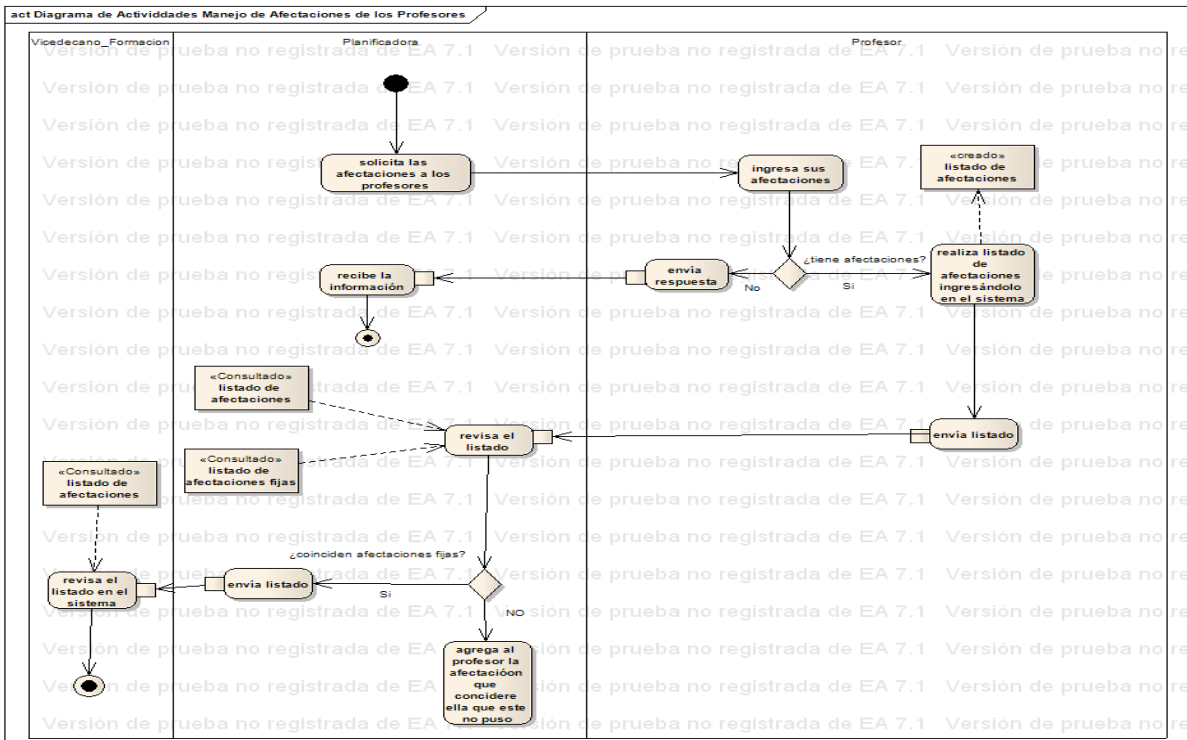


1.5.2 Diagrama de actividad del proceso “Evaluar Horario”

Capítulo 2: Modelación del Negocio



1.5.3 Diagrama de actividad del proceso “ Manejo de Afectaciones de los profesores ”



1.6 Requerimientos del sistema

En esta etapa se derivaron los requerimientos del sistema que va a soportar la aplicación, con lo cual se refuerza la idea de que sea el propio negocio lo que determine los requisitos. Esta etapa se refiere además a la disciplina análisis y diseño, donde a través de los artefactos más importantes de este flujo de trabajo, se modelan los principales casos de uso seleccionados para la iteración del producto propuesto a desarrollar.

1.6.1 Captura e identificación de los Requerimientos

Un requerimiento es una condición o capacidad necesaria para que un usuario resuelva un problema o alcance un objetivo. Una condición o capacidad que debe encontrarse o estar en un sistema o componente para satisfacer un contrato, norma, especificación u otro documento impuesto formalmente. El conjunto de todas las necesidades es el fundamento para el consiguiente desarrollo del sistema o componente. (31)

Los requerimientos por lo general se dividen en dos grupos: requerimientos funcionales (capacidades o condiciones que el sistema debe cumplir) y requerimientos no funcionales (son propiedades o cualidades que el producto debe tener).

1.6.1.1 - Definición de los requisitos funcionales

Los requisitos funcionales son aquellos que, desde el punto de vista de las necesidades del usuario, debe cumplir el sistema y que están fuertemente ligados a las opciones del programa. Para cumplir con los objetivos propuestos se prevé que el sistema tenga las siguientes funcionalidades:

RF 1 Gestionar las afectaciones de los profesores.

- 1.1 Adicionar afectación.
- 1.2 Modificar afectación.
- 1.4 Eliminar afectación.

RF 2 Gestionar locales.

- 2.1 Adicionar local.

2.2 Eliminar local.

2.3 Modificar local.

RF 3 Visualizar Horario por el personal de la Facultad 7.

RF 4 Listar las afectaciones de los profesores.

RF 5 Gestionar la configuración del sistema.

5.1 Crear tipo de usuario (editor, visualizador)

5.2 Modificar tipo de usuario.

5.3 Eliminar tipo de usuario.

RF 6 Gestionar profesores de la Facultad 7.

6.1 Adicionar profesor.

6.2 Eliminar profesor.

6.3 Modificar profesor.

RF 7 Gestionar asignatura.

7.1 Adicionar asignatura.

7.2 Eliminar asignatura.

7.3 Modificar asignatura.

RF 8 Gestionar grupo.

3.1 Adicionar grupo.

3.2 Eliminar grupo.

3.3 Modificar grupo.

RF 9 Autenticar.

RF 10 Buscar horario.

10.1 Por grupo.

10.2 Por semana.

10.3 Por semestre.

RF 11 Gestionar turno.

- 11.1 Adicionar turno.
- 11.2 Eliminar turno.
- 11.3 Modificar turno.

RF 12 Gestionar actividad.

- 12.1 Adicionar actividad
- 12.2 Eliminar actividad
- 12.3 Modificar actividad

RF 13 Gestionar semestre

- 12.1 Adicionar semestre
- 12.2 Eliminar semestre
- 12.3 Modificar semestre

1.6.1.2 - Definición de los requisitos no funcionales.

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. Normalmente están vinculados a requerimientos funcionales, es decir una vez se conozca lo que el sistema debe hacer podemos determinar cómo ha de comportarse, qué cualidades debe tener o cuán rápido o grande debe ser. (36) A continuación se muestran los requerimientos no funcionales:

- **Usabilidad.**

El sistema podrá ser usado por cualquier personal de la Facultad 7 que acceda a él, fácil de usar, con un glosario de términos informáticos que tiene el sistema para que sean entendibles por los clientes.

- **Rendimiento.**

- El sistema deberá ser capaz de gestionar la información referente a la planificación docente y dar respuesta a las solicitudes lo más rápido posible.
- Debe ser eficiente a la hora de gestionar las solicitudes logrando que sin mucha navegación por el sitio se obtenga los resultados deseados.

- Debe estar disponible las 24 horas del día.
- El sistema debe soportar la conexión simultánea de más de 500 usuarios.

- **Soporte.**
 - Se deben realizar pruebas al software para comprobar su funcionalidad.
 - El personal que trabaja con el módulo debe contar con el nivel técnico requerido mediante adiestramiento de servicio.
 - Se requiere que el producto reciba mantenimiento ante cualquier fallo que ocurra.
 - Se instalará el gestor de base de datos PostgreSQL 8.2 para soportar grandes volúmenes de datos y lograr que la velocidad de procesamiento sea elevada.
 - 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.

- **Portabilidad.**
 - El producto puede ser usado bajo el S.O. Windows o Linux.

- **Seguridad.**
 - El sistema se encargará de controlar los niveles de acceso y funcionalidad de usuarios, de identificar al usuario antes de que pueda realizar cualquier acción sobre el sistema. Se hacen validaciones de la información, los usuarios acceden de manera rápida y operativa al sistema sin que los requerimientos de seguridad se conviertan en un retardo para ellos.
 - Dentro de este requisito se puede mencionar: integridad, confidencialidad, disponibilidad y confiabilidad.

- ✓ **Integridad.**
 - La información manejada por el sistema que se desea desarrollar será objeto de cuidadosa protección contra la corrupción y estados inconsistentes teniendo una base de datos segura donde solo aquellos que tienen el permiso de acceso podrán hacer cambios en la información y publicar las misma.
 - Prevenir posibles fallos y recuperarse ante ellos.

✓ **Confiabilidad.**

- Capaz de restaurarse de las fallas rápidamente.
- La información manejada por el sistema está protegida de acceso no autorizado.

✓ **Disponibilidad**

- A los usuarios autorizados se les garantizará el acceso a la información y los mecanismos utilizados para lograr la seguridad no ocultarán o retrasará a los usuarios para obtener los datos deseados. Si el usuario no se autentica, no puede acceder a la funcionalidad del sistema.
- Disponible en todo momento debido a su importancia para planificar las clases.

✓ **Confidencialidad**

- La autenticación será la primera acción del usuario en el sistema y consistirá en suministrar un nombre de usuario único y una contraseña que debe ser de conocimiento exclusivo de la persona que se autentica.
- Si el usuario autenticado no se encuentra registrado se debe reportar un error de acceso.
- La información manejada por el sistema está protegida de acceso no autorizado.

• **Interfaz Externa**

- La herramienta propuesta debe tener una interfaz amigable y fácil de usar, de manera que no sea una dificultad para el usuario el uso de ella.
- Diseño sencillo y claro, con reconocimiento visual a través de elementos visibles que identifiquen cada una de sus funcionalidades.

• **Software**

- Para el funcionamiento del sistema en el servidor es necesario el S.O. Windows 98 o superior, Linux o Unix, en sus versiones de S.O. servidores.
- Para el funcionamiento del sistema en las terminales cliente es necesario el S.O. Windows 95 o superior, Linux o Unix.
- Los clientes tendrán acceso a la aplicación a través de cualquier navegador Web, recomendado Mozilla 1.5, Internet Explorer 5.0 o superior.



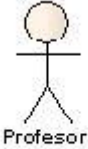
1.7 Actores del sistema

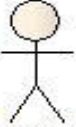

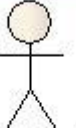
Los actores no son ninguna parte del sistema, ellos representan a cualquiera o algo que debe interactuar con el sistema. Un actor puede que:

- Sólo brinde información de entrada al sistema.
- Sólo reciba la información del sistema.
- Brinde y reciba información. (32)

Los actores que interactúan con el sistema se definen a continuación:

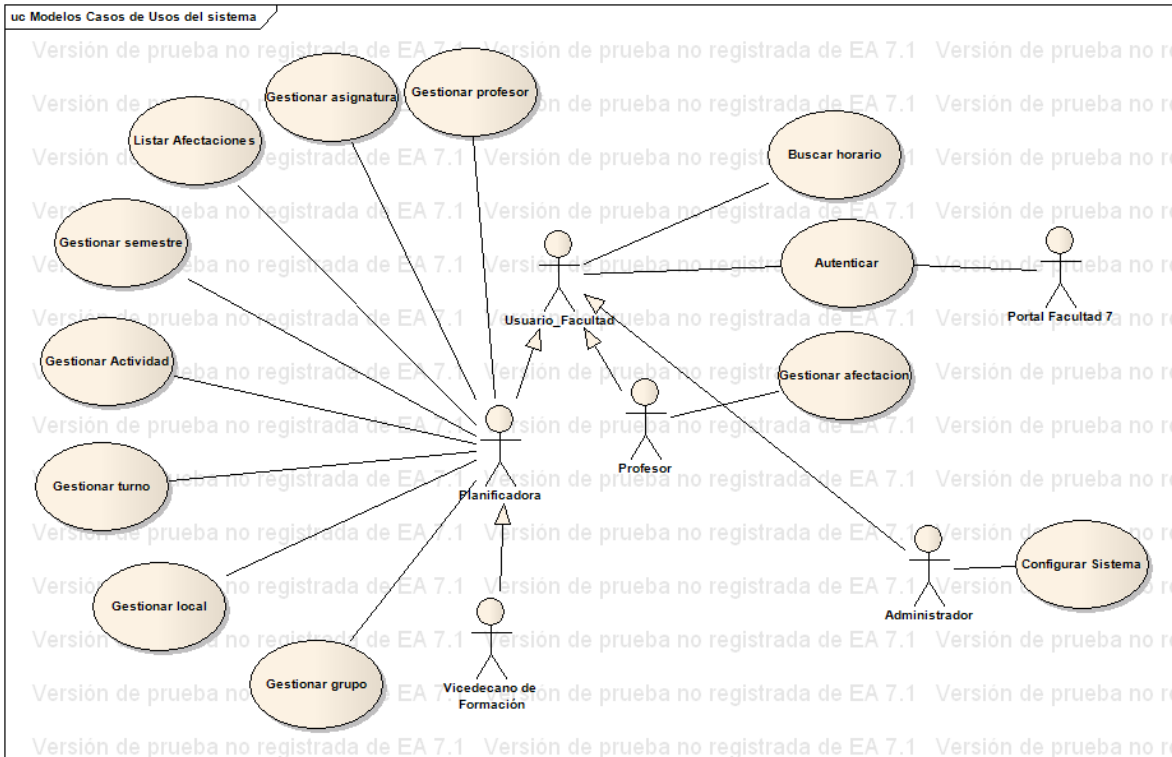
-Vicedecano de Formación – Profesores – Planificadora – Portal Facultad 7 – Administrador – Usuario_Facultad.

Actores	Justificación
 <p>Vicedecano de Formación</p>	<p>Vicedecano de Formación es quien da la orientación de hacer modificaciones en el horario e interactúa con este proceso para evitar contradicciones. Es el encargado de gestionar los datos referentes a los profesores de la Facultad 7 y las asignaturas que estos imparten. Además de tener las funciones del planificador debido a que está a cargo de la planificación y el control del proceso docente.</p>
 <p>Planificadora</p>	<p>Persona encargada de hacer la entrada de todos los datos como: los grupos de la facultad, los turnos de clases, las semanas del semestre, las afectaciones de los profesores, listado de locales destinados a la docencia, teniendo el acceso para adicionar, modificar y eliminar cualquiera de estos datos. Visualiza afectaciones profesores.</p>
 <p>Profesor</p>	<p>Responsable de ingresar sus afectaciones personales (Ej. Maestría, cursos de superación) para que no coincidan con sus turnos de clases.</p>

 Portal Facultad 7	Actor externo que permite registrar el personal que va a trabajar en la aplicación dígame profesores y estudiantes, planificador o vicedecano de formación.
 Administrador	Es el encargado de crear la estructura de cómo van a ser organizados los servicios del negocio en concreto otorgando los privilegios de los usuarios que interactúan con el sistema.
 Usuario_Facultad	Es una generalización de los cinco primeros actores. Interviene en todos los procesos.

1.7.1 Diagrama de Casos de Usos del Sistema a Informatizar.

El modelo de Casos de Uso presenta las funciones del sistema y los actores que hacen uso de ellas.



1.7.2 Descripción de los Casos de Uso.

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

1.7.2.1 - Descripción de Casos de Uso Gestionar Afectación.

Caso de Uso:	Gestionar Afectación
Actores:	Profesor
Resumen:	El CU se inicia cuando el profesor solicita incluir sus afectaciones a tener en cuenta a la hora de elaborar el horario; para esto el sistema le muestra un formulario donde ingresará sus afectaciones.
Precondiciones:	El Profesor debe estar previamente autenticado.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Negocio

Capítulo 2: Modelación del Negocio

<p>1. El profesor solicita la interfaz correspondiente a la gestión de las afectaciones para insertar, eliminar o modificar los datos de alguna afectación.</p>	<p>1.1 El sistema ejecuta alguna de las siguientes acciones:</p> <p>a) Para ingresar en el sistema una nueva afectación: ir a la sección “Adicionar”</p> <p>b) Para eliminar del sistema alguna de las afectaciones: ir al icono “Eliminar”</p> <p>c) Para modificar en el sistema una de las afectaciones: ir al icono “Editar”</p>
---	--

Flujos Alternos

Sección “Adicionar Afectación”

Acción del Actor	Respuesta del Negocio
<p>1. Solicita la interfaz adicionar.</p> <p>2. Adiciona todas a las afectaciones que puede tener.</p>	<p>1.1 Muestra la interfaz correspondiente, donde habrá una tabla para llenar los datos referentes a las afectaciones.</p> <p>2.1 Guarda las afectaciones y lo muestra en la lista de todas las afectaciones.</p>

Flujo Normal de Eventos

Sección “Eliminar Afectación”

Acción del Actor	Respuesta del sistema
<p>1. Selecciona la opción de eliminar.</p> <p>2. Selecciona la afectación que va a eliminar.</p>	<p>1.1 El sistema muestra una interfaz visual con el listado de todas las afectaciones y un icono de eliminar.</p> <p>2.1 El sistema elimina la afectación que está seleccionada con el icono eliminar mostrando un mensaje confirmando si desea eliminar.</p>

Flujo Normal de Eventos

Sección “Modificar Afectación”

Acción del actor	Respuesta del sistema

Capítulo 2: Modelación del Negocio

<ol style="list-style-type: none"> 1. Selecciona la opción de editar afectación. 2. Selecciona una afectación a editar. 3. Modifica los campos que desee y acepta la acción 	<ol style="list-style-type: none"> 1.1 El sistema muestra una interfaz visual con el listado de todas las afectaciones registradas. 2.1 El sistema muestra una interfaz visual con los campos para modificar. 3.1 El sistema verifica todos los datos. 3.2 Si los datos están correctos el sistema los modifica.
--	--

1.7.2.2 - Descripción de Casos de Uso Gestionar Asignatura.

Caso de Uso:	Gestionar Asignatura.
Actores:	Vice-Formación
Resumen:	El Vice-Formación tiene la posibilidad de insertar, modificar o eliminar cada una de las asignaturas que posteriormente formarán parte del horario.
Precondiciones:	El Vice-Formación debe estar previamente autenticado.
Referencias	

Flujo Normal de Eventos

Acción del Actor	Respuesta del Negocio
<ol style="list-style-type: none"> 1. El Vice-Formación solicita la interfaz correspondiente a la gestión de las asignaturas para insertar, eliminar o modificar los datos de alguna asignatura. 	<ol style="list-style-type: none"> 1.1 El sistema ejecuta alguna de las siguientes acciones: <ol style="list-style-type: none"> a) Para ingresar en el sistema una nueva asignatura: ir a la sección "Adicionar" b) Para eliminar del sistema alguna de las asignaturas: ir al icono "Eliminar" c) Para modificar en el sistema una de las asignaturas: ir al icono "Editar"

Flujos Alternos

Sección "Insertar Asignatura"	
Acción del Actor	Respuesta del Negocio

Capítulo 2: Modelación del Negocio

<p>1. Solicita la interfaz adicionar.</p> <p>2. Adiciona la asignatura deseada.</p>	<p>1.1 Muestra la interfaz correspondiente, donde habrá una tabla para llenar los datos referentes a las afectaciones.</p> <p>2.1 Guarda la asignatura y lo muestra en la lista de todas las asignaturas.</p>
Flujo Alternativo	
Acción del Actor	Respuesta del sistema.
	2.1 Si la asignatura ya se encuentra registrada el sistema muestra un mensaje de error
Flujo Normal de Eventos	
Sección “Eliminar Asignatura”	
Acción del Actor	Respuesta del sistema
<p>1. Selecciona la opción de eliminar.</p> <p>2. Selecciona la asignatura que va a eliminar.</p>	<p>1.1 El sistema muestra una interfaz visual con el listado de todas las asignaturas y un icono de eliminar.</p> <p>2.1 El sistema elimina la asignatura que está seleccionada con el icono eliminar mostrando un mensaje confirmando si desea eliminar.</p>
Flujo Alternativo	
Acción del actor	Respuesta del sistema
	2.1 Si no hay ninguna asignatura seleccionada el sistema muestra un mensaje de error.
Flujo Normal de Eventos	
Sección “Modificar Asignatura”	

Capítulo 2: Modelación del Negocio

Acción del actor	Respuesta del sistema
<ol style="list-style-type: none"> 1. Selecciona la opción de editar asignatura. 2. Selecciona una asignatura a editar. 3. Modifica los campos que desee y acepta la acción 	<ol style="list-style-type: none"> 1.1 El sistema muestra una interfaz visual con el listado de todas las asignaturas registradas. 2.1 El sistema muestra una interfaz visual con los campos para modificar. 3.1 El sistema verifica todos los datos. 3.2 Si los datos están correctos el sistema los modifica.
Flujo Alternativo	
Acción de actor	Respuesta del sistema
	<ol style="list-style-type: none"> 3.2 Si algún dato no está correcto el sistema muestra un mensaje de error y cuáles son los campos incorrectos.

1.7.2.3 - Descripción de Casos de Uso Gestionar Grupos.

Caso de Uso:	Gestionar Grupos
Actores:	Planificadora
Resumen:	El sistema permite que la planificadora realice algunas operaciones, estas son insertar, modificar o eliminar los grupos que pertenecen a la Facultad 7.
Precondiciones:	La Planificadora debe estar previamente autenticada.
Referencias	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Negocio

Capítulo 2: Modelación del Negocio

<p>1. La Planificadora solicita la interfaz correspondiente para la gestión de los grupos de la facultad para insertar, eliminar o modificar los mismos.</p>	<p>1.1 El sistema ejecuta alguna de las siguientes acciones:</p> <p>a) Para ingresar en el sistema un nuevo grupo: ir a la sección “Adicionar”</p> <p>b) Para eliminar del sistema algún grupo: ir al icono “Eliminar”</p> <p>c) Para modificar en el sistema algún grupo: ir al icono “Editar”</p>
Flujos Alternos	
Sección “Insertar Grupos”	
Acción del Actor	Respuesta del Negocio
<p>1. Solicita la interfaz adicionar.</p> <p>2. Adiciona el grupo deseado.</p>	<p>1.1 Muestra la interfaz correspondiente, donde habrá una tabla para llenar los datos referentes a los grupos.</p> <p>2.1 Guarda el grupo y lo muestra en la lista de todos los grupos.</p>
Flujo Alterno	
Acción del Actor	Respuesta del sistema
	<p>2.2 Si los datos se encuentran registrados, el sistema muestra un mensaje indicando que ya esa operación se realizó.</p>
Flujo Normal de Eventos	
Sección “Eliminar Grupos”	
Acción del Actor	Respuesta del sistema
<p>1. Selecciona la opción de eliminar.</p> <p>2. Selecciona el grupo que va a eliminar.</p>	<p>1.1 El sistema muestra una interfaz visual con el listado de todos los grupos y un icono de eliminar.</p> <p>2.1 El sistema elimina el grupo que está seleccionado con el icono eliminar mostrando un mensaje confirmando si</p>

	desea eliminar.
Flujo Alterno	
Acción del actor	Respuesta del sistema
	1.2 Si no hay ningún grupo seleccionado el sistema muestra un mensaje de error.
Flujo Normal de Eventos	
Sección “Modificar Grupos”	
Acción del actor	Respuesta del sistema
1. Selecciona la opción de editar grupo.	1.1 El sistema muestra una interfaz visual con el listado de todos los grupos registrados.
2. Selecciona un grupo a editar.	2.1 El sistema muestra una interfaz visual con los campos para modificar.
3. Modifica los campos que desee y acepta la acción	3.1 El sistema verifica todos los datos. 3.2 Si los datos están correctos el sistema los modifica.
Flujo Alterno	
Acción de actor	Respuesta del sistema
	3.2 Si algún dato no está correcto el sistema muestra un mensaje de error.

1.7.2.4 - Descripción de Casos de Uso Gestionar Locales.

Caso de Uso:	Gestionar Locales.
Actores:	Planificadora
Resumen:	El sistema permite que la planificadora realice algunas operaciones, estas son insertar, modificar o eliminar los locales que le fueron asignados a la Facultad 7.

Precondiciones:	La Planificadora debe estar previamente autenticada.	
Referencias		
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Negocio	
1 La planificadora solicita la interfaz correspondiente a la gestión de locales de la facultad para insertar, eliminar o modificar los mismos.	1.1 El sistema ejecuta alguna de las siguientes acciones: a) Para ingresar en el sistema un nuevo grupo: ir a la sección “Adicionar” b) Para eliminar del sistema algún grupo: ir al icono “Eliminar” c) Para modificar en el sistema un grupo: ir al icono “Editar”	
Flujos Alternos		
Sección “Insertar Local”		
Acción del Actor	Respuesta del Negocio	
1. Solicita la interfaz adicionar. 2. Adiciona el local deseado.	1.1 Muestra la interfaz correspondiente, donde habrá una tabla para llenar los datos referentes a los locales. 2.1 Guarda el local adicionado y lo muestra en la lista de todos los locales.	
Flujo Alterno		
Acción del Actor	Respuesta del sistema	
	2.2 Si dicho local se encuentra registrado, el sistema muestra un mensaje indicando que ya esa operación se realizó.	
Flujo Normal de Eventos		
Sección “Eliminar Local”		
Acción del Actor	Respuesta del sistema	
1. Selecciona la opción de eliminar.	1.1 El sistema muestra una interfaz visual con el listado de todos los locales y un	

Capítulo 2: Modelación del Negocio

2. Selecciona el local que va a eliminar.	<p>icono de eliminar.</p> <p>2.1 El sistema elimina el local que está seleccionado con el icono eliminar mostrando un mensaje confirmando si desea eliminar.</p>
Flujo Alterno	
Acción del actor	Respuesta del sistema
	2.1 Si no hay ningún local seleccionado el sistema muestra un mensaje de error.
Flujo Normal de Eventos	
Sección “Modificar Local”	
Acción del actor	Respuesta del sistema
<p>1. Selecciona la opción de editar local.</p> <p>2. Selecciona un local a editar.</p> <p>3. Modifica los campos que desee y acepta la acción</p>	<p>1.1 El sistema muestra una interfaz visual con el listado de todos los locales registrados.</p> <p>2.1 El sistema muestra una interfaz visual con los campos para modificar.</p> <p>3.1 El sistema verifica todos los datos.</p> <p>3.2 Si los datos están correctos el sistema los modifica.</p>
Flujo Alterno	
Acción de actor	Respuesta del sistema
	3.2 Si algún dato no se encuentra correcto el sistema muestra un mensaje de error.

1.7.2.5 - Descripción de Casos de Uso Gestionar Profesores.

Caso de Uso:	Gestionar Profesores.
Actores:	Vicedecano de Formación

Capítulo 2: Modelación del Negocio

Resumen:	El Vicedecano de Formación tiene la posibilidad de insertar, modificar o eliminar cada una de los profesores que pertenecen a la facultad 7.
Precondiciones:	El Vicedecano de Formación debe estar previamente autenticado.
Referencias	

Flujo Normal de Eventos

Acción del Actor	Respuesta del Negocio
1. El Vicedecano de Formación solicita la interfaz correspondiente a la gestión de profesores para insertar, eliminar o modificar los datos sobre alguno de ellos.	<p>El sistema ejecuta alguna de las siguientes acciones:</p> <p>a) Para ingresar en el sistema un nuevo profesor: ir a la sección “Adicionar”</p> <p>b) Para eliminar del sistema algún profesor: ir al icono “Eliminar”</p> <p>c) Para modificar en el sistema un profesor: ir al icono “Editar”</p>

Flujos Alternos

Sección “Insertar Profesor”

Acción del Actor	Respuesta del Negocio
1. Solicita la interfaz adicionar.	1.1 Muestra la interfaz correspondiente, donde habrá una tabla para llenar los datos referentes a los profesores.
2. Adiciona el profesor deseado.	2.1 Guarda el profesor y lo muestra en la lista de todos los profesores.

Flujo Alternativo

Acción del Actor	Respuesta del sistema
	2.2 Si hay datos incorrectos el sistema muestra un mensaje de error con un asterisco rojo.

Flujo Normal de Eventos

Sección “Eliminar Profesor”

Capítulo 2: Modelación del Negocio

Acción del Actor	Respuesta del sistema
<ol style="list-style-type: none"> 1. Selecciona la opción de eliminar. 2. Selecciona el profesor que va a eliminar. 	<p>1.1 El sistema muestra una interfaz visual con el listado de todos los profesores y un icono de eliminar.</p> <p>2.1 El sistema elimina el profesor que está seleccionado con el icono eliminar mostrando un mensaje confirmando si desea eliminar.</p>
Flujo Alterno	
Acción del actor	Respuesta del sistema
	2.1 Si no hay ningún profesor seleccionado el sistema muestra un mensaje de error.
Flujo Normal de Eventos	
Sección “Modificar Profesor”	
Acción del actor	Respuesta del sistema
<ol style="list-style-type: none"> 1. Selecciona la opción de editar profesor. 2. Selecciona un profesor a editar. 3. Modifica los campos que desee y acepta la acción 	<p>1.1 El sistema muestra una interfaz visual con el listado de todos los profesores registrados.</p> <p>2.1 El sistema muestra una interfaz visual con los campos para modificar.</p> <p>3.1 El sistema verifica todos los datos.</p> <p>3.2 Si los datos están correctos el sistema los modifica.</p>
Flujo Alterno	
Acción de actor	Respuesta del sistema
	3.2 Si algún dato no está correcto el sistema muestra un mensaje de error.

1.7.2.6 - Descripción de Casos de Uso Buscar Horario.

Caso de Uso:	Buscar Horario.	
Actores:	Usuario_Facultad.	
Resumen:	El CU se inicia cuando cualquier persona que accede al sistema solicita ver el horario y este lo muestra según la opción que seleccione; esta puede ver horario por grupo, por año o semestre.	
Precondiciones:	El horario docente debe haber sido publicado previamente.	
Referencias		
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Negocio	
1 El usuario solicita la interfaz correspondiente para consultar el horario docente de la facultad.	1.1 El sistema ejecuta alguna de las siguientes acciones: a) Para consultar todos los horarios: ir a la sección "Buscar" b) Para consultar el horario de un grupo: ir al icono mostrar turnos correspondiente a ese grupo.	
Flujos Normal de Eventos		
Sección "Buscar Horario por Grupo"		
Acción del Actor	Respuesta del Negocio	
1 El usuario selecciona el icono mostrar turnos correspondiente al grupo deseado.	1.1 El sistema muestra una interfaz visual con el horario docente correspondiente a ese grupo.	
Flujo Normal de Eventos		
Sección "Consultar Horario Por Año"		
Acción del Actor	Respuesta del sistema	
1. Se solicita observar el horario de un año determinado.	1.1 Muestra un listado con los años.	
2. El usuario selecciona el año del cual desea consultar el horario.	2.1 Muestra el horario requerido.	

Flujo Normal de Eventos	
Sección “Consultar Horario por Semestre”	
Acción del Actor	Respuesta del sistema
1. El usuario selecciona el icono correspondiente al semestre deseado de un grupo determinado.	1.1 El sistema muestra una interfaz visual con el horario docente correspondiente a ese grupo.

1.7.2.6 - Descripción de Casos de Uso Gestionar Turnos.

Caso de Uso:	Gestionar Turnos.
Actores:	Vicedecano de Formación
Resumen:	El Vicedecano de Formación tiene la posibilidad de insertar, modificar o eliminar cada una de los turnos que pertenecen a la facultad 7.
Precondiciones:	El Vicedecano de Formación debe estar previamente autenticado.
Referencias	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Negocio
1. El Vicedecano de Formación solicita la interfaz correspondiente a la gestión de turnos para insertar, eliminar o modificar los datos sobre alguno de ello.	El sistema ejecuta alguna de las siguientes acciones: a) Para ingresar en el sistema un nuevo turno: ir a la sección “Adicionar” b) Para eliminar del sistema algún turno: ir al icono “Eliminar” c) Para modificar en el sistema un turno: ir al icono “Editar”
Flujos Alternos	
Sección “Insertar Turno”	

Capitulo 2: Modelación del Negocio

Acción del Actor	Respuesta del Negocio
1. Solicita la interfaz adicionar. 2. Adiciona el turno deseado.	1.1 Muestra la interfaz correspondiente, donde habrá una tabla para llenar los datos referentes a los turnos. 2.1 Guarda el turno y lo muestra en la lista de todos los turnos.
Flujo Alternativo	
Acción del Actor	Respuesta del sistema
	2.2 Si hay datos incorrectos el sistema muestra un mensaje de error con un asterisco rojo.
Flujo Normal de Eventos	
Sección "Eliminar Turno"	
Acción del Actor	Respuesta del sistema
1. Selecciona la opción de eliminar. 2. Selecciona el turno que va a eliminar.	1.1 El sistema muestra una interfaz visual con el listado de todos los turnos y un icono de eliminar. 2.1 El sistema elimina el turno que está seleccionado con el icono eliminar mostrando un mensaje confirmando si desea eliminar.
Flujo Alternativo	
Acción del actor	Respuesta del sistema
	2.1 Si no hay ningún turno seleccionado el sistema muestra un mensaje de error.
Flujo Normal de Eventos	
Sección "Modificar Turno"	
Acción del actor	Respuesta del sistema

Capítulo 2: Modelación del Negocio

1. Selecciona la opción de editar turno.	1.1 El sistema muestra una interfaz visual con el listado de todos los turnos registrados.
2. Selecciona un turno a editar.	2.1 El sistema muestra una interfaz visual con los campos para modificar.
3. Modifica los campos que desee y acepta la acción	3.1 El sistema verifica todos los datos. 3.2 Si los datos están correctos el sistema los modifica.
Flujo Alternativo	
Acción de actor	Respuesta del sistema
	3.2 Si algún dato no está correcto el sistema muestra un mensaje de error.

Conclusiones.

En el presente capítulo se inició el desarrollo de la solución propuesta en este trabajo. Se obtuvieron, mediante el análisis profundo de los procesos del negocio, todas las funciones que el sistema debe tener. Los mismos se representaron a través un Diagrama de Casos de Uso, y se describieron todas sus acciones con sus respectivos actores para finalmente comenzar a construir el sistema siguiendo todos los requerimientos establecidos en el capítulo.

CAPÍTULO 3. ANÁLISIS Y DISEÑO DEL SISTEMA.

Introducción.

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

Este capítulo tiene el objetivo de plantear la concepción general del diseño del sistema propuesto y cómo se implementa este. Se presentan los diagramas de clases de diseño Web que detallan la interacción de las distintas páginas; se estructura la información que se desea que persista a través del diseño de la base de datos; se describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. Son también descritos los estándares de diseño y programación.

1.2 Principios de diseño.

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

Sirve de abstracción de la implementación y es utilizada como entrada fundamental de las actividades de implementación. (33)

1.2.1 Diagrama de clases del diseño

Un diagrama de clases presenta las clases del sistema con sus relaciones estructurales y de herencia, todo el código que irá creando las páginas, así como el contenido dinámico de estas una vez que estén en el navegador del cliente. En el caso de las aplicaciones Web, el diagrama de clases representa las colaboraciones que ocurren entre las páginas, donde cada página lógica puede ser representada como una clase, es muy importante pues estos son los artefactos que se necesitan modelar para que el desarrollador los implemente y obtener así el producto final con la calidad requerida.

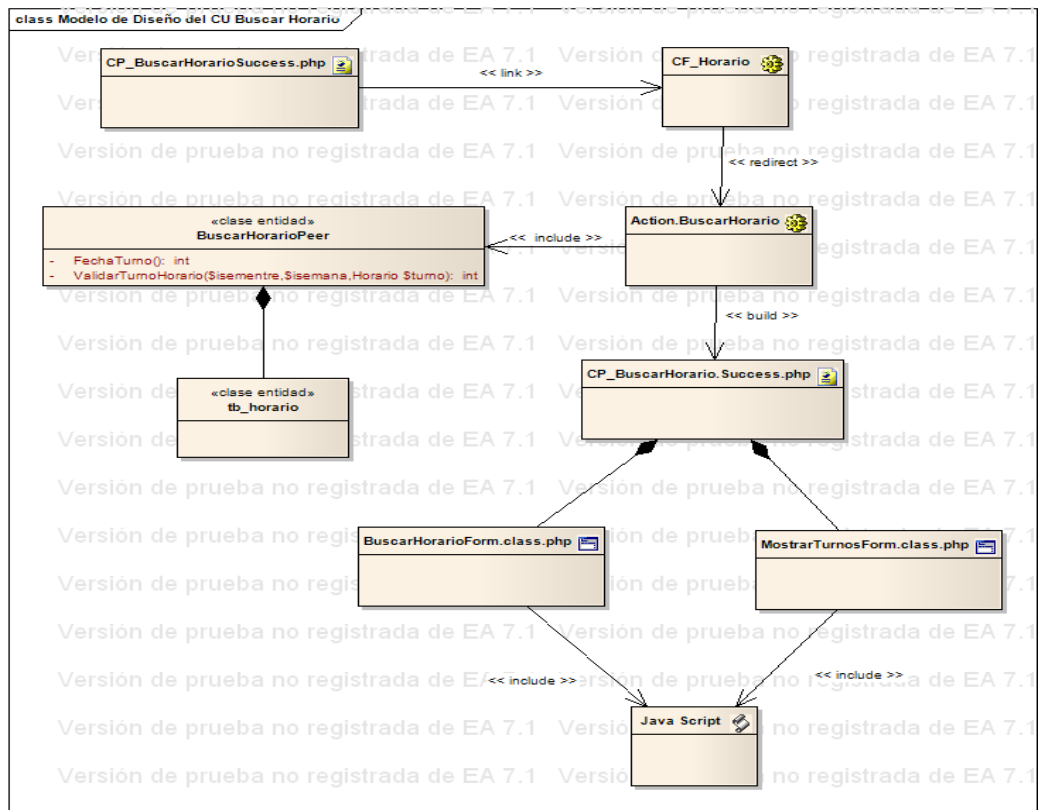
Al tratar de utilizar el diagrama de clases tradicional para modelar aplicaciones Web surgen varios problemas, por lo cual los especialistas del Rational plantearon la creación

Capítulo 3: Análisis y Diseño del Sistema

de una extensión al modelo de análisis y diseño que permitiera representar el nivel de abstracción adecuado y la relación con los restantes artefactos de UML. (34)

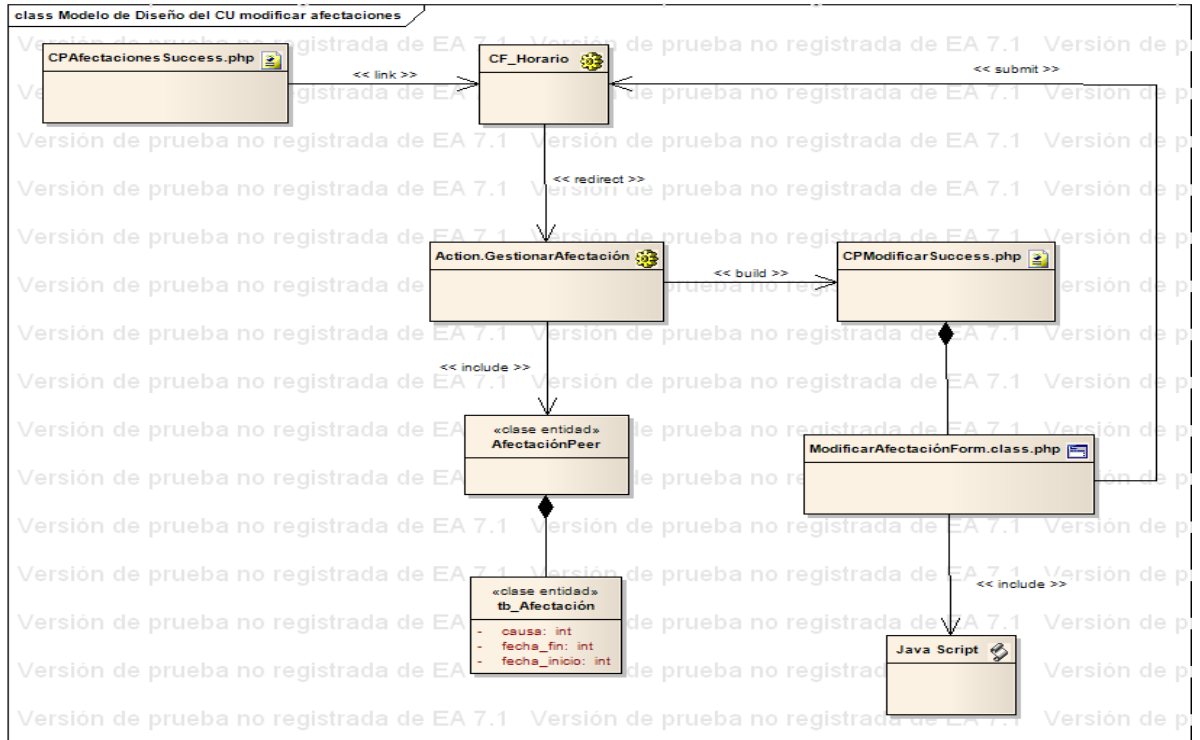
El diagrama de clases Web, fue definido, a partir de los diferentes casos de uso del sistema y empleando las extensiones de UML para Web, a continuación se muestran los diagramas de clases para los distintos Casos de Uso.

1.2.1.1 Modelo de clases del diseño del CU Buscar Horario.

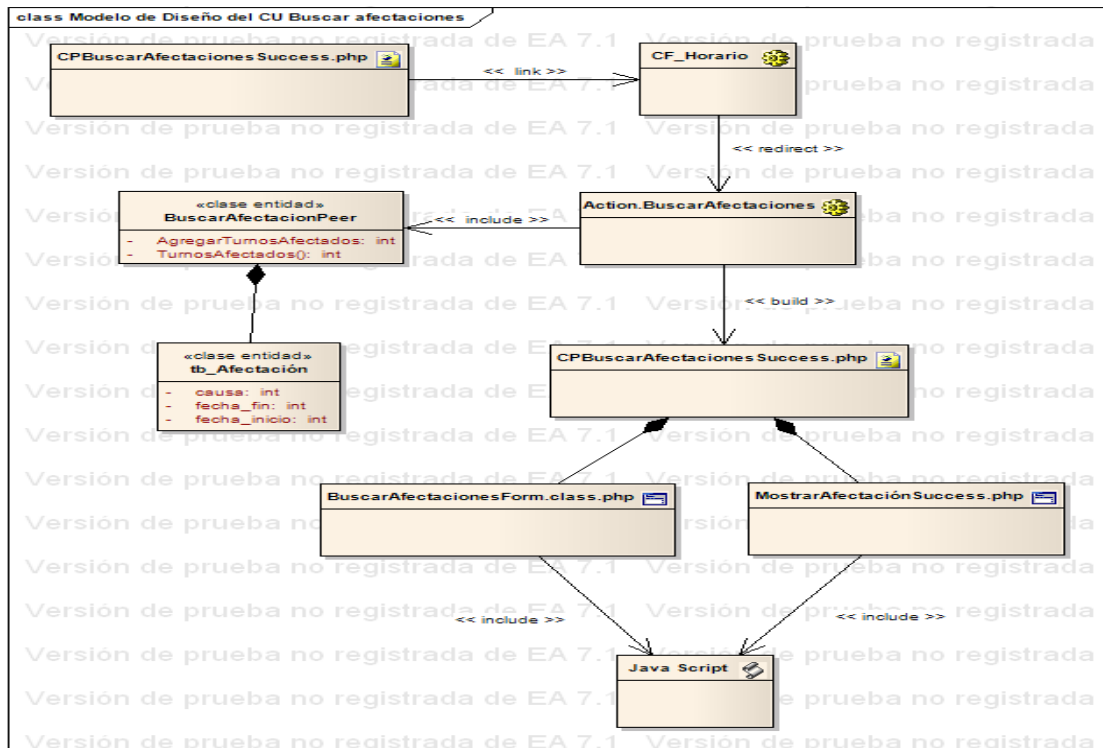


1.2.1.2 Modelo de clases del diseño del CU Modificar afectaciones.

Capítulo 3: Análisis y Diseño del Sistema



1.2.1.3 Modelo de clases del diseño del CU Buscar afectaciones



1.3 Descripción de las clases del Diseño.

1.3.1 Descripción de las clases Interfaz para Gestionar Horario:

Nombre: Buscar Horario	
Tipo de clase: interfaz	
Atributo	Tipo
grupo	<<HTML Select>>
semana	<<HTML Input>>
semestre	<<HTML Select>>
año	<<HTML Select>>
Para cada responsabilidad:	
Nombre:	Buscar horario.
Descripción:	Esta clase muestra una interfaz para Buscar los Horarios de los estudiantes.

Nombre: Modificar Horario	
Tipo de clase: interfaz	
Atributo	Tipo
Día	<<HTML Select>>
local	<<HTML Select>>
Tipo de actividad	<<HTML Select>>
Turno	<<HTML Select>>
Asignatura	<<HTML Select>>
profesor	<<HTML Input>>
Para cada responsabilidad:	
Nombre:	Modificar horario.
Descripción:	Esta clase muestra una interfaz para Modificar los Horarios de los estudiantes.

Nombre: Adicionar Horario	
Tipo de clase: interfaz	
Atributo	Tipo
local	<<HTML Select>>
asignatura	<<HTML Select>>
Profesor	<<HTML Select>>
Tipo de actividad	<<HTML Select>>
turno	<<HTML Select>>
día	<<HTML Select>>
Para cada responsabilidad:	
Nombre:	Adicionar horario.
Descripción:	Esta clase muestra una interfaz para Adicionar los Horarios de los estudiantes.

1.3.2 Descripción de las clases Interfaz para Gestionar Afectaciones:

Nombre: Buscar Afectaciones	
Tipo de clase: interfaz	
Atributo	Tipo
profesor	<<HTML Select>>
fecha	<<HTML Select>>
causa	<<HTML Input>>
Para cada responsabilidad:	
Nombre:	Buscar afectaciones
Descripción:	Esta clase muestra una interfaz para Buscar las afectaciones de los profesores.

Nombre: Nueva Afectación	
Tipo de clase: interfaz	
Atributo	Tipo
profesor	<<HTML Select>>
fecha inicio	<<HTML Select>>
Turnos afectados	<<HTML Select>>
Fecha fin	<<HTML Select>>
causa	<<HTML Input>>
Para cada responsabilidad:	
Nombre:	nueva afectación
Descripción:	Esta clase muestra una interfaz para adicionar una nueva afectación

Nombre: Modificar	
Tipo de clase: interfaz	
Atributo	Tipo
profesor	<<HTML Select>>
fecha inicio	<<HTML Select>>
Turnos afectados	<<HTML Select>>
Fecha fin	<<HTML Select>>
causa	<<HTML Input>>
Para cada responsabilidad:	
Nombre:	modificar
Descripción:	Esta clase muestra una interfaz para modificar una afectación

1.3.3 Descripción de las clases Interfaz para Gestionar Local:

Nombre: Modificar Local	
Tipo de clase: interfaz	
Atributo	Tipo
nombre	<<HTML Input>>
capacidad	<<HTML Input>>
tipo	<<HTML Select>>
Para cada responsabilidad:	
Nombre:	Modificar Local
Descripción:	Esta clase muestra una interfaz para modificar un local.

Nombre: Nuevo Local	
Tipo de clase: interfaz	
Atributo	Tipo
nombre	<<HTML Input>>
capacidad	<<HTML Input>>
tipo	<<HTML Select>>
Para cada responsabilidad:	
Nombre:	Nuevo Local
Descripción:	Esta clase muestra una interfaz para adicionar un local.

1.3.4 Descripción de las clases Interfaz para Gestionar Turno:

Nombre: Modificar Turno	
Tipo de clase: interfaz	
Atributo	Tipo
número	<<HTML Input>>
Hora inicio	<<HTML Select>>
Hora fin	<<HTML Select>>
Para cada responsabilidad:	
Nombre:	Modificar Turno
Descripción:	Esta clase muestra una interfaz para modificar un turno.

Nombre: Adicionar Turno	
Tipo de clase: interfaz	
Atributo	Tipo
número	<<HTML Input>>
Hora inicio	<<HTML Select>>

Hora fin	<<HTML Select>>
Para cada responsabilidad:	
Nombre:	Adicionar Turno
Descripción:	Esta clase muestra una interfaz para adicionar un turno.

1.3.5 Descripción de las clases Interfaz para Gestionar Asignatura:

Nombre: Adicionar Asignatura	
Tipo de clase: interfaz	
Atributo	Tipo
Nombre	<<HTML Input>>
Año	<<HTML Select>>
disciplina	<<HTML Select>>
semestre	<<HTML Select>>
Para cada responsabilidad:	
Nombre:	Adicionar Asignatura
Descripción:	Esta clase muestra una interfaz para adicionar una asignatura.

Nombre: Modificar Asignatura	
Tipo de clase: interfaz	
Atributo	Tipo
Nombre	<<HTML Input>>
Año	<<HTML Select>>
disciplina	<<HTML Select>>
semestre	<<HTML Select>>
Para cada responsabilidad:	
Nombre:	Modificar Asignatura
Descripción:	Esta clase muestra una interfaz para modificar una asignatura.

1.3.6 Descripción de las clases Interfaz para Gestionar Profesores:

Nombre: Modificar Profesores	
Tipo de clase: interfaz	
Atributo	Tipo
No Identidad	<<HTML Input>>
Nombre	<<HTML Input>>
Primer apellido	<<HTML Input>>

Segundo apellido	<<HTML Input>>
Asignatura que imparte	<<HTML Select>>
Grupos a que imparte	<<HTML Select>>
Usuario	<<HTML Input>>
Cargo	<<HTML Select>>
tipo	<<HTML Select>>
Para cada responsabilidad:	
Nombre:	Modificar Profesores
Descripción:	Esta clase muestra una interfaz para modificar un profesor.

Nombre: Adicionar Profesores	
Tipo de clase: interfaz	
Atributo	Tipo
No Identidad	<<HTML Input>>
Nombre	<<HTML Input>>
Primer apellido	<<HTML Input>>
Segundo apellido	<<HTML Input>>
Asignatura que imparte	<<HTML Select>>
Grupos a que imparte	<<HTML Select>>
Usuario	<<HTML Input>>
Cargo	<<HTML Select>>
tipo	<<HTML Select>>
Para cada responsabilidad:	
Nombre:	Adicionar Profesores
Descripción:	Esta clase muestra una interfaz para adicionar un profesor.

1.3.7 Descripción de las clases Interfaz para Gestionar Semestre:

Nombre: Adicionar Semestre	
Tipo de clase: interfaz	
Atributo	Tipo
Fecha inicio	<<HTML Input>>
Fecha fin	<<HTML Input>>
número	<<HTML Select>>
Para cada responsabilidad:	
Nombre:	Adicionar Semestre
Descripción:	Esta clase muestra una interfaz para adicionar un semestre.

Nombre: Modificar Semestre	
Tipo de clase: interfaz	
Atributo	Tipo
Fecha inicio	<<HTML Input>>

Fecha fin	<<HTML Input>>
número	<<HTML Select>>
Para cada responsabilidad:	
Nombre:	Modificar Semestre
Descripción:	Esta clase muestra una interfaz para modificar un semestre.

1.3.8 Descripción de las clases Interfaz para Gestionar Tipo de Actividad:

Nombre: Modificar Tipo de Actividad	
Tipo de clase: interfaz	
Atributo	Tipo
nombre	<<HTML Input>>
Para cada responsabilidad:	
Nombre:	Modificar Tipo de Actividad
Descripción:	Esta clase muestra una interfaz para modificar una actividad.

Nombre: Adicionar Tipo de Actividad	
Tipo de clase: interfaz	
Atributo	Tipo
nombre	<<HTML Input>>
Para cada responsabilidad:	
Nombre:	Adicionar Tipo de Actividad
Descripción:	Esta clase muestra una interfaz para adicionar una actividad.

1.3.9 Descripción de las clases controladoras de Gestión.

Nombre: Afectaciones	
Tipo de clase: controladora	
Para cada responsabilidad:	
Nombre:	Insertar_Afectación() Buscar_Afectación_Asignatura_Grupo(Id_Asignatura,Id_Grupo) Eliminar_Afectación()
Descripción:	Esta clase permite gestionar todo lo que tiene que ver con el módulo de afectaciones.

Nombre: Asignaturas	
Tipo de clase: controladora	
Para cada responsabilidad:	
Nombre:	Existe_Asignatura() Insertar_Asignatura() Buscar_Asignatura()
Descripción:	Esta clase permite gestionar todo lo que tiene que ver con el módulo de asignaturas.

Nombre: Grupo Horario	
Tipo de clase: controladora	
Para cada responsabilidad:	
Nombre:	Buscar_Horario_año Buscar_Horario_grupo Buscar_Horario_semestre Buscar_Horario_semana
Descripción:	Esta clase permite gestionar todo lo que tiene que ver con el módulo de los horarios.

Nombre: Grupo	
Tipo de clase: controladora	
Para cada responsabilidad:	
Nombre:	Insertar_Grupo() Buscar_Todos_Grupos() Buscar_Grupos() Actualizar_Grupo() Eliminar_Grupo()
Descripción:	Esta clase permite gestionar todo lo que tiene que ver con el módulo de los grupos.

Nombre: Horario	
Tipo de clase: controladora	
Para cada responsabilidad:	
Nombre:	Reporte por local() Insertar_Horario() Actualizar_Horario_estudiantes()

Capítulo 3: Análisis y Diseño del Sistema

Descripción:	Esta clase permite gestionar todo lo que tiene que ver con el módulo del horario.
--------------	---

Nombre: Locales	
Tipo de clase: controladora	
Para cada responsabilidad:	
Nombre:	RegistrarLocal() Buscar_Local() Buscar_Locales_disponibles(Id_Semana,Id_Día,Id_turno) ExisteLocal()
Descripción:	Esta clase permite gestionar todo lo que tiene que ver con el módulo de locales.

Nombre: Profesores	
Tipo de clase: controladora	
Para cada responsabilidad:	
Nombre:	Asig_Profes_Asignatura()
Descripción:	Esta clase permite gestionar todo lo que tiene que ver con el módulo de profesores.

Nombre: Semestre	
Tipo de clase: controladora	
Para cada responsabilidad:	
Nombre:	RegistrarSemestre() BuscarSemestrel()
Descripción:	Esta clase permite gestionar todo lo que tiene que ver con el módulo de profesores.

Nombre: Turno	
Tipo de clase: controladora	
Para cada responsabilidad:	
Nombre:	RegistrarTurno() ModificarTurno() EliminarTurno()
Descripción:	Esta clase permite gestionar todo lo que tiene que ver con el módulo de tipo de actividad.

1.4 Principios de diseño empleados para el formato de la aplicación

En el éxito o el fracaso de una aplicación web, ya sea esta pequeña, mediana o grande, tienen gran influencia el correcto diseño de la interfaz, la elaboración de una completa y detallada ayuda y el tratamiento de los errores. Por lo que a continuación son descritos los principios de diseño utilizados para el desarrollo de la presente aplicación.

1.4.1 Estándares en la interfaz de la aplicación

El correcto diseño de una interfaz de usuario es una tarea que ha adquirido una gran relevancia en el desarrollo de sistemas, pudiera definirse como interfaz de usuario a: “el conjunto de trabajos y pasos que seguirá el usuario durante todo el tiempo que se encuentre en relación con el software, detallando lo que verá y escuchará en cada momento, las acciones que realizará, así como las respuestas que el software brindará”.

La calidad con que sea desarrollada la interfaz de usuario es uno de los principales motivos para lograr el éxito o el fracaso de una aplicación, es por ello que en gran medida uno de los principales aspectos de la usabilidad es la consistencia de su interfaz de usuario y por tanto la misma debe ser lo más amigable posible y debe proporcionarle funciones fáciles de aprender al usuario para que se sienta mejor identificada con la misma, siendo esta, el medio por el cual se interactúa con el sistema.

Para el desarrollo de la interfaz de usuario de la siguiente aplicación se tuvieron en cuenta los siguientes aspectos:

- Reducir la carga a la memoria.
- Lograr que los usuarios realicen las acciones de forma fácil.
- Obtención en todo momento de información de retroalimentación.
- Interfaz sencilla para que el usuario en ningún momento se sienta desorientado.
- Previsiones de errores y manejo de errores de entrada de datos por parte del usuario.
- Se utilizaron colores gris, blanco y negro, estándares y acordes a las normas de diseño de la Facultad.
- Se utilizó una hoja de estilo para guardar la configuración del diseño para todas las páginas, para los botones y las líneas se utilizaron estos estilos, eliminando así el número de imágenes que ralenticen la presentación de la página.
- La letra usada fue tipo Arial de tamaño 11, porque permite una separación entre las letras óptima para presentar información.

1.4.2 Tratamiento de errores

En el sistema propuesto se evitan, minimizan y tratan los posibles errores, con el fin de garantizar la integridad y confiabilidad de la información que en este se registra y muestra. Los errores se tratan en una página especial que incluye el fichero de configuración general, y está preparada para recoger el número del error y presentar la pantalla con el error que le corresponde a ese código. En algunos casos incluye la forma de solucionar el error, como es el caso de la sesión, y la autenticación de los usuarios.

Algunos errores serán generados por funciones Java Script para evitar la ejecución de la página en vano y otros en forma de mensajes de texto en la misma página donde se ejecutó la acción, de forma que el usuario pueda corregir más fácilmente y continuar. Se utilizan mensajes de confirmación, para acciones que son irreversibles como es el caso de las eliminaciones.

Los mensajes de error que emite el sistema se muestran en un lenguaje de fácil comprensión para los usuarios. Cuando se introduce información en un formulario y faltan datos, sale un cuadro de alerta indicando el campo o dato que falta. Similar ocurre cuando se introduce información errónea en un campo numérico o e-mail.

1.4.3 Seguridad

El sistema se encarga de controlar los diferentes niveles de acceso y funcionalidad de usuarios en la aplicación, identificando al usuario antes de que este pueda realizar cualquier acción sobre el mismo; garantizando que la información sea vista y manipulada únicamente por quien tiene derecho a hacerlo.

1.5 Diseño de la Base de Datos

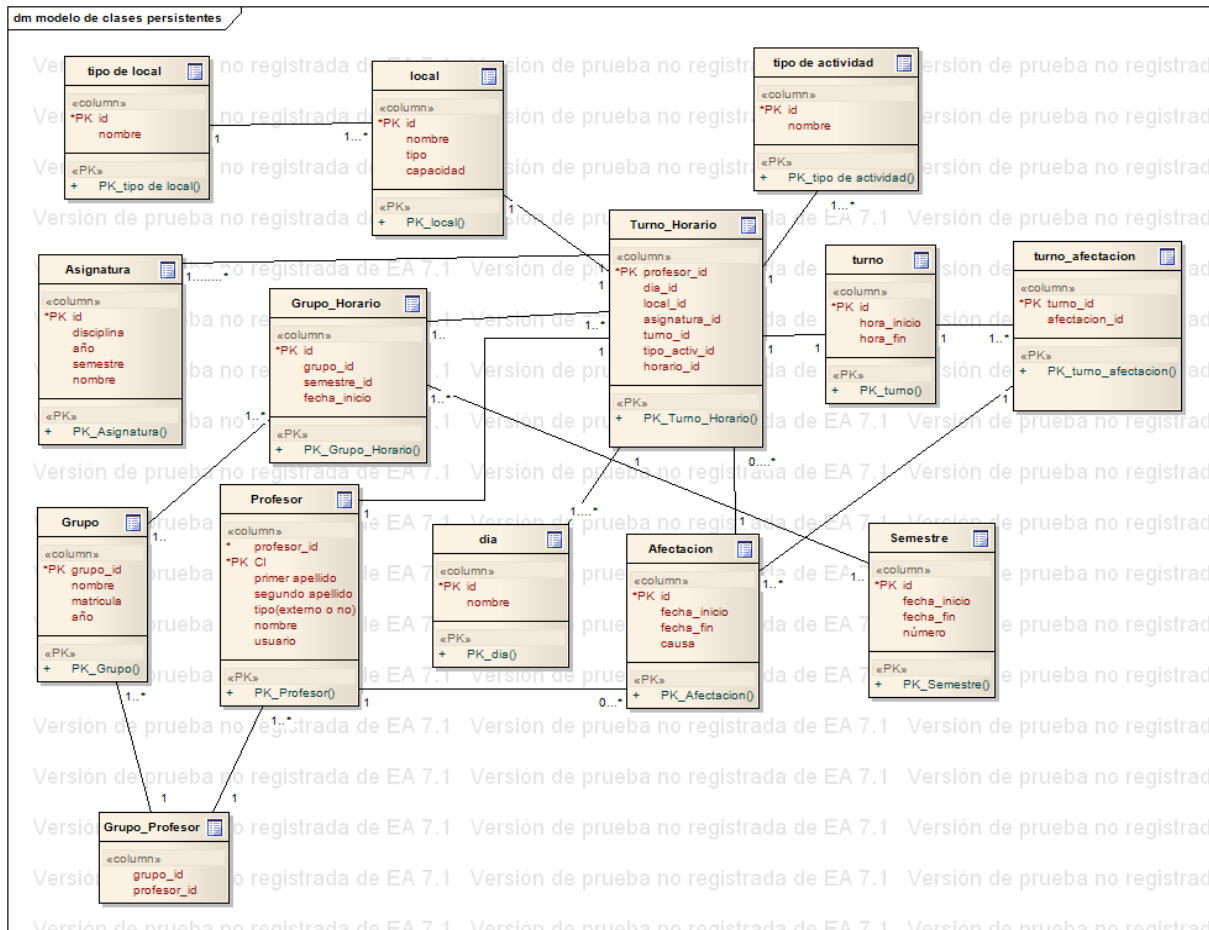
Para el diseño de la base de dato, se creó una base de datos general que une a todos los módulos referentes a los sistemas para el Control de la guardia estudiantil, al de la Gestión de pases y al de Control de la asistencia.

A continuación se muestran las tablas referentes al modelo de datos del sistema Gestión de los Horarios Docentes (Clases persistentes):

El modelo de clases persistentes está formado por las tablas:

1. Grupo
2. Asignatura
3. Profesor
4. Local
5. Horario
6. Semestre
7. Día
8. Turno
9. Tipo de actividad
10. Afectación

1.5.1 Diagrama del Modelo de Datos



1.5.2 Descripción de las tablas del Modelo de Datos.

-Tabla Grupo.

Nombre: Grupo		
Descripción: En esta tabla se recoge toda la información referente a los grupos docentes de la Facultad 7.		
Atributo	Tipo	Descripción
1-Nombre	1-Varchar	1-Nombre del grupo docente
2-Matrícula	2-Integer	2-La matrícula que tiene el grupo docente.
3-Sesión	3- Integer	3-Sesión en la que da clases el grupo docente.
4-Año	4- Integer	4-Año del grupo docente.
5-id_grupo	5-integer	5- Es un identificador.

-Tabla Asignatura

Nombre: Asignatura		
Descripción: En esta tabla se recoge toda la información referente a las asignaturas que se imparten en la Facultad 7.		
Atributo	Tipo	Descripción
1-Nombre 2-Semestre 3-Disciplina 4-Año 5-id_asignatura	1-Varchar 2-Integer 3- Integer 4- Integer 5-Integer	1-Nombre de la asignatura 2-Semestre en el que se imparte la asignatura. 3-Disciplina (Refiriéndose al departamento)a la que pertenece la asignatura. 4-Año que recibe la asignatura 5- Es un identificador.

-Tabla Profesor

Nombre: Profesor		
Descripción: En esta tabla se recoge toda la información referente a los profesores que pertenecen a la Facultad 7.		
Atributo	Tipo	Descripción
1-Nombre 2-Primer apellido 3-Segundo apellido 4-Tipo 5-Usuario 6-CI 7-id_profesor	1-Varchar 2-Varchar 3-Varchar 4- Integer 5-Varchar 6-Numeric 7-Integer	1-Nombre del profesor. 2-Primer apellido del profesor. 3-Segundo apellido del profesor 4-Si es externo o no el profesor. 5-Usuario del profesor 6-El carnet de identidad del profesor. 7- Es un identificador.

-Tabla Local

Nombre: Local		
Descripción: En esta tabla se recoge toda la información referente a los locales que se destinan para la docencia en la Facultad 7.		
Atributo	Tipo	Descripción
1-Nombre 2-Capacidad 3-Tipo 4-id_local	1-Varchar 2-Integer 3- Integer 4-Integer	1-Nombre del local(Numeración que se le pone) 2-Capacidad del local. 3-Tipo de local(Aula, Salón, laboratorio u otro). 4- Es un identificador.

-Tabla Horario

Nombre: Horario		
Descripción: En esta tabla se recoge toda la información referente a los datos que se necesitan para confeccionar el horario. Todas las tablas se relacionan con esta.		
Atributo	Tipo	Descripción
1-id_asignatura	1-Integer	1-Para acceder a toda la información

Capítulo 3: Análisis y Diseño del Sistema

2-id_horariosemana 3-id_local 4-id_profesor 5-id_turno 6-id_dia 7-id_tipo_actividad 8-id_horario	2-Integer 3-Integer 4-Integer 5-Integer 6-Integer 7-Integer 8-Integer	de esta tabla. 2-Para acceder a toda la información de esta tabla. 3-Para acceder a toda la información de esta tabla. 4-Para acceder a toda la información de esta tabla. 5-Para acceder a toda la información de esta tabla. 6-Para acceder a toda la información de esta tabla. 7-Para acceder a toda la información de esta tabla. 8-Para acceder a toda la información de esta tabla.
--	---	---

-Tabla Semestre

Nombre: Semestre		
Descripción: En esta tabla se recoge toda la información referente a los semestres de clases que se san en la Facultad 7		
Atributo	Tipo	Descripción
1-Fecha inicio 2-Fecha fin 3-Número 4-id_semestre	1-Date 2-Date 3- Integer 4- Integer	1-Fecha en la que inicia un semestre. 2-fecha en la que culmina un semestre. 3-Si el semestre uno o dos. 4- Es un identificador.

-Tabla Día

Nombre: Día		
Descripción: En esta tabla se recoge toda la información referente a los días de clases de los grupos de la Facultad 7		
Atributo	Tipo	Descripción
1-Nombre 2-Dia 3-id_dia	1-Varchar 2-Integer 3- Integer	1-Nombre del día(Lunes, Martes, Miércoles, Jueves ,Viernes, Sábado) 2-Número del día. 3- Es un identificador.

-Tabla Turno

Nombre: Turno		
Descripción: En esta tabla se recoge toda la información referente a los turnos de clases.		
Atributo	Tipo	Descripción
1-Número 2-Horalnicio 3-HoraFin 4-id_turno	1-Integer 2-Time 3- Time 4-Integer	1-Número del turno. 2-Hora de inicio del turno. 3-Hora de terminar el tuno. 4- Es un identificador.

Capítulo 3: Análisis y Diseño del Sistema

--	--	--

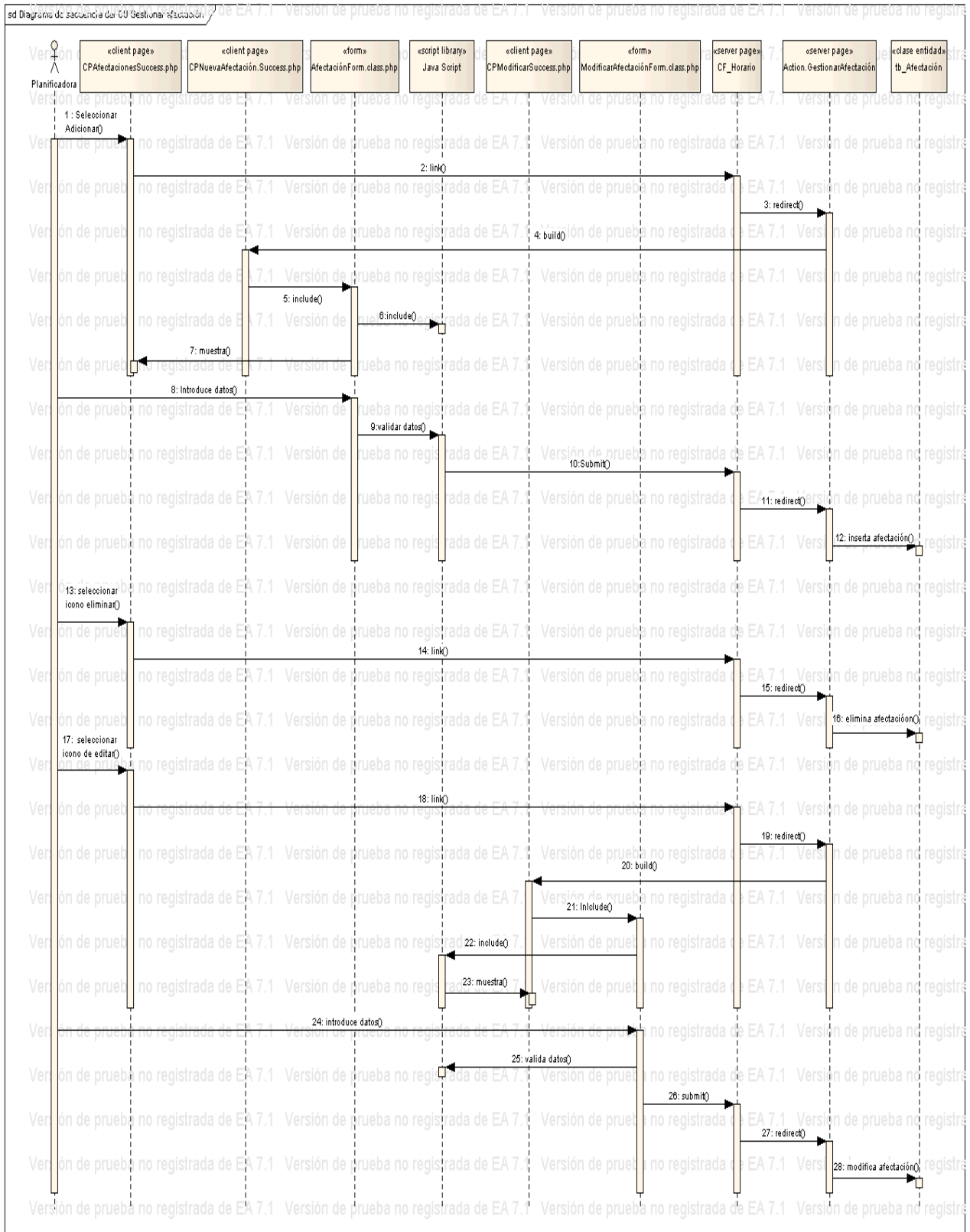
-Tabla Afectación

Nombre: Afectación		
Descripción: En esta tabla se recoge toda la información referente a las afectaciones de los profesores de la Facultad 7.		
Atributo	Tipo	Descripción
1-FechaInicio	1-Date	1-Fecha en la que inicia la afectación.
2-FechaFin	2-Date	2-Fecha en la que culmina la afectación.
3-Causa	3- Text	3-La causa de la afectación.
4-id_afectación	4-Integer	4- Es un identificador.
5-id_profesor	5-Integer	5-Es el identificador del profesor por la relación entre las dos tablas.

-Tabla Tipo de actividad

Nombre: Tipo de actividad		
Descripción: En esta tabla se recoge toda la información referente a las afectaciones de los profesores de la Facultad 7.		
Atributo	Tipo	Descripción
1-Nombre	1-Varchar	1-Nombre de la actividad(Clase práctica, Conferencia, Laboratorio u otras)
2-id_actividad	2-Integer	2-Es un identificador.

3.6 Diagrama de secuencia del diseño



1.7 Patrones de diseños utilizados

1.7.1 Alta Cohesión

Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme.

El patrón Alta Cohesión es la meta principal que ha de tenerse en cuenta en cada momento en todas las decisiones de diseño. Es un patrón evaluativo que el desarrollador aplica al valorar sus decisiones de diseño. La cohesión es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. Una clase con baja cohesión hace muchas cosas no afines o un trabajo excesivo. Además son difíciles de entender, de reutilizar y de mantener. Son clases las cuales constantemente las afectan los cambios.

- Muy baja cohesión: Una clase es la única responsable de muchas cosas en áreas funcionales muy heterogéneas.
- Baja cohesión: Una clase tiene responsabilidad exclusiva de una tarea compleja dentro de un área funcional.
- Alta cohesión: Una clase tiene responsabilidades moderadas en un área funcional y colabora con las otras para compartir el esfuerzo y llevar a cabo las tareas.
- Cohesión moderada: Una clase tiene responsabilidades exclusivas en unas cuantas áreas que están relacionadas lógicamente con el concepto de clase, pero no entre ellas.

Con el uso de este patrón mejoran la claridad y la facilidad con que se entiende el diseño. Se simplifican el mantenimiento y las mejoras en funcionalidad. A menudo se genera un bajo acoplamiento. La ventaja de una gran funcionalidad soporta una mayor capacidad de reutilización, porque una clase muy cohesiva puede destinarse a un propósito muy específico.

1.7.2 Bajo Acoplamiento.

El bajo acoplamiento es un principio que se debe tener siempre en cuenta durante las decisiones de diseño. Es un patrón evaluativo que el diseñador aplica al juzgar sus decisiones de diseño. Este patrón estimula asignar una responsabilidad de modo que su colocación no incremente el acoplamiento tanto que produzca los resultados negativos propios de un alto acoplamiento.

Soporta el diseño de clases más independientes, que reducen el impacto de los cambios, y también más reutilizables, que acrecientan la oportunidad de una mayor productividad. No puede considerarse en forma independiente de otros patrones como Experto o Alta cohesión, sino que más bien ha de incluirse como uno de los principios del diseño que influyen en la decisión de asignar responsabilidades.

Con el uso de este patrón los componentes no se afectan por cambios de otros componentes, son fáciles de entender por separado y fáciles de reutilizar.

Conclusiones.

En este capítulo fueron tratadas las concepciones básicas para el análisis y diseño del sistema. Se desarrollaron los diagramas de clases del análisis de diseño, además de describir los principios de diseño seguidos, los estándares de la interfaz y la concepción del tratamiento de errores.

CAPÍTULO 4 IMPLEMENTACIÓN

Introducción

En el capítulo anterior se modeló los diagramas de clases, los diagramas de clases Web. En este capítulo se pasa a la implementación del Software en el lenguaje de programación escogido, se modelarán los diagramas de Despliegues y Componentes para una mayor vista para la implementación del producto.

La implementación es la fase más esperada en un proceso de desarrollo de un producto software, es donde se hacen realidad todas las ideas y artefactos que han sido modelados por el equipo de trabajo responsable de la solución; no constituye una etapa independiente y formalmente delimitada en el proceso, ya que la metodología de desarrollo utilizada permite que partes de la solución que conceptualmente estaban definidas y que se podrían ir prototipando de forma funcional se le diera luz verde para implementación mucho antes de que estos artefactos fueran modelados.

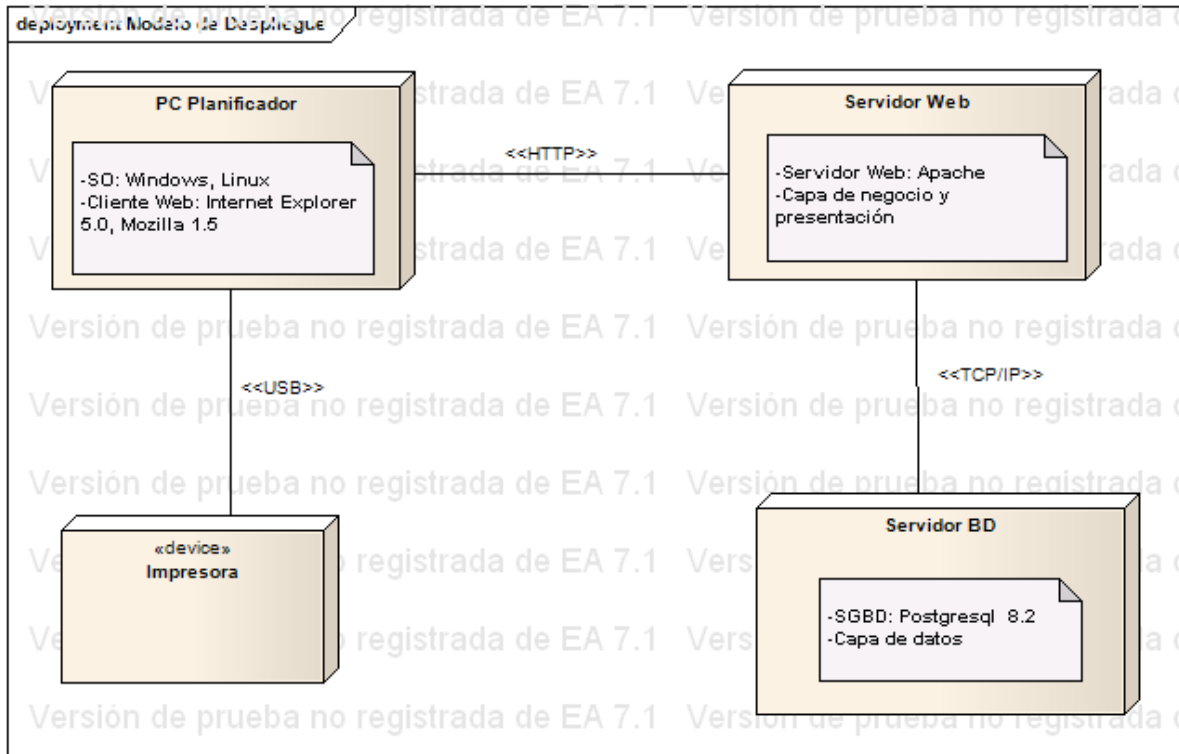
4.1 Diagrama de despliegue.

El modelo de despliegue contiene los nodos que conforman la topología de hardware sobre la que se ejecuta el sistema. Muestra las relaciones entre el hardware y el software en el sistema final. Se representa como un grafo de nodos unidos por conexiones de comunicación.

El diagrama de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. Es una colección de nodos y arcos; donde cada nodo representa un recurso de cómputo, normalmente un procesador o un dispositivo de hardware similar. (35)

Muestra la configuración de los componentes hardware, los procesos, los elementos de procesamiento en tiempo de ejecución y los objetos que existen en tiempo de ejecución. En este tipo de diagramas intervienen nodos, asociaciones de comunicación, componentes dentro de los nodos y objetos que se encuentran a su vez dentro de los

componentes. Un nodo es un objeto físico en tiempo de ejecución, es decir una máquina que se compone habitualmente de, por lo menos, memoria y capacidad de procesamiento, a su vez puede estar formada por otros componentes. (36)



El modelo despliegue que se propone por tres computadoras una de ellas haciendo función de servidor de Base de Datos utilizando como SGBD Postgres 8.2, otra máquina como PC cliente que sería la PC Planificador, la cual usará como sistema operativo Windows o Linux y como navegadores Internet Explorer y Mozilla. La aplicación como tal brindará una interfaz de web por lo que necesitará una computadora con conexión a servidores externos.

La conexión entre la impresora y la PC cliente es mediante puerto USB, la de la PC Cliente y la PC Servidor Web es mediante el protocolo HTTP, así como la relación entre las PC servidoras es con el protocolo TCP/IP.

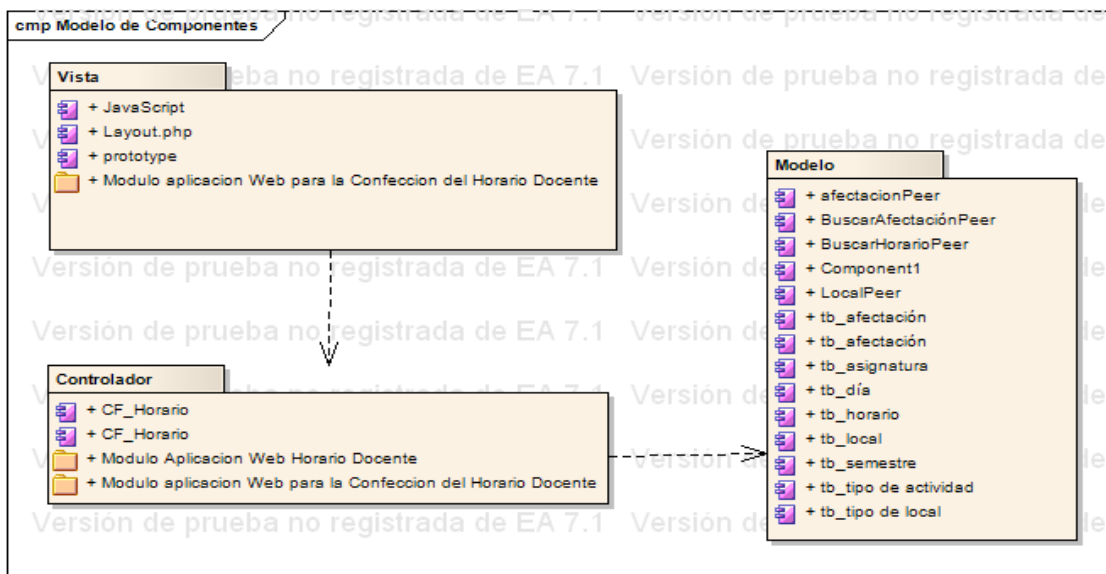
4.2 Diagrama de Componentes

Un diagrama de componentes muestra las organizaciones y dependencias lógicas entre componentes software, sean éstos componentes de código fuente, binarios o ejecutables.

Normalmente contienen componentes, interfaces y relaciones entre ellos y como todos los diagramas, también puede contener paquetes utilizados para agrupar elementos del modelo. (37)

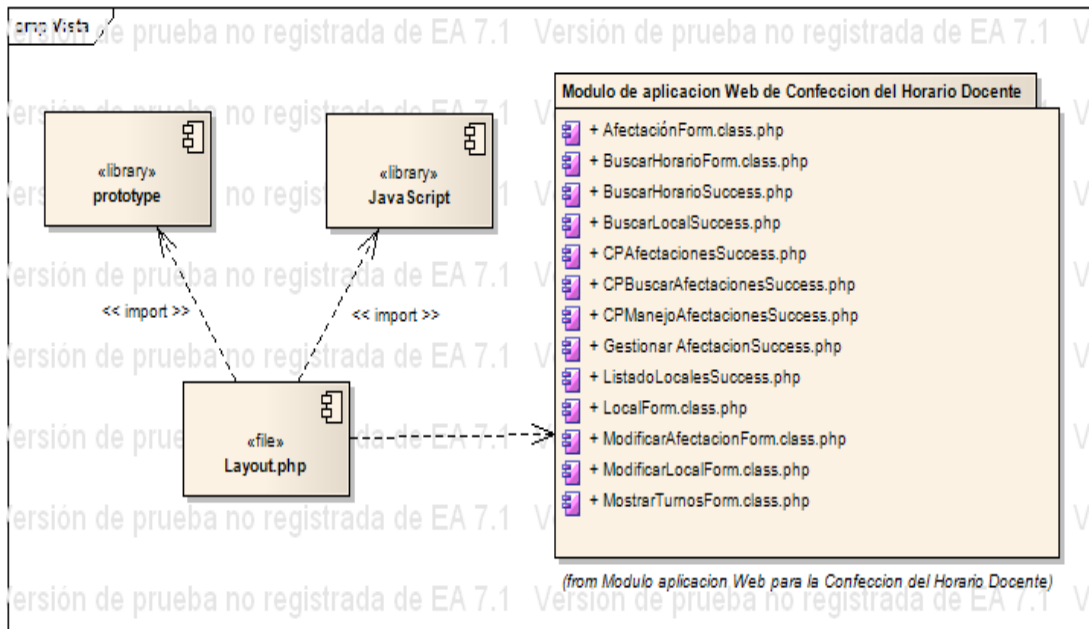
Los elementos de modelado dentro de un diagrama de componentes serán componentes y paquetes. En cuanto a los componentes, sólo aparecen tipos de componentes, ya que las instancias específicas de cada tipo se encuentran en el diagrama de despliegue.

Los diagramas de componentes muestran los componentes software que constituyen una parte reusable, sus interfaces, y sus interrelaciones, en muchos aspectos se puede considerar que un diagrama de componentes es un diagrama de clases a gran escala. Cada componente en el diagrama debe ser documentado con un diagrama de componentes más detallado, un diagrama de clases, o un diagrama de casos de uso.

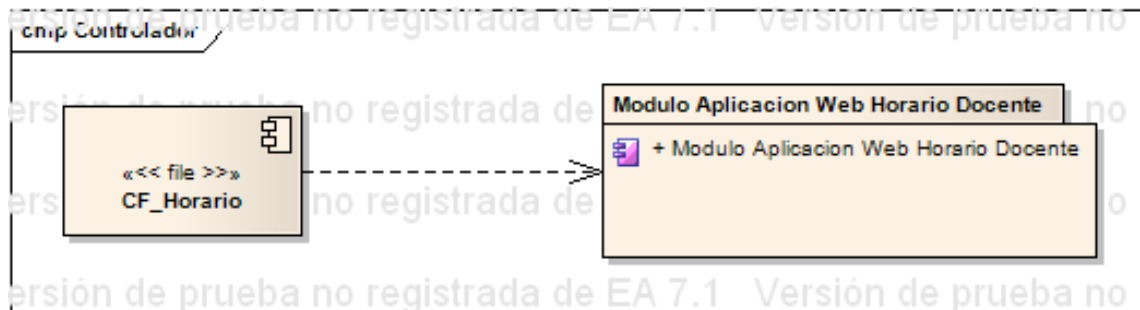


Un diagrama de componentes se representa como un grafo de componentes software unidos por medio de relaciones de dependencia (generalmente de compilación). Puede mostrar también que un componente software contiene una interfaz, es decir, la soporta. En la arquitectura en capas con Symfony se modela por capas.

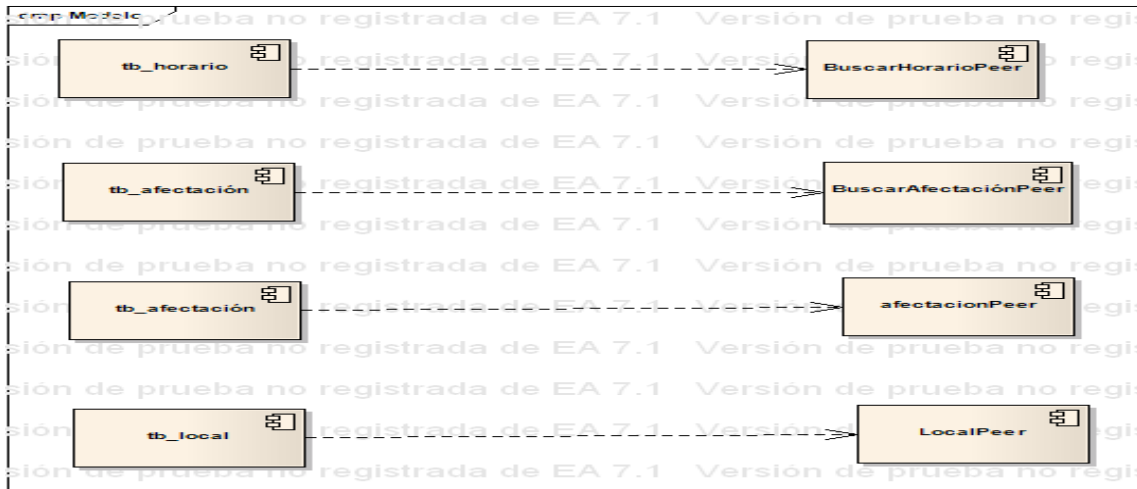
Dentro de la vista



Dentro del controlador



Dentro del Modelo



Normalmente los diagramas de componentes se utilizan para modelar código fuente, versiones ejecutables, bases de datos físicas, entre otros.

Conclusiones

En este capítulo se mostró la topología de hardware sobre la que corre el sistema a través del modelo de despliegue y fue definida la organización de las clases y objetos en componentes. A partir de este momento el sistema cuenta con funcionalidad operativa.

CONCLUSIONES

El presente trabajo brinda una solución al problema de la gestión de horarios docentes en la Facultad 7 de la Universidad de las Ciencias Informáticas implementando una aplicación web para la confección del horario docente.

Se desarrolla el análisis y diseño para que la aplicación sea capaz de generar horarios y brindar muchas otras facilidades a estudiantes y profesores, proponiendo una interfaz sencilla haciendo uso de las facilidades del servicio de intranet para que sea posible obtener la información demandada en tiempo real, garantizando la solidez de los datos.

El sistema se desarrolló siguiendo la metodología RUP, y se utilizaron representaciones UML para la modelación de todas las fases del proyecto. El sistema resultante está provisto de un ambiente cómodo, fácil de entender, que cumple los estándares de diseño y utiliza técnicas modernas de programación.

Por todo lo anterior se concluye que los objetivos propuestos para el presente trabajo han sido cumplidos satisfactoriamente.

RECOMENDACIONES

1. Implementar en próximas versiones una funcionalidad que genere el horario de forma inteligente para de esta forma optimizar el proceso de confección del Horario Docente.
2. Implementar un reporte para obtener el Horario Docente por los diferentes departamentos de la facultad.
3. Proponer, tras corroborar un desempeño exitoso, la utilización y generalización de este sistema en las diferentes facultades de la UCI.

REFERENCIAS BIBLIOGRÁFICAS

1. [En línea][Citado el:13 de enero 2009] <http://www.mic.gov.cu//hinfosoc.aspx>
2. Valdés Acosta, Aneli y Guardia Macías, María de Dolores. Sistema Automatizado para la Gestión de Horarios Docentes. 2007.
3. [En línea][Citado el 17 de 01 2009] <https://sigacad.unanleon.edu.ni/principal.php>
4. [En línea] [Citado el: 17 de 01 de 2009.] <http://www.penalara.com/>
5. [En línea] [Citado el: 17 de 01 de 2009.] <http://www.kronowin.es/>
6. [En línea] [Citado el: 17 de 01 de 2009.] <http://www.asctimetables.com>
7. [En línea] [Citado el: 17 de 01 de 2009.] <http://www.mimosa.softonic.com> y www.mimosasoftware.com
8. Tamarit Cutiño, Jorge Luis y Santiesteban Díaz, Maykel. Asistente de Ayuda para la Confección de Horarios Docentes. 2007.
9. Zamora González, Amado y Arce Fernández, Ronnie. Herramienta de ayuda para la confección del Horario Docente en la Facultad #7 2008.
10. Valdés Acosta, Aneli y Guardia Macías, María de Dolores. Sistema Automatizado para la Gestión de Horarios Docentes. 2007.
11. En línea][Citado el 22 de 02 2009]
http://www.programacion.com/articulo/grafos_asignacion/#grafos_asignacion_alg
- 12 Téllez Enríquez, Emanuel.Tesis-Tellez <http://delta.cs.cinvestav.mx/~ccoello/tesis/tesis-tellez.pdf.gz> 2007)
13. Gómez Gallego, Juan Pablo. FUNDAMENTOS DE LA METODOLOGIA RUP <http://www.scribd.com/doc/297224/RUP> y Enterprise Architect 7.1 //Ayuda del Enterprise Architect
14. [En línea] [Citado el: 26 de 02 de 2009]
<http://www.sparxsystems.com.ar/products/ea.html>
15. [En línea] [Citado el: 26 de 02 de 2009]
http://tienda.ramblainf.com/epages/tienda_ramblainf_com.sf/es_ES/?ObjectPath=/Shops/tienda_ramblainf_com/Products/ENTARC/SubProducts/EAPRO
16. [En línea] [Citado el: 26 de 02 de 2009]
http://www.taringa.net/posts/downloads/933938/Enterprise-Architect-7_0_813+-+Medicina.html

17. [En línea] [Citado el: 1 de 03 de 2009]
<http://www.apexnet.com.ar/index.php/product/viewProducts/24/sl=0>
18. [En línea] [Citado el: 8 de 03 de 2009]
http://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n
19. [En línea] [Citado el: 26 de 02 de 2009]
<http://www.monografias.com/trabajos28/reingenieria-procesos-uml20/reingenieria-procesos-uml20.shtml>
20. Verdecia Álvarez, Yoandris y Hurtado Portal, Yoisbel Módulo para la gestión de la planificación docente facultad 8.
21. [En línea] [Citado el: 11 de 03 de 2009]
<http://www.programacion.com/php/noticia/1550/>
- 22 [En línea] [Citado el: 25 de 02 de 2009.]
23. [En línea] [Citado el: 25 de 02 de 2009.]
http://www.error500.net/garbagecollector/archives/categorias/bases_de_datos/sistema_gestor_de_base_de_datos_sgbd.php
24. [En línea][Citado el:25 de 02 2009]
http://www.igac.gov.co:8080/igac_web/UserFiles/File/ciaf/TutorialSIG_2005_26_02/paginas/ctr_sistemasdegestiondebasededatos.htm
25. [En línea] [Citado el: 25 de 02 de 2009]
<http://www.buayacorp.com/archivos/postgresql-82/>
- 26 [En línea] [Citado el: 25 de 03 de 2009]
<http://www.proactivacalidad.com/java/patrones/mvc.html>
27. [En línea] [Citado el: 25 de 03 de 2009]
<http://www.desarrolloweb.com/articulos/332.php>
28. [En línea] [Citado el: 26 de 03 de 2009]
<http://www.macromedia.com/es/software/dreamweaver/>
- 29.Potiencer, Fabien. Symfony, la guía definitiva.2008 <http://www.librosweb.es>
30. Clases de Ingeniería del Software I, curso 2008-2009, UCI.
31. BOCH Grady, RUMBAUGH James, JACOBSON Ivar. “El proceso unificado de desarrollo de software”.2000. Addison Wesley.
32. Tamarit Cutiño, Jorge Luis y Santiesteban Díaz, Maikel Asistente de ayuda para la confección de horarios docents.2007
33. Herrera J., Lizka Johany. “Ingeniería de Requerimientos – Ingeniería de Software”, <http://www.monografias.com/trabajos6/resof>

Referencias Bibliográficas

34. BOCH Grad., RUMBAUGH James, JACOBSON Ivar. "El lenguaje unificado de modelado. Manual de referencia".2000. Addison Wesley
35. JACOBSON, Ivar; RUMBAUGH, James; BOOCH, Grady, "El proceso unificado de desarrollo".2000. Addison Wesley. Capítulos 9 Páginas 205-254.
36. Clases de Ingeniería del Software I, curso 2008-2009, UCI.
37. Valdés Acosta, Aneli y Guardia Macías, María de Dolores. Sistema Automatizado para la Gestión de Horarios Docentes. 2007.

BIBLIOGRAFÍA

- 1--[En línea][Fecha de consulta 31 marzo 2009].Algoritmos genéticos. Disponible en http://es.wikipedia.org/wiki/Algoritmo_gen%C3%A9tico.
- 2--[En línea][Fecha de consulta 6 abril 2009].Algoritmos genéticos y computación evolutiva. Adam Marczyk.2004 Disponible en: <http://the-geek.org/docs/algen/>
- 3--Tamarit Cutiño, J. L., y Santiesteban Díaz, M. (2007). Asistente de Ayuda para la Confección de Horarios Docentes.
- 4--BOCH Grady, RUMBAUGH James, JACOBSON Ivar. “El proceso unificado de desarrollo de software”.2000. Addison Wesley.
- 5--[En línea][Fecha de consulta 2 febrero 2009].Características de un Diagrama de Despliegue. Disponible En: <http://www-gris.det.uvigo.es/~avilas/UML/node50.html>
- 6-- [Fecha de consulta 24 febrero 2009].Clases de Ingeniería del Software I, curso 2008-2009, UCI.
- 7--[En línea][Fecha de consulta 2 febrero 2009].Cronos Disponible en: <http://cronos.sisvenca.com/Cronos.html>
- 8--[Fecha de consulta 12 febrero 2009].Clases de Ingeniería del Software I, curso 2008-2009, UCI.
- 9--[Fecha de consulta 12 febrero 2009].Clases de Ingeniería del Software I, curso 2008-2009, UCI.
- 10--[En línea][Fecha de consulta 6 abril 2009].Colonia de Hormigas. Disponible en <http://delta.cs.cinvestav.mx/~ccoello/tesis/tesis-tellez.pdf.gz>.
Clases de Ingeniería del Software I, curso 2008-2009, UCI.
- 11-- [En línea][Fecha de consulta 27 abril 2009] Disponible en <http://www.penalara.com/>
- 12--[En línea][Fecha de consulta 7 mayo 2009] Disponible en: <https://sigacad.unanleon.edu.ni/principal.php>
- 13-- [En línea][Fecha de consulta 15 mayo 2009] Disponible en: <http://www.kronowin.es>
- 14--[En línea][Fecha de consulta 17 mayo 2009] Disponible en: <http://www.asctimetables.com>
- 15--[En línea][Fecha de consulta 24 mayo 2009] Disponible en: <http://www.mic.gov.cu/hinfosoc.aspx>

- 16--[En línea][Fecha de consulta 27 mayo 2009] Disponible en:
<http://www.desarrolloweb.com/articulos/332.php>
http://es.wikipedia.org/wiki/Adobe_Dreamweaver
- 17--[En línea][Fecha de consulta 21 abril 2009].Uml. Disponible en
<http://es.tldp.org/Tutoriales/doc-modelado-sistemas-UML/multiple-html/c12.html>.
<http://www.osmosislatina.com/lenguajes/uml/basico.htm>.
- 18--Fowler, Martín. "UML Gota a Gota". Primera edición. Addison Wesley Longman. 1999.
- 19--[En línea][Fecha de consulta 8 abril 2009].Fundamentos de la Teoría de Grafos. Disponible en <http://www.sol.com/algoritmos/pagina03.asp>.
- 20--[En línea][[Fecha de consulta 30 enero 2009].]. Generador de Horarios para Centros Docentes 7.0. Disponible en:
<http://generador-de-horarios-para-centros-docentes.softonic.com/ie/22084>.
- Gómez Gallego, Juan Pablo. FUNDAMENTOS DE LA METODOLOGIA RUP <http://www.scribd.com/doc/297224/RUP> y Enterprise Architect 7.1 //Ayuda del Enterprise Architect.
- 21-- Herrera J., Lizka Johany. "Ingeniería de Requerimientos – Ingeniería de Software", <http://www.monografias.com/trabajos6/resof>
- 22-- [En línea][Citado el:13 de enero 2009] <http://www.mic.gov.cu/hinfosoc.aspx>
- 23-- [En línea][Citado el 17 de 01 2009] <https://sigacad.unanleon.edu.ni/principal.php>
- 24-- [En línea] [Citado el: 17 de 01 de 2009.] <http://www.penalara.com/>
- 25-- [En línea] [Citado el: 17 de 01 de 2009.] <http://www.kronowin.es/>
- 26--[En línea] [Citado el: 17 de 01 de 2009.] <http://www.asctimetables.com>
- 27-- [En línea] [Citado el: 17 de 01 de 2009.] <http://www.mimosa.softonic.com> y www.mimosasoftware.com
- 28--En línea][Citado el 22 de 02 2009]
http://www.programacion.com/articulo/grafos_asignacion/#grafos_asignacion_alg
- 29--[En línea] [Citado el: 26 de 02 de 2009]
<http://www.sparxsystems.com.ar/products/ea.html> [En línea] [Citado el: 26 de 02 de 2009]
http://tienda.ramblainf.com/epages/tienda_ramblainf_com.sf/es_ES/?ObjectPath=/Shops/tienda_ramblainf_com/Products/ENTARC/SubProducts/EAPRO
- 30--[En línea] [Citado el: 26 de 02 de 2009]
http://www.taringa.net/posts/downloads/933938/Enterprise-Architect-7_0_813--Medicina.html

- 31-- [En línea] [Citado el: 1 de 03 de 2009]
<http://www.apexnet.com.ar/index.php/product/viewProducts/24/sl=0>
- 32-- [En línea] [Citado el: 8 de 03 de 2009]
http://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n
- 33--[En línea] [Citado el: 26 de 02 de 2009]
<http://www.monografias.com/trabajos28/reingenieria-procesos-uml20/reingenieria-procesos>
- 34-- [En línea] [Citado el: 11 de 03 de 2009]
<http://www.programacion.com/php/noticia/1550/>
http://www.error500.net/garbagecollector/archives/categorias/bases_de_datos/sistema_gestor_de_base_de_datos_sgbd.php
- 35-- [En línea][Citado el:25 de 02 2009]
http://www.igac.gov.co:8080/igac_web/UserFiles/File/ciaf/TutorialSIG_2005_26_02/paginas/ctr_sistemasdegestiondebasededatos.htm
- 36-- [En línea] [Citado el: 25 de 02 de 2009]
<http://www.buayacorp.com/archivos/postgresql-82/>
- 37--[En línea] [Citado el: 25 de 03 de 2009]
<http://www.proactivacalidad.com/java/patrones/mvc.html>
- 38--[En línea] [Citado el: 25 de 03 de 2009]
<http://www.desarrolloweb.com/articulos/332.php>
- 39-- [En línea] [Citado el: 25 de 02 de 2009.] 28. [En línea] [Citado el: 26 de 03 de 2009]
<http://www.macromedia.com/es/software/dreamweaver>
- 40--[En línea][Fecha de consulta 8 abril 2009].Introducción a los Algoritmos Genéticos. Disponible en <http://www.redcientifica.com/doc/doc199904260011.html<28/3/2007>.
- 41--JACOBSON, Ivar; RUMBAUGH, James; BOOCH, Grady, “El proceso unificado de desarrollo”.2000. Addison Wesley. Capítulos 9 Páginas 205-254.
- 42-- [En línea][Fecha de consulta 12 febrero 2009].Metodologías De Desarrollo De Software. Disponible en:
http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html .
- 43-- [En línea][Fecha de consulta 5 febrero 2009].Patrón Modelo-Vista-Controlador. Disponible
Potiencer, Fabien. Symfony, la guía definitiva.2008 <http://www.librosweb.es>

- 44--[En línea][Fecha de consulta 4 marzo 2009].PostgreSQL 8.1.x. Disponible en:
<http://www.postgresql.cl/>
- 45--[En línea][Fecha de consulta 14 abril 2009].Postgres SQL. Disponible en
<http://es.wikipedia.org/wiki/PostgreSQL>
en:<http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/MTJ_3317.as
- 46--[En línea][Fecha de consulta 2 febrero 2009].Principios de Diseño. Disponible en
<http://www.sidar./recur/desdi/traduc/es/visitable/quees/dcu.htm>
- 47-- [En línea][Fecha de consulta 7 marzo 2009].Programación en Capas. Disponible
En:< http://es.wikipedia.org/wiki/Programaci%C3%B3n_por_capas
- 48--Pressman, Roger. Ingeniería del Software, Enfoque Práctico. Ciudad Habana : Felix
Varela , 2005.
- 49--Pressman, Roger S. Ingeniería de Software, un enfoque práctico. Cuarta
Edición.
- 50--Pressman, Roger. Ingeniería del Software, Enfoque Práctico. Ciudad Habana : Felix
Varela , 2005.
- 51--[En línea][Fecha de consulta 14 marzo 2009] Publicaciones en español.
<http://es.tldp.org/Tutoriales/doc-modeladosistemasUML/multiple-html/x121.htm>
- 52--[En línea][Fecha de consulta 17 febrero 2009].RATIONAL ROSE: PROCEDIMIENTOS
BÁSICOS PARA DESARROLLAR UN PROYECTO CON UML. Disponible
En:<<http://www.vico.org/TallerRationalRose.pdf>.30--[En línea][Fecha de consulta 14 abril
- 53--[En línea][Fecha de consulta 31 enero 2009].Refusiones para el Generador de
Horarios y Generador de Grupos Milenio-6. Disponible en:
<http://www.adossis.es/REFUGHM6.htm>
- 2009].Rup. Disponible en http://es.wikipedia.org/wiki/Rational_Unified_Process.40--[En
línea][Fecha de consulta 3 mayo 2009].Reynoso, Carlos. Introducción a la Arquitectura
del Software.
http://www.microsoft.com/spanish/msdn/arquitectura/roadmap_arq/intro.msp#EAG.
- 54--[En línea][Fecha de consulta 27 abril 2009].Sistemas gestores de Base de Datos
libres. Disponible en: http://es.wikipedia.org/wiki/DBMS#SGBD_libres1--[En
línea][Consultado:el 30 de enero 2009]. Sistema Generador de Horarios.
Disponiblehttp://lcc.ens.uabc.mx/sh2007_1/index.jsp.
- 55--[En línea][Fecha de consulta 4 marzo 2009].Sistema Gestor de base de datos SGBD.
Disponible

En:http://www.error500.net/garbagecollector/bases_de_datos/sistema_gestor_de_base_de_datos.html

56--Valdés Acosta, A., y Guardia Macías, M. d. (2007). Sistema Automatizado para la Gestión de Horarios Docentes.

57--[En línea][Fecha de consulta 21 marzo 2009].Sitio de Ing. de software.

<http://www.dcc.uchile.cl/~psalinas/uml/casosuso.html#actor>

58--[En línea][Fecha de consulta 21 marzo 2009].Sitio de Ing. de software.

<http://www.unp.edu.ar/fce/documentos/trelew/analisis%20sistemas%20I/UML2.pdf>.

59--Sanchez, Alfredo. Paradigmas y estilos de interacción. 2007.

Tamarit Cutiño, Jorge Luis y Santiesteban Díaz, Maykel. Asistente de Ayuda para la Confección de Horarios Docentes. 2007

60--[En línea][Fecha de consulta 12 febrero 2009].Tutorial de UML. Disponible en:

<http://www.dcc.uchile.cl/~psalinas/uml/introduccion.html>

61--[En línea][Fecha de consulta 8 marzo 2009].Tutorial de PostgreSQL. Disponible en:

<http://es.tldp.org/Postgresql-es/web/navegable/tutorial/tutorial.html>

62--[En línea][Fecha de consulta 14 abril 2009].Tabú Search. Disponible en

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

Téllez Enríquez, Emanuel.Tesis-Tellez <http://delta.cs.cinvestav.mx/~ccoello/tesis/tesis-tellez.pdf.gz> 2007)

63--[En línea][Fecha de consulta 27 marzo 2009].Universidad de Córdoba, “Glosarios de Internet”, <http://www.uco.es/ccg/glosario>.

64--Un estudio para automatizar Horarios. Toma de A Survey of Automated

Timetabling. Andrea Schaerf, pág. 35

65--[En línea][Fecha de consulta 21 abril 2009].Vértice-Coloración. Disponible en

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

66-- Valdés Acosta, Aneli and Guardia Macías, María de Dolores. Sistema Automatizado para la Gestión de Horarios Docentes. 2007.

Verdecia Álvarez, Yoandris y Hurtado Portal, Yoisbel Módulo para la gestión de la planificación docente facultad 8.

67--Zamora González, Amado y Arce Fernández, Ronnie. Herramienta de ayuda para la confección del Horario Docente en la Facultad #7 2008.

GLOSARIO DE TÉRMINOS

Autónoma: Trabaja por cuenta propia

Artefactos: Todo lo que se realiza para un mayor conocimiento del trabajo

Accesibilidad: Facilidad con la que algo puede ser usado, visitado o accedido en general por todas las personas.

Actor: Alguien o algo, fuera del sistema o negocio que interactúa con el sistema o negocio.

AGs: Algoritmos Genéticos

CUS: Caso de uso del sistema.

Ciclos: Son los periodos de enseñanza en los que se estructuran las titulaciones

CMS: Significa sistema de administración de contenido (en inglés Content Management System), un sistema de este tipo funciona para la creación y administración de contenido, actualmente se usa para denominar a los sistemas que sirven para administrar el contenido de páginas Web.

CASE: Computer Aided Software Engineering.

Client Page (Página Cliente): Las páginas clientes representan páginas web con formato HTML y son interpretadas por el navegador.

Desarrollo Sostenible: Se llama desarrollo sostenible aquél desarrollo que es capaz de satisfacer las necesidades actuales sin comprometer los recursos y posibilidades de las futuras generaciones. Intuitivamente una actividad sostenible es aquella que se puede mantener.

Expertos: Simplemente conoce sobre un campo delimitado del saber.

Estandarizado: Ajustado a un tipo, modelo o norma

GHC: Generador de Horarios para Centros Docentes.

HTML: Hypertext Markup Language. Lenguaje usado para escribir documentos para servidores World Wide Web. Es una aplicación de la ISO Standard 8879:1986. Es un lenguaje de marcas. Los lenguajes de marcas no son equivalentes a los lenguajes de programación aunque se definan igualmente como "lenguajes". Son sistemas complejos de descripción de información, normalmente documentos, que se pueden controlar desde cualquier editor ASCII.

Hito: Punto de control de objetivo intermedio antes de que el proyecto finalice

HTTP: Hipertexto Transfer Protocol. Protocolo de Transferencia de Hipertextos. Modo de comunicación para solicitar páginas Web.

Herramientas CASE: Herramientas utilizadas para el desarrollo de proyectos de Ingeniería de Software.

Hardware: Componentes electrónicos, tarjetas, periféricos y equipo que conforman un sistema de computación; se distinguen de los programas (software) porque son tangibles.

Horario Docente: Constituye el instrumento organizativo de las actividades docentes, que permite brindar una visión general de la distribución cuantitativa de las asignaturas contenidas en el plan de estudio, según el número de horas clases cada año. HDF7:

Horario Docente Facultad 7.

Iterativo: Algo repetitivo.

Interfaz: Frontera convencional entre dos sistemas o dos unidades, que permite intercambio de informaciones.

Informatización de la Sociedad: La Informatización de la Sociedad es el proceso de utilización ordenada y masiva de las Tecnologías de la Información y las Comunicaciones en la vida cotidiana, para satisfacer las necesidades de todas las esferas de la sociedad, en su esfuerzo por lograr cada vez más eficacia y eficiencia en todos los procesos y por consiguiente mayor generación de riqueza y aumento en la calidad de vida de los ciudadanos.

Metodología: Metodología se refiere a los métodos de investigación en una ciencia. Aun cuando el término puede ser aplicado a las artes cuando es necesario efectuar una observación o análisis más riguroso o explicar una forma interpretar la obra de arte.

Microsoft: Compañía que manufactura los sistemas de operación DOS y Windows.

MySQL: Es un sistema de gestión de bases de datos relacional que cuentan con todas las características de un motor de BD comercial: transacciones atómicas, triggers, replicación, llaves foráneas entre otras. Su ingeniosa arquitectura lo hace extremadamente rápido y fácil de personalizar.

Nodo: Un nodo es un elemento físico que dispone de memoria y con frecuencia capacidad de almacenamiento.

Óptimos: Algo que se realiza lo mejor posible

Online: Si está conectado a una red.

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

HTML y scripts. Con el objetivo de crear aplicaciones potentes.

PostgreSQL: Es un Sistema de Gestión de Bases de Datos Objeto-Relacionales (ORDBMS) libre.

P1: Planificación de las asignaturas. Documento donde se organiza la secuencia de las actividades docentes que se realizarán en un semestre, entiéndase por actividades: conferencias, clases prácticas, seminarios, talleres, laboratorios y evaluaciones. El documento mostrará el título del tema a desarrollar y en qué semana del horario docente tendrá lugar cada actividad.

RUP: Rational Unified Process (Proceso Unificado de desarrollo). Metodología para el desarrollo de Software.

Roles: En los ciclos de publicación, es necesaria la definición y manejo de roles como por ejemplo editor, autorizador, publicador, administrador, etc. Un administrador de contenido maneja esta funcionalidad como base para el sistema de flujo de documentos.

Software: Programas de sistema, utilerías o aplicaciones expresados en un lenguaje de máquina.

SQL: Structured Query Language. Es un lenguaje declarativo de acceso a bases de datos que permite especificar diversos tipos de operaciones sobre las mismas. Aúna características del álgebra y el cálculo relacional permitiendo lanzar consultas con el fin de recuperar información de interés de una base de datos.

SGBD: Sistema de Gestión de Bases de Datos. Es el software que permite la utilización y/o la actualización de los datos almacenados en una (o varias) base(s) de datos por uno o varios usuarios desde diferentes puntos de vista y a la vez.

Servidor: Es un computador potente o un software que provee una clase especial de servicio a los software clientes que están corriendo en otros computadores y que lo accedan para realizar una función determinada. Un computador funcionando como servidor puede tener operando varios software servidores para prestar servicios, por ejemplo: servidor de www, servidor de FTP, de Mail, etc.

Software: Palabra en inglés utilizada para indicar a los programas de computadoras, a las aplicaciones.

SGA: Sistema de Gestión Académica.

Server Page (Página Servidora): Las páginas servidoras representan páginas web que tienen código que se ejecuta en el servidor.

Trazabilidad: Término que se utiliza en los procesos industriales o actividades económicas que requieren la certificación de su buena práctica. Por ejemplo en los sistemas de gestión de calidad, de gestión medioambiental y sistemas de control conocidos como cadena de custodia.

TCP/IP: Este protocolo de red. En realidad son dos acrónimos distintos; TCP son las siglas de "*Transmisión Control Protocol*", mientras que IP significa "*Internetwork Protocol*". Fueron propuestos originariamente por ARPA (Defensa Americana) como estándar de comunicaciones para intercomunicar las diversas redes que existían.

UML: Unified Modeling Language. Es una notación estándar para modelar objetos del mundo real como primer paso en el desarrollo de programas orientados a objetos. Es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema de software.

UCI: Universidad de las Ciencias Informáticas.

XML: Extensible Markup Language. Es un lenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium. Orientado principalmente al almacenamiento, procesamiento y transmisión de mensajes.

ANEXOS

Diagrama de clases del análisis: Adicionar afectación.

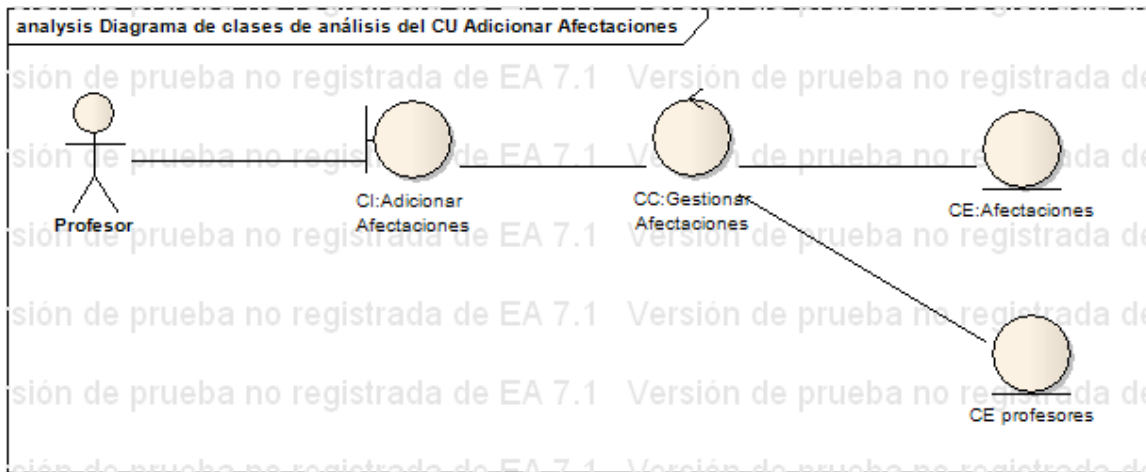


Diagrama de clases del análisis: Eliminar afectación.

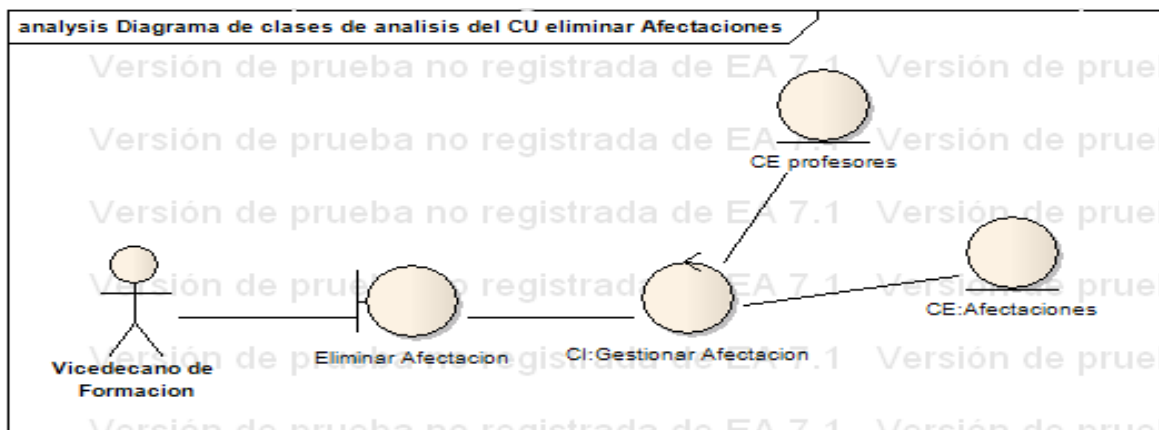


Diagrama de clases del análisis: Modificar afectación.

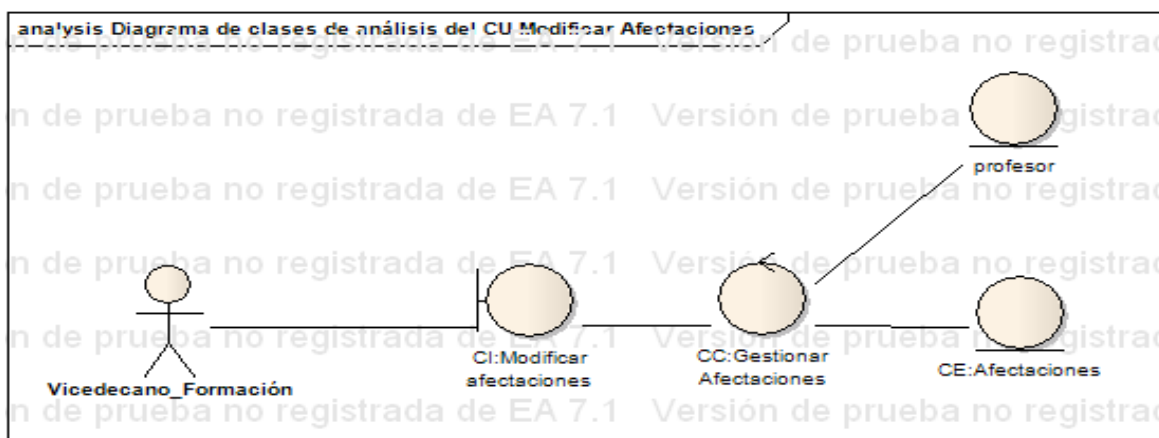


Diagrama de clases del análisis: Adicionar profesor.



Diagrama de clases del análisis: Eliminar profesor

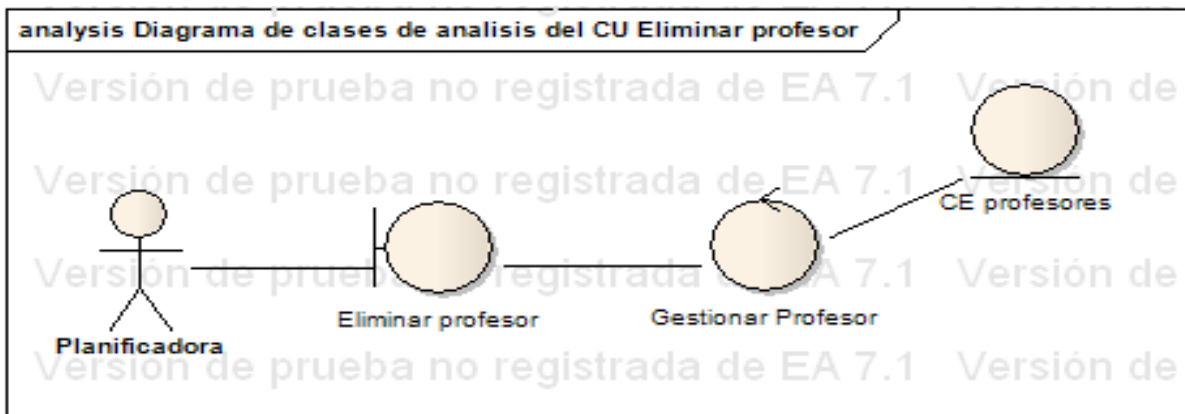


Diagrama de clases del análisis: Modificar profesor.



Diagrama de clases del análisis: Buscar horario.

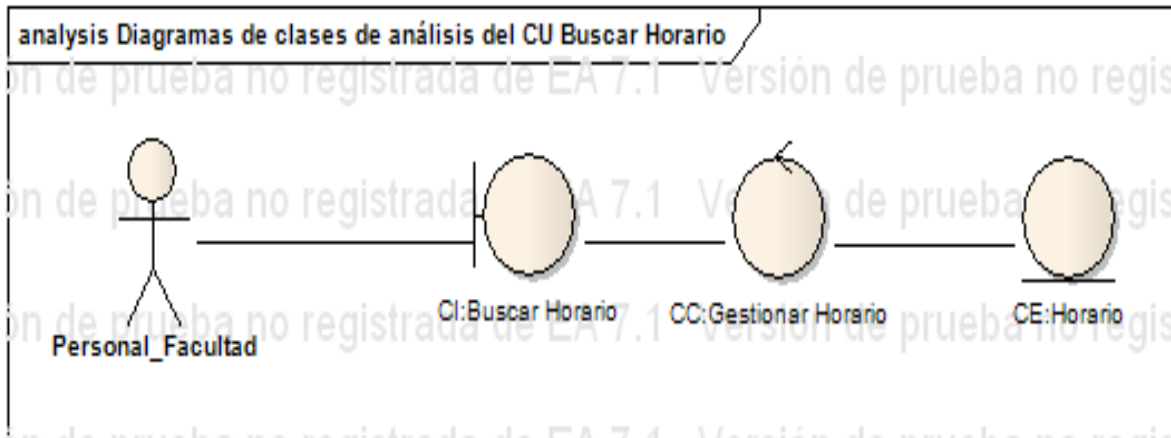


Diagrama de clases del análisis: Confeccionar horario.

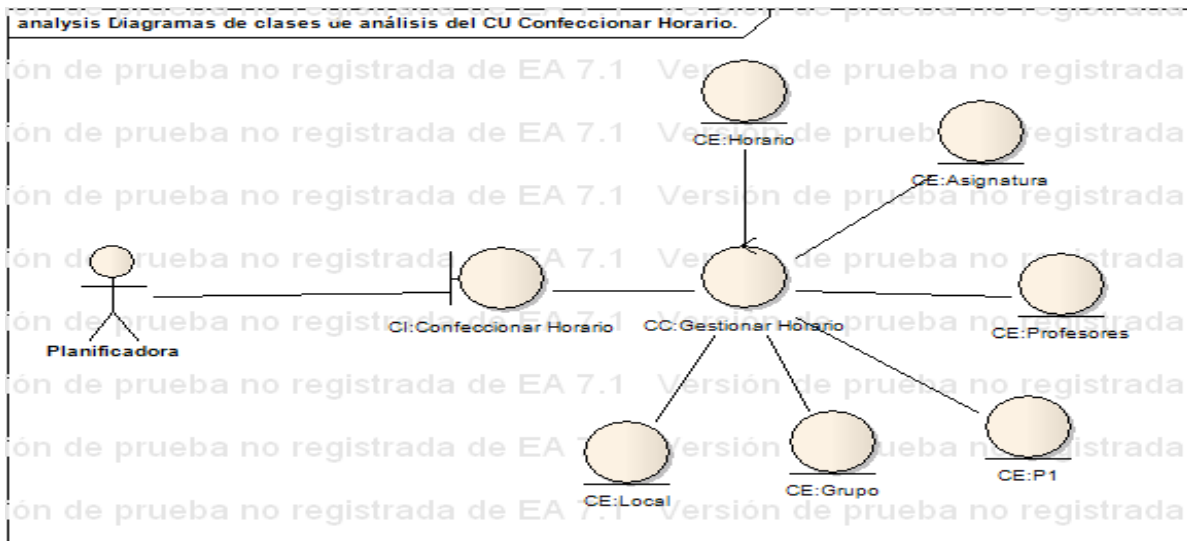


Diagrama de clases del análisis: Consultar horario.

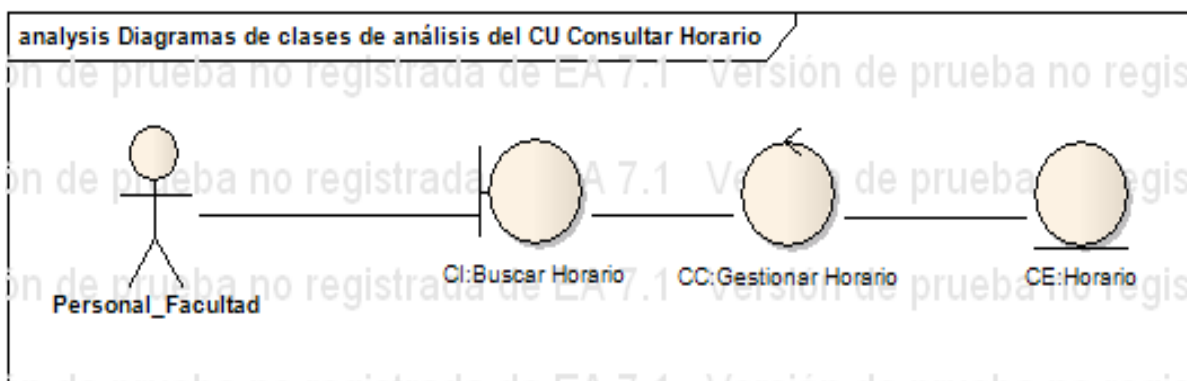


Diagrama de clases del análisis: Modificar horario.

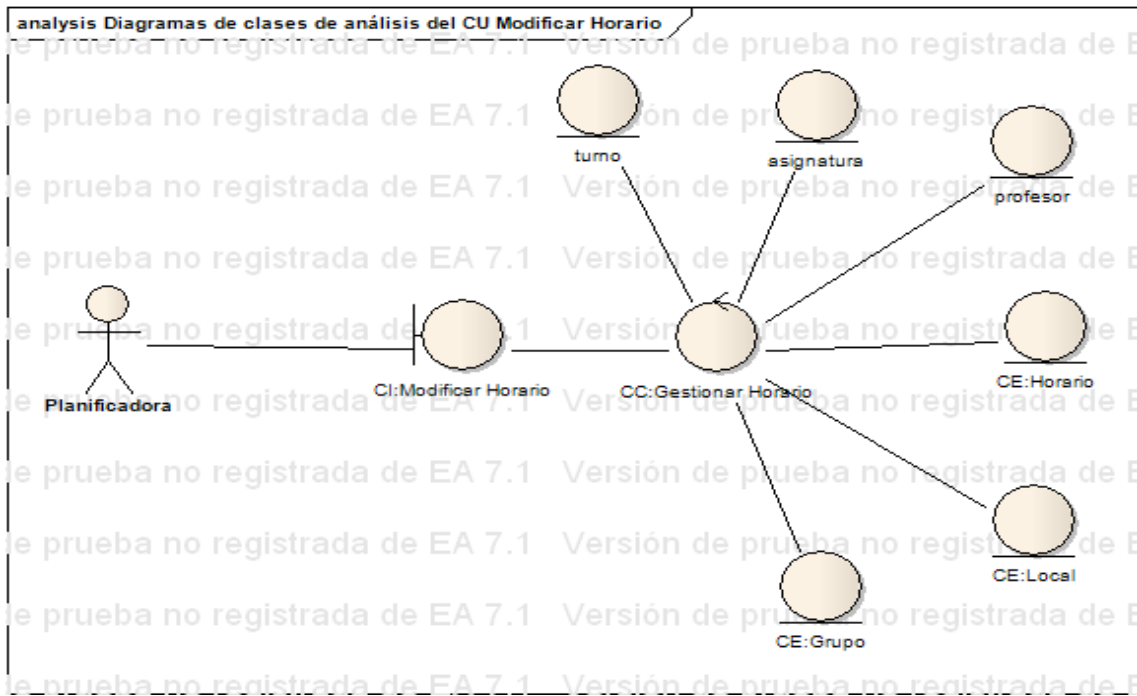


Diagrama de secuencia del análisis del CU: Adicionar local

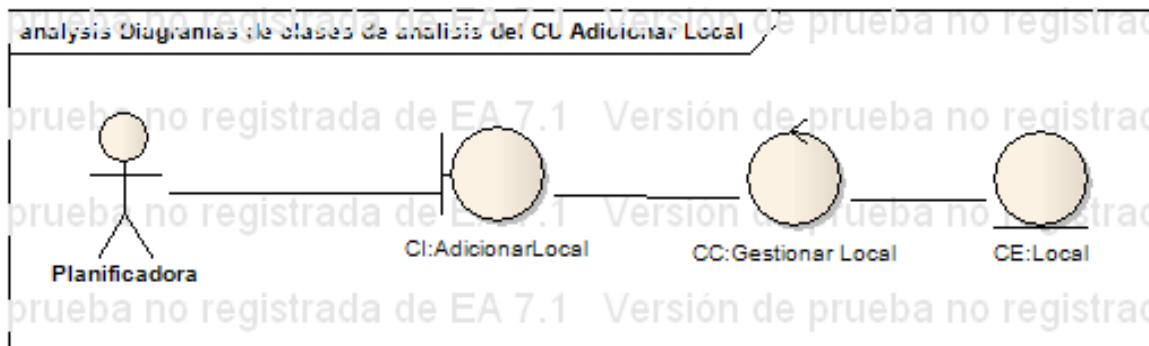


Diagrama de secuencia del análisis del CU: Gestionar Afectación

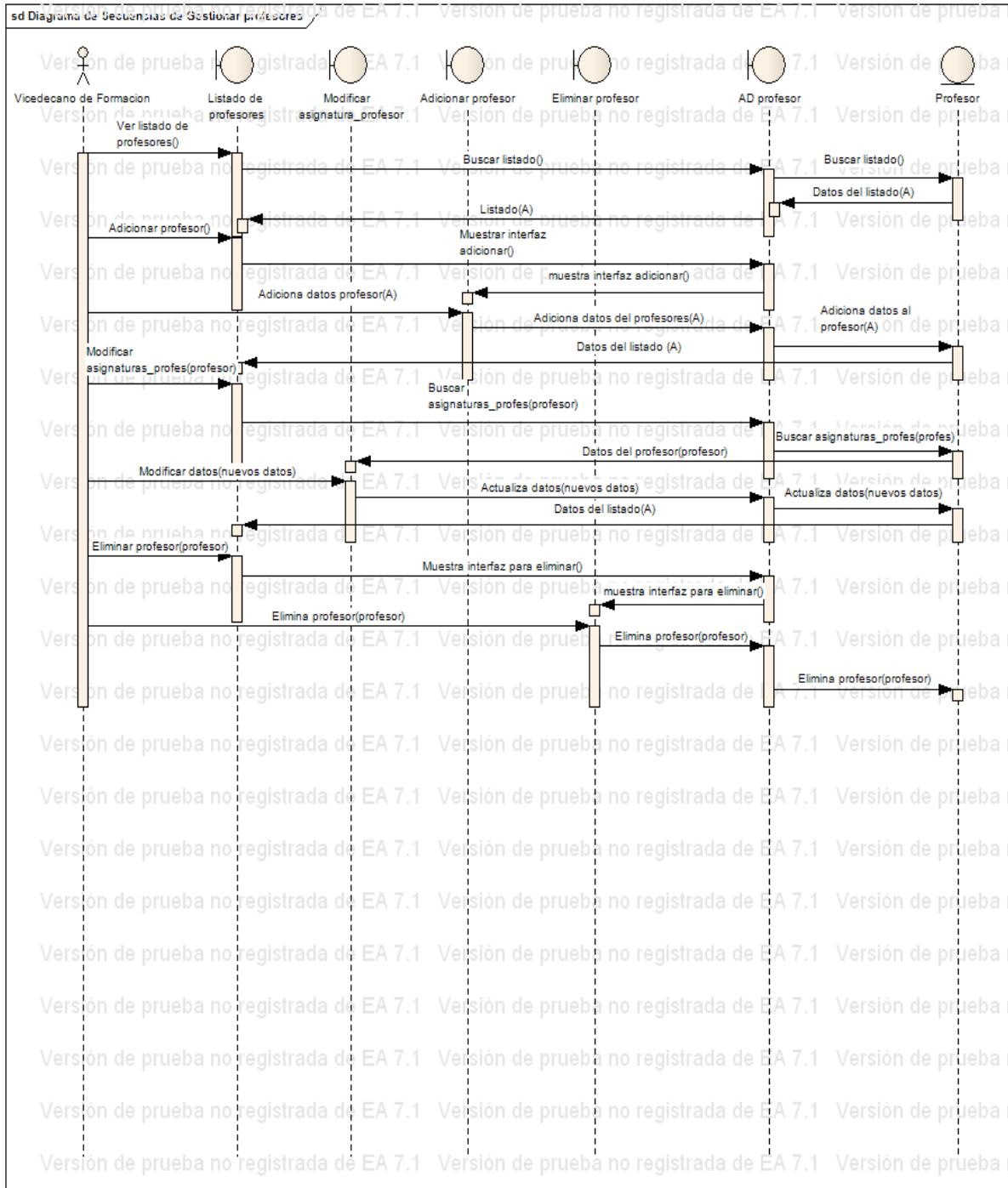
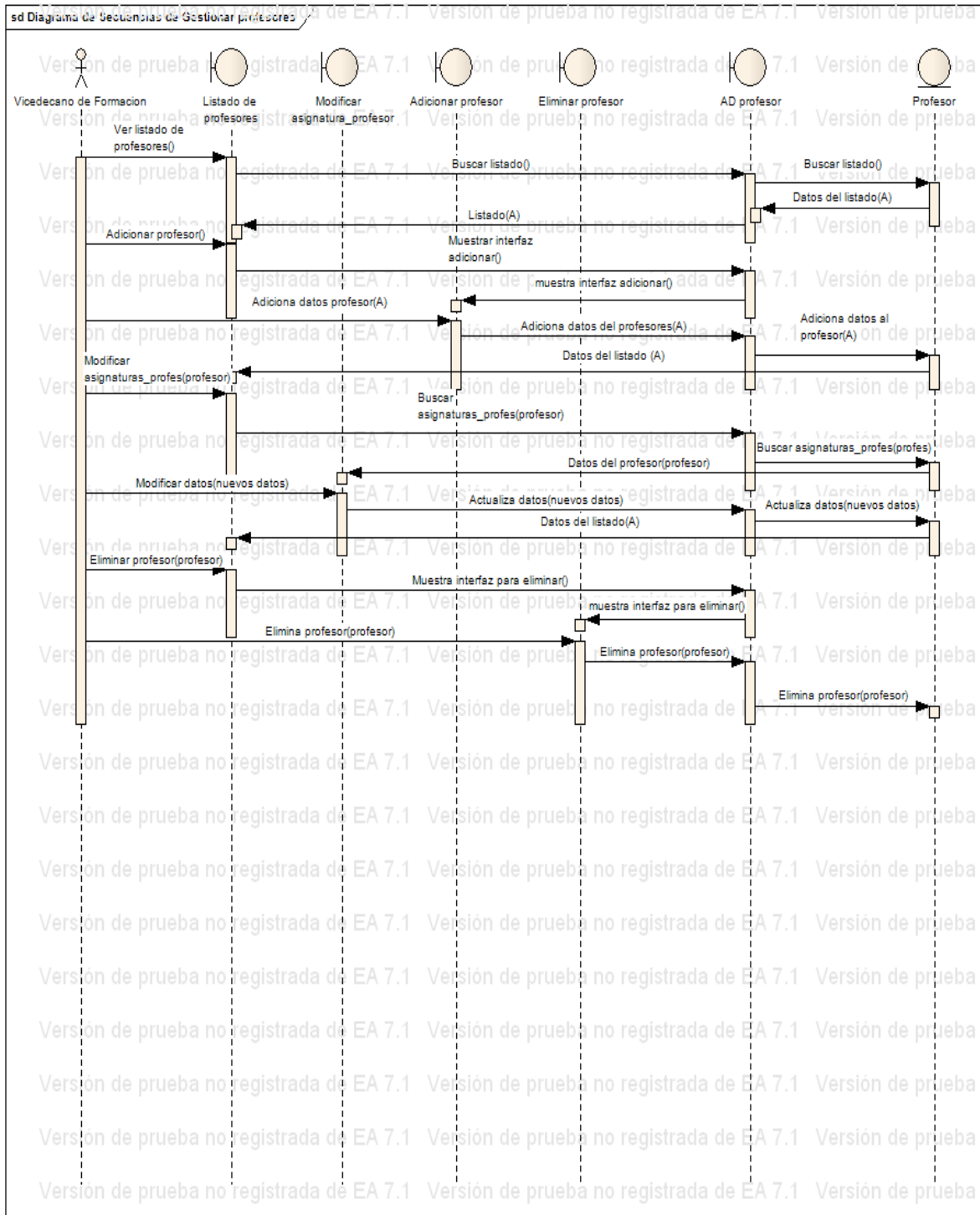
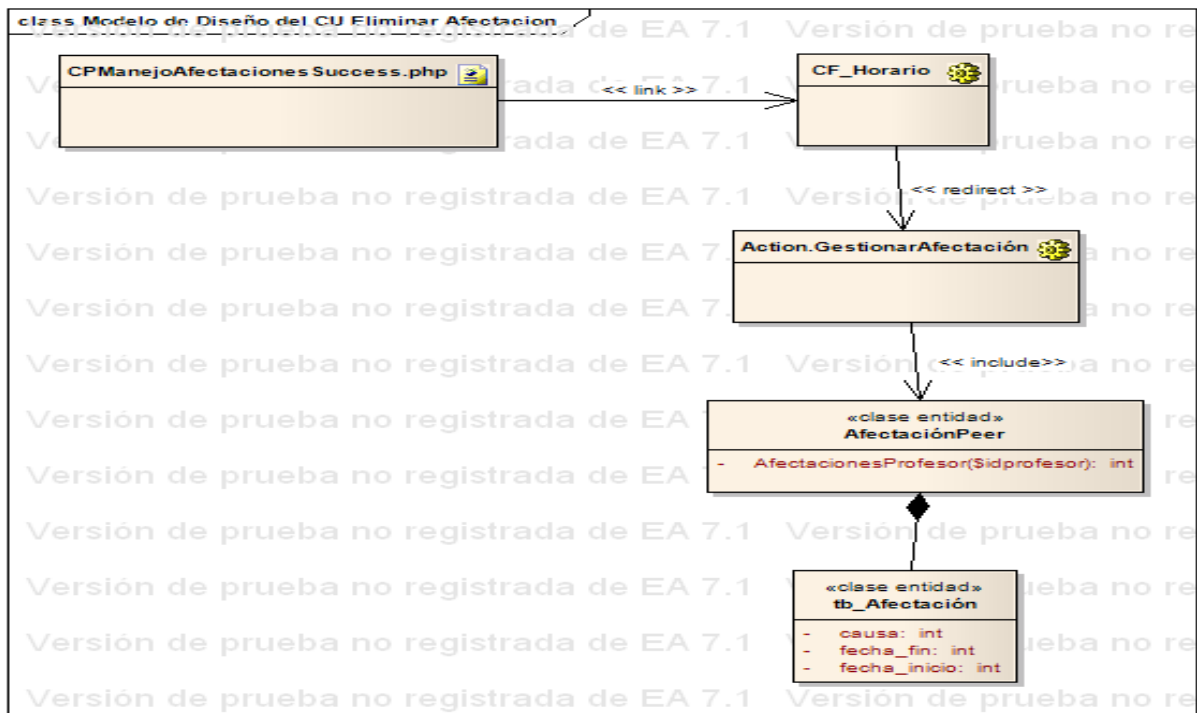


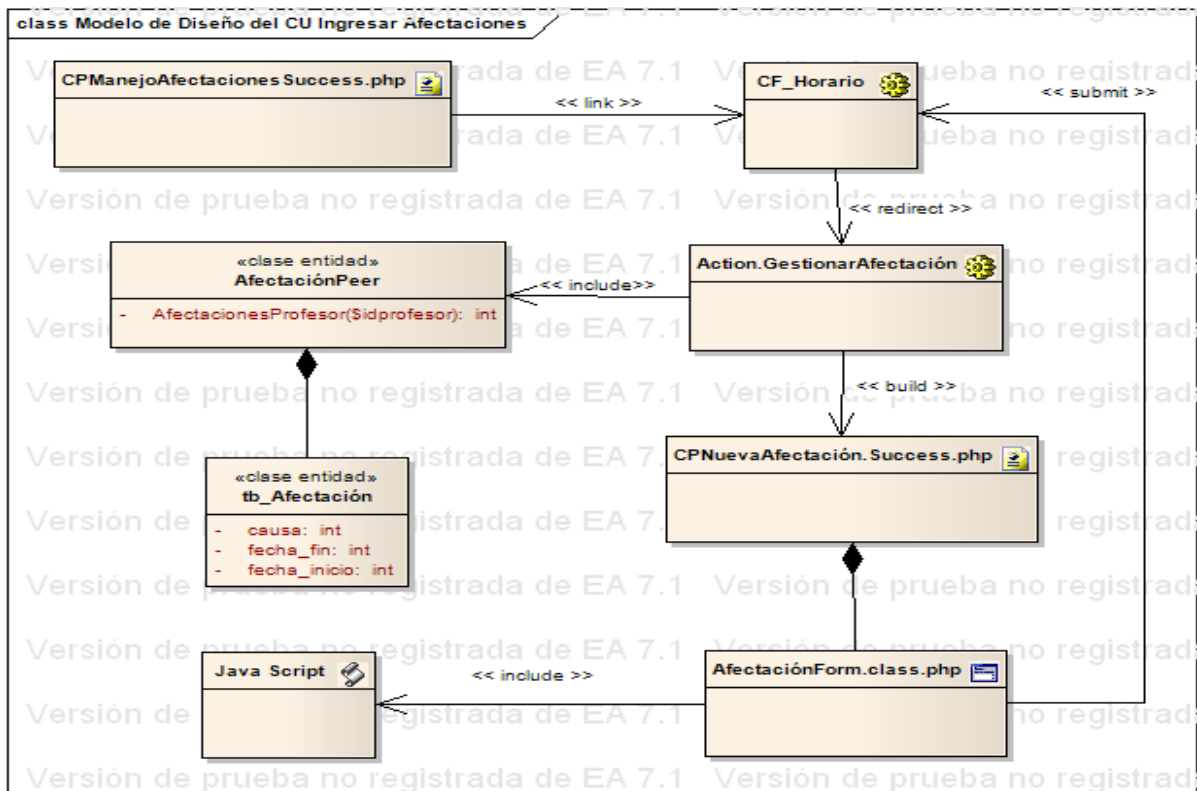
Diagrama de secuencia del análisis del CU: Gestionar Profesores.



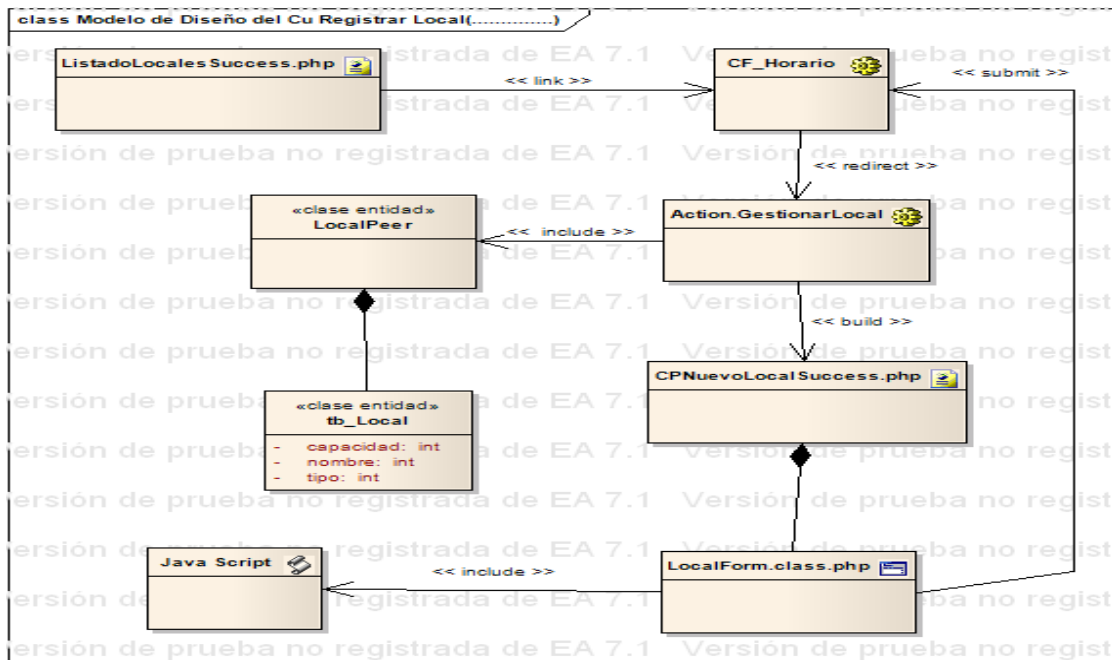
Modelo de clases del diseño del CU Eliminar afectaciones



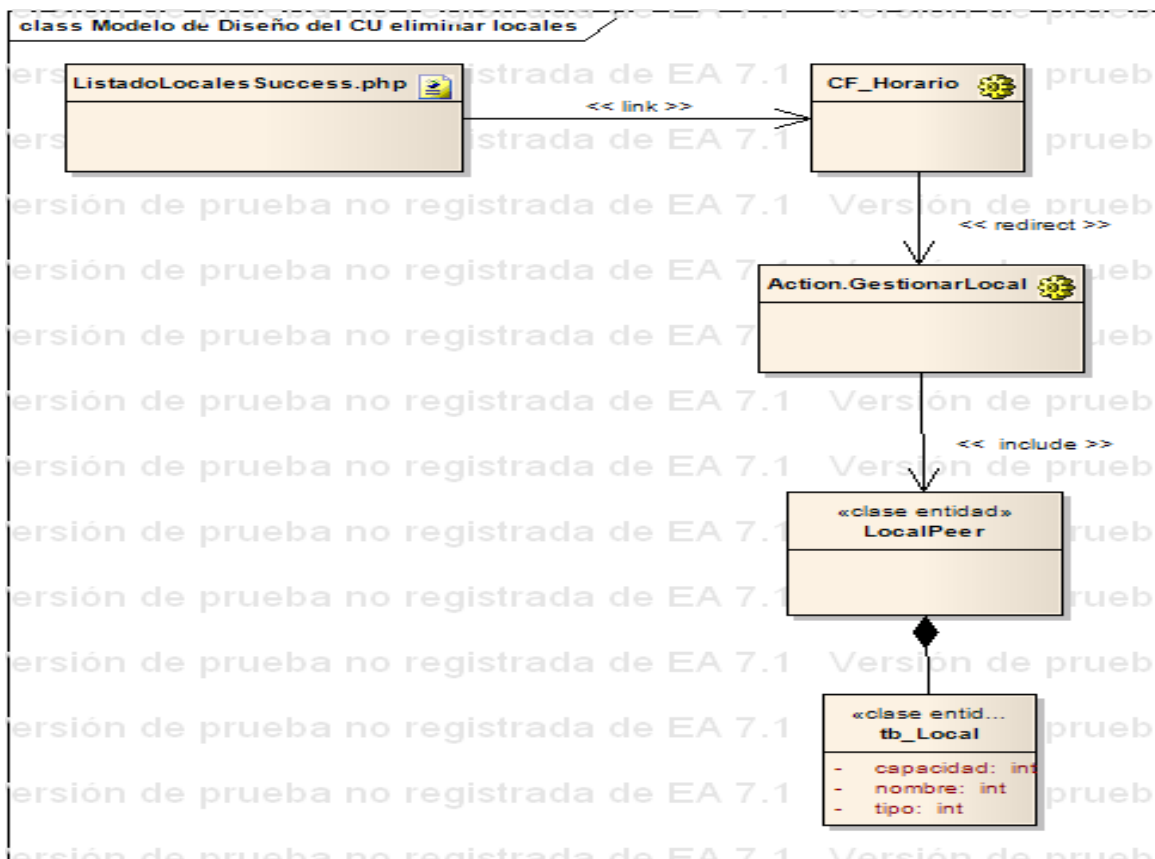
Modelo de clases del diseño del CU Adicionar afectaciones



Modelo de clases del diseño del CU Adicionar local



Modelo de clases del diseño del CU Eliminar local



Modelo de clases del diseño del CU Modificar local

