

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Facultad 3



Título: Desarrollo del sistema *“Academic Schedule”*.

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

Autores: Yudel Rosales Echemendia.

Juan Carlos Pérez García.

Tutor: Lic. Rolan Rober Bullain Dieguez.

Ciudad de La Habana, Junio 2009.

“Año del 50 aniversario del triunfo de la Revolución”



DECLARACIÓN DE AUTORÍA

Declaramos ser únicos autores del presente trabajo y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales del mismo, con carácter exclusivo.

Para que así conste firmo el presente a los ____ días del mes de _____ del año 2009.

Yudel Rosales Echemendia.

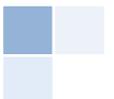
(Autor)

Juan Carlos Pérez García.

(Autor)

Lic. Rolan Rober Bullain Dieguez.

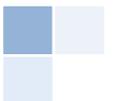
(Tutor)





*“...Todo hombre debe decidir una vez en la vida
si se lanza a triunfar arriesgándolo todo,
o se sienta a contemplar el paso de los triunfadores...”*

Benedetti



**DEDICATORIA Y AGRADECIMIENTOS DE JUAN CARLOS**

Son pocas las palabras que pudieran demostrar los sentimientos que se apoderan del corazón de alguien que se siente agradecido de tantas personas que de una forma u otra han contribuido a la formación de un profesional que sin ustedes la vida no tendría sentido, nadie lo hace todo en la vida por hacer sentir bien a si mismo, siempre existe algo por lo que vale la pena luchar, por lo que vale la pena seguir adelante y por lo que vale la pena vivir, todas esas razones son ustedes. Dedico y agradezco estos 5 años de estudio y sacrificio....

A mi mamá que siempre me ha guiado e iluminado desde donde está, te mereces este título y todo lo bueno de mi corazón, ojalá hubieses podido leer estas líneas que te dedico, te amo.

A mi papá por ser mi confianza, mi amigo, mi guía, mi ejemplo a seguir y apoyarme en todas las decisiones que he tomado en la vida, eres el padre que todo hijo desearía tener, te amo.

A mi abuelita Pipe y a Papi Cuca por haberme sabido guiar y enseñarme a dar los primeros pasos de la vida, gracias a ustedes soy como soy, no existen palabras para expresar lo que significan para mi, mi corazón late por ustedes, son mi vida.

A mi abuela Nena por ser la persona que más me ayudó en toda esta etapa de la universidad, nunca lo olvidaré, gracias por estar siempre pendiente de mis cosas, gracias por tu confianza, te quiero mucho mi vieja y a mi abuelo Guango, te quiero mucho también mi viejito gruñón.

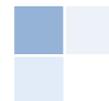
A mama Odafys por saber ser mi mamá, gracias por quererme tanto y ser para ti un hijo, me han pasado muchas cosas buenas en la vida pero ninguna mejor que tú.

A mama Bulys por apoyarme y ayudarme en todo lo que ha podido, gracias por todo lo bueno que me has brindado, te quiero con la vida.

A mi tía Tata, te agradezco el cariño que siempre me has brindado, te quiero muchísimo, eres alguien especial en mi vida.

A mis hermanos el Pepe y el Seca por soportar lo quisquilloso que soy, gracias de todo corazón por su cariño y apoyo, los quiero con mi vida, que no quede la menor duda.

A mis tíos y tías: tío Jorge, tío Gilbe, Ta, tía Lisset, Ernesto, Cristi y Papi Doctor, los quiero a todos por igual, gracias por su ayuda y apoyo todo este tiempo.





A mis primas y primos: Yiny, Kiky, Yayai, Carby, el Dayo y Jorgito, saben que la palabra primos solo significa que son hijos de mis tíos, para mi son como mis hermanos, los quiero a todos con la vida.

A mis primitos: Jorgitico, Claudia, José Carlos, Adan, José Ernesto y la Lore, los quiero mucho, sigan el ejemplo del primo y dedíquenme algún día un título como este.

A mi tutor Rolan, por el apoyo y la ayuda brindada durante el desarrollo de la tesis, gracias por todo, eres un ejemplo a seguir como profesional.

A la gente del Entronque: abuelo José y abuela América, gracias por ayudarme tanto en todo este tiempo, los quiero mucho, a mi madre Amparo, Prieta y Nieves

A mis cuñadas, por saber cuidar de mis hermanos, tontina te quiero mucho, eres la razón del aire que respira Frank, siempre estaré agradecido de eso.

A mi amigo y compañero de tesis el colorao, Yuyo la UCI hubiese sido mucho más difícil sin tu ayuda, gracias por tu confianza mi hermano, únicamente yo te pude soportar estos 5 años.

A mi amigo el Prieto, arrastraste conmigo en el pre y aquí te tocó de nuevo, gracias por ayudarme tanto en la tesis, sin tu ayuda no se que hubiese pasado, te quiero mi hermanito prieto.

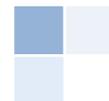
A mi amigo Jairol, sabes que te quiero como un hermano, gracias por todos estos años de fiesta y jodederas.

A mi amigo Flavio, eres otro de mis hermanos, gracias por tu confianza se que siempre seremos amigos.

A ti Rider, la palabra amigo se queda corta para expresar lo que significas para mi, gracias por tu confianza, tus consejos y ánimo, eres mi hermano.

A mi socio Arlan, muchas gracias mi hermano por ayudarme en todo este tiempo y resolverme los problemas siempre que acudía a ti, sobre todo en la tesis claro.

A mi gente de la UCI, esas buenas amistades que supieron quererme durante todo este tiempo, no puedo mencionar tantos nombres porque se me quedarían muchos, Lisi eres especial, tu si vas con todas las letras, gracias por tu confianza y por ese cariño tan sano y lindo, te amo, así como mi socio Gustavo, nadie en la vida me dio tanto chuchio como tú, a mi socio el asqueroso (Yaidel) que fue el último borracho que conocí en la UCI, a mi junta la Pilluela, gracias por escucharme siempre que te necesité, a mis socios Ale, a PPCSI





Ramiro, Tabique, Margelíng, el negro Bladimir, Pescao, a la loca de Yasma y todos los demás del 3103, a Roxi, Mary, Nilodo, Serguey y Dayrien, gracias a todos por su amistad.

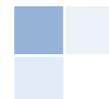
A mi socio y barbero Osnielito, a Oriél y su papá, gracias por su amistad.

A Diana, se que estás lejos pero gracias por tu amistad y confianza, te quiero mucho, a Aidely, eres como mi hermana, te quiero mucho y agradezco tu amistad.

A mi flaca Yurianna que me ayudó mucho cuando fue mi profe, gracias mi vida, te quiero mucho y al piquete de su apto, la Yura y Milena por soportarme y matarme el hambre de vez en cuando, las quiero también.

A Abdiel y al Mode, gracias por ayudarme en la tesis y dedicar parte de su tiempo para poder terminar en tiempo.

A todas esas personas que cuando me veían me preguntaron ¿Cómo va la tesis? Especialmente Ari, Ayeris, Rosaida y la Nena, muchas gracias.



**DEDICATORIA Y AGRADECIMIENTOS DE YUDEL**

Dedico y agradezco a mi abuela MAMA, por su preocupación, apoyo, por quererme tanto y darme lo mejor de ella, por su amor y comprensión, por saber guiarme, eres la razón por lo que valió la pena el sacrificio de estos 5 años, por eso te los dedico a ti, TQ mucho.

A mi mamá por ayudarme, preocuparse y darme su apoyo incondicional a lo largo de toda mi vida y en especial por estos últimos 5 años tan importante para mi, por lo que también esta dedicado a ti.

A toda mi familia, en especial a mis hermanitos Wandé y Dani, que los adoro, a mi papá, a mis tías Juanita, Mariela y Esperanza.

A mi tutor por ayudarme y enseñarme a lo largo de toda mi carrera, gracias por su confianza, y su apoyo, ahí y por darnos el tema de tesis. . . ., muchas gracias.

Le agradezco a todas las amistades que me han dado su ayuda en la realización de esta tesis, que sin la misma hubiese sido muy difícil esta labor, gracias Gerandys, Abdiel, Modesto y Arlan.

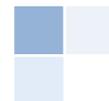
A mi amigo y compañero de tesis TONTIN, por haberme soportado estos 5 años, por estar en los momentos buenos y malos, por ser buen amigo. Gracias por tu amistad, por querer hacer esta tesis conmigo.

A Emilio por ser de gran ayuda y apoyo en estos últimos momentos tan difíciles, muchas gracias por estar ahí.

Le agradezco al TAMBÚ por su gran apoyo en los momentos más difíciles que he tenido en la universidad, por todo lo que ha hecho por mi y por darme su apoyo en todo momento, muchas gracias por estar a mi lado estos 5 años.

Le agradezco a mi amiga Liu, por darme su apoyo y confianza. Porque a pesar de que es tan insoportable como yo se que me quiere mucho al igual yo a ella.

A Lisi por saberse ganar mi cariño, con esa cantidad de tonterías, y por tener una risa tan expresiva, ya saben porque la tiene. Te quiero mucho.





A Jairo! sabes que te considero un gran amigo, gracias por tu apoyo y por ser como eres. Has sido un hermano para mí en estos 5 años.

A las KETCHUP, Malena, Ivonne, Ida y Ana por ser como, por soportarme todo este tiempo y por el apoyo que me han dado, las quiero.

A Pepe por su ayuda en todo este tiempo, por soportarme y por quererme, gracias.

A Dalía por toda su ayuda, gracias por apoyarme y confiar en mí, te aprecio mucho.

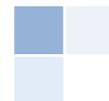
A mis compañeros de grupo, Gustavo, Any, a la loca de Yasma, Margelys, al Blade, Kenny, Yoney, Arianna(Pesqui), Yamilet, muchas gracias a todos.

A mis amigos Rey, Genaro, Ricardo, Luis, gracias por haberme ayudado en todo momento.

A Ernesto, Lázaro y Ronald, gracias por su amistad.

A mi piquete del monte Danilo, Tito, Yaumel, Yaubel, Magaly, gracias por su amistad.

A todas las personas que me han ayudado y me han servido de guía durante estos 5 años tanto en mi vida estudiantil como en lo personal, muchas gracias a todos.

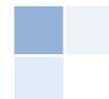




RESUMEN

La gestión de la información se ha convertido en un proceso importante en las instituciones y empresas, debido a que esta permite organizar y poner en uso los recursos de información de la organización, esto ha permitido un progresivo desarrollo tecnológico, que posibilita que las instituciones desarrollen sistemas que le permita gestionar grandes volúmenes de información. En la Facultad 3 de la Universidad de las Ciencias Informáticas (UCI, en lo adelante) los procesos que se realizan en la Planificación Docente no están automatizados, la información que se maneja se hace manualmente, los datos son recogidos y archivados en copia dura. Entiéndase como información que se maneja todo lo referido a las afectaciones de los profesores, a los cambios de turnos que estos solicitan y a la solicitud de locales para la realización de una actividad fuera del horario docente. Hasta el momento tramitar toda la información de estos procesos se hace lenta y engorrosa para el personal encargado de realizar la Planificación Docente. El presente trabajo está centrado en la implementación de un sistema de ayuda para la planificación docente en la Facultad 3, destinado a mejorar la eficiencia del proceso de planificación teniendo en cuenta las características específicas de la misma.

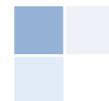
Para ello se estudian otros sistemas existentes de Gestión de la Información Docente en Cuba y a nivel internacional, se realiza un análisis de las tecnologías y lenguajes de programación más utilizados actualmente en Cuba y el mundo para el desarrollo de aplicaciones Web. También se realiza un estudio acerca de conceptos y factores que influyen en la planificación, se llevó a cabo un análisis y selección de herramientas de desarrollo de software a utilizar, quedando finalmente plasmados los resultados en la implementación del sistema y recomendándose algunos aspectos para el mejoramiento futuro de versiones que puedan ser producidas.





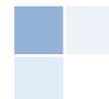
ÍNDICE DE CONTENIDO

Introducción -----	1
Capítulo 1: Fundamentación Teórica. -----	4
1.1. Introducción. -----	4
1.2. La planificación docente. -----	4
Factores que inciden y participan en la planificación docente. -----	4
Sistemas de planificación académica similares. -----	6
1.3. Metodologías para el desarrollo de Software. -----	9
Programación Extrema.-----	10
Microsoft Solution Framework.-----	11
Rational Unified Process.-----	12
1.4. Paradigmas de programación. -----	16
Programación estructurada. -----	16
Programación modular. -----	18
Programación concurrente. -----	18
Programación lógica. -----	19
Programación con Restricciones.-----	19
Programación Orientada a Objetos. -----	21
1.5. Lenguajes. -----	22
Lenguajes de modelado. -----	22
Lenguajes del lado del cliente. -----	23
Lenguajes de programación del lado del servidor. -----	25
1.6. Análisis de herramientas a utilizar en el sistema. -----	26
Herramientas CASE. -----	26
1.7. Tecnologías del lado del cliente. -----	29
1.8. Tecnologías del lado del servidor. -----	29
Sistemas Gestores de Bases de Datos. -----	30
Servidor Web. -----	32
IDE de desarrollo. -----	33
1.9. EXT JS -----	35
1.10. Sistema de Gestión de Contenidos. -----	36
1.11. Conclusiones. -----	39
Capítulo 2: Descripción y Análisis de la Solución Propuesta. -----	41





2.1	Introducción.	41
2.2	Valoración crítica del diseño propuesto por el analista.	41
2.3	Requisitos Funcionales.	44
2.4	Diagrama de Casos de Uso del Sistema.	46
2.5	Descripción de los Casos de Uso Críticos.	47
2.6	Diagramas de Clases del Diseño.	55
2.7	Diagrama de Clases Persistentes.	60
2.8	Modelo de la Base de Datos.	61
2.9	Descripción de las Clases.	62
2.10	Conclusiones.	65
Capítulo 3: Propuesta de Solución.		66
3.1	Introducción.	66
3.2	Modelo de Despliegue.	66
3.3	Modelo de Implementación.	67
3.4	Modelo de Pruebas.	70
3.5	Casos de Prueba de Funcionalidad.	70
3.6	Conclusiones.	75
Conclusiones		76
Recomendaciones		77
Bibliografía.		78
Anexos		82
3.7	Anexo # 1. Descripción de Casos de Uso.	82
3.8	Anexo # 2. Diagramas de Clases del Diseño.	92
3.9	Anexo # 3. Descripción de las Clases a Implementar.	98
3.10	Anexo # 4. Diagramas de Componentes.	104
3.11	Anexo # 5: Casos de Prueba.	107
3.12	Anexo # 6. Carta de aceptación del cliente.	116
Glosario de Términos		117





ÍNDICE DE IMÁGENES

<i>Imagen 1.1: Proceso Unificado de Rational (RUP).</i>	15
<i>Imagen 2.1: Diagrama de Paquetes. (Sixto, y otros, 2008)</i>	41
<i>Imagen 2. 2: Diagrama de Paquetes</i>	43
<i>Imagen 2. 3: Diagrama de Casos de Uso del Sistema</i>	46
<i>Imagen 2. 4: Diagrama de Clases del Diseño del Caso de Uso del Sistema – Gestionar Asignatura</i>	55
<i>Imagen 2. 5: Diagrama de Clases del Diseño del Caso de Uso del Sistema – Gestionar Grupo</i>	56
<i>Imagen 2. 6: Diagrama de Clases del Diseño del Caso de Uso del Sistema – Gestionar Local</i>	57
<i>Imagen 2. 7: Diagrama de Clases del Diseño del Caso de Uso del Sistema – Gestionar P1 Asignatura</i>	58
<i>Imagen 2. 8: Diagrama de Clases del Diseño del Caso de Uso del Sistema – Planificar Horario</i>	59
<i>Imagen 2. 9: Diagrama de Clases Persistentes</i>	60
<i>Imagen 2. 10: Modelo de la Base de Datos</i>	61
<i>Imagen 3. 1: Diagrama de Despliegue. Configuración de nodos de procesamiento en tiempo de ejecución y los componentes que residen en ellos</i>	67
<i>Imagen 3. 2: Diagrama de Componentes del Caso de Uso Gestionar Horario</i>	68
<i>Imagen 3.3: Diagrama de Componentes del Caso de Uso Gestionar Asignatura</i>	69
<i>Imagen 3. 4: Diagrama de Componentes del Caso de Uso Gestionar Grupo</i>	69
<i>Imagen 3. 5: Diagrama de Componentes del Caso de Uso Gestionar P1 Asignatura</i>	69
<i>Imagen 3. 6: Diagrama de Componentes del Caso de Uso Gestionar Local</i>	70

ÍNDICE DE TABLAS

<i>Tabla 2. 1: Descripción del Casos de Uso del Sistema - Gestionar Grupos</i>	48
<i>Tabla 2. 2: Descripción del Casos de Uso del Sistema - Gestionar P1 Asignatura</i>	50
<i>Tabla 2. 3: Descripción del Casos de Uso del Sistema - Gestionar Locales</i>	51
<i>Tabla 2. 4: Descripción del Casos de Uso del Sistema - Gestionar Asignatura</i>	53
<i>Tabla 2. 5: Descripción del Casos de Uso del Sistema - Planificar Horario</i>	54
<i>Tabla 2. 6: Descripción de la Clase Entidad del CMS Joomla JView</i>	63
<i>Tabla 2. 7: Descripción de la Clase Entidad del CMS Joomla JModel</i>	64
<i>Tabla 2. 8: Descripción de la Clase Controladora del CMS Joomla Controller</i>	65
<i>Tabla 3. 1: Descripción del Caso de Prueba del Caso de Uso-Gestionar Asignatura</i>	72

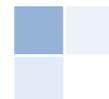
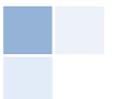




Tabla 3. 2: Descripción del Caso de Prueba del Caso de Uso-Gestionar Grupo..... 72
Tabla 3. 3: Descripción del Caso de Prueba del Caso de Uso-Gestionar P1 Asignatura. 74
Tabla 3. 4: Descripción del Caso de Prueba del Caso de Uso-Gestionar Local..... 75





Introducción

El perfeccionamiento de la actividad educacional, ha demostrado que se requiere de un alto desarrollo de la ciencia y la tecnología en todos los subsistemas de educación. Las universidades no escapan de este contexto, constituyen la mayor fuente de formación de futuros profesionales que cubren todas las esferas de la vida; las cuales hacen uso intensivo de las Tecnologías de la Información y las Comunicaciones (TIC) para enfrentar los problemas de planificación docente, proceso sumamente complejo sobre todo en grandes centros universitarios donde se desarrollan múltiples actividades docentes en un corto plazo de tiempo.

En Cuba existen centros universitarios que realizan la planificación docente de forma manual. La UCI es en nuestro país la institución docente que más tecnología avanzada posee, combinándola con el proceso de enseñanza–aprendizaje de los alumnos y profesores del centro, en la cual la docencia y la producción se encuentran estrechamente relacionadas generando un flujo de actividades muy elevado, afectando directamente el proceso de planificación docente.

La UCI está compuesta por 10 facultades y en la facultad 3, que no está excluida de esta situación, el proceso de planificación del horario docente es realizado de forma manual variando semanalmente, lo cual además de provocar un exceso de trabajo para el planificador, afecta el proceso de actualización, calidad, distribución de locales, planificación de turnos de clases y el flujo de información que debe existir para con el personal docente.

En (Sixto, y otros, 2008) se realizó el Análisis y Diseño del sistema “Academic Schedule”, que pretende solucionar en gran medida los problemas de planificación docente de la facultad 3.

Debido a las razones que anteriormente se explican se hace necesaria la implementación de un sistema que garantice un mejor flujo de la información con relación a la planificación docente de la facultad 3 basándonos en el Análisis y Diseño previamente realizado de este sistema (Sixto, y otros, 2008). De esta forma se puede definir que la **situación problemática** gira en torno a que el mecanismo actualmente existente no permite agilizar los procesos de planificación docente en la facultad, al mismo tiempo es una manera ineficiente de manejar la distribución y actualización de los locales docentes, representa una





sobrecarga de trabajo para el planificador que no esté especializado en esta tarea; así como gastos de materiales de oficina y de tiempo útil de trabajo de la dirección de la facultad en lo que a docencia se refiere. Para resolver esta problemática no existe actualmente una solución informática que haga viable la gestión de estos procesos en la facultad permitiendo el intercambio de información con el personal docente en general.

Ante la situación problemática existente, se define el siguiente **problema**: ¿Cómo desarrollar un sistema de ayuda para la planificación docente de la facultad 3, que reduzca los errores y agilice el proceso de planificación? Por consiguiente el **objeto de estudio** se centra en el proceso de desarrollo de software. El **campo de acción** está comprendido en la implementación como fase específica dentro del proceso de desarrollo de software de una aplicación que permita la gestión de la planificación docente en la facultad 3.

Para darle solución al problema planteado se define como **objetivo general**: implementar un sistema informático que permita la gestión del proceso de planificación docente en la facultad 3.

Cuyos **objetivos específicos** son:

- Realizar el estado del arte de la planificación académica y los paradigmas de programación.
- Realizar un estudio del Análisis y Diseño de la solución a la cual se le dará continuidad.
- Realizar el modelo de implementación.
- Implementar los componentes del diseño.
- Probar los componentes implementados.

Para un correcto desempeño de este trabajo se definen las siguientes **tareas**:

- Definición de conceptos referentes a la planificación académica.
- Estudio de las metodologías y herramientas a utilizar en la solución.
- Estudio de los flujos de trabajo que propone RUP en cada una de sus fases.
- Documentación de los cambios realizados a la propuesta definida.
- Elaboración del diagrama de despliegue de la solución.
- Elaboración de los diagramas de componentes.





- Elaboración de los diseños de casos de prueba de funcionalidad.
- Ejecución de las pruebas de funcionalidad.

Como **idea a defender** se plantea que con la implementación del sistema “Academic Schedule” se logrará una mayor eficiencia y rapidez en la planificación e intercambio de información con el personal docente de la Facultad 3.

A partir de los objetivos planteados se espera que el sistema permita al personal docente conocer la disponibilidad de locales, profesores con sus afectaciones y datos de la facultad con respecto a la docencia y la detección de conflictos en la planificación de los horarios docentes.

En aras de una mejor comprensión del presente trabajo de diploma se decidió estructurar el mismo en tres capítulos de los cuales se hará un resumen a continuación: “*Capítulo 1: Fundamentación Teórica*”, se abordan los principales conceptos y tendencias referentes a la planificación docente en el ámbito internacional. Se realiza el estado del arte de las posibles metodologías, paradigmas de programación, herramientas y tecnologías a utilizar en esta solución informática, seleccionando las más apropiadas y argumentando el por qué de esta selección. “*Capítulo 2: Descripción y análisis de la solución propuesta*”, se hace una valoración crítica del diseño propuesto por el analista. Análisis de posibles implementaciones, componentes o módulos ya existentes que puedan ser rehusados y la descripción de las nuevas clases u operaciones. “*Capítulo 3: Propuestas de Solución*”, se realiza el modelo de implementación, con su respectivo diagrama de despliegue y diagramas de componentes, así como el diseño y aplicación de las pruebas de funcionalidad.





Capítulo 1: Fundamentación Teórica.

1.1. *Introducción.*

En este capítulo se definen los principales conceptos relacionados con el tema a desarrollar, se abordará sobre la existencia de sistemas similares en el ámbito internacional y nacional, de las metodologías de desarrollo de software, características de los paradigmas de programación, herramientas y tecnologías para el desarrollo de este tipo de aplicación.

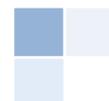
1.2. *La planificación docente.*

La planificación docente organiza el proceso de enseñanza-aprendizaje para que este se realice con el mínimo esfuerzo y la máxima eficacia, implica un proceso consciente de estudio y selección del mejor curso de acción a seguir, estableciendo prioridades frente a una variedad de alternativas posibles y factibles de acuerdo a los recursos con que se cuenta; es la organización coherente y funcional de un conjunto de actividades destinadas al eficaz aprendizaje de los estudiantes tratando de lograr el óptimo uso de los recursos disponibles.

Factores que inciden y participan en la planificación docente.

Se menciona el tema de la planificación docente ideándola como la primera función administrativa que se emplea como base para otras funciones, determinando por anticipado cuáles son los objetivos que deben cumplirse y qué debe hacerse para lograrlos.

La planificación docente comienza por implantar los objetivos y detallar los planes necesarios para llegar a ellos de la mejor forma. Determina a dónde se pretende llegar, qué debe hacerse, cómo, cuándo y en qué orden deben suceder los acontecimientos. Luego de un estudio realizado, se concibe dicha actividad como un proceso complejo, pues se debe tener en cuenta muchos aspectos de singular importancia referentes a la actividad docente, es mediante esta donde se logra la asimilación de conocimientos científicos y la formación de habilidades correspondientes, objetivo y resultado según Maria Onelia Chiang Molina en su





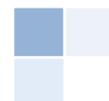
libro "Higiene de la actividad docente", la cual es una forma de actividad cognoscitiva dirigida mediante el proceso de enseñanza en la escuela, regida por un conjunto de características según (Molina, 1995), entre las que se encuentran:

- ❖ Basada en contenidos previamente determinados en el plan de estudio de la carrera y en programas establecidos.
- ❖ Puede hacerse por bloques lectivos, ciclos o niveles, en dependencia de la carrera, el curso o el nivel de enseñanza.
- ❖ Planificación adecuada de medios y materiales de enseñanza, así como de volumen de la información.

Unido a esto debe considerarse la organización de la actividad docente desde el punto de vista higiénico, la cual está muy relacionada con uno de los principios más importantes de la organización científica del trabajo: el principio de la optimización, el cual exige la selección de una variante óptima de la actividad para evitar trastornos a los estudiantes. Esta organización supone:

- ❖ La existencia de un balance de carga docente normalizado (equilibrio de clases para un grupo de estudiantes en un período).
- ❖ Equilibrio de los tipos de actividades docentes, pues está comprobado que muchas actividades evaluativas alteran los biorritmos de aprovechamiento académico, de alimentación, sueño y descanso, llegando a ocasionar alteraciones en el calibre de los vasos sanguíneos y en la respiración.

“Planificar la enseñanza significa tomar en consideración las determinaciones legales (los descriptores), tomar en consideración los contenidos básicos de nuestra disciplina, tomar en consideración el marco curricular en que se ubica la disciplina (en qué plan de estudio, en relación a qué perfil profesional, en qué curso, con qué duración), tomar en consideración nuestra propia visión de la disciplina y de su didáctica (nuestra experiencia docente y nuestro estilo personal), tomar en consideración las características de nuestros alumnos (su número, su preparación anterior, sus posibles intereses) y tomar en consideración los recursos disponibles” (Zabalza, 2003)





Sistemas de planificación académica similares.

La dinámica en el proceso docente-educativo de las universidades exige cada día una planificación docente que se ajuste a las exigencias de las mismas, por lo que se ha incentivado el desarrollo de sistemas de planificación en todo el mundo cada vez más eficientes e “inteligentes”, entre los que se pueden destacar trabajos como:

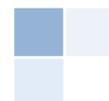
Ámbito Internacional.

La Universidad de Chile cuenta con un Sistema de Información de Gestión Académica (SIGA), permitiendo la constatación y medición de procesos y acciones que realiza la Universidad. SIGA permite recopilar y tratar la información de los distintos niveles institucionales, lo que relaciona la labor docente, investigativa, de creación artística y extensión, y permite conocer cuantitativa y cualitativamente las iniciativas desarrolladas por los distintos organismos de la institución, además de generar indicadores y reportes de Actividad Académica, de Investigación y Extensión. (SIGA, 2008)

Este software no se ajusta a las necesidades que tiene la facultad 3 respecto a la planificación docente, siendo el mismo de carácter propietario imposibilitando su adquisición.

UniTime 3.1 es un sistema automatizado para la construcción de horarios de los estudiantes; sistema distribuido el cual permite que varios departamentos universitarios puedan modificar el programa de tal forma que responda a las necesidades de la organización, minimizando los conflictos que puedan existir durante el curso académico. Además cuenta con la capacidad de gestión de la entrega de un paquete completo que cubre todas las necesidades de horarios de una universidad o cualquier otra institución educativa. Cuenta con un total de 9 000 clases, 570 habitaciones, 2 400 exámenes y 39 000 estudiantes, lo cual provoca una demanda de 190 000 solicitudes de curso. (2008)

Dicho sistema está preparado para generar horarios automáticamente, de cierta forma se ajusta a las necesidades de la facultad en cuanto a la planificación docente, pero está creado para gestionar un cúmulo de informaciones muy elevado, característica que se convierte en una debilidad al ser ajustado a las





necesidades de la facultad 3 en cuanto a la planificación docente, impidiendo su uso como solución a los problemas existentes en la misma.

Existe además un Sistema de Administración Académica (SAAC), realizado por el Centro de Servicios Informáticos (CSI) de la Escuela Superior Politécnica del Litoral (ESPOL), es una aplicación que automatiza todo el proceso de gestión académica de un instituto de educación mediante ágiles opciones que permiten registrar y compartir información de estudiantes, profesores, cursos, aulas y la planificación de cada periodo en ejecución. (CSI, 2008)

Este sistema está diseñado para uso exclusivo de la ESPOL, siendo un software propietario lo cual impide su uso para responder a las necesidades que existen en la facultad 3.

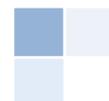
Ámbito Nacional.

En el caso de Cuba no se ha registrado la existencia de sistemas de gran magnitud que respondan a los problemas de planificación docente, a pesar de contar con un elevado por ciento de jóvenes realizando estudios universitarios en las decenas de institutos de educación superior con que cuenta el país, pero si existen sistemas que de alguna forma están inmersos en la importante tarea de la gestión y planificación académica como los casos que se muestran a continuación.

Sistema Automatizado de Información Docente (SAIDO).

Fue uno de los primeros sistemas diseñados e implantados por el Centro de Cibernética Aplicada a la Medicina (CECAM). A partir de su explotación se han incorporado nuevos módulos y subsistemas y se han elaborado otros sistemas relacionados con él, de acuerdo a las necesidades de los usuarios y al avance de las tecnologías computacionales. **SAIDO** cuenta con paquetes de sistemas de Información Docente específicamente el paquete **SAIDO-F**: (Sistema automatizado de información docente, facultades). Es un sistema de base de datos que contiene y procesa información sobre los estudiantes. Incluye módulos de matrícula, actualización y notas.

Dicho sistema solo contiene información acerca de los estudiantes, por lo que es imposible utilizarlo como solución puesto que no cumple con los requisitos que se deben cumplir.





Sistema para la Gestión Académica (GestAcad).

GestAcad fue creado en la Facultad de Informática de la Universidad de Matanzas “Camilo Cienfuegos”. El sistema desde su creación ha dado lugar a la aparición de dos versiones. La primera donde la concepción fue la creación de módulos Stand Alone para la actualización de los datos y un sitio Web destinado a mostrar reportes de la información almacenada. La segunda, ya en explotación en la Universidad de Matanzas, está concebida para desprenderse de aquellas dependencias de los módulos programados en un lenguaje propietario y la migración total hacia plataformas de Software Libre. Esta versión está completamente basada en plataforma Web y está totalmente programada en PHP. Ya incluye los módulos correspondientes de administración, de secretaría y de matrícula.

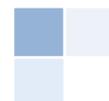
En la actualidad, según (DELGADO), el sistema es capaz de realizar diversas acciones y brindar numerosos reportes como son:

- ❖ Listado de estudiantes por grupo.
- ❖ Reportes dinámicos de la información existente.
- ❖ Reporte de notas por asignatura y grupo.
- ❖ Tabla con los resultados docentes de un grupo en un semestre.
- ❖ Reporte de los resultados académicos de un estudiante en toda su carrera.
- ❖ Actas de exámenes de las diferentes asignaturas.
- ❖ Dar baja a un estudiante.

Dicho software no cuenta con un módulo que se encargue de la planificación del horario docente, característica que lo hace inútil como solución a los problemas de la facultad 3.

AKADEMOS.

Es el sistema de gestión académica de la UCI, pone a disposición de los alumnos y profesores un nuevo módulo desarrollado a partir de la necesidad de conocer las calificaciones obtenidas por un estudiante durante su carrera.





Este sistema permite gestionar la información de los estudiantes, está dividido por módulos y define niveles de acceso de tal manera que cada cual pueda visualizar solo aquella información relacionada con su función; cuenta con numerosas funcionalidades que permiten la gestión de la información, se almacenan los datos por facultad de los resultados obtenidos por los estudiantes en las pruebas realizadas, la matrícula con que se cuenta, asistencia a clases, las causas que describen la situación del ausente, los resultados obtenidos en los cursos optativos y la evaluación de los estudiantes en los diferentes cortes; permitiendo así mantener informado a los estudiantes sobre su desempeño académico y a los profesores sobre los resultados de sus estudiantes.

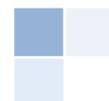
Akademos no cuenta con funcionalidades que gestionen la planificación de los horarios docentes de ninguna facultad de la UCI, razón por la que no es utilizada como solución de los problemas existentes en la facultad 3.

En esta solución informática no se pretende implementar algoritmos que den solución a la generación del horario, sino que sirva de ayuda al planificador al realizar esta tarea, minimizando tiempo, recursos, ganando en calidad y que permita mejorar el flujo de información para con el personal docente.

1.3. Metodologías para el desarrollo de Software.

Para el desarrollo de un proyecto de software la metodología define Quién debe hacer Qué, Cuándo y Cómo debe hacerlo, por lo que es de suma importancia una selección adecuada de la metodología a utilizar. En la actualidad a nivel mundial no existe una metodología estándar para la producción de software. Entre las metodologías más usadas se encuentran: Extreme Programming (XP, por sus siglas en inglés), Microsoft Solution Framework (MSF, por sus siglas en inglés) y Rational Unified Process (RUP, por sus siglas en inglés).

Algunas de estas metodologías ya fueron abordadas en el Análisis y Diseño previamente realizado al sistema en cuestión en (Sixto, y otros, 2008), puesto que al mismo se le está dando continuidad; aun así, es válido tratar estas metodologías para lograr una mayor documentación y adquirir suficientes conocimientos acerca de las mismas.





Programación Extrema.

La Programación Extrema es una metodología ligera de desarrollo de software que se basa en la simplicidad al desarrollar y codificar los módulos del sistema, la comunicación entre los usuarios y los desarrolladores y la realimentación o reutilización del código desarrollado. Extreme Programming (XP) se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. (RUMBAUGH, y otros, 2000)

La principal suposición que se realiza en XP es la posibilidad de disminuir la mítica curva exponencial del costo del cambio a lo largo del proyecto, lo suficiente para que el diseño evolutivo funcione. XP apuesta por un crecimiento lento del costo del cambio y con un comportamiento asintótico. Esto se consigue gracias a las tecnologías disponibles para ayudar en el desarrollo de software y a la aplicación disciplinada de las prácticas que se describen a continuación.

❖ El juego de la planificación

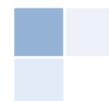
Es un espacio frecuente de comunicación entre el cliente y los programadores. El equipo técnico realiza una estimación del esfuerzo requerido para la implementación de las historias de usuario y los clientes deciden sobre el ámbito y tiempo de las entregas y de cada iteración.

❖ Metáfora

En XP no se enfatiza la definición temprana de una arquitectura estable para el sistema. Dicha arquitectura se asume evolutiva y los posibles inconvenientes que se generarían por no contar con ella explícitamente en el comienzo del proyecto se solventan con la existencia de una metáfora. El sistema es definido mediante una metáfora o un conjunto de metáforas compartidas por el cliente y el equipo de desarrollo. Una metáfora es una historia compartida que describe cómo debería funcionar el sistema.

❖ Pruebas

La producción de código está dirigida por las pruebas unitarias; estas son establecidas antes de escribir el código y son ejecutadas constantemente ante cada modificación del sistema.





❖ Programación en parejas

Toda la producción de código debe realizarse con trabajo en parejas de programadores. Según Cockburn y Williams en un estudio realizado para identificar los costos y beneficios de la programación en parejas, las principales ventajas de introducir este estilo de programación son: muchos errores son detectados conforme son introducidos en el código (inspecciones de código continuas), por consiguiente la tasa de errores del producto final es más baja, los diseños son mejores y el tamaño del código menor (continua discusión de ideas de los programadores), los problemas de programación se resuelven más rápido y posibilita la transferencia de conocimientos de programación entre los miembros del equipo. (Cockburn, y otros, 2000)

❖ 40 horas por semana

Se debe trabajar un máximo de 40 horas por semana. No se trabajan horas extras en dos semanas seguidas. Si esto ocurre, probablemente está ocurriendo un problema que debe corregirse. El trabajo extra desmotiva al equipo.

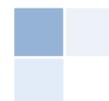
❖ Cliente in-situ

El cliente tiene que estar presente y disponible todo el tiempo para el equipo. Gran parte del éxito del proyecto XP se debe a que es el cliente quien conduce constantemente el trabajo hacia lo que aportará mayor valor de negocio y los programadores pueden resolver de manera inmediata cualquier duda asociada.

Esta metodología no será usada debido a que está diseñada para proyectos muy cambiantes y con grandes posibilidades a dificultades técnicas, característica que no presenta el sistema en cuestión.

Microsoft Solution Framework.

MSF es un compendio de las mejores prácticas en cuanto a administración de proyectos se refiere. Más que una metodología rígida de administración de proyectos, MSF es una serie de modelos que puede adaptarse a cualquier proyecto de tecnología de información. (GPI, 2006)





Es una metodología que proporciona un sistema de modelos, principios y pautas para dar soluciones a empresas que diseñan y desarrollan de una manera que se asegure de que todos los elementos de un proyecto, tales como gente, procesos, y herramientas, puedan ser manejados con éxito.

El modelo de proceso Microsoft Solution Framework (MSF, por sus siglas en inglés) combina los mejores principios del modelo en cascada y del modelo en espiral. Combina la claridad que planea el modelo en cascada y las ventajas de los puntos de transición del modelo en espiral.

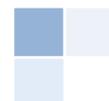
La principal desventaja es que se torna un trabajo bastante largo, ya que para cada fase se debe documentar profundamente todo lo que se realice, lo cual nos haría perder mucho tiempo en documentación que puede ser empleado en el desarrollo de la aplicación por lo que esta metodología no será usada en el desarrollo del sistema.

Rational Unified Process.

El Proceso Unificado de Rational (RUP, por sus siglas en inglés) es un proceso de ingeniería del software. Proporciona un acercamiento disciplinado a la asignación de tareas y responsabilidades en una organización de desarrollo. Su propósito es asegurar la producción de software de alta calidad que se ajuste a las necesidades de sus usuarios finales con unos costos y calendario predecibles.

RUP es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas, diferentes áreas de aplicación, tipos de organizaciones, niveles de aptitud y tamaños de proyecto. (RUMBAUGH, y otros, 2000)

En definitiva RUP es una metodología de desarrollo de software que intenta integrar todos los aspectos a tener en cuenta durante todo el ciclo de vida del software, con el objetivo de hacer abarcables tanto pequeños como grandes proyectos software. Además Rational proporciona herramientas para todos los pasos del desarrollo así como documentación en línea para sus clientes. Las características principales de RUP son:





❖ Guiado/Manejado por casos de uso

La razón de ser de un sistema es servir a usuarios ya sean humanos u otros sistemas; un caso de uso es una facilidad que el software debe proveer a sus usuarios. Los casos de uso reemplazan la antigua especificación funcional tradicional y constituyen la guía fundamental establecida para las actividades a realizar durante todo el proceso de desarrollo incluyendo el diseño, la implementación y las pruebas del sistema.

❖ Centrado en arquitectura

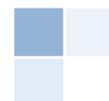
La arquitectura involucra los elementos más significativos del sistema y está influenciada entre otros por plataformas software, sistemas operativos, manejadores de bases de datos, protocolos, consideraciones de desarrollo como sistemas heredados y requerimientos no funcionales. Es como una radiografía del sistema que se está desarrollando, lo suficientemente completa como para que todos los implicados en el desarrollo tengan una idea clara de qué es lo que están construyendo.

❖ Iterativo e Incremental

Para hacer más manejable un proyecto se recomienda dividirlo en ciclos. Para cada ciclo se establecen fases de referencia, cada una de las cuales debe ser considerada como un mini proyecto cuyo núcleo fundamental está constituido por una o más iteraciones de las actividades principales básicas de cualquier proceso de desarrollo. En concreto RUP divide el proceso en cuatro fases, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor hincapié en las distintas actividades. Véase en (*Imagen 1.1*).

❖ Desarrollo basado en componentes

La creación de sistemas intensivos en software requiere dividir el sistema en componentes con interfaces bien definidas, que posteriormente serán ensamblados para generar el sistema. Esta característica en un proceso de desarrollo permite que el sistema se vaya creando a medida que se obtienen o se desarrollan y maduran sus componentes.





❖ Utilización de un único lenguaje de modelado

UML es adoptado como único lenguaje de modelado para el desarrollo de todos los modelos.

❖ Proceso Integrado

Se establece una estructura que abarque los ciclos, fases, flujos de trabajo, mitigación de riesgos, control de calidad, gestión del proyecto y control de configuración; el proceso unificado establece una estructura que integra todas estas facetas. Además esta estructura cubre a los vendedores y desarrolladores de herramientas para soportar la automatización del proceso, soportar flujos individuales de trabajo, para construir los diferentes modelos e integrar el trabajo a través del ciclo de vida y a través de todos los modelos.

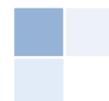
❖ Esta metodología divide en 4 fases el desarrollo de software:

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

Los 6 primeros son conocidos como flujos de ingeniería y los tres últimos como de apoyo.

❖ Flujos de trabajo:

- **Modelado del negocio:** Describe los procesos de negocio, identificando quiénes participan y las actividades que requieren automatización.
- **Requerimientos:** Define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.
- **Análisis y diseño:** Describe cómo el sistema será realizado a partir de la funcionalidad prevista y las restricciones impuestas (requerimientos), por lo que indica con precisión lo que se debe programar.
- **Implementación:** Define cómo se organizan las clases y objetos en componentes, cuáles nodos se utilizarán y la ubicación en ellos de los componentes y la estructura de capas de la aplicación.





- **Prueba (Testeo):** Busca los defectos a lo largo del ciclo de vida.
- **Instalación:** Produce release del producto y realiza actividades (empaquete, instalación, asistencia a usuarios) para entregar el software a los usuarios finales.
- **Administración del proyecto:** Involucra actividades con las que se busca producir un producto que satisfaga las necesidades de los clientes.
- **Administración de configuración y cambios:** Describe cómo controlar los elementos producidos por todos los integrantes del equipo de proyecto en cuanto a: utilización/actualización concurrente de elementos, control de versiones.
- **Ambiente:** Contiene actividades que describen los procesos y herramientas que soportarán el equipo de trabajo del proyecto; así como el procedimiento para implementar el proceso en una organización.

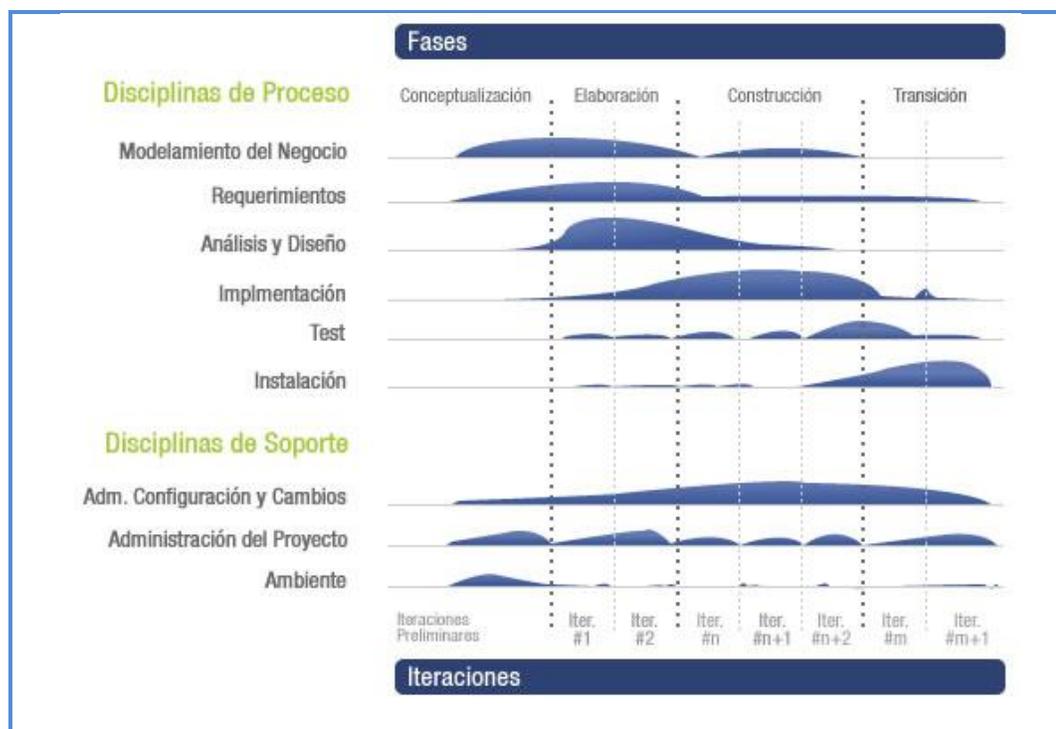


Imagen1.1: Proceso Unificado de Rational (RUP).

En los proyectos de desarrollo de software se hace indispensable la elección de una metodología a emplear, pues estas propician la calidad y mantenimiento del producto final. Estas metodologías no





pueden aplicarse como una secuencia de pasos lógicos, sino que deben adaptarlas a las necesidades que presentan los proyectos en su desarrollo.

Para el desarrollo del Sistema de Academic Schedule, se decidió mantener el uso de Rational Unified Process (RUP), argumentado que es la misma metodología utilizada en el Análisis y Diseño previamente realizado para este sistema, puesto que el mismo constituye la continuidad de dicho trabajo de diploma. (Sixto, y otros, 2008)

Esta metodología, también está preparada para desarrollar grandes y pequeños proyectos. Al estar basada en componentes da la posibilidad de construir un sistema formado por componentes interconectados a través de interfaces bien definidas.

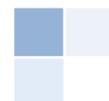
1.4. Paradigmas de programación.

Los paradigmas de programación son bastante variados. En la mayoría de los casos, los paradigmas se centran en programación modular y programación estructurada, pero existen otros tipos de programación. Algunos de los paradigmas de programación existentes serán analizados a continuación:

Programación estructurada.

La programación estructurada es un paradigma de programación basado sobre el concepto de la unidad y el alcance. Un programa estructurado se compone de una o más unidades o módulos — escrito por el usuario, o sacado de una librería; cada módulo se compone de una o más procedimientos, también llamado una función, una rutina, un subprograma, o un método, dependiendo del lenguaje de programación. Es posible que un programa estructural tenga niveles múltiples o alcances, con los procedimientos definidos dentro de otros procedimientos. Cada alcance puede contener las variables que no se pueden considerar en alcances externos. (PE, 2007)

La programación estructurada está compuesta por un conjunto de técnicas que han ido evolucionando considerablemente la productividad del programa reduciendo el tiempo de depuración y mantenimiento del mismo. Esta programación utiliza un número limitado de estructuras de control, reduciendo así considerablemente los errores.





❖ Este paradigma incorpora:

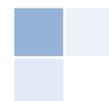
- Recursos abstractos (simplicidad): consiste en descomponer las acciones complejas en otras más simples capaces de ser resueltas con mayor facilidad.
- Estructuras básicas: existen tres tipos de estructuras básicas:
 - Estructuras secuenciales: cada acción sigue a otra acción secuencialmente. La salida de una acción es la entrada de otra.
 - Estructuras selectivas: en estas estructuras se evalúan las condiciones y en función del resultado de las mismas se realizan unas acciones u otras. Se utilizan expresiones lógicas.
 - Estructuras repetitivas: son secuencias de instrucciones que se repiten un número determinado de veces.

❖ Las principales ventajas de la programación estructurada son:

- Los programas son más fáciles de entender
- Se reduce la complejidad de las pruebas
- Aumenta la productividad del programador
- Los programas queden mejor documentados internamente.

Un programa está estructurado si posee un único punto de entrada y sólo uno de salida, existen de "1 a n" caminos desde el principio hasta el fin del programa, y por último todas las instrucciones son ejecutables sin que aparezcan bucles infinitos. (PE, 2007)

El principal inconveniente de este paradigma de programación, es que se obtiene un único bloque de programa, que cuando se hace demasiado grande puede resultar problemático su manejo, debido al tamaño del sistema este paradigma se hace insuficiente para satisfacer las necesidades por lo que no será usado.





Programación modular.

La programación modular es uno de los métodos más conocidos para resolver un problema, consiste en dividir el problema en problemas más pequeños, llamados sub-problemas. De esta manera, en lugar de resolver una tarea compleja y tediosa, se resuelven otras más sencillas y a partir de ellas se llega a la solución. Esta técnica se usa mucho en programación ya que programar no es más que resolver problemas, y se le suele llamar diseño descendente, metodología del divide y vencerás o programación top-down. (PM, 2006)

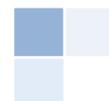
Un módulo es cada una de las partes de un programa que resuelve uno de los sub-problemas en que se divide el problema complejo original. Cada uno de estos módulos tiene una tarea bien definida y algunos necesitan de otros para poder operar. En caso que un módulo necesite de otro, puede comunicarse con éste mediante una interfaz de comunicación que también debe estar bien definida.

Si bien un módulo puede entenderse como una parte de un programa en cualquiera de sus formas y variados contextos, en la práctica es común representarlos con procedimientos y funciones. Adicionalmente, también pueden considerarse módulos las librerías que pueden incluirse en un programa o, en programación orientada a objetos, la implementación de un tipo de dato abstracto.

A medida que se divide el problema en sub-problemas, el número de módulos crece y se produce un incremento de las interfaces, por lo que no será usada en el desarrollo del sistema.

Programación concurrente.

La programación concurrente es la ejecución simultánea de múltiples tareas interactivas. Estas tareas pueden ser un conjunto de procesos o hilos de ejecución creados por un único programa. Las tareas se pueden ejecutar en una sola unidad central de proceso (multiprogramación), en varios procesadores o en una red de computadoras distribuidas; la misma está relacionada con la programación paralela, pero enfatiza más la interacción entre tareas. Así, la correcta secuencia de interacciones o comunicaciones entre los procesos y el acceso coordinado de recursos que se comparten por todos los procesos o tareas son las claves de esta disciplina.





La misma no será usada en el desarrollo del sistema debido a que no se implementarán funciones que utilicen estas técnicas de programación.

Programación lógica.

Consiste en la aplicación del corpus de conocimiento sobre lógica para el diseño de lenguajes de programación; no debe confundirse con la disciplina de la lógica computacional. La programación lógica comprende dos paradigmas de programación: la programación declarativa y la programación funcional. (MU, y otros)

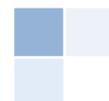
- ❖ La programación declarativa gira en torno al concepto de predicado, o relación entre elementos.
- ❖ La programación funcional se basa en el concepto de función (evolución de los predicados), de corte más matemático.

Se suele utilizar en la inteligencia artificial y pequeños programas infantiles. Se trata de una programación basada en el cálculo de predicados (una teoría matemática que permite lograr que un ordenador basándose en hecho y reglas lógicas, pueda dar soluciones inteligentes).

Debido al alcance que presenta el sistema en cuestión no se utilizarán técnicas de inteligencia artificial, por lo que no será usado este paradigma.

Programación con Restricciones.

La programación con restricciones es un paradigma utilizado para la descripción y posterior resolución efectiva de grandes y complejos problemas, particularmente combinatorios, de muchas áreas de la vida real. Muchos de estos problemas pueden modelarse como problemas de satisfacción de restricciones y resolverse usando técnicas de programación de restricciones. Esto incluye problemas de áreas tales como inteligencia artificial, investigación operativa, bases de datos y sistemas expertos. Algunos ejemplos son scheduling, planificación, razonamiento temporal, diseño en la ingeniería, problemas de empaquetamiento, criptografía, diagnóstico y toma de decisiones. (Barber, y otros)





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

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

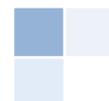
La diferencia entre ambos radica principalmente en sus estilos y enfoques en el modelado del mundo. Para ciertos problemas es más natural (y por ende más simple) escribirlos como programas lógicos, mientras que en otros es más natural escribirlos como programas con restricciones.

El enfoque de la programación con restricciones se basa principalmente en buscar un estado en el cual una gran cantidad de restricciones sean satisfechas simultáneamente. Un problema se define típicamente como un estado de la realidad en el cual existe un número de variables con valor desconocido. Un programa basado en restricciones busca dichos valores para todas las variables.

Algunos dominios de aplicación de este paradigma son:

- ❖ Dominios booleanos, donde solo existen restricciones del tipo verdadero/falso.
- ❖ Dominios en variables enteras y racionales.
- ❖ Dominios lineales, donde sólo se describen y analizan funciones lineales.
- ❖ Dominios finitos, donde las restricciones son definidas en conjuntos finitos.
- ❖ Dominios mixtos, los cuales involucran dos o más de los anteriores.

Los lenguajes de programación con restricciones son típicamente ampliaciones de otro lenguaje. El primer lenguaje utilizado a tal efecto fue Prolog. Por esta razón es que este campo fue llamado inicialmente Programación Lógica con Restricciones. Ambos paradigmas comparten características muy similares, tales como las variables lógicas, o el back-tracking.





La programación con restricciones puede ser implementada como un lenguaje propio o como bibliotecas para ser usadas en algún lenguaje de programación imperativo. Algunos lenguajes populares de programación con restricciones son:

- ❖ B-Prolog (Basado en Prolog, propietario).
- ❖ CHIP V5 (Basado en Prolog, también existen bibliotecas en C y C++, propietario).
- ❖ Ciao Prolog (Basado en Prolog, software libre: GPL/LGPL).
- ❖ ECLiPSe (Basado en Prolog, propietario).
- ❖ GNU Prolog (Basado en Prolog, software libre).
- ❖ SWI Prolog Un entorno Prolog que contiene varias librerías para soluciones con restricciones (LGPL).

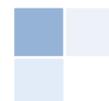
La programación con restricciones es empleada principalmente en sistemas muy engorrosos y difíciles, sobre todo en los que se generen calendarios o en problemas relacionados con la inteligencia artificial, por lo que no será usado para darle solución al sistema en cuestión.

Programación Orientada a Objetos.

La programación orientada a objetos (POO, en lo adelante) centra su diseño en objetos y sus interacciones para diseñar aplicaciones y programas de computadora. Está basado en varias técnicas, incluyendo herencia, modularidad, polimorfismo y encapsulamiento. Se trata de un paradigma que aumenta considerablemente la velocidad de desarrollo de los programas gracias a la reutilización de los objetos. (POO, 2006)

El elemento principal de la programación orientada a objetos es el objeto. El objeto es un conjunto complejo de datos y programas que poseen estructuras y forman parte de una organización. Contiene varios datos bien estructurados y pueden ser visibles o no dependiendo del programador y las acciones del programa en ese momento.

La POO expresa un programa como un conjunto de objetos, que colaboran entre ellos para realizar tareas. Esto permite hacer los programas y módulos más fáciles de escribir, mantener y reutilizar.





De esta forma, un objeto contiene toda la información que permite definirlo e identificarlo frente a otros objetos pertenecientes a otras clases e incluso frente a objetos de una misma clase, al poder tener valores bien diferenciados en sus atributos. A su vez, los objetos disponen de mecanismos de interacción llamados métodos que favorecen la comunicación entre ellos. Esta comunicación favorece a su vez el cambio de estado en los propios objetos. Esta característica lleva a tratarlos como unidades indivisibles, en las que no se separan ni deben separarse el estado y el comportamiento.

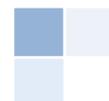
Los métodos (comportamiento) y atributos (estado) están estrechamente relacionados por la propiedad de conjunto. Esta propiedad destaca que una clase requiere de métodos para poder tratar los atributos con los que cuenta. El programador debe pensar indistintamente en ambos conceptos, sin separar ni darle mayor importancia a ninguno de ellos, hacerlo podría producir el hábito erróneo de crear clases contenedoras de información por un lado y clases con métodos que manejen a las primeras por el otro. De esta manera se estaría realizando una programación estructurada camuflada en un lenguaje de programación orientado a objetos. (POO, 2006)

En la actualidad existen muchos paradigmas de programación que son empleados en el desarrollo de sistemas. Cada uno brinda sus ventajas, pero debido a la exigencia que demandan los productos en cuando a reutilización de código, reparación y complejidad de los mismos, el paradigma de programación que responde a estas exigencias es la POO, este paradigma fomenta la reutilización y extensión del código, permite crear sistemas más complejos, relacionarlos con el mundo real y agiliza el desarrollo del software. Basándose en todas estas ventajas se determina que la POO es la mejor opción para el desarrollo del sistema propuesto.

1.5. Lenguajes.

Lenguajes de modelado.

La estandarización de un lenguaje de modelado es invaluable, ya que es la parte principal del proceso de comunicación que requieren todos los agentes involucrados en un proyecto informático. Si se quiere discutir un diseño con alguien más, ambos deben conocer el lenguaje de modelado y no así el proceso que se siguió para obtenerlo.





Este lenguaje unificado que cumple con estos requerimientos, es ciertamente UML, el cual cuenta con una notación estándar y semánticas esenciales para el modelado de un sistema orientado a objetos, el cual es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema.

El Lenguaje de Modelado Unificado (UML: Unified Modeling Language) es la sucesión de una serie de métodos de análisis y diseño orientadas a objetos que aparecen a fines de 1980 y principios del 90. UML es llamado un lenguaje de modelado, no un método, fusiona los conceptos de la orientación a objetos aportados por Booch, incrementa la capacidad de lo que se puede hacer con otros métodos de análisis y diseño orientados a objetos. Los autores de UML apuntaron también al modelado de sistemas distribuidos y concurrentes para asegurar que el lenguaje maneje adecuadamente estos dominios. (Booch, y otros, 1999)

Por lo antes expuesto se utilizará la versión UML 2.0 la cual constituye una de las versiones más recientes publicada.

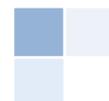
Lenguajes del lado del cliente.

Los lenguajes de lado cliente son aquellos que pueden ser directamente asimilados por el navegador Web, (entre los cuales se encuentra el HTML, Java y JavaScript que son simplemente incluidos en el código HTML).

DHTML

DHTML no es precisamente un lenguaje de programación. Más bien se trata de una nueva capacidad de la que disponen los navegadores modernos, por la cual se puede tener un mayor control sobre la página que antes. (de la Torre)

Cualquier página que responde a las actividades del usuario y realiza efectos y funcionalidades se puede englobar dentro del DHTML, pero en este caso permite referirse más a efectos en el navegador por los cuales se pueden mostrar y ocultar elementos de la página, se puede modificar su posición, dimensiones, color. Da más control sobre la página, gracias a que los navegadores modernos incluyen una nueva





estructura para visualizar en páginas web denominada capa. Las capas se pueden ocultar, mostrar, desplazar.

Para realizar las acciones sobre la página, como modificar la apariencia de una capa, es necesario un lenguaje de programación del lado del cliente como Javascript o VBScript. En la actualidad, DHTML también puede englobar la programación en el servidor. Dentro de su concepto se engloban también las Hojas de Estilo en Cascada o CSS (Cascade Style Sheets). (de la Torre)

HTML

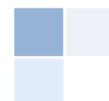
HTML, siglas de *HyperText Markup Language (Lenguaje de Marcas de Hipertexto)*, es el lenguaje de marcado predominante para la construcción de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. Este se escribe en forma de etiquetas, rodeadas por corchetes angulares (<,>). También puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir un script (por ejemplo Javascript), el cual puede afectar el comportamiento de navegadores web y otros procesadores de HTML. (de la Torre)

JAVASCRIPT

Javascript es un lenguaje de programación utilizado para crear pequeños programas encargados de realizar acciones dentro del ámbito de una página web. Se trata de un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento. Su uso se basa fundamentalmente en la creación de efectos especiales en las páginas y la definición de interactividades con el usuario. (de la Torre)

La utilización del lenguaje JavaScript según (BRANDENDAUGH, 2000), brinda diferentes ventajas como:

- ❖ **Rápido y potente:** Es ideal para agregar ciertas funciones rápidas a una página Web. Es un lenguaje muy potente de alto nivel. No necesita una fase de compilación como Java o C, solo hay que crear el código y cargarlo.
- ❖ **Usabilidad:** JavaScript es uno de los lenguajes de programación que más se utiliza en la Web. Hay publicadas millones de páginas Web que incorporan elementos que lo usan. Una de las principales





razones por la que los desarrolladores Web han adoptado este lenguaje es porque reduce la carga del servidor.

- ❖ **Reducción de la Carga del Servidor:** JavaScript se puede hacer cargo de gran parte de las funciones del cliente de las cuales se encargaba el servidor. Uno de los mejores ejemplos es la validación. Con él es posible validar los elementos antes de que el usuario los envíe al servidor. De esta forma se reduce la cantidad de transacciones que se efectúan a través de http y las posibilidades de que se genere un error durante la inserción de datos.

Lenguajes de programación del lado del servidor.

Los lenguajes de lado del servidor son aquellos que son reconocidos, ejecutados e interpretados por el propio servidor y que se envían al cliente en un formato comprensible para él. Debido a las necesidades y requisitos que posee el proyecto en cuestión se decidió utilizar como lenguaje de programación el PHP del cual se abordará posteriormente.

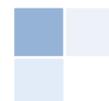
PERL

Perl es un lenguaje de programación interpretado, al igual que otros lenguajes como Javascript o ASP. Esto quiere decir que el código de los scripts en Perl no se compila sino que cada vez que se quiere ejecutar se lee el código y se pone en marcha interpretando lo que hay escrito. Además es extensible a partir de otros lenguajes, ya que desde Perl se puede hacer llamadas a subprogramas escritos en otros lenguajes. También desde otros lenguajes se pueden ejecutar código Perl. (de la Torre)

Este lenguaje no se considera potente para la implementación de sitios web, por lo que no se empleará para implementar el sistema.

PHP

PHP (HyperText Pre-Processor) es un lenguaje interpretado de propósito general ampliamente usado y que está diseñado especialmente para desarrollo web y puede ser incrustado dentro de código HTML. Generalmente se ejecuta en un servidor web, tomando el código en PHP como su entrada y creando





páginas web como salida. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno.

PHP según (WELLING, y otros) posee una serie de características que conllevaron a la elección de su uso en nuestro sistema, las cuales se muestran a continuación:

- ❖ Es un lenguaje multiplataforma.
- ❖ Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad.
- ❖ Capacidad de expandir su potencial utilizando la enorme cantidad de módulos.
- ❖ Posee una amplia documentación en su página oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- ❖ Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- ❖ Permite las técnicas de Programación Orientada a Objetos.
- ❖ Biblioteca nativa de funciones sumamente amplia e incluida.
- ❖ No requiere definición de tipos de variables.
- ❖ Tiene manejo de excepciones.

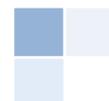
1.6. *Análisis de herramientas a utilizar en el sistema.*

Herramientas CASE.

Las herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador) un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del Ciclo de Vida de desarrollo de un Software (Investigación Preliminar, Análisis, Diseño, Implementación e Instalación.).

Las herramientas CASE son también definidas como: el conjunto de métodos, utilidades y técnicas que facilitan el mejoramiento del ciclo de vida del desarrollo de sistemas de información; completamente o en alguna de sus fases (). Ventajas de las herramientas CASE:

- ❖ Mejorar la productividad en el desarrollo y mantenimiento del software.





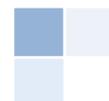
- ❖ Aumentar la calidad del software.
- ❖ Mejorar el tiempo y coste de desarrollo y mantenimiento de los sistemas informáticos.
- ❖ Mejorar la planificación de un proyecto.
- ❖ Aumentar la biblioteca de conocimiento informático de una empresa ayudando a la búsqueda de soluciones para los requisitos.
- ❖ Automatizar, desarrollo del software, documentación, generación de código, pruebas de errores y gestión del proyecto.
- ❖ Ayuda a la reutilización del software, portabilidad y estandarización de la documentación.
- ❖ Gestión global en todas las fases de desarrollo de software con una misma herramienta.
- ❖ Facilitar el uso de las distintas metodologías propias de la ingeniería del software.

Rational Rose y el Visual Paradigm, constituyen dos de las herramientas más utilizadas para el modelado visual de sistemas, estas herramientas fueron tratadas en el Análisis y Diseño del sistema en cuestión previamente realizado como se ha venido diciendo. Debido a la importancia que se le atribuye a dichas herramientas, se hace necesario hacer una descripción y argumentar acerca de las mismas para lograr una mayor documentación a través de la cual se puedan adquirir bastos conocimientos.

Rational Rose es una de las más poderosas herramientas de modelado visual para sistemas basados en objetos. Se utiliza para modelar un sistema antes de proceder a construirlo. Cubre todo el ciclo de vida de un proyecto: concepción y formalización del modelo, construcción de los componentes, transición a los usuarios y certificación de las distintas fases. También utiliza un proceso de desarrollo iterativo controlado, donde el desarrollo se lleva a cabo en una secuencia de iteraciones. Otra característica importante es que usando Rose se puede generar código en distintos lenguajes de programación a partir de un diseño en UML.

Características adicionales incluidas:

- ❖ Característica de control por separado de componentes modelo que permite una administración más granular y el uso de modelos.
- ❖ Capacidad de análisis de calidad de código.



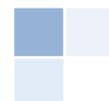


- ❖ Modelado UML para trabajar en diseños de base de datos, con capacidad de representar la integración de los datos y los requerimientos de aplicación a través de diseños lógicos y físicos.
- ❖ Publicación web y generación de informes para optimizar la comunicación dentro del equipo. (BLANCO, y otros)

Otra herramienta case sumamente interesante y a tomar en consideración es **Visual Paradigm** la cual da soporte para el modelado visual tanto con UML 2.0 como 2.1. La misma soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Visual Paradigm también ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Las siguientes características demuestran las verdaderas potencialidades de esta herramienta: entorno de creación de diagramas para UML 2.0; diseño centrado en casos de uso y enfocado al negocio que generan un software de mayor calidad; uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación; capacidades de ingeniería directa e inversa; modelo y código que permanece sincronizado en todo el ciclo de desarrollo; disponibilidad de múltiples versiones, para cada necesidad; disponibilidad de integrarse en los principales IDEs (Integrated Development Environment); disponibilidad en múltiples plataformas; ofrece un mecanismo general para la organización de los modelos/subsistemas/capas agrupando elementos de modelado y versión gratuita (licencia para Community Edition).

No obstante a todas las facilidades que ofrece Visual Paradigm se decidió seguir modelando el sistema con Rational Rose, por haber sido la herramienta empleada en el Análisis y Diseño de este sistema anteriormente mencionando.

Rational Rose constituye además la herramienta líder en el mundo de modelación visual para el proceso de modelación del negocio, análisis de requerimientos y diseño de arquitectura de componentes. Cubre todo el ciclo de vida de un proyecto: concepción y formalización del modelo, construcción de los componentes, transición a los usuarios y certificación de las distintas fases, características que ratifican su uso en esta solución informática.





1.7. *Tecnologías del lado del cliente.*

FLASH

Flash es una tecnología, y un programa, para crear efectos especiales en páginas web. Con Flash también conseguimos hacer páginas dinámicas del lado del cliente. Flash en realidad no es un lenguaje; Sin embargo, si hubiera que catalogarlo en algún sitio quedaría dentro del ámbito de las páginas dinámicas de cliente. (de la Torre)

CSS

CSS son las siglas de **Cascading Style Sheets**, en español **Hojas de estilo en Cascada**. Es una tecnología que permite crear páginas web de una manera más exacta. Gracias a las CSS se logra ser mucho más dueños de los resultados finales de la página, pudiendo hacer muchas cosas que no se podía hacer utilizando solamente HTML, como incluir márgenes, tipos de letra, fondos, colores. Incluso se puede definir los propios estilos en un archivo externo a las páginas; así, si en algún momento se quiere cambiar alguno de ellos, automáticamente se actualizarán todas las páginas vinculadas al sitio. (de la Torre)

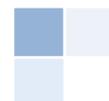
1.8. *Tecnologías del lado del servidor.*

ASP

ASP (Active Server Pages) es la tecnología desarrollada por Microsoft para la creación de páginas dinámicas del servidor. ASP se escribe en la misma página web, utilizando el lenguaje Visual Basic Script o Jscript (Javascript de Microsoft). (de la Torre)

JSP

JSP es un acrónimo de Java Server Pages, que sería Páginas de Servidor Java. Es una tecnología orientada a crear páginas web con programación en Java.





Con JSP se puede crear aplicaciones web que se ejecuten en variados servidores web, de múltiples plataformas, ya que Java es en esencia un lenguaje multiplataforma. Por tanto, las JSP se pueden escribir con un editor HTML/XML. (de la Torre)

Sistemas Gestores de Bases de Datos.

Los **Sistemas Gestores de Bases de Datos (SGBD)** se definen como el conjunto de programas que administran y gestionan la información contenida en una base de datos. (Alvarez)

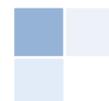
Los SGBD persiguen de forma general los siguientes objetivos:

- ❖ **Abstracción de la información:** Los SGBD ahorran a los usuarios detalles acerca del almacenamiento físico de los datos.
- ❖ **Consistencia:** En aquellos casos en los que no se ha logrado eliminar la redundancia, será necesario vigilar que aquella información que aparece repetida se actualice de forma coherente.
- ❖ **Seguridad:** La información almacenada en una Base de Datos puede llegar a tener un gran valor. Los SGBD deben garantizar que esta información se encuentra segura frente a usuarios malintencionados, que atenten en contra de la información almacenada.
- ❖ **Integridad:** Se trata de proteger los datos ante fallos de hardware, datos introducidos por usuarios descuidados, o cualquier otra circunstancia capaz de corromper la información almacenada.
- ❖ **Control de la Concurrencia:** En la mayoría de los entornos, lo más habitual es que muchas personas accedan a una base de datos simultáneamente, por lo que los SGBD deben controlar este acceso concurrente a la información, que podría derivar en inconsistencias.

Existen varios SGBD como: PostgreSQL, MySQL, Firebird, SQLite, Oracle, Microsoft SQL Server, Paradox, a continuación se abordará acerca de tres de los más usados a nivel internacional, en Cuba y en la Universidad.

Oracle: Es un Sistema de Gestión de Base de Datos relacional, desarrollado por Oracle Corporation.

Se considera a Oracle como uno de los sistemas de bases de datos más completos, destacando su:





- ❖ Soporte de transacciones.
- ❖ Estabilidad.
- ❖ Escalabilidad.
- ❖ Soporte multiplataforma.

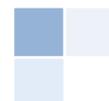
Ha sido criticada por algunos especialistas la seguridad de la plataforma y las políticas de suministro de parches de seguridad, modificadas a comienzos de 2005 y que incrementan el nivel de exposición de los usuarios. En los parches de actualización provistos durante el primer semestre de 2005 fueron corregidas 22 vulnerabilidades públicamente conocidas, algunas de ellas con una antigüedad de más de dos años.

Aunque su dominio en el mercado de servidores empresariales ha sido casi total hasta hace poco, recientemente sufre la competencia del Microsoft SQL Server de Microsoft y de la oferta de otros RDBMS con licencia libre como PostgreSQL, MySQL o Firebird. Las últimas versiones de Oracle han sido certificadas para poder trabajar bajo Linux.

PostgreSQL: Es un Sistema de Gestión de Base de Datos relacional orientada a objetos de software libre. Posee una alta concurrencia, mediante un sistema denominado MVCC (Acceso Concurrente Multiversión, por sus siglas en inglés) PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Esta estrategia es superior al uso de bloqueos por tabla o por filas común en otras bases, eliminando la necesidad del uso de bloqueos explícitos.

PostgreSQL provee nativamente soporte para, números de precisión arbitraria, texto de largo ilimitado, figuras geométricas (con una variedad de funciones asociadas), direcciones IP (IPv4 e IPv6), bloques de direcciones estilo CIDR, direcciones MAC, y arrays. (2005)

MySQL: Es un sistema para la administración de Bases de Datos relacional (RDBMS) rápido y sólido. Las Bases de Datos permiten almacenar, buscar, ordenar y recuperar datos de forma eficiente. El servidor de MySQL controla el acceso a los datos para garantizar el uso simultáneo de varios usuarios, para proporcionar acceso a dichos datos y para asegurarse de que solo obtienen acceso a ellos los usuarios con autorización. Por lo tanto, es un servidor multiusuario y de subprocesamiento múltiple. (WELLING, y otros)





Ofrece un sistema de contraseñas y privilegios seguro mediante verificación basada en el host y el tráfico de contraseñas está cifrado al conectarse a un servidor. Soporta gran cantidad de datos. MySQL Server tiene Bases de Datos de hasta 50 millones de registros. Los clientes se conectan al servidor MySQL usando sockets TCP/IP en cualquier plataforma. En sistemas Windows se pueden conectar usando named pipes y en sistemas Unix usando ficheros socket Unix.

El uso de este SGBD en el sistema en cuestión fue valorado debido a las siguientes características que posee el mismo según (WELLING, y otros):

- ❖ Múltiples motores de almacenamiento lo cual permite al usuario escoger el que sea más adecuado para cada tabla de la base de datos.
- ❖ Agrupación de transacciones, es decir reúne múltiples transacciones de varias conexiones para así incrementar el número de transacciones realizadas por segundo.

MySQL cuenta con las siguientes ventajas:

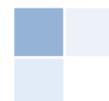
- ❖ Alto Rendimiento: Es muy rápido.
- ❖ Facilidad de Configuración y Aprendizaje.

Las Bases de Datos más modernas utilizan SQL. MySQL resulta además más sencillo de configurar que otros productos similares.

- ❖ Portabilidad: Se puede utilizar en una gran cantidad de sistemas Unix diferentes, así como bajo Microsoft Windows.

Servidor Web.

Un **Servidor Web** es un programa que se ejecuta continuamente en un computador, manteniéndose a la espera de peticiones de ejecución que le hará un cliente o un usuario de Internet. El servidor web se encarga de contestar a estas peticiones de forma adecuada, entregando como resultado una página web o información de todo tipo de acuerdo a los comandos solicitados. (2006)





Algunos servidores web importantes son el, Apache, IIS y Cherokee aunque el más utilizado es el **Apache**, el cual es un software libre de código abierto para plataformas Unix (BSD, GNU/Linux), Windows y Macintosh fundamentalmente, que implementa el protocolo HTTP y la noción de sitio virtual. (Martin Maldonado)

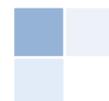
Apache presenta mensajes de error altamente configurables, Bases de Datos de autenticación y negociado de contenido, aunque carece de una interfaz gráfica que ayude en su configuración.

Posee características según (Martin Maldonado), que hacen ratificar su uso como son:

- ❖ Multiplataforma.
- ❖ Es un servidor de web conforme al protocolo HTTP/1.1.
- ❖ Modular: Puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona, y con la API de programación de módulos, para el desarrollo de módulos específicos.
- ❖ Basado en hebras en la versión 2.0.
- ❖ Incentiva la realimentación de los usuarios, obteniendo nuevas ideas, informes de fallos y parches para la solución de los mismos.
- ❖ Se desarrolla de forma abierta.
- ❖ Extensible: gracias a ser modular se han desarrollado diversas extensiones entre las que destaca PHP, un lenguaje de programación del lado del servidor.
- ❖ Soporte para los lenguajes perl, python, tcl y PHP.

IDE de desarrollo.

Un **Entorno de Desarrollo Integrado** o, en inglés, **Integrated Development Environment ('IDE')**, es un programa compuesto por un conjunto de herramientas para el programador, que suelen incluir, un buen editor de código, administrador de proyectos y archivos, enlace transparente a compiladores e integración con sistemas controladores de versiones. (Luciano)





Es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica GUI. Los IDEs pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes.

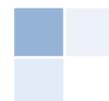
Algunos de los Entornos de Desarrollo Integrados para PHP más conocidos o habituales son:

Zend Studio o Zend Development Environment: es un completo entorno integrado de desarrollo para el lenguaje de programación PHP. El programa, además de servir de editor de texto para páginas PHP, proporciona una serie de ayudas que pasan desde la creación y gestión de proyectos hasta la depuración de código. (Alvarez)

Junto con su contraparte Zend Platform, son la propuesta de Zend Technologies para el desarrollo de aplicaciones Web utilizando PHP, actuando Zend Studio como la parte cliente y Zend Platform como la parte servidora.

Características:

- ❖ No requiere la instalación previa de PHP ni del entorno de ejecución de Java.
- ❖ Soporte para PHP 4 y PHP 5.
- ❖ Resaltado de sintaxis, autocompletado de código, ayuda de código y lista de parámetros de funciones y métodos de clase.
- ❖ Plegado de código (comentarios, bloques de phpDoc, cuerpo de funciones y métodos e implementación de clases).
- ❖ Inserción automática de paréntesis y corchetes de cierre.
- ❖ Detección de errores de sintaxis en tiempo real.
- ❖ Instalación de barras de herramientas para Internet Explorer y Mozilla Firefox (opcional).
- ❖ Soporte para gestión de grandes proyectos de desarrollo.
- ❖ Manual de PHP integrado.
- ❖ Soporte para navegación en bases de datos y ejecución de consultas SQL.
- ❖ Zend Studio fue diseñado para usarse con el lenguaje PHP; sin embargo ofrece soporte básico para otros lenguajes Web, como HTML, Javascript y XML.





Aptana: es líder en una serie de infraestructuras para el desarrollo de sitios y aplicaciones Web que usan lenguajes de programación como JavaScript, Ruby on Rails y PHP. Aptana Studio, un entorno integrado de desarrollo basado en Eclipse para sitios y aplicaciones Web 2.0, ha sido descargado por casi 1.400.000 desarrolladores y cada mes se añaden más de 100.000 nuevos usuarios. Aptana también es el creador de Aptana Jaxer(TM), el primer servidor de Ajax del mundo. A través del uso del código fuente abierto del servidor Aptana Jaxer, los desarrolladores pueden escribir aplicaciones enteras y capas de presentación usando Ajax, usar JavaScript desde el lugar del servidor para generar o manipular páginas Web, interactuar directamente con Bases de Datos, sistemas de archivos y redes, y ejecutar lógica de aplicación en el servidor. (2008)

Incorpora características completas, sincronización y administración de proyectos. Permite incorporarle funciones mediante plugins. Soporte para las plataformas Microsoft Windows, Mac y Linux. Cuenta con una Edición gratis para la comunidad y otra Profesional de pago.

Características

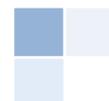
- ❖ Desarrollo HTML, CSS, Javascript, soporte para AJAX, incluye librerías AJAX más populares (jQuery, Prototype, YUI, Spry).
- ❖ Características de Edición Profesional: editor JSON (Javascript Object Notation), protocolos de Comunicación FTPS, SFTP, soporte en Forum.

Ventajas: permite comprobar la compatibilidad de las funciones con los diferentes navegadores, multiplataforma, sincronización con carpetas locales y remotas.

En cuanto a las aplicaciones clientes, Aptana provee al programador de funcionalidades muy ricas para el desarrollo de aplicaciones gráficas, definición y manipulación de modelos de software, aplicaciones web; por lo que será utilizado como IDE de desarrollo en el sistema propuesto.

1.9. EXT JS

EXT 2.2 es una librería de javascript, basado originalmente en **YUI (The Yahoo! UI Library)** pero que actualmente es independiente del framework que se utilice. Ofrece una gran cantidad de artefactos para





crear interfaces de usuario complejas. Permite potenciar la capa UI de Javascript en las aplicaciones. Ayuda a la comunicación entre el cliente y el servidor mediante JSON y XML.

Brinda un conjunto de ventajas tales como:

- ❖ Código reutilizable.
- ❖ Independiente o adaptable a frameworks diferentes (prototype, jquery, YUI).
- ❖ El API es homogeneizado independientemente del adaptador usado. Los controles siempre se verán igual.
- ❖ Soporte comercial.

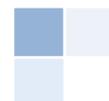
La librería ofrece:

- ❖ Controles de GUI.
- ❖ Layouts.
- ❖ Paneles.
- ❖ Grids.
- ❖ Ventanas DHTML.
- ❖ Tooplips.
- ❖ Pestañas.
- ❖ Controles de formulario.
- ❖ Menus.

Esta librería será usada en el desarrollo de la aplicación debido a las potenciales ventajas que proporciona, con los EXT se logra una interfaz visual amigable, donde se pueden ejecutar una serie de acciones para que el sistema funcione correctamente.

1.10. Sistema de Gestión de Contenidos.

Un **Sistema de Gestión de Contenidos** (**Content Management System** en inglés, abreviado **CMS**) es un programa que permite crear una estructura de soporte (framework) para la creación y administración de contenidos por parte de los participantes principalmente en páginas web. (2008)





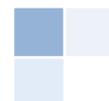
Consiste en una interfaz que controla una o varias Bases de Datos donde se aloja el contenido del sitio. El sistema permite manejar de manera independiente el contenido y el diseño. Así, es posible manejar el contenido y darle en cualquier momento un diseño distinto al sitio sin tener que darle formato al contenido de nuevo, además de permitir la fácil y controlada publicación en el sitio a varios editores. Un ejemplo clásico es el de editores que cargan el contenido al sistema y otro de nivel superior que permite que estos contenidos sean visibles a todo el público.

Ofrece la plataforma necesaria para desarrollar e implementar aplicaciones que den solución a necesidades específicas. Pueden servir para construir soluciones de gestión de contenidos, soluciones de comercio electrónico, blogs, portales. Ejemplos: Plone, MODx, OpenCMS, TYPO3, Apache lenya, Joomla, Drupal, Nuxeo, Content-SORT. A continuación se aborda sobre los más usados en la UCI.

Drupal: es un CMS que permite, tanto a individuos como a comunidades de usuarios, la publicación de contenidos en Internet, de manera sencilla, y sin tener conocimientos de lenguaje de marcas. Desarrollado dentro de la filosofía del Software Libre y de código fuente abierto bajo licencia GPL, Drupal es un sistema dinámico, puesto que el contenido de las páginas y otras configuraciones se pueden editar online mediante las herramientas que él mismo facilita. (2009)

Permite publicar artículos, imágenes u otros archivos y servicios añadidos como foros, encuestas, votaciones, blogs, administración de usuarios y permisos. Se usa en intranets de compañías, enseñanza en línea, comunidades de arte y administración de proyectos. La relevancia de Drupal en las comunidades de usuarios es lo que lo hace destacarse de la competencia.

Joomla: es un sistema de administración de contenidos de código libre y abierto. Se usa para realizar publicaciones en Internet e intranets utilizando una Base de Datos MySQL. En Joomla se incluyen características como: hacer caché de páginas para mejorar el rendimiento, indexamiento web, versiones imprimibles de páginas, flash con noticias, blogs, foros, encuestas, calendarios, búsqueda en el sitio web, e internacionalización del lenguaje. Es compatible con las siguientes plataformas: GNU/Linux, Windows y Mac OSX.



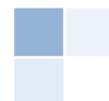


De forma similar a otros proyectos, mantiene dos versiones de la aplicación: una estable y otra Beta o en desarrollo. La versión estable es la que se considera para usuarios y a medida que aparecen errores se corrigen, esta versión se publica sin nuevas funcionalidades. La versión Beta, incluye nuevas funcionalidades y mejoras a las fallas reportadas en versiones anteriores, de igual forma se tiene que en este tipo de versión se reflejan las directivas del proyecto para usuarios avanzados y desarrolladores.

El paquete está diseñado para ser instalado fácilmente. Una vez que se ha instalado y configurado, resulta muy sencillo para usuarios sin conocimientos técnicos el añadir o editar contenido, subir imágenes, y gestionar los datos principales de su compañía u organización.

Joomla, según (2009) posee un conjunto de características tales como:

- ❖ Organización del sitio web: Está preparado para organizar eficientemente los contenidos de su sitio en secciones y categorías, lo que facilita la navegabilidad para los usuarios y permite crear una estructura sólida, ordenada y sencilla para los administradores. Desde el panel administrador de Joomla usted podrá crear, editar y borrar las secciones y categorías de su sitio de la manera en que más le convenga.
- ❖ Publicación de Contenidos: Se pueden crear páginas ilimitadas y editarlas desde un sencillo editor que permite formatear los textos con los estilos e imágenes deseados. Los contenidos son totalmente editables y modificables.
- ❖ Escalabilidad e implementación de nuevas funcionalidades: Ofrece la posibilidad de instalar, desinstalar y administrar componentes y módulos, que agregarán servicios de valor a los visitantes de su sitio web, por ejemplo: galerías de imágenes y foros.
- ❖ Administración de usuarios: Permite almacenar datos de usuarios registrados y también la posibilidad de enviar E-mails masivos a todos los usuarios. La administración de usuarios es jerárquica, y los distintos grupos de usuarios poseen diferentes niveles de facultades/permisos dentro de la gestión y administración del sitio.
- ❖ Diseño y aspecto estético del sitio: Es posible cambiar todo el aspecto del sitio web, gracias al sistema de plantillas que utiliza.
- ❖ Navegación y menú: Totalmente editables desde el panel administrador que posee.





- ❖ Administrador de Imágenes: Contiene una utilidad para subir imágenes al servidor y usarlas en todo el sitio.
- ❖ Disposición de módulos modificable: La posición de módulos puede acomodarse como se prefiera.
- ❖ Encuestas: Contiene sistema de votaciones y encuestas dinámicas con resultados en barras porcentuales.
- ❖ Feed de Noticias: Trae incorporado un sistema de sindicación de noticias por RSS/XMS de generación automática.
- ❖ Publicidad: Es posible hacer publicidad en el sitio usando el Administrador de Banners.

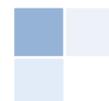
Tanto Joomla como Drupal poseen características y funciones similares por lo que se basan los autores para seleccionar a Joomla, en su compatibilidad con el SGBD, y por ser además un requerimiento del usuario.

1.11. Conclusiones.

En este capítulo se dieron a conocer una serie de características que intervienen en la planificación docente, se abordaron una serie de conceptos para una mejor comprensión para el desarrollo del sistema. Se realizó un análisis de algunos de los sistemas de planificación similares al que se desarrollará. Se llevó a cabo un estudio riguroso de las principales metodologías, paradigmas de programación, herramientas y tecnologías más usadas a nivel internacional, nacional y en la universidad, para lograr un elevado nivel de conocimiento y productividad en el desarrollo del sistema propuesto.

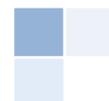
De esta forma se arriban a las siguientes conclusiones:

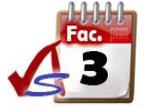
- ❖ Utilizar como metodología de desarrollo de software RUP en lo referente al flujo de trabajo de Implementación y Prueba, por ser la metodología usada en el Análisis y Diseño previamente realizado del sistema en cuestión.
- ❖ Emplear como herramienta CASE Rational Rose para elaborar el Modelo de Implementación y seguir modelando el sistema en la misma herramienta que se modeló el Análisis y Diseño previamente realizado.





- ❖ Emplear como sistema gestor de bases de datos a MySQL debido a que posee gran portabilidad, es una herramienta libre que ofrece seguridad y soporta un número ilimitado de Bases de Datos.
- ❖ Utilizar como servidor Web Apache por ser una tecnología gratuita, multiplataforma y altamente configurable.
- ❖ Emplear como lenguaje del lado servidor PHP en vista de que es software libre, rápido, multiplataforma y posee abundante documentación.
- ❖ Aptana como IDE de desarrollo, pues facilita el trabajo y permite al desarrollador un mejor desempeño en las tarea que debe desarrollar.
- ❖ Se seleccionó como lenguaje de programación del lado del cliente JavaScript por su rapidez, sencillez y el contenido de la librería EXT JS, así como hacer uso de HTML para dar cierta interactividad al sistema.
- ❖ Se seleccionó Joomla como CMS por las facilidades que brinda para el desarrollo de la interfaz, a su compatibilidad con MySQL, y por ser un requerimiento del usuario.





Capítulo 2: Descripción y Análisis de la Solución Propuesta.

2.1 Introducción.

En este capítulo se propone la solución que dará paso a la implementación del sistema, se hace además un análisis crítico del diseño propuesto por el analista, se definen nuevos requisitos funcionales, así como la descripción de los casos de uso críticos del sistema y se hará una descripción de las clases que serán usadas en el desarrollo de esta solución informática.

2.2 Valoración crítica del diseño propuesto por el analista.

El diseño propuesto por el analista está compuesto por paquetes como se describe a continuación y véase en (*Imagen. 2.1*).

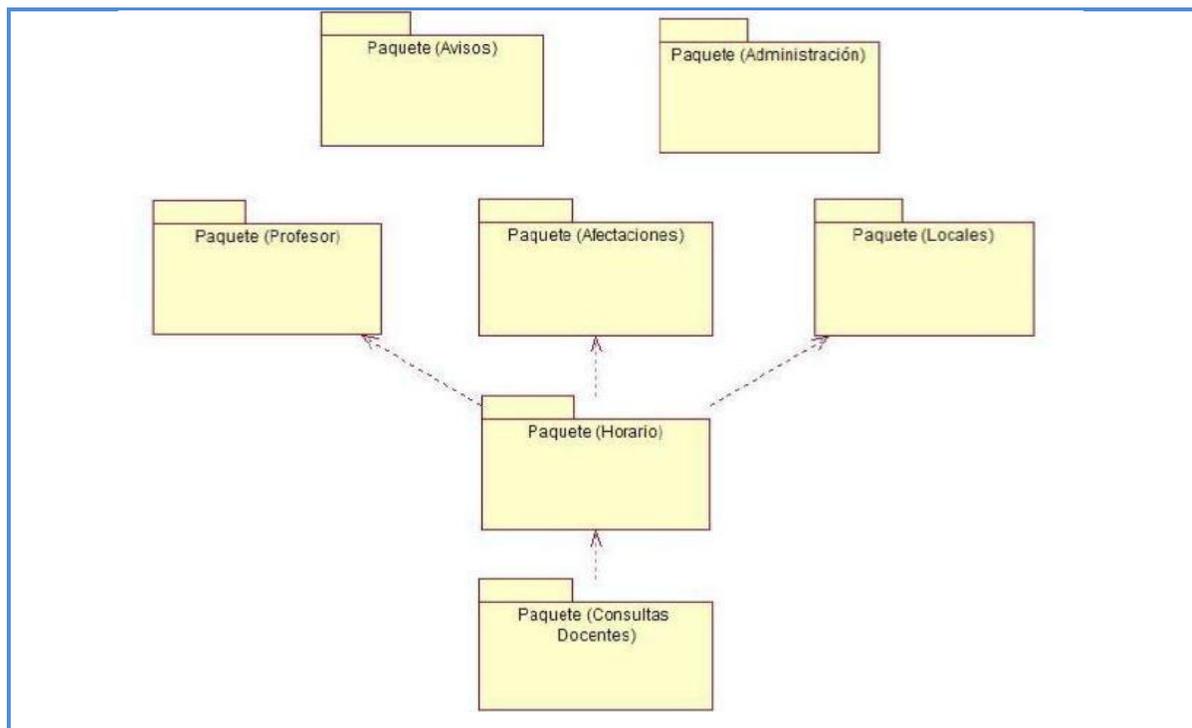
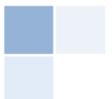


Imagen 2.1: Diagrama de Paquetes. (Sixto, y otros, 2008)





Paquete Consultas Docentes: Este paquete se encarga de brindar información al usuario sobre el personal docente.

Paquete Avisos: Este paquete permite a los administradores del sistema brindar al usuario información de último momento a través de la aplicación web.

Paquete Administración: Este paquete es el encargado de controlar la administración del sistema.

Paquete Profesor: Este paquete se encarga de gestionar todos los datos concernientes a los profesores.

Paquete Afectaciones: Este paquete gestiona las posibles afectaciones que presenta el personal docente, lo cual permite disminuir la coincidencia de las actividades docentes a efectuarse.

Paquete Local: Este paquete se encarga de gestionar los diferentes tipos de locales, si están siendo o no utilizados, evitando que coincidan actividades en los mismos.

Paquete Horario: Este paquete es el más importante de todos, pues gestiona todo lo concerniente a la actividad docente, vinculando las mismas con el personal docente y los locales, teniendo en cuenta un conjunto de restricciones a la hora de realizar la planificación docente.

El diseño propuesto por el analista, en el momento que se realizó este análisis cumplía con las necesidades del cliente, este posee siete paquetes, lo cual es innecesario ya que algunos de ellos incluyen un solo caso de uso y el objetivo es agrupar dos o más para una mejor organización, por lo que se propone un nuevo diseño, estructurado también en forma de paquetes (*Imagen 2.2*), con el fin de darle soporte y funcionalidad al sistema, permitiendo la implementación de todos los requisitos, así como la posibilidad de mantener el sistema sin necesidad de cambiar la arquitectura del mismo. En este nuevo diseño se agruparon los paquetes según las relaciones entre las diferentes funcionalidades que tendrá el sistema y se crean dos nuevos grupos, los cuales se describen posteriormente.



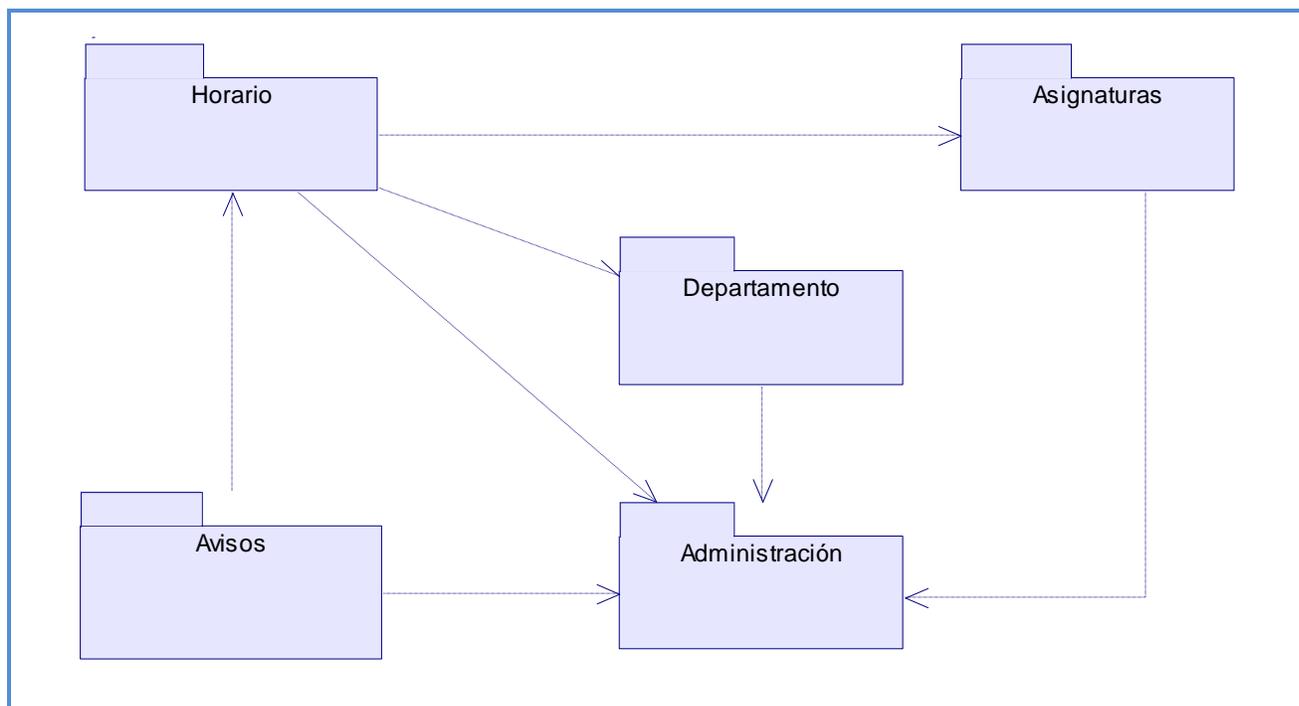


Imagen 2. 2: Diagrama de Paquetes.

Paquete Asignaturas: Este paquete es el encargado de gestionar todo lo relacionado con las asignaturas que se imparten en cada semestre, así como los tipos de actividades docentes que tiene la asignatura minimizando los errores a la hora de planificar las actividades docentes en la semana.

Paquete Departamento: Este paquete se encarga de la gestión de la información concerniente a los profesores y los departamentos a los cuales pertenecen dichos profesores.

Como resultado de entrevistas realizadas al cliente se identificaron 13 requisitos funcionales, los cuales se muestran a continuación.





2.3 Requisitos Funcionales.

1. El sistema debe permitir Autenticar Usuario.

- 1.1. Verificar usuario y contraseña con las existentes en la base de datos.

2. El sistema debe permitir Gestionar Actividad Docente.

- 2.1. Adicionar Actividad Docente.
- 2.2. Modificar Actividad Docente.
- 2.3. Eliminar Actividad Docente.
- 2.4. Buscar Actividad Docente.

3. El sistema debe permitir Gestionar Asignatura.

- 3.1. Adicionar Asignatura.
- 3.2. Modificar Asignatura.
- 3.3. Eliminar Asignatura.
- 3.4. Buscar Asignatura.

4. El sistema debe permitir Gestionar Departamento.

- 4.1. Adicionar Departamento.
- 4.2. Modificar Departamento.
- 4.3. Eliminar Departamento.
- 4.4. Buscar Departamento.

5. El sistema debe permitir Gestionar Grupo.

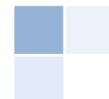
- 5.1. Adicionar Grupo.
- 5.2. Eliminar Grupo.
- 5.3. Buscar Grupo.

6. El sistema debe permitir Gestionar Local.

- 6.1. Adicionar Local.
- 6.2. Eliminar Local.
- 6.3. Buscar Local

7. El sistema debe permitir Gestionar Profesor.

- 7.1. Adicionar Profesor.
- 7.2. Modificar Profesor.





- 7.3. Eliminar Profesor.
- 7.4. Vincular Profesor – Asignatura.
- 7.5. Vincular Profesor – Asignatura – Grupo.
- 7.6. Eliminar Vínculo Profesor – Asignatura.
- 7.7. Eliminar Vínculo Profesor – Asignatura – Grupo.
- 7.8. Buscar Profesor.

8. El sistema debe permitir Gestionar P1 Asignatura.

- 8.1. Adicionar Semana al P1 Asignaturas.
- 8.2. Adicionar Actividades a la semana.
- 8.3. Eliminar Actividades de la Semana.
- 8.4. Eliminar Asignatura del P1.

9. El sistema debe permitir Gestionar Solicitud.

- 9.1. Adicionar Solicitud.
- 9.2. Modificar Solicitud.
- 9.3. Eliminar Solicitud.
- 9.4. Estado de las Solicitudes.

10. El sistema debe permitir Gestionar Afectación.

- 10.1. Aprobar o Rechazar Solicitud.
- 10.2. Eliminar Afectación.

11. El sistema debe permitir Consultar Reporte.

- 11.1. Selección del Reporte deseado.

12. El sistema debe permitir Publicar Horario.

- 12.1. Publicar Horario de la semana.

13. El sistema debe permitir Planificar Horario.

- 13.1. Insertar Datos en el Horario.

Después de haber definido los requisitos funcionales del sistema se muestra el Diagrama de Casos de Uso del Sistema (DCUS), (*Imagen 2.3*), en el cual se representa la relación que existe entre los actores y los Casos de Uso.





2.4 Diagrama de Casos de Uso del Sistema.

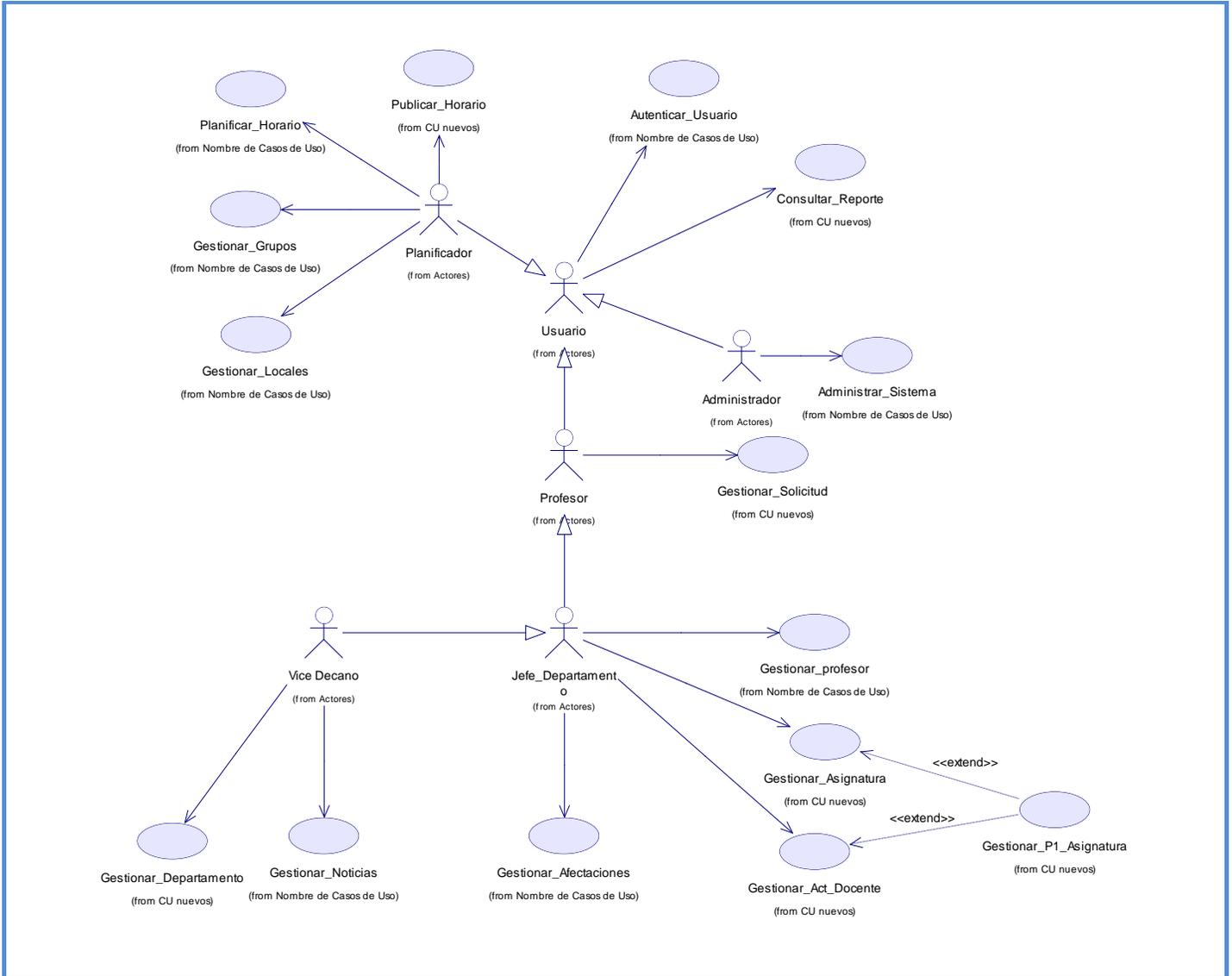
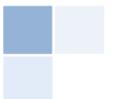


Imagen 2.3: Diagrama de Casos de Uso del Sistema.





El surgimiento de los requisitos planteados anteriormente conllevó a definir nuevos casos de uso del sistema mostrados en el DCUS y los que ya estaban definidos sufrieron algunos cambios de forma tal que se ajustaran a las nuevas necesidades del producto, algunos se muestran a continuación, los restantes se encuentran en los [Anexos](#), **Anexo #1**.

2.5 Descripción de los Casos de Uso Críticos.

Nombre del Caso de Uso	Gestionar Grupos
Actores	Planificador.
Propósito	Este caso de uso tiene como propósito gestionar los grupos que pertenecen a la facultad.
Resumen	El caso de uso comienza cuando el Planificador accede a adicionar, eliminar y/o modificar un grupo.
Referencias	RF 5.
Precondiciones	-
Poscondiciones	Se adiciona, modifica o se elimina un grupo perteneciente a la facultad.
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema
1. El Planificador dirige al menú “Gestionar Datos, Grupo”.	1.1 El sistema le brinda la posibilidad de “Adicionar Grupo” o “Eliminar Grupo” o “Modificar Grupo”.
2. El Planificador selecciona una de las opciones presentadas.	2.1 Si selecciona “Adicionar Grupo” ver sección de Adicionar Grupo . - Si selecciona “Eliminar Grupo” ver sección de Eliminar Grupo . - Si selecciona “Modificar Grupo” ver sección de Modificar Grupo .
Sección “Adicionar Grupo”	
Acciones del Actor	Respuesta del Sistema
3. El Planificador despliega el campo Adicionar Grupo.	3.1 El sistema muestra una interfaz para adicionar un nuevo grupo.
4. El Planificador entra los datos del grupo (Año y Nombre del Grupo) para adicionar un nuevo grupo al sistema.	4.1 El sistema verifica que todos los campos obligatorios estén llenos. 4.2 El sistema verifica que el nombre no aparezca en la base de datos. 4.3 Se insertan los datos del grupo en el sistema. 4.4 Se muestra un mensaje informando que fue adicionado un nuevo Grupo y termina el Caso de Uso.
Curso alternativo	
Acciones del Actor	Respuesta del Sistema
	44.1 El sistema muestra un mensaje informando que no se llenaron todos los campos.





	4.2 El Sistema muestra un mensaje informando que el grupo ya existe.
Sección “Eliminar Grupo”	
Acciones del Actor	Respuesta del Sistema
3. El Planificador despliega el campo Eliminar Grupo.	3.1 El sistema muestra una interfaz para eliminar un Grupo.
4. El Planificador selecciona el grupo y oprime el botón eliminar.	4.1 El sistema muestra un mensaje informando que el grupo fue eliminado y termina el caso de uso.
Curso alternativo	
Acciones del Actor	Respuesta del Sistema
	4.1 El sistema muestra un mensaje informando que no ha seleccionado ningún grupo para eliminar.
Sección “Modificar Grupo”	
Acciones del Actor	Respuesta del Sistema
3. El Planificador despliega el campo Modificar Grupo.	3.1 El sistema muestra una interfaz para modificar un nuevo grupo.
4. El Planificador entra los datos del grupo (Año y Nombre del Grupo) para modificar al grupo.	4.1 El sistema verifica que todos los campos obligatorios estén llenos. 4.2 El sistema verifica que el nombre no aparezca en la base de datos 4.3 Se modifican los datos del grupo en el sistema. 4.4 Se muestra un mensaje informando que fue modificado el Grupo y termina el Caso de Uso.
Curso alternativo	
Acciones del Actor	Respuesta del Sistema
	4.1 El sistema muestra un mensaje informando que no se llenaron todos los campos. 4.2 El Sistema muestra un mensaje informando que el grupo ya existe.
Prioridad	Crítico.

Tabla 2. 1: Descripción del Casos de Uso del Sistema - Gestionar Grupos.

Nombre del Caso de Uso	Gestionar P1 Asignatura
Actores	Jefe de Departamento.
Propósito	El caso de uso tiene como propósito gestionar el P1 de cada una de las asignaturas.
Resumen	El caso de uso comienza cuando el Jefe de Departamento accede a adicionar semana al P1, adicionar actividades a la semana, eliminar actividades de la semana y/o eliminar el P1 de las asignaturas.
Referencias	RF 8.
Precondiciones	Tienen que estar creada las asignaturas y las actividades docentes.
Poscondiciones	Se adiciona las semanas al P1, se adicionan las actividades, se eliminan las actividades de la semana o se elimina el P1 de las Asignaturas.
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema



1. El Jefe de Departamento dirige al menú “Gestionar Datos, P1 Asignaturas”	1.1 El sistema le brinda la posibilidad de “Adicionar Semana al P1 de la Asignatura” o “Adicionar Actividades a la Semana” o “Eliminar P1 de la Asignatura” o “Actividades de la Semana”.
2. El Jefe de Departamento selecciona una de las opciones presentadas.	<p>2.1 Si selecciona “Adicionar Semana al P1 de la Asignatura” ver sección de Adicionar Semana al P1 de la Asignatura.</p> <p>- Si selecciona “Adicionar Actividades a la Semana” ver sección de Adicionar Actividades a la Semana.</p> <p>- Si selecciona “Eliminar P1 de la Asignatura” ver sección de Eliminar P1 de la Asignatura.</p> <p>- Si selecciona “Eliminar Actividades de la Semana” ver sección de Eliminar Actividades de la Semana.</p>
Sección “Adicionar Semana al P1 de la Asignatura”	
Acciones del Actor	Respuesta del Sistema
3. El Jefe de Departamento despliega el campo Adicionar Semana al P1 de la Asignatura.	3.1 El sistema muestra una interfaz para adicionar Semana al P1 de la Asignatura.
4. El Jefe de Departamento selecciona la asignatura y la semana y oprime el botón Aceptar.	<p>4.1 El sistema verifica que todos los campos obligatorios estén llenos.</p> <p>4.2 Se insertan los datos de la en el sistema.</p> <p>4.5 Se muestra un mensaje informando que fue adicionado una nueva semana al P1 de la Asignatura y termina el Caso de Uso.</p>
Curso alternativo	
Acciones del Actor	Respuesta del Sistema
	4.1 El sistema emite un mensaje informando que no todos los campos no están llenos.
Sección “Adicionar Actividades a la Semana”	
Acciones del Actor	Respuesta del Sistema
3. El Jefe de Departamento despliega el campo Adicionar Actividades a la Semana.	3.1 El sistema muestra una interfaz para adicionar Actividades a la Semana.
4. El Jefe de Departamento selecciona la asignatura, semana, actividad docente, cantidad de actividades y si son consecutivas y oprime el botón Aceptar.	<p>4.1 El sistema verifica que todos los campos obligatorios estén llenos.</p> <p>4.2 Se insertan los datos de la en el sistema.</p> <p>4.5 Se muestra un mensaje informando que fue adicionado una nueva Actividad a la Semana y termina el Caso de Uso.</p>
Curso alternativo	
Acciones del Actor	Respuesta del Sistema
	4.1 El sistema emite un mensaje informando que no todos los campos no están llenos.
Sección “Eliminar P1 de la Asignatura”	





Acciones del Actor		Respuesta del Sistema
3. El Jefe de Departamento despliega el campo Eliminar P1 de la Asignatura.		3.1 El sistema muestra una interfaz para eliminar el P1 de la Asignatura.
4. El Jefe de Departamento selecciona la asignatura del P1 y oprime el botón eliminar.		4.1 El sistema muestra un mensaje informando que el P1 de la Asignatura fue eliminado y termina el caso de uso.
Curso alternativo		
Acciones del Actor		Respuesta del Sistema
		4.1 El sistema muestra un mensaje informando que no ha seleccionada ningún P1 de la Asignatura para eliminar.
Sección “Eliminar Actividades de la Semana”		
Acciones del Actor		Respuesta del Sistema
3. El Jefe de Departamento despliega el campo Eliminar Actividades de la Semana.		3.1 El sistema muestra una interfaz para eliminar Actividades de la Semana.
4. El Jefe de Departamento selecciona la Asignatura P1 y el Número de la Semana y oprime el botón eliminar.		4.1 El sistema muestra un mensaje informando que la Actividades de la Semana fue eliminada y termina el caso de uso.
Curso alternativo		
Acciones del Actor		Respuesta del Sistema
		4.1 El sistema muestra un mensaje informando que no todos los campos están llenos.
Prioridad:	Crítico.	

Tabla 2. 2: Descripción del Casos de Uso del Sistema - Gestionar P1 Asignatura.

Nombre del Caso de Uso	Gestionar Locales
Actores	Planificador.
Propósito	Este caso de uso tiene como propósito gestionar los locales que pertenecen a la facultad.
Resumen	El caso de uso comienza cuando el Planificador accede a adicionar y/o eliminar un Local.
Referencias	RF 6.

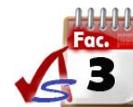




Precondiciones	Tiene que haberse definido el Tipo de Local
Poscondiciones	Se adiciona o se elimina un local perteneciente a la facultad.
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema
1. El Planificador dirige al menú “Gestionar Datos, Locales”	1.1 Ver caso de uso Gestionar Tipo Locales. 1.2 El sistema le brinda la posibilidad de “Adicionar Local” o “Eliminar Local”.
2. El Planificador selecciona una de las opciones presentadas.	2.1 Si selecciona “Adicionar Local” ver sección de Adicionar Local . - Si selecciona “Eliminar Local” ver sección de Eliminar Local .
Sección “Adicionar Local”	
Acciones del Actor	Respuesta del Sistema
3. El Planificador despliega el campo Adicionar Local.	3.1 El sistema muestra una interfaz para adicionar un nuevo Local.
4. El Planificador entra los datos del Local (Número y Tipo de Local) para adicionar un nuevo Local al sistema.	4.1 El sistema verifica que todos los campos obligatorios estén llenos. 4.2 El sistema verifica que no aparezca un local con dos números iguales en la base de datos. 4.3 Se insertan los datos del local en el sistema. 4.4 Se muestra un mensaje informando que fue adicionado un nuevo local y termina el Caso de Uso.
Curso alternativo	
Acciones del Actor	Respuesta del Sistema
	4.1 El sistema muestra un mensaje informando que no se llenaron todos los campos. 4.2 El Sistema muestra un mensaje informando que el Local ya existe
Sección “Eliminar Local”	
Acciones del Actor	Respuesta del Sistema
3. El Planificador despliega el campo Eliminar Local.	3.1 El sistema muestra una interfaz para eliminar un Local.
4. El Planificador selecciona el Local y oprime el botón eliminar.	4.1 El sistema muestra un mensaje informando que el Local fue eliminado y termina el caso de uso.
Curso alternativo	
Acciones del Actor	Respuesta del Sistema
	4.1 El sistema muestra un mensaje informando que no ha seleccionado ningún Local para eliminar.
Prioridad:	Crítico.

Tabla 2. 3: Descripción del Casos de Uso del Sistema - Gestionar Locales.

Nombre del Caso de Uso	Gestionar Asignatura
Actores	Jefe Departamento.
Propósito	Este caso de uso tiene como propósito gestionar las asignaturas que serán



	impartidas en cada uno de los semestres.
Resumen	El caso de uso comienza cuando el Jefe Departamento accede a adicionar, modificar y/o eliminar una Asignatura de su Departamento.
Referencias	RF 3.
Precondiciones	Tienen que estar creados los departamentos.
Poscondiciones	Se adiciona, modifica o se elimina una asignatura perteneciente a su departamento.
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema
1. El Jefe de Departamento dirige al menú “Gestionar Datos, Asignatura”	1.1 El sistema le brinda la posibilidad de “Adicionar Asignatura” o “Eliminar Asignatura” o “Modificar Asignatura”.
2. El Jefe de Departamento selecciona una de las opciones presentadas.	2.1 Si selecciona “Adicionar Asignatura” ver sección de Adicionar Asignatura . - Si selecciona “Eliminar Asignatura” ver sección de Eliminar Asignatura . - Si selecciona “Modificar Asignatura” ver sección de Modificar Asignatura .
Sección “Adicionar Asignatura”	
Acciones del Actor	Respuesta del Sistema
3. El Jefe de Departamento despliega el campo Adicionar Asignatura.	3.1 El sistema muestra una interfaz para adicionar una Asignatura.
4. El Jefe de Departamento entra los datos de la Asignatura (Nombre, Abreviatura y Departamento) para adicionar una Asignatura al sistema.	4.1 El sistema verifica que todos los campos obligatorios estén llenos. 4.2 El sistema verifica que no aparezcan nombres iguales en la base de datos. 4.3 El sistema verifica que no aparezcan abreviaturas iguales en la base de datos. 4.4 Se insertan los datos de la asignatura en el sistema. 4.5 Se muestra un mensaje informando que fue adicionado una nueva asignatura y termina el Caso de Uso.
Curso alternativo	
Acciones del Actor	Respuesta del Sistema
	4.1 El sistema muestra un mensaje informando que no se llenaron todos los campos. 4.2 El Sistema muestra un mensaje informando que el Nombre ya existe. 4.3 El Sistema muestra un mensaje informando que la abreviatura ya existe.
Sección “Eliminar Asignatura”	
Acciones del Actor	Respuesta del Sistema
3. El Jefe de Departamento despliega el campo Eliminar Asignatura.	3.1 El sistema muestra una interfaz para eliminar una Asignatura.



4. El Jefe de Departamento selecciona la Asignatura y oprime el botón eliminar.	4.1 El sistema muestra un mensaje informando que la Asignatura fue eliminada y termina el caso de uso.
Curso alternativo	
Acciones del Actor	Respuesta del Sistema
	4.1 El sistema muestra un mensaje informando que no ha seleccionada ninguna Asignatura para eliminar.
Sección “Modificar Asignatura”	
Curso Normal de los Eventos	Curso Normal de los Eventos
3. El Jefe de Departamento despliega el campo Modificar Asignatura.	3.1 El sistema muestra una interfaz para modificar una Asignatura.
4. El Jefe de Departamento entra los datos de la Asignatura (Nombre, Abreviatura y Departamento) para modificar una Asignatura en el sistema.	4.1 El sistema verifica que todos los campos obligatorios estén llenos. 4.2 El sistema verifica que no aparezcan nombres iguales en la base de datos. 4.3 El sistema verifica que no aparezcan abreviaturas iguales en la base de datos. 4.4 Se actualizan los datos de la asignatura en el sistema. 4.4 Se muestra un mensaje informando que fue modificado la asignatura y termina el Caso de Uso.
Curso alternativo	
Acciones del Actor	Respuesta del Sistema
	4.1 El sistema muestra un mensaje informando que no se llenaron todos los campos. 4.2 El Sistema muestra un mensaje informando que el Nombre ya existe. 4.3 El Sistema muestra un mensaje informando que la abreviatura ya existe.
Prioridad:	Crítico.

Tabla 2. 4: Descripción del Casos de Uso del Sistema - Gestionar Asignatura.

Nombre del Caso de Uso	Planificar Horario
Actores	Planificador.
Propósito	Este caso de uso tiene como propósito gestionar el horario en cada una de las semanas de cada semestre.
Resumen	El caso de uso comienza cuando el planificador accede a planificar un horario para cada semana del semestre.
Referencias	RF 13
Precondiciones	Tienen que estar definidos los locales, grupos, asignaturas y el P1 de cada asignatura.
Poscondiciones	Se planifica el horario de cada una de las semanas.
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema
1. El planificador se dirige al menú “Gestionar	1.1 El sistema le brinda la posibilidad de “Planificar Horario”



Datos, Planificar Horario”.	según la semana.
2. El planificador selecciona la semana a planificar.	2.1 El sistema muestra le brinda la posibilidad de planificar el horario por “año” o por “Locales”.
3. El planificador selecciona una de las opciones presentadas.	3.1 Si selecciona “año” ver sección de Año . - Si selecciona “Locales” ver sección de Locales .
Sección “Año”	
Acciones del Actor	Respuesta del Sistema
4. El Planificador selecciona el año, turno, día de la semana y da un clic.	4.1 El sistema muestra una interfaz para seleccionar las actividades del turno de clase.
5. El planificador selecciona la Actividad Doc, Asignatura, Locales y oprime el botón Aceptar.	5.1 El sistema verifica que no se quede ningún campo vacío. 5.2 Se insertan los datos en el sistema. 5.3 Se muestran los datos insertados en el sistema y termina el caso de uso.
Curso alternativo	
Acciones del Actor	Respuesta del Sistema
	5.1 El sistema emite un mensaje informando que no todos los campos están llenos.
Sección “Locales”	
Acciones del Actor	Respuesta del Sistema
4. El Planificador selecciona el local, día de la semana, turno y da un clic.	4.1 El sistema muestra una interfaz para seleccionar las actividades del turno de clase.
5. El planificador selecciona la Actividad Doc, Asignatura, Locales y oprime el botón Aceptar.	5.1 El sistema verifica que no se quede ningún campo vacío. 5.2 Se insertan los datos en el sistema. 5.3 Se muestran los datos insertados en el sistema y termina el caso de uso.
Curso alternativo	
Acciones del Actor	Respuesta del Sistema
	5.1 El sistema emite un mensaje informando que no todos los campos están llenos.
Prioridad:	Crítico.

Tabla 2. 5: Descripción del Casos de Uso del Sistema - Planificar Horario



También se realizaron los diagramas de clases del diseño para dichos casos de uso los cuales se muestran a continuación y los restantes se encuentran en los [Anexos, Anexo #2](#).

2.6 Diagramas de Clases del Diseño.

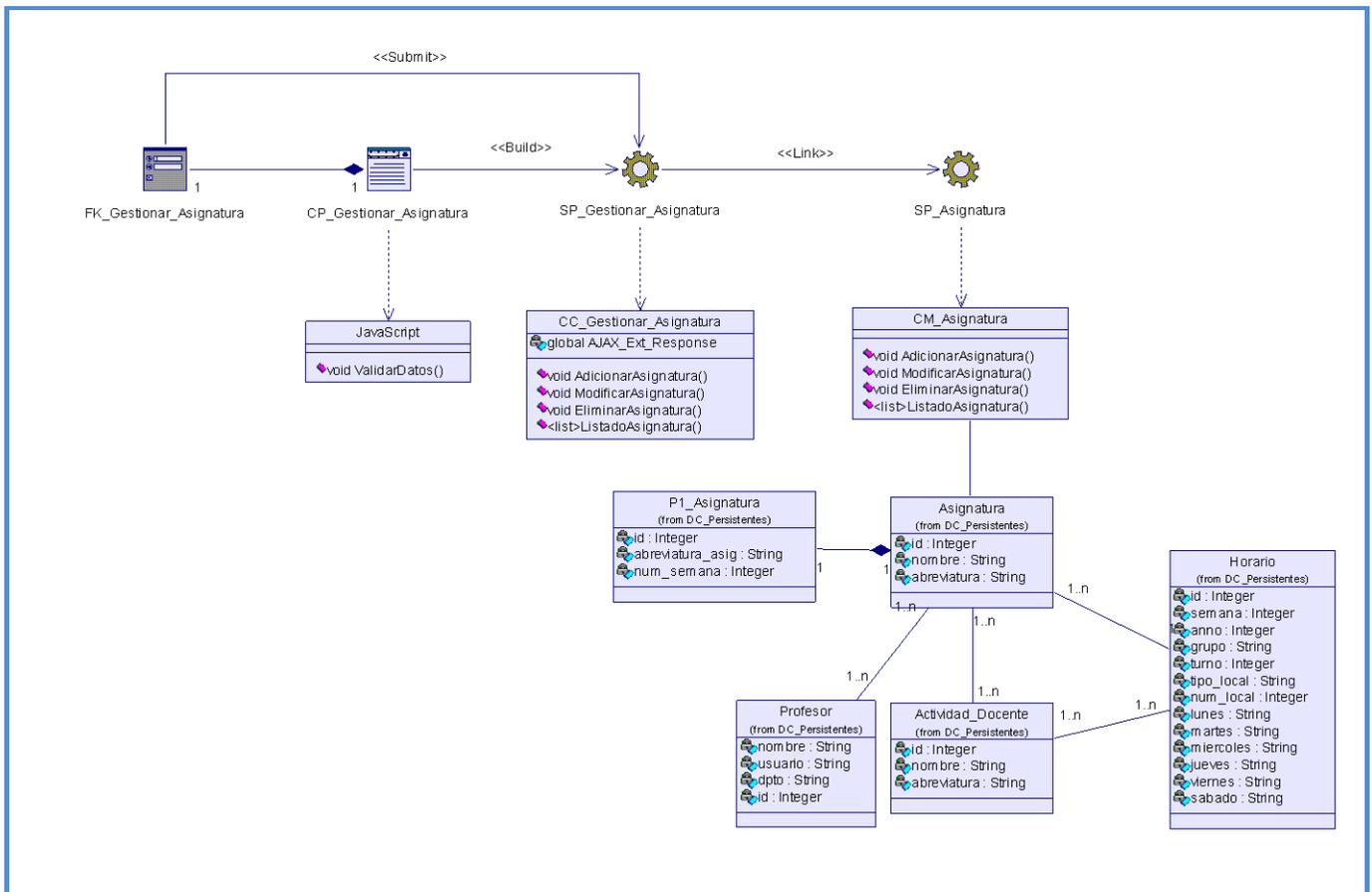
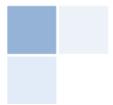


Imagen 2. 4: Diagrama de Clases del Diseño del Caso de Uso del Sistema – Gestionar Asignatura.



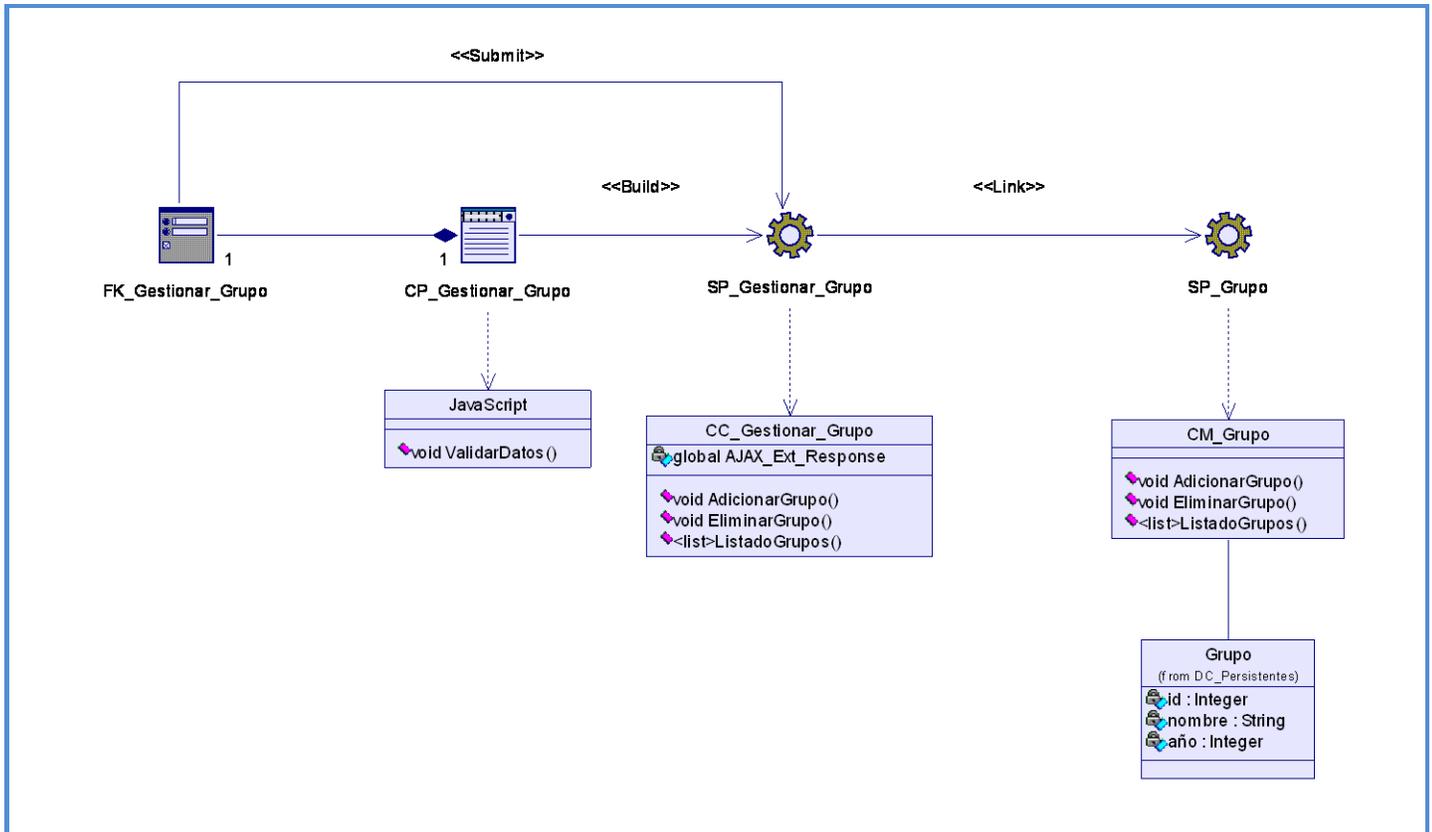
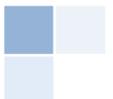


Imagen 2. 5: Diagrama de Clases del Diseño del Caso de Uso del Sistema – Gestionar Grupo.



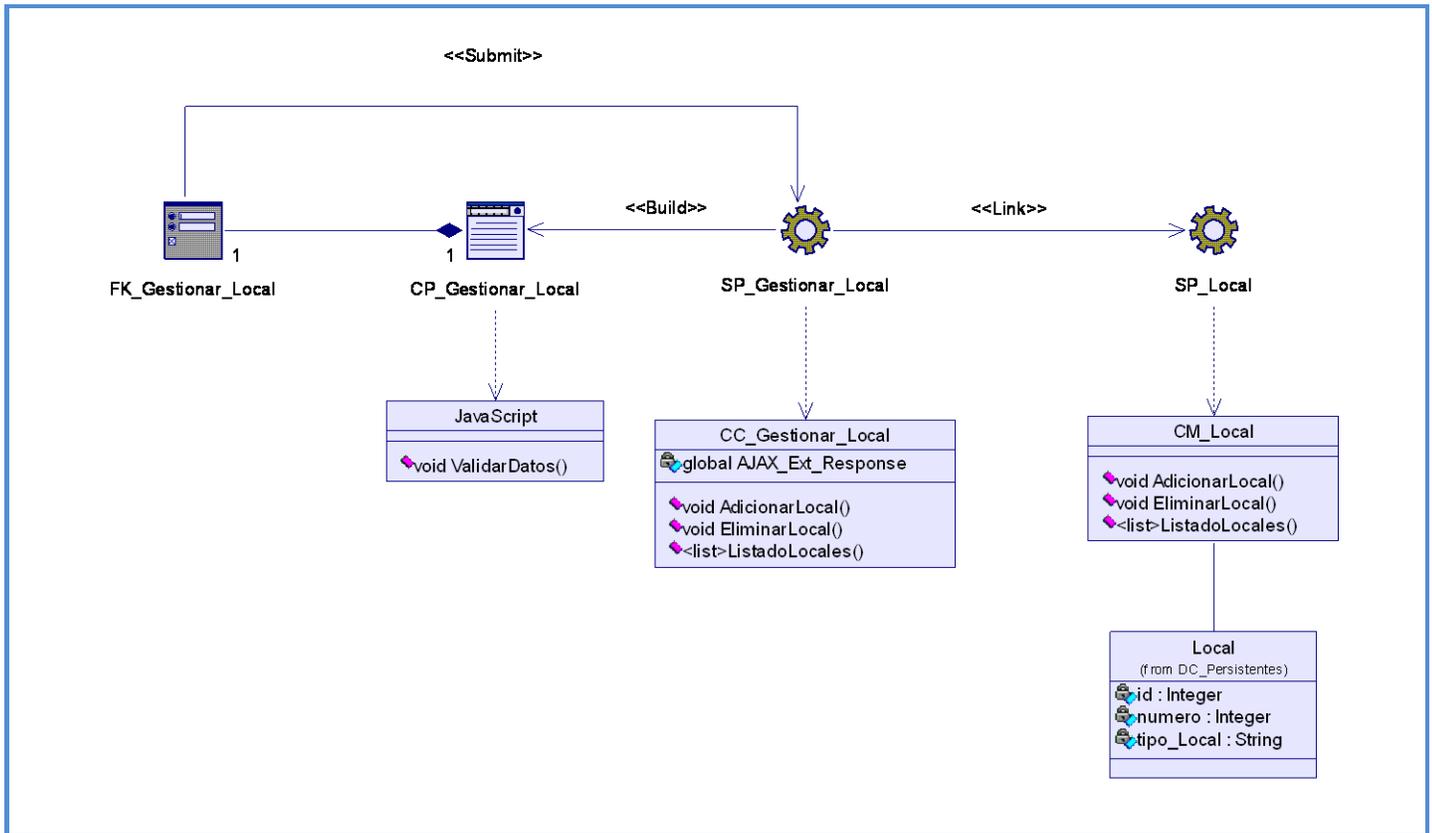
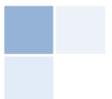


Imagen 2. 6: Diagrama de Clases del Diseño del Caso de Uso del Sistema – Gestionar Local.



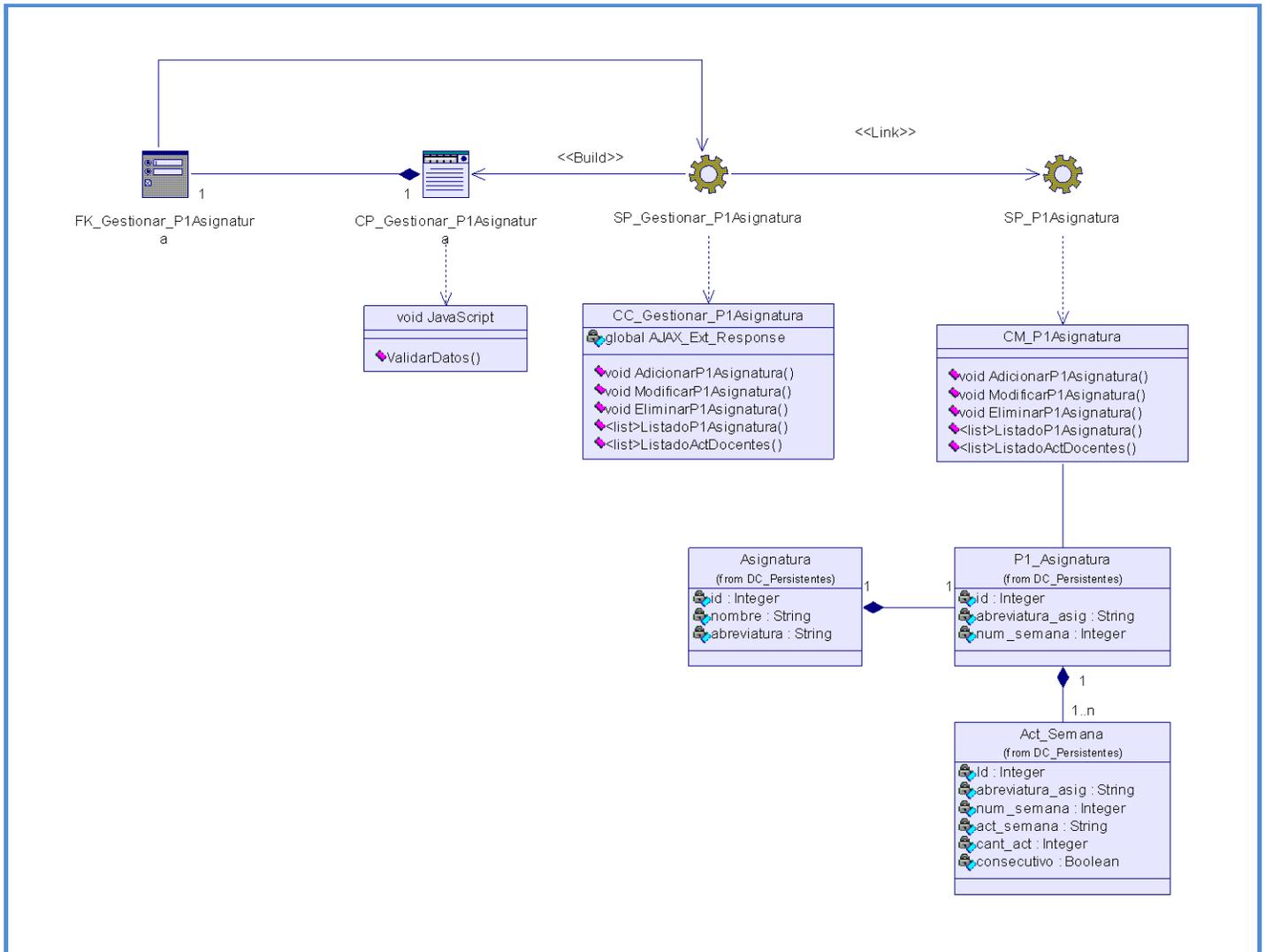
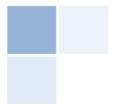


Imagen 2. 7: Diagrama de Clases del Diseño del Caso de Uso del Sistema – Gestionar P1 Asignatura.



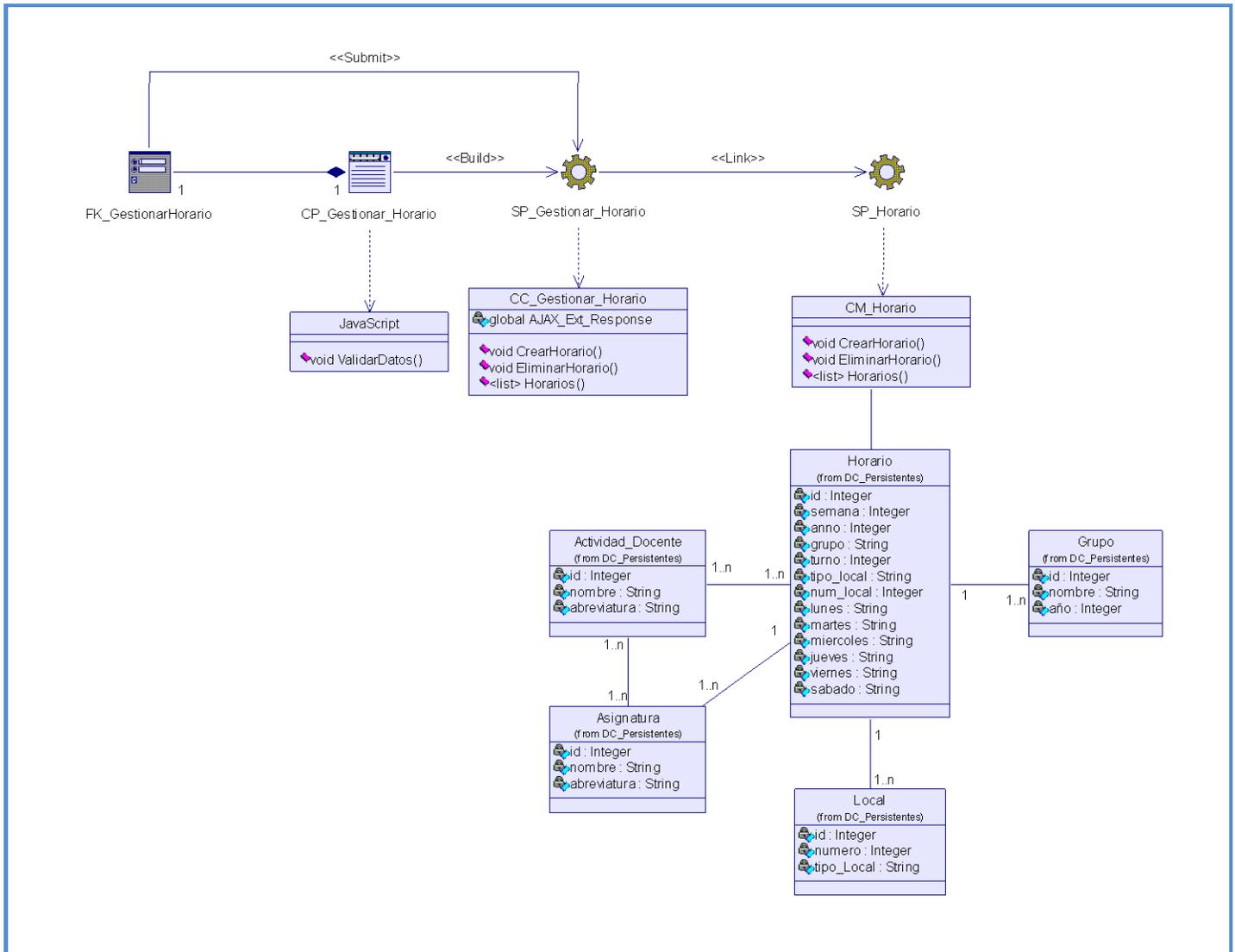
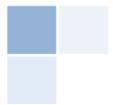


Imagen 2. 8: Diagrama de Clases del Diseño del Caso de Uso del Sistema – Planificar Horario.



2.7 Diagrama de Clases Persistentes.

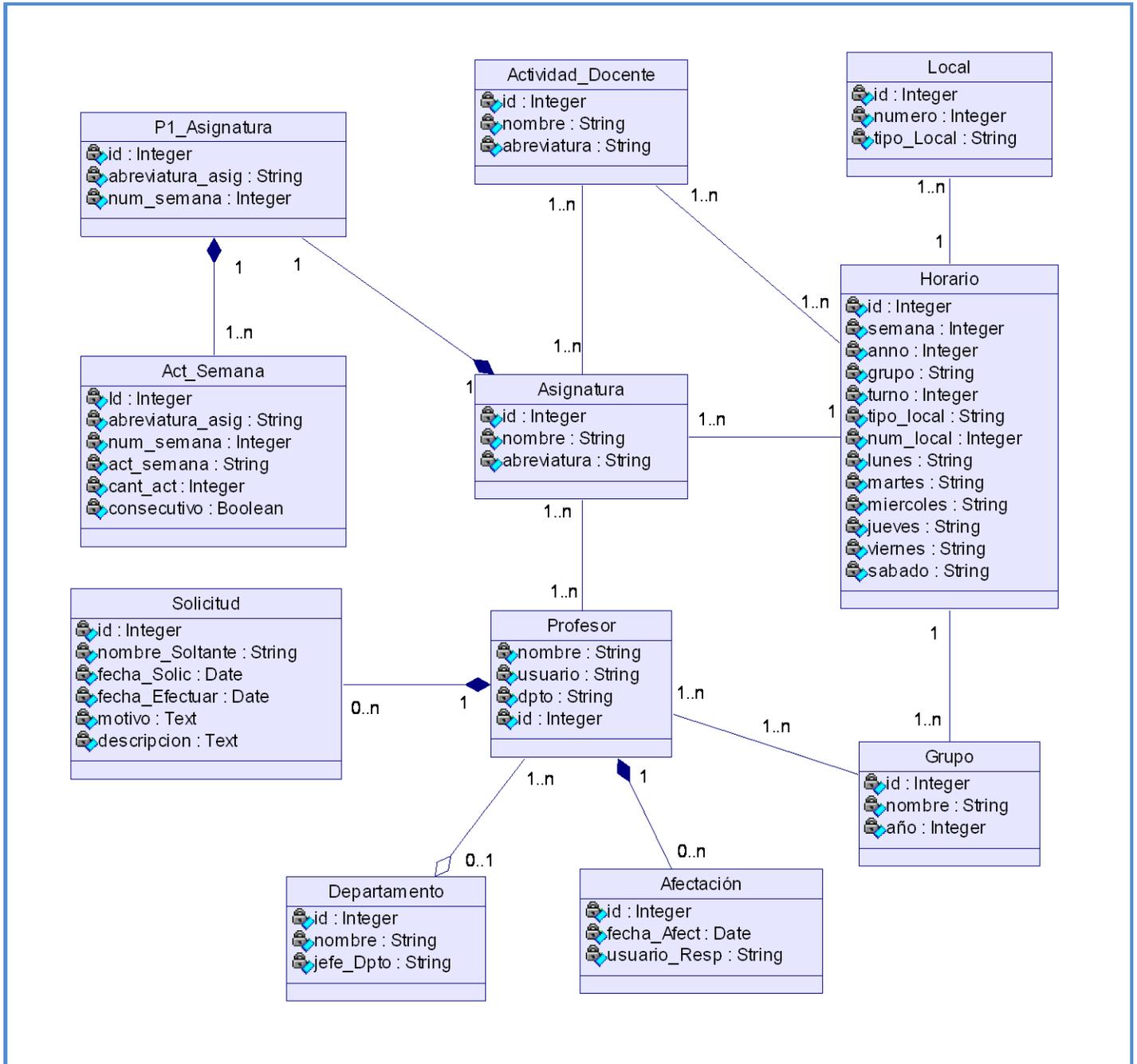
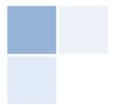


Imagen 2. 9: Diagrama de Clases Persistentes.





2.8 Modelo de la Base de Datos.

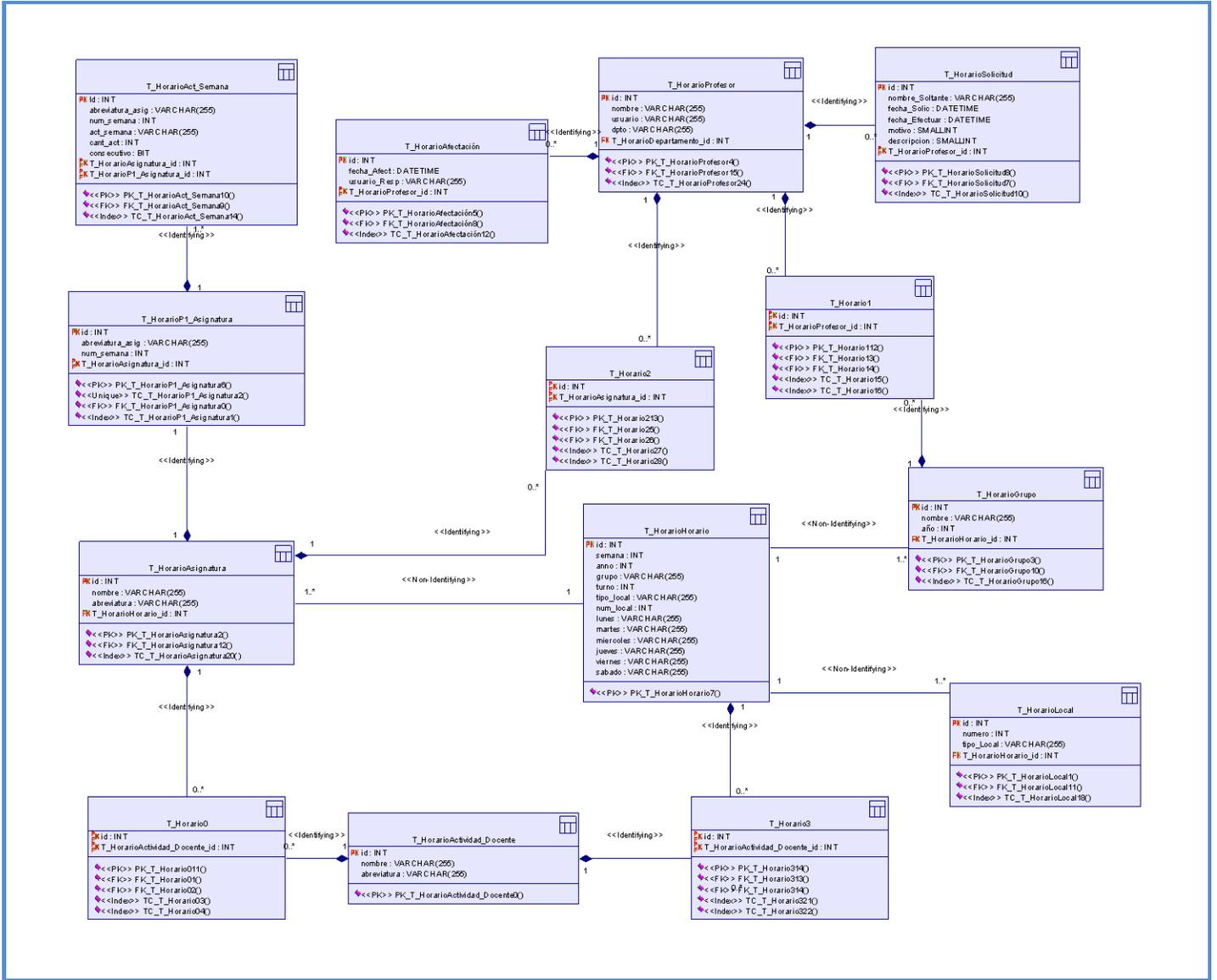


Imagen 2.10: Modelo de la Base de Datos.



2.9 Descripción de las Clases.

Joomla consta de tres clases bases que responden a la implementación del patrón de arquitectura en tres capas Modelo - Vista - Controlador. La clase JView tiene como función definir la apariencia que tiene el sitio. La clase JModel es la encargada de realizar todas las operaciones sobre la Base de Datos y la clase JController es el intermediario entre las clases mencionadas anteriormente, la cual maneja todas las funciones del sitio web.

Nombre: JView	
Tipo de Clase: Entidad	
Atributos	Tipo
\$basePath	string
\$defaultModel	string
\$layout	string
\$layoutExt	string
\$models	string
\$name	string
\$path	string
Responsabilidades	
Nombre	Descripción
JView construct ([\$config = array()])	Permite crear una nueva instancia en las clases descendientes.
void addHelperPath (string/array \$path)	Permite adicionar un script de ayuda a la pila en orden LIFO.
void addTemplatePath (string/array \$path)	Permite adicionar un template a la pila en orden LIFO.
bool assign ()	Permite asignar valores a una variable.
bool assignRef (string \$key, mixed &\$val)	Permite asignar valor a una variable por referencia.
void display ([string \$tpl = null])	Ejecuta y muestra un template.
Mixed escape (mixed \$var)	Devuelve un valor de salida de una variable.
Mixed get (string \$property, [string \$default = null])	Devuelve un dato contenido en el modelo o una propiedad de la vista.
string getLayout ()	Devuelve el nombre del layout.
mixed getModel ([string \$name = null])	Devuelve el nombre del modelo.





string getName ()	Devuelve el nombre de la vista.
boolean loadHelper ([\$hlp = null], string \$tpl)	Carga un archivo de ayuda.
string loadTemplate ([string \$tpl = null])	Carga un template.
void setEscape (mixed \$spec)	Cambia la función escape()
string setLayout (\$layout, string \$template)	Cambia el nombre del layout
string setLayoutExt (string \$value)	Permite el uso de una extensión diferente para el layout.
object setModel (&\$model, [boolean \$default = false], object \$model)	Adiciona un modelo a la vista.
void addPath (\$type, string/array \$path)	Adiciona un camino de búsqueda para los templates y recursos.
void setPath (string \$type, string/array \$path)	Cambia el arreglo de caminos de búsqueda para templates o recursos.

Tabla 2. 6: Descripción de la Clase Entidad del CMS Joomla JView.

Nombre: JModel	
Tipo de Clase: Entidad	
Atributos	Tipo
\$db	object
\$name	string
\$state	string
Responsabilidades	
Nombre	Descripción
void addTablePath (string/array \$path)	Adiciona una tabla al modelo.
JModel construct ([\$config = array()])	Define el constructor de la clase.
array addIncludePath ([string \$path = "])	Adiciona un directorio de búsqueda.
Object getDBO ()	Devuelve el objeto de conexión de la base de datos.
mixed getInstance (string \$type, [string \$prefix = "], [array \$config = array()])	Devuelve una referencia del objeto del modelo.
String getName ()	Devuelve el nombre del modelo.
object getState ([string \$property = null])	Devuelve la variable de estado del modelo.
object getTable ([string \$name = "], [string \$prefix = 'Table'], [array \$options = array()])	Devuelve el la instancia de una tabla.
void setDBO (&\$db, object \$db)	Cambia el objeto de conexión de la base de datos.
Mixed setState (string \$property, [mixed \$value = null])	Cambia la variable de estado del modelo.
Array getList (string \$query, [int \$limitstart = 0], [int \$limit = 0])	Devuelve la instancia de una lista
int getListCount (string \$query)	Devuelve el número de campos obtenidos de ejecutar una





	consulta.
--	-----------

Tabla 2. 7: Descripción de la Clase Entidad del CMS Joomla JModel.

Nombre: JController	
Tipo de Clase: Controladora	
Atributos	Tipo
\$acoSection	string
\$acoSectionValue	string
\$basePath	string
\$doTask	string
\$message	string
\$messageType	string
\$methods	array
\$name	array
\$path	array
\$redirect	string
\$task	string
\$taskMap	array
Responsabilidades	
Nombre	Descripción
void addModelPath (string/array \$path)	Adiciona a la pila un modelo en orden LIFO.
void addViewPath (string/array \$path)	Adiciona una o más vistas a la pila del controlador.
JController construct ([array \$config = array()])	Implementa el constructor de la clase.
boolean authorize (string \$task)	Chequea la autorización.
void display ([string \$cachable = false])	Cambia la función de salida.
Mixed execute (string \$task)	Ejecuta una tarea.
object getModel ([string \$name = "], [string \$prefix = "], [array \$config = array()])	Devuelve una instancia del modelo.
string getName ()	Devuelve el nombre del controlador.
string getTask ()	Devuelve la instancia de la última tarea.
array getTasks ()	Devuelve la tarea que se está ejecutando en el controlador.
object getView ([string \$name = "], [string \$type= "], [string \$prefix = "], [array \$config = array()])	Devuelve una referencia de la vista actual.





boolean redirect ()	Re direcciona la búsqueda.
void registerDefaultTask (string \$method)	Registra una tarea definida por defecto.
void registerTask (string \$task, string \$method)	Registra una tarea.
void setAccessControl (string \$section, [string \$value = null])	Cambia el nivel de control de acceso.
string setMessage (string \$text)	Cambia un mensaje interno.
void setRedirect (string \$url, [string \$msg = null], [string \$type = 'message'])	Cambia la URL para la búsqueda.
void addPath (string \$type, string/array \$path)	Adiciona el arreglo de caminos de búsqueda para templates y recursos.
void setPath (string \$type, string/array \$path)	Cambia el arreglo de caminos de búsqueda para templates y recursos.

Tabla 2. 8: Descripción de la Clase Controladora del CMS Joomla Controller.

2.10 Conclusiones.

En este capítulo se analizó el diseño propuesto por el analista, lo cual permitió definir un nuevo diseño para la solución que se está desarrollando, se definieron un conjunto de requisitos funcionales bajo los cuales el sistema será implementado, se definió un nuevo DCUS en el cual se muestran los actores del sistema con los respectivos casos de uso que se encuentran relacionados, se describieron los casos de uso generados por los requisitos funcionales, se diseñaron las clases de diseño, se realizó el modelo de base de datos y se describieron las clases que serán implementadas en esta solución informática.





Capítulo 3: Propuesta de Solución.

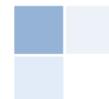
3.1 *Introducción.*

En el presente capítulo se realizará el modelo de despliegue e implementación, en el cual se desarrollan los diagramas de despliegue y componentes respectivamente. También se efectuarán las pruebas de funcionalidad al sistema como colofón imprescindible para verificar que el software cumpla con los requerimientos definidos en el capítulo anterior.

3.2 *Modelo de Despliegue.*

El **Modelo de Despliegue** es un modelo que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. El Modelo de Despliegue se utiliza como entrada fundamental de las actividades de diseño e implementación debido a que la distribución del sistema tiene una influencia principal en su diseño. En dicho modelo cada nodo representa un recurso de cómputo, normalmente un procesador o un dispositivo de hardware similar. Los nodos poseen relaciones que representan medios de comunicación entre ellos.

La aplicación necesitará tres procesadores: el servidor de la aplicaciones, el cual se conectará directamente al servidor de base de datos mediante el protocolo de comunicación TCP/IP y las PC clientes se conectarán a dicho servidor mediante el protocolo HTTP. Al menos una de las PCs clientes, utilizará una impresora la cual estará conectada mediante el puerto USB y en otro caso esta impresora será un recurso compartido en la red utilizando un Print Server mediante el protocolo TCP/IP. El diagrama que se muestra a continuación representa un modelo compuesto por cuatro nodos y la conexión entre ellos.



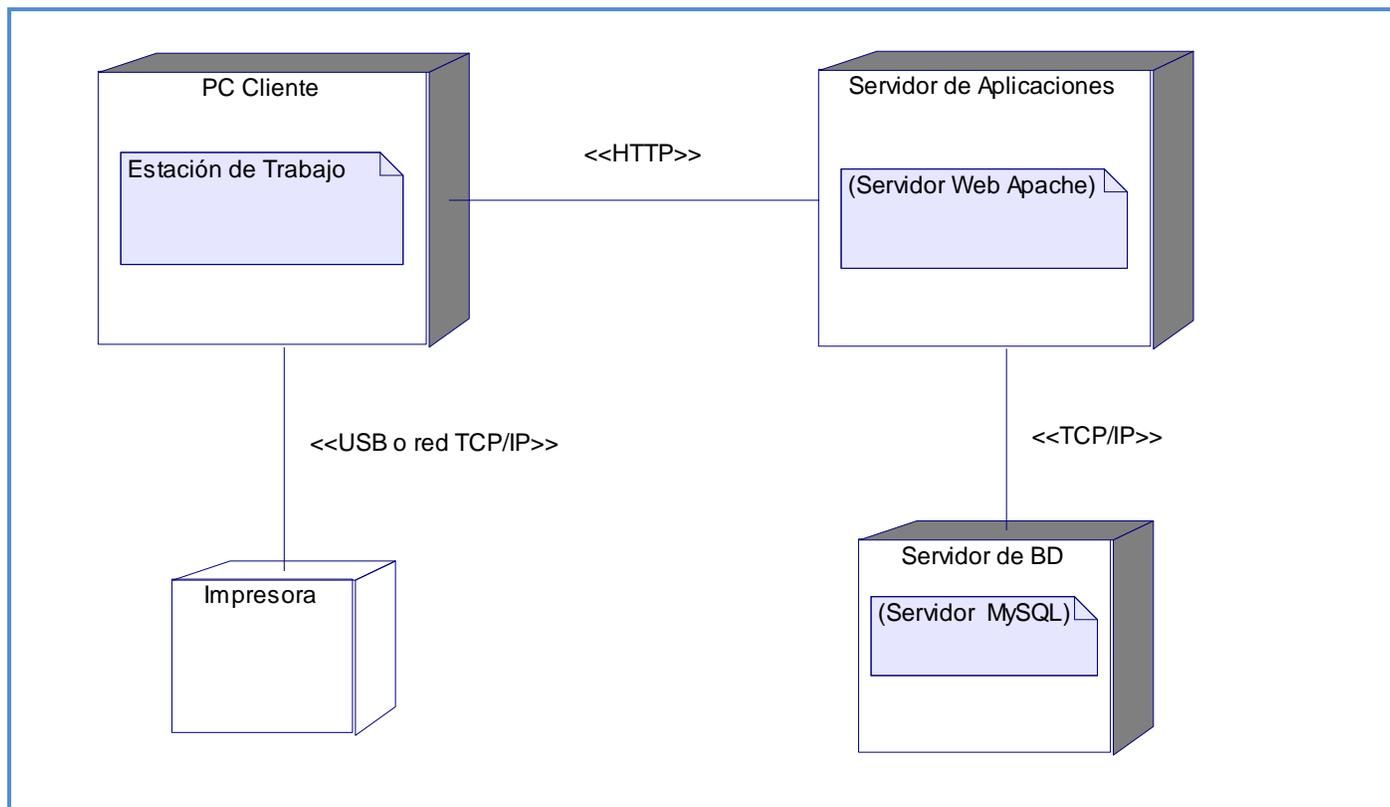


Imagen 3. 1: Diagrama de Despliegue. Configuración de nodos de procesamiento en tiempo de ejecución y los componentes que residen en ellos.

3.3 Modelo de Implementación.

El **Modelo de Implementación** describe como los elementos del diseño se implementan en términos de componentes: ficheros de código fuente y ejecutables. Este modelo describe como se organizan los componentes de acuerdo a los mecanismos de estructuración, el lenguaje o lenguajes de programación utilizados y como dependen los componentes unos de otros.

Un **componente** se corresponde con una o varias clases, interfaces o colaboraciones. Los componentes pertenecen al mundo físico, es decir, representan un bloque de construcción al modelar aspectos físicos de un sistema. Una característica básica de un componente es que: “debe definir una abstracción precisa con





una interfaz bien definida, y permitiendo reemplazar fácilmente los componentes más viejos con otros más nuevos y compatibles.” En UML todos los elementos físicos se modelan como componentes.

Los **diagramas de componentes** se utilizan para modelar la vista estática de un sistema, muestran las organizaciones y dependencias lógicas entre componentes software que formarán un sistema, sean éstos librerías, componentes de código fuente , binarios o ejecutables. Cuando se realiza un diagrama de componentes se tienen en consideración los requisitos relacionados con la facilidad de desarrollo, la gestión del software, la reutilización, y las restricciones impuestas por los lenguajes de programación y las herramientas utilizadas en el desarrollo.

Es importante señalar que no es necesario realizar solo un diagrama de componentes que incluya todos los componentes del sistema sino que este puede dividirse en varios diagramas. En este trabajo los Diagramas de Componentes se realizarán por Casos de Uso.

A continuación se muestran los diagramas de componentes de los casos de uso críticos que presenta el sistema, los cuales son muy similares, cambiando solamente el nombre del caso de uso.

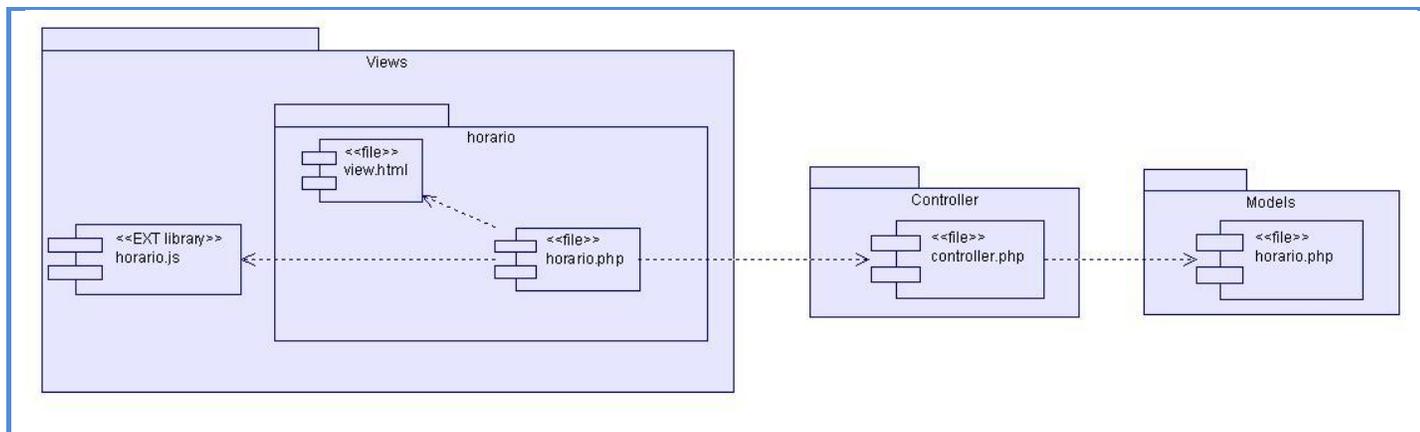


Imagen 3. 2: Diagrama de Componentes del Caso de Uso Gestionar Horario.



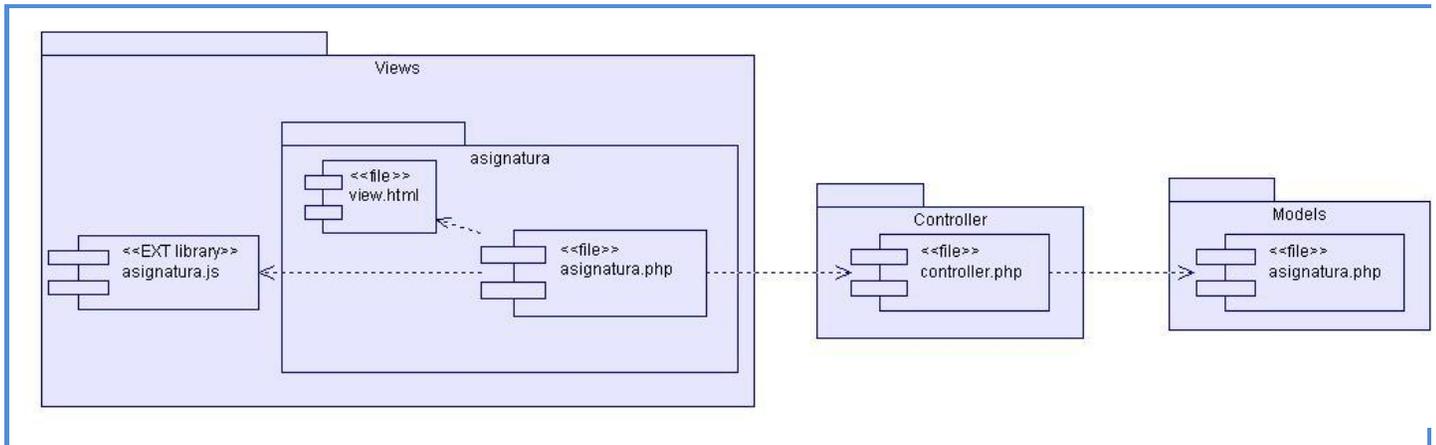


Imagen 3.3: Diagrama de Componentes del Caso de Uso Gestionar Asignatura.

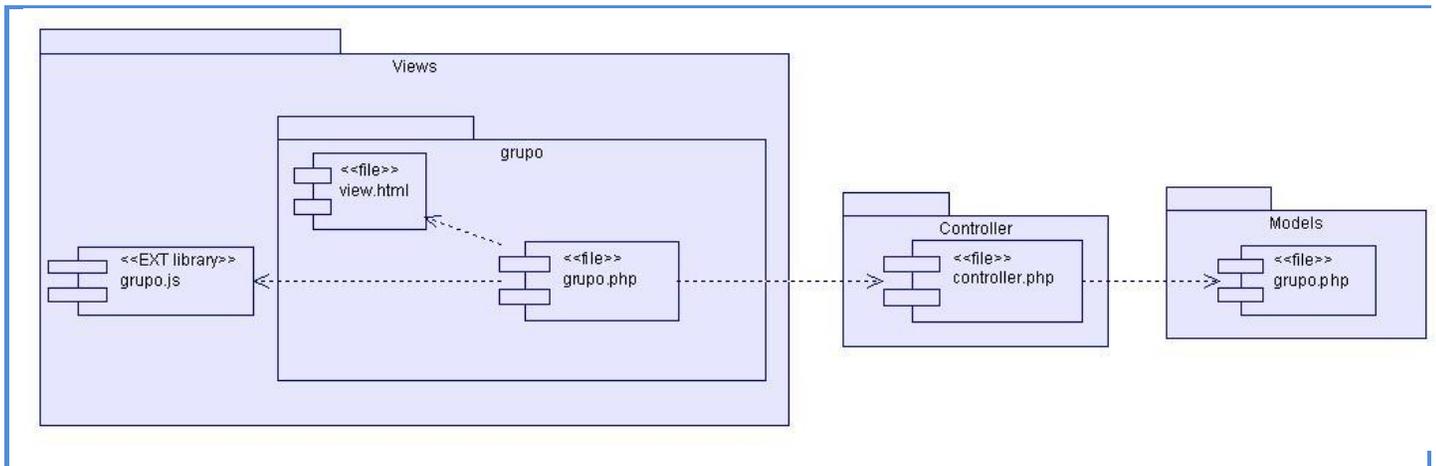


Imagen 3.4: Diagrama de Componentes del Caso de Uso Gestionar Grupo.

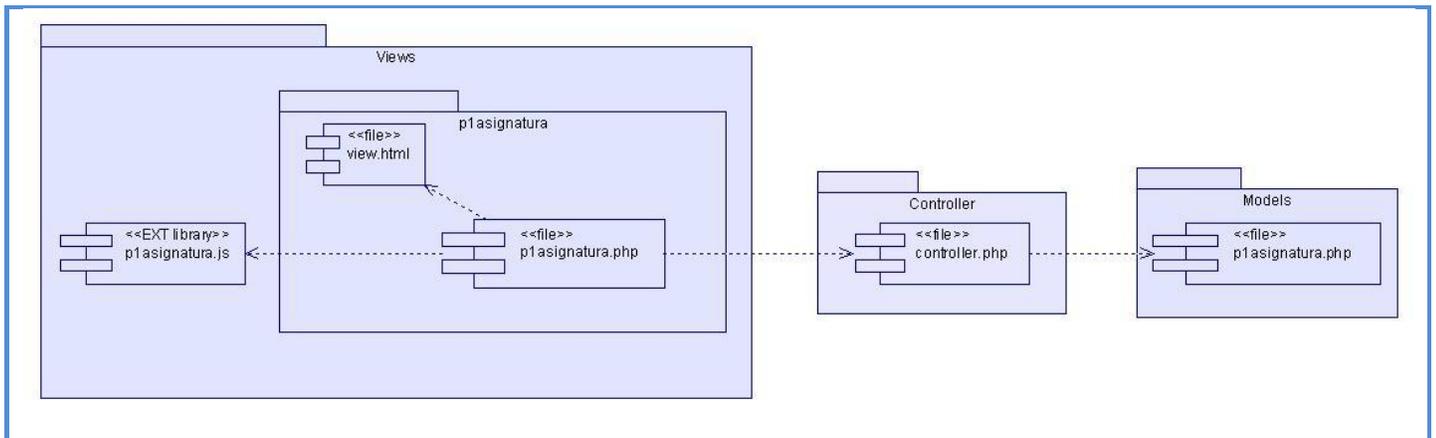
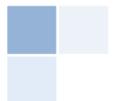


Imagen 3.5: Diagrama de Componentes del Caso de Uso Gestionar P1 Asignatura.



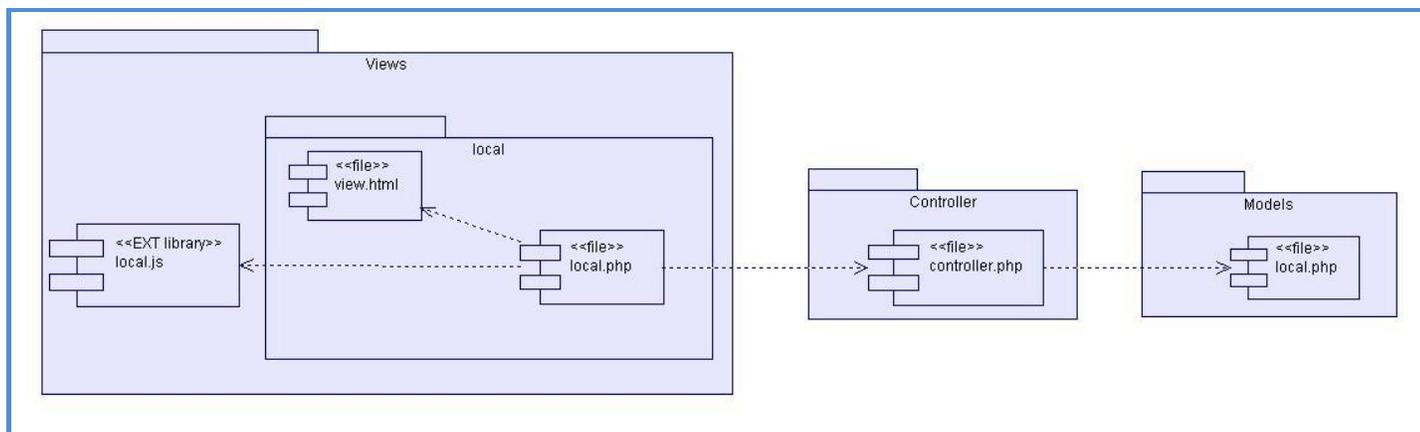


Imagen 3. 6: Diagrama de Componentes del Caso de Uso Gestionar Local.

3.4 Modelo de Pruebas.

Las **Pruebas** son actividades en las cuales un sistema o componente es ejecutado bajo unas condiciones o requerimientos específicos. Los resultados son observados y registrados, y se realiza una evaluación del sistema o componente. Las pruebas verifican los resultados de la implementación del sistema, las cuales no se realizan con el objetivo de demostrar que el sistema funciona correctamente, sino de detectar los errores que tenga el mismo. El modelo de prueba describe cómo han de ser probados aspectos específicos del sistema.

A continuación se muestran los casos de prueba de funcionalidad aplicados al sistema en cuestión.

3.5 Casos de Prueba de Funcionalidad.

Escenario	Nombre.	Abreviatura.	Año.	Departamento.	Asignatura	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
Adicionar Asignatura.	V = Programación 1	= V = P1	V = Se selecciona "1"	I = No seleccionar ningún departamento		El sistema debe mostrar un mensaje de error informando que se debe llenar el campo.	Satisfactorio.	1. Gestionar Datos a. Gestionar Asignatura. b. Adicionar Asignatura.





Escenario	Nombre.	Abreviatura.	Año.	Departamento.	Asignatura	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
	V = Idioma extranjero	V = IE1	I = No se selecciona ningún	V = Seleccionar “Humanidades”		El sistema debe mostrar un mensaje de error informando que se debe llenar el campo.	Satisfactorio.	
	V = Matemática2	I = zlxchhsdfnddffd 89dfdsf98v8cuv9zux	V = Selecciona “2”	V = Seleccionar “Ciencias Básicas”		El sistema debe mostrar un mensaje diciendo que hay un error en la entrada de los datos o que la cadena esta fuera del rango definido.	Satisfactorio.	
	I = khdfkihsdkfksdaaTD 7ASTF7STDFt/t/Tdstus	V = P2	V = Selecciona “2”	V = Seleccionar “Especialidad”		El sistema debe mostrar un mensaje diciendo que hay un error en la entrada de los datos o que la cadena esta fuera del rango definido.	Satisfactorio.	
Eliminar Asignatura					I = No seleccionar ninguna asignatura.	El sistema debe mostrar un mensaje de error informando que se debe llenar el campo.	Satisfactorio.	1. Gestionar Datos a. Gestionar Asignatura. b. Eliminar Asignatura.
Modificar Asignatura.	V = Física1	V = F1	V = Selecciona “2”	V = Seleccionar “Ciencias Básicas”	I = No seleccionar ninguna asignatura	El sistema debe mostrar un mensaje de error informando que se debe llenar el campo.	Satisfactorio.	1. Gestionar Datos a. Gestionar Asignatura. b. Modificar Asignatura.
	V = Maquinas Computadoras	V = MC	V = Selecciona “2”	I = No seleccionar ningún departamento	V = Seleccionar “Maquinas Computadoras”	El sistema debe mostrar un mensaje de error informando que se debe llenar el campo.	Satisfactorio.	
	V = Matemática 3	V = M3	I = No seleccionar ningún	V = Seleccionar “Ciencias Básicas”	V = Selecciona “Ingles”	El sistema debe mostrar un mensaje de error informando que se debe llenar el campo.	Satisfactorio.	





Escenario	Nombre.	Abreviatura.	Año.	Departamento.	Asignatura	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
	V = Filosofía y Sociedad	I = kkjsaieixkc kx668e663 4838434yo liwocy3l	V = Selecciona "1"	V = Selecciona "Humanidades"	V = Selecciona "Filosofía y Sociedad"	El sistema debe mostrar un mensaje diciendo que hay un error en la entrada de los datos o que la cadena esta fuera del rango definido.	Satisfactorio.	
	I = kjc7e4w378w yb8ow48883 838c3y8y3c8	V = FS	V = Selecciona "1"	V = Selecciona "Humanidades"	V = Selecciona "Filosofía y Sociedad"	El sistema debe mostrar un mensaje diciendo que hay un error en la entrada de los datos o que la cadena esta fuera	Satisfactorio.	

Tabla 3. 1: Descripción del Caso de Prueba del Caso de Uso-Gestionar Asignatura.

Escenario	Año	Nombre del Grupo	Grupo	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
Adicionar Grupo	V = 1	I = No seleccionar ningún grupo		El sistema debe mostrar un mensaje de error informando que se debe llenar el campo.	Satisfactorio.	1. Gestionar Datos a. Gestionar Grupo b. Adicionar Grupo
	I = No seleccionar ningún grupo	V = Seleccionar "3101"		El sistema debe mostrar un mensaje de error informando que se debe llenar el campo.	Satisfactorio.	
Eliminar Grupo			V Seleccionar "3204"	El sistema debe mostrar un mensaje notificando que se ha eliminado un grupo docente.	Satisfactorio.	1. Gestionar Datos a. Gestionar Grupo b. Eliminar Grupo
			I = No seleccionar ningún grupo	El sistema debe mostrar un mensaje de error.	Satisfactorio.	

Tabla 3. 2: Descripción del Caso de Prueba del Caso de Uso-Gestionar Grupo.

Escenario	Asignatura	No. Semana	Asignatura P1	Actividad Doc	Cantidad de Actividades	Respuesta del Sistema	Resultado de la Prueba	Flujo Central





Escenario	Asignatura	No. Semana	Asignatura P1	Actividad Doc	Cantidad de Actividades	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
Adicionar semana al P1 de la Asignatura	V = Seleccionar "F1"	I = No seleccionar ningún número de semana				El sistema debe mostrar un mensaje de error informando que se debe llenar el campo.	Satisfactorio.	1. Gestionar Datos a. Gestionar P1 Asignatura. b. Adicionar Semana al P1 de la Asignatura.
	I = No seleccionar ninguna asignatura	V = Seleccionar "1"				El sistema debe mostrar un mensaje de error informando que se debe llenar el campo.	Satisfactorio.	
Adicionar Actividades a la semana.		V = Seleccionar "3"	V = Seleccionar "PPD"	V = Seleccionar "C"	I = No seleccionar ninguna cantidad de actividades	El sistema debe mostrar un mensaje de error informando que se debe llenar el campo.	Satisfactorio.	1. Gestionar Datos a. Gestionar P1 Asignatura. b. Adicionar Actividades a la Semana.
		V = Seleccionar "3"	V = Seleccionar "PPD"	I = No seleccionar ninguna cantidad	V = Seleccionar "1"	El sistema debe mostrar un mensaje de error informando que se debe llenar el campo.	Satisfactorio.	
		V = Seleccionar "3"	I = No seleccionar ninguna	V = Seleccionar "C"	V = Seleccionar "1"	El sistema debe mostrar un mensaje de error informando que se debe llenar el campo.	Satisfactorio.	
		I = No seleccionar "2"	V = Seleccionar "PPD"	V = Seleccionar "S"	V = Seleccionar "1"	El sistema debe mostrar un mensaje de error informando que se debe llenar el campo.	Satisfactorio.	





Escenario	Asignatura	No. Semana	Asignatura P1	Actividad Doc	Cantidad de Actividades	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
Eliminar Actividades de la Semana.		V = Seleccionar "3"	I = No seleccionar ninguna asignatura			El sistema debe mostrar un mensaje de error informando que se debe llenar el campo.	Satisfactorio.	1. Gestionar Datos a. Gestionar P1 Asignatura. b. Eliminar Actividades de la Semana.
		I = No seleccionar "2"	V = Seleccionar "PPD"			El sistema debe mostrar un mensaje de error informando que se debe llenar el campo.	Satisfactorio.	
Eliminar Asignatura del P1.			I = No seleccionar ninguna asignatura			El sistema debe mostrar un mensaje de error informando que se debe llenar el campo.		1. Gestionar Datos a. Gestionar P1 Asignatura. b. Eliminar Asignatura del

Tabla 3. 3: Descripción del Caso de Prueba del Caso de Uso-Gestionar P1 Asignatura.

Escenario	Numero	Tipo de local	Local	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
Adicionar Local	V = 101	I = No seleccionar ningún tipo local		El sistema debe mostrar un mensaje de error informando que se deben llenar los campos.	Satisfactorio.	1. Gestionar Datos a. Gestionar Local. b. Adicionar Local.
	I = 788745848458	V = Seleccionar "Aula"		El sistema debe mostrar un mensaje diciendo que hay un error en la entrada de los datos o que la cadena esta fuera del rango definido.	Satisfactorio.	





Escenario	Numero	Tipo de local	Local	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
Eliminar Local			I = No seleccionar ningún tipo local	El sistema debe mostrar un mensaje de error informando que se deben llenar los campos.	Satisfactorio.	1. Gestionar Datos a. Gestionar Local. b. Eliminar Local.

Tabla 3. 4: Descripción del Caso de Prueba del Caso de Uso-Gestionar Local.

Las pruebas realizadas al sistema resultaron ser de gran utilidad, pues el mismo responde a las necesidades del cliente y cumple con una serie de requisitos que permiten ejecutar sus funcionalidades sin grandes riesgos y márgenes de errores.

3.6 Conclusiones.

En este capítulo se realizó el modelo de despliegue de la aplicación, en el que se describieron los nodos por los que está compuesto el diagrama de despliegue y las conexiones entre ellos. Se realizó el modelo de implementación en el que se realizaron los diagramas de componentes, así como los diseños de casos de pruebas de funcionalidad con sus respectivas pruebas.



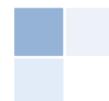


Conclusiones

La realización de este trabajo responde a la necesidad de buscar una alternativa en la Facultad 3 de la Universidad de las Ciencias Informáticas para efectuar la gestión del proceso de planificación docente en la misma, con el objetivo de lograr una solución lo más óptima posible.

Con el desarrollo de la investigación se arrojaron las siguientes conclusiones:

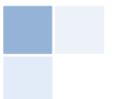
- ❖ Con el estudio crítico y seguimiento de la propuesta de solución realizada anteriormente, se logró desarrollar una aplicación que cumple con los requisitos definidos para la misma.
- ❖ Se elaboraron satisfactoriamente los artefactos necesarios referentes a cada flujo de trabajo determinado por la metodología de desarrollo utilizada.
- ❖ Se diseñaron y aplicaron las pruebas de funcionalidad, permitiendo validar los resultados de la propuesta y chequear que el sistema cumple con los requisitos establecidos por el cliente.
- ❖ Se implementó el sistema “Academic Schedule” como ayuda a los problemas existentes con respecto a la planificación del horario en la Facultad 3.





Recomendaciones

- ❖ Poner a prueba el sistema durante el período de tiempo necesario para comprobar su desempeño y que las funcionalidades del sistema se corresponden con los requisitos del usuario.
- ❖ Ampliar las funcionalidades del sistema como resultado de nuevos requisitos a cumplir.
- ❖ Realizar refactorizaciones al código para lograr más claridad y mejoras en el mismo.
- ❖ Implementar la generación del horario de forma automática en versiones futuras.





Bibliografía.

2008. *Aptana*. [En línea] 2008. <http://www.aptana.com>.

ALIAGA, F y BARTOLOMÉ, A. 2005. El Impacto de las Nuevas Tecnologías en Educación. [En línea] 2005. <http://www.uv.es/aliaga/curriculum/Aliaga&Bartolome-2005-borrador.pdf>..

Alvarez, Miguel Angel. Zend Studio. [En línea] <http://www.desarrolloweb.com/articulos/1178.php>.

Alvarez, Sara. Sistemas Gestores de Bases de Datos. [En línea] <http://www.desarrolloweb.com/articulos/sistemas-gestores-bases-datos.html>.

Arquitectura del servidor Apache. [En línea] <http://www.desarrolloweb.com/articulos/1112.php>..

Barber, Federico y Salido, Miguel A. *Introduccion a la Programacion de Restricciones*.

BLANCO, R y TOBALINA, S P. Introducción a Rational Rose. [En línea] <http://ateam.lsi.upc.es/>.

Booch, G, Rumbaugh, J y Jacobson, I. 1999. The unified modeling language user guide. s.l. : Addison-Wesley, 1999.

BRANDENDAUGH, J. 2000. Aplicaciones JavaScript. [En línea] 2000. <http://bibliodoc.uci.cu/pdf/reg00009.pdf> .

2009. Características de Joomla. [En línea] 2009. <http://www.joomlaos.net/caracteristicas-de-joomla.php>.

Cockbun, A. 2001. *Agile Software Development*. s.l. : Addison-Wesley, 2001.

Cockbun, A y Williams, L. 2000. *The Costs and Benefits of Pair Programming*. s.l. : Humans and Technology Technical Report, 2000.

2008. Comprehensive Academic Timetabling Solutions. *University Timetabling*. [En línea] 2008. <http://www.unitime.org/>.

2009. Comunidad Joomla! [En línea] 2009. <http://comunidadjoomla.org/>.

Conferencia 1. Introducción a la Ingeniería de Software. [En línea] http://teleformacion.uci.cu/file.php/42/Clases_Curso_2007-2008/conferencias/Conferencia_1/Profesores/Conferencia_1.pdf..

2008. CSI Centro de Servicios Informaticos. [En línea] 2008. <http://www.csi.espol.edu.ec/ui/es/content/sistema/sistema.aspx?op=toshow&id=105>.





de la Torre, Aníba. Lenguajes del lado servidor o cliente. [En línea]

http://www.adelat.org/media/docum/nuke_publico/lenguajes_del_lado_servidor_o_cliente.html.

DELGADO, Y. SISTEMA PARA LA GESTIÓN ACADÉMICA. *GESTACAD*. [En línea]

<http://www.posgrados.frc.utn.edu.ar/congreso/trabajos/7.doc..>

—. SISTEMA PARA LA GESTIÓN ACADÉMICA. *GESTACAD*. [En línea]

<http://www.posgrados.frc.utn.edu.ar/congreso/trabajos/7.doc..>

2009. Drupal. [En línea] 2009. <http://drupal.org/>.

2009. Drupal, una herramienta de gestión de contenido con vocabulario. [En línea] 2009.

<http://hugodonairemiles.blogspot.com/2006/11/accesibilidad-y-cms-drupal-una.html>.

2009. eclipsecon. [En línea] 2009. <http://www.edipse.org/>.

ESCRIBANO, G. Introducción a Extreme Programming. [En línea]

<http://www.dsi.udm.es/asignaturas/42551/trabajosAnteriores/Presentacion-XP.pdf..>

GONZÁLEZ, C. Base de Datos PostgreSQL, SQL avanzado y PHP. [En línea]

<http://www.usabilidadweb.com.ar/postgre.php..>

GPI, Consultores. 2006. Disciplina de administración del proyecto - M.S.F. [En línea] 2006. [Citado el: 21 de marzo de 2009.] <http://www.gpicr.com/msf.aspx>.

GSInnova. *GSInnova*. [En línea] <http://www.rational.com.ar/herramientas/roseenterprise.html>.

HEREDIA, M. 2007. Los mejores IDEs para Php. [En línea] 2007. <http://www.tufuncion.com/ide-php>.

Herramientas CASE. [En línea] <http://www.monografias.com/trabajos24/herramientas-case/herramientas-case.shtml>.

2009. Joomla! Hispano. [En línea] 2009. <http://www.joomlahispano.org/>.

Los mejores IDEs para Php. [En línea] <http://www.tufuncion.com/ide-php>.

Luciano. Entornos de Desarrollo Integrado para Java. [En línea] <http://luauf.com/2008/05/13/entornos-de-desarrollo-integrado-para-java/>.

Manual del Usuario de PostgreSQL. *POSTGRESQL*. [En línea] <http://okulto.net/Postgresql-es/web/navegable/user/user.html..>





- Martin Maldonado, Daniel.** Aplicaciones Empresariales. [En línea]
<http://www.aplicacionesempresariales.com/apache-el-servidor-web-mas-reconocido.html>.
- 2006.** Mis Respuestas.com. [En línea] 2006. <http://www.misrespuestas.com/que-es-un-servidor-web.html>.
- Molina, Maria Onelia Chiang. 1995.** *Higiene de la actividad docente*. La Habana : Editorial Pueblo y Educación, 1995.
- MU, WordPress y Hosted.** Cuademo de Lógica. [En línea] <http://blogs.ua.es/ignaciolog/programacion-logica/>.
- Open-Source PHP Web Framework. [En línea] <http://www.symfony-project.org/>.
- Pagina web de UPEDU (Unified Proces for EDUcation). [En línea] <http://www.yoopeedoo.com/upedu/>.
- Paradigmas de Lenguajes de Programación. [En línea] <http://www-2.dc.uba.ar/materias/plp/20071C/homepage.html>.
- PAVÓN, E. 2007.** Visual Paradigm, una herramienta de lo más útil. [En línea] 2007.
<http://sliion2000.blogspot.com/2007/04/visual-paradigm-una-herramienta-de-lo.html>.
- Philippe, Kruchten. 2001.** *The Rational Unified*. s.l. : Addison Wesley, 2001.
- Pressman, Roger. 2005.** *Ingeniería del software: Un enfoque práctico*. s.l. : Félix Varela, 2005.
- PROAÑO, D. 2006.** ANALISIS COMPARATIVO DE BASES DE DATOS DE CODIGO ABIERTO. [En línea] 2006.
<http://www.mysql-hispano.org/articulos/num43/analisis-comparativo.pdf>.
- 2007.** Programacion Estructurada. [En línea] 2007. <http://www.mitecnologico.com/Main/ProgramacionEstructurada>.
- 2006.** Programacion Modular. [En línea] 2006.
<http://teleformacion.edu.aytolacoruna.es/PASCAL/document/modular.htm#intro>.
- 2006.** Programación Orientada a Objetos. [En línea] 2006.
<http://www.monografias.com/trabajos/objetos/objetos.shtml>.
- RUBIO, F y SANTOS, C.** Metodologías de Desarrollo de Software. [En línea] http://alarcos.inf-cr.uclm.es/per/fgarcia/isoftware/doc/tema3_1xh.pdf.
- RUMBAUGH, J y JACOBSON, I. 2000.** El Proceso Unificado de Desarrollo de Software. [En línea] 2000.
<http://bibliodoc.uci.cu/pdf/reg00060.pdf>.
- 2008.** Scribd. [En línea] 2008. <http://www.scribd.com/doc/9784583/Que-es-un-CMS>.





Sistema de Información de Gestión Académica. [En línea] http://alpes.stg.uchile.cl/~svcen01/censo/index_ie.html.

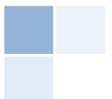
Sixto, Ramón y Saltaren Montiel, Alfredo. 2008. *Análisis y Diseño del Sistema de Ayuda para la planificación docente en la facultad 3*. Cuba : s.n., 2008.

VALDÉS, D. Los diferentes lenguajes de programación para la web. [En línea]
<http://www.maestrosdelweb.com/principiantes/los-diferentes-lenguajes-de-programacion-para-la-web/>.

WELLING, L y THOMSON, L. Desarrollo Web con PHP y MySQL. [En línea] <http://bibliodoc.uci.cu/pdf/reg02819.pdf>.

Zabalza, M A. 2003. Competencias docentes del profesorado universitario. Madrid : Narcea, 2003, pág. 73.

Zend Framework: Estado del Arte. [En línea] <http://bolsadeideas.cl/zsamer/>.





Anexos

3.7 Anexo # 1. Descripción de Casos de Uso.

Nombre del Caso de Uso	Gestionar Afectaciones
Actores	Jefe de Departamento.
Propósito	Este caso de uso tiene como propósito aprobar o rechazar las solicitudes emitidas por los profesores.
Resumen	El caso de Uso comienza cuando el Jefe de Departamento accede a aceptar o rechazar las solicitudes emitidas por los profesores.
Referencias	RF 10.
Precondiciones	Tiene que haber sido emitida una solicitud por al menos un profesor.
Poscondiciones	Se aceptará o se rechazara la solicitud emitida por un profesor.
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema
1. El Jefe de Departamento se dirige al menú “Gestionar Datos, Gestionar Afectación”.	1.1 El sistema le brinda la posibilidad de “Aprobar o Rechazar Solicitud” o “Eliminar Afectación”.
2. El Jefe de Departamento selecciona una de las Opciones presentadas.	2.1 Si selecciona “Aprobar o Rechazar Solicitud” ver sección de Aprobar o Rechazar Solicitud . - Si selecciona “Eliminar Afectación” ver sección de Eliminar Afectación .
Sección “Aprobar o Rechazar Solicitud”	
Acciones del Actor	Respuesta del Sistema
3. El Jefe de Departamento despliega el campo Aprobar o Rechazar Solicitud.	3.1 El sistema muestra la lista de solicitudes realizadas por los profesores.
4. El Jefe de Departamento selecciona la solicitud a aprobar o rechazar y oprime el botón deseado.	4.1 El sistema muestra un mensaje informando que fue aprobada o rechazada la solicitud y termina el Caso de Uso.
Curso alternativo	
Acciones del Actor	Respuesta del Sistema
	4.1 El sistema muestra un mensa informando que no fue seleccionada ninguna solicitud.
Sección “Eliminar Afectación”	
Acciones del Actor	Respuesta del Sistema
3. El Jefe de Departamento despliega el campo Eliminar Solicitud.	3.1 El sistema muestra la lista de Afectaciones.
4. El Jefe de Departamento selecciona la afectación a eliminar y oprime el botón Eliminar.	4.1 El sistema muestra un mensaje informando que fue eliminada la afectación y termina el Caso de Uso.
Curso alternativo	
Acciones del Actor	Respuesta del Sistema
	4.1 El sistema muestra un mensa informando que no fue seleccionada ninguna afectación.
Prioridad:	Secundario.

Tabla 1 : Descripción del Casos de Uso del Sistema - Gestionar Afectaciones.





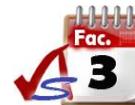
Nombre del Caso de Uso	Gestionar Departamento	
Actores	Vicedecano.	
Propósito	Este caso de uso tiene como propósito gestionar los departamentos de la facultad.	
Resumen	El caso de Uso comienza cuando el vicedecano accede a adicionar, eliminar y/o modificar un departamento de la facultad.	
Referencias	RF 4.	
Precondiciones	Tienen que estar definidos los departamentos docentes en la facultad.	
Poscondiciones	Se adiciona, elimina y/o eliminan los departamentos docentes de la facultad.	
Curso Normal de los Eventos		
Acciones del Actor	Respuesta del Sistema	
1. El vicedecano se dirige al menú “Gestionar Datos, Departamento”.	1.1 El sistema le brinda la posibilidad de “Adicionar Departamento” o “Eliminar Departamento” o “Modificar Departamento”	
2. El vicedecano selecciona una de las Opciones presentadas.	2.1 Si selecciona “Adicionar Departamento” ver sección de Adicionar Departamento . - Si selecciona “Eliminar Departamento” ver sección de Eliminar Departamento . - Si selecciona “Modificar Departamento” ver sección de Modificar Departamento .	
Sección “Adicionar Departamento”		
Acciones del Actor	Respuesta del Sistema	
3. El vicedecano despliega el campo Adicionar Departamento.	3.1 El sistema muestra una interfaz para que se cree el nuevo Departamento.	
4. El vicedecano entra los datos de departamento (Nombre del Departamento, Nombre y Apellidos del Jefe de Departamento y Usuario del Jefe de Departamento) para adicionar el departamento al sistema.	4.1 El sistema verifica que todos los campos obligatorios estén llenos. 4.2 El sistema verifica que el Nombre Departamento no exista en la base de datos. 4.3 El sistema verifica que el Jefe de Departamento no sea Jefe de otro Departamento. 4.4 El Sistema verifica que el usuario se corresponda con el Usuario del Jefe de Departamento. 4.5 Se insertan los datos del Departamento en el sistema. 4.6 Se muestra un mensaje informando que fue adicionado el nuevo Departamento y termina el Caso de Uso.	
Curso alternativo		
Acciones del Actor	Respuesta del Sistema	





	<p>4.1 Se emite un mensaje informando que no todos los campos están llenos.</p> <p>4.2 Se mite un mensaje informando que el Nombre del Departamento ya existe.</p> <p>4.3 Se emite un mensaje informando que el Jefe de Departamento es jefe de otro departamento.</p> <p>4.4 Se emite un mensaje informando que el usuario no corresponde al Jefe de Departamento</p>
Sección “Modificar Departamento”	
Acciones del Actor	Respuesta del Sistema
3. El vicedecano despliega el campo Modificar Departamento.	3.1 El sistema muestra una interfaz con la lista de Departamentos a Modificar.
4. El vicedecano selecciona el Departamento y entra los datos de departamento (Nombre del Departamento, Nombre y Apellidos del Jefe de Departamento y Usuario del Jefe de Departamento) para modificar el departamento en el sistema.	<p>4.1 El sistema verifica que todos los campos obligatorios estén llenos.</p> <p>4.2 El sistema verifica que el Nombre Departamento no exista en la base de datos.</p> <p>4.3 El sistema verifica que el Jefe de Departamento no sea Jefe de otro Departamento.</p> <p>4.4 El Sistema verifica que el usuario se corresponda con el Usuario del Jefe de Departamento.</p> <p>4.5 Se insertan los datos del Departamento en el sistema.</p> <p>4.6 Se muestra un mensaje informando que fue modificado el Departamento ye termina el Caso de Uso.</p>
Curso alternativo	
Acciones del Actor	Respuesta del Sistema
	<p>4.1 Se emite un mensaje informando que no todos los campos están llenos.</p> <p>4.2 Se mite un mensaje informando que el Nombre del Departamento ya existe.</p> <p>4.3 Se emite un mensaje informando que el Jefe de Departamento es jefe de otro departamento.</p> <p>4.4 Se emite un mensaje informando que el usuario no corresponde al Jefe de Departamento</p>
Sección “Eliminar Departamento”	
Acciones del Actor	Respuesta del Sistema
3. El vicedecano despliega el campo Eliminar Departamento.	3.1 El sistema muestra una interfaz con la lista de Departamentos.



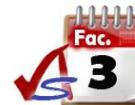


4. El vicedecano selecciona el Departamento y oprime el botón eliminar.	4.1 El sistema muestra un mensaje informando que el departamento fue eliminado y termina el caso de uso.
Curso alternativo	
Acciones del Actor	Respuesta del Sistema
	4.1 El sistema muestra un mensaje informando no fue seleccionado ningún Departamento y termina el caso de uso.
Prioridad:	Secundario.

Tabla 2: Descripción del Caso de Uso del Sistema - Gestionar Departamento.

Nombre del Caso de Uso	Gestionar Solicitud
Actores	Profesor.
Propósito	Este caso de uso tiene como propósito gestionar las solicitudes emitidas por cada uno de los profesores.
Resumen	El caso de uso comienza cuando el profesor accede a adicionar, eliminar y/o modificar una solicitud.
Referencias	RF 9.
Precondiciones	-
Poscondiciones	Se adiciona, modifica o se elimina una solicitud creada por un profesor.
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema
1. El Profesor se dirige al menú “Gestionar Datos, Realizar Solicitud”	1.1 El sistema le brinda la posibilidad de “Adicionar Solicitud” o “Eliminar Solicitud” o “Modificar Solicitud”
2. El Profesor selecciona una de las Opciones presentadas.	2.1 Si selecciona “Adicionar Solicitud” ver sección de Adicionar Solicitud . - Si selecciona “Eliminar Solicitud” ver sección de Eliminar Solicitud . - Si selecciona “Modificar Solicitud” ver sección de Modificar Solicitud .
Sección “Adicionar Solicitud”	
Acciones del Actor	Respuesta del Sistema
3. El profesor despliega el campo Adicionar Solicitud.	3.1 El sistema muestra una interfaz para que se cree la nueva Solicitud.
4. El profesor entra los datos de Solicitud (Fecha de Solicitud, Fecha a Efectuarse, Motivo de la solicitud y Breve descripción) para adicionar la solicitud al sistema.	4.1 El sistema verifica que todos los campos obligatorios estén llenos. 4.2 Se insertan los datos de la solicitud en el sistema. 4.3 Se muestra un mensaje informando que fue adicionada una nueva Solicitud y termina el Caso de Uso.
Curso alternativo	
Acciones del Actor	Respuesta del Sistema
	4.1 El sistema muestra un mensaje informando que no todos los campos están llenos.

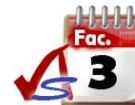




Sección “Eliminar Solicitud”	
Acciones del Actor	Respuesta del Sistema
3. El profesor despliega el campo Eliminar Solicitud.	3.1 El sistema muestra una interfaz con la lista de Solicitudes.
4. El profesor selecciona la solicitud y oprime el botón eliminar.	4.1 El sistema muestra un mensaje informando que la solicitud fue eliminada y termina el caso de uso.
Curso alternativo	
Acciones del Actor	Respuesta del Sistema
	4.1 El sistema muestra un mensa informando que no fue seleccionada ninguna solicitud.
Sección “Modificar Solicitud”	
Acciones del Actor	Respuesta del Sistema
3. El profesor despliega el campo Modificar Solicitud.	3.1 El sistema muestra una interfaz para que se modifique la Solicitud.
4. El profesor entra los datos de Solicitud (Fecha de Solicitud, Fecha a Efectuarse, Motivo de la solicitud y Breve descripción) para modificar la solicitud en el sistema.	4.1 El sistema verifica que todos los campos obligatorios estén llenos. 4.2 Se actualizan los datos de la solicitud en el sistema. 4.3 Se muestra un mensaje informando que fue modificada la Solicitud y termina el Caso de Uso.
Curso alternativo	
Acciones del Actor	Respuesta del Sistema
	4.1 El sistema muestra un mensaje informando que no todos los campos están llenos.
Prioridad:	Secundario.

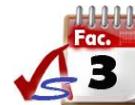
Tabla 3: Descripción del Caso de Uso del Sistema - Gestionar Solicitud.

Nombre del Caso de Uso	Gestionar Profesor
Actores	Jefe Departamento
Propósito	Este caso de uso tiene como propósito gestionar los profesores que pertenecen a la facultad.
Resumen	El caso de uso comienza cuando el Jefe de Departamento accede a adicionar, eliminar y/o modificar un profesor. Así como vincular profesor con asignatura, vincular profesor –asignatura con grupos, eliminar los vínculos realizados.
Referencias	RF 7.
Precondiciones	Tienen que estar creado los departamentos, las asignaturas y los grupos docentes.
Poscondiciones	Se adiciona, se modifica o elimina un profesor así como se crearan y eliminaran los vínculos con los grupos y las asignaturas.
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema
1. El Jefe de Departamento dirige al menú “Gestionar Datos, Profesor”.	1.1 El sistema le brinda la posibilidad de “Adicionar Profesor” o “Eliminar Profesor” o “Modificar Profesor” o “Vincular Profesor Asignatura” o “Vincular Profesor Asignatura Grupo” o Eliminar Vínculo Profesor



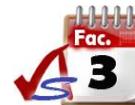
	Asignatura” o “Eliminar Vínculo Profesor Asignatura Grupo”.
2. El Jefe de Departamento selecciona una de las opciones presentadas.	<p>2.1 Si selecciona “Adicionar Solicitud” ver sección de Adicionar Profesor.</p> <p>- Si selecciona “Eliminar Solicitud” ver sección de Eliminar Profesor.</p> <p>- Si selecciona “Modificar Solicitud” ver sección de Modificar Profesor.</p> <p>- Si selecciona “Vincular Profesor Asignatura” ver sección de Vincular Profesor Asignatura.</p> <p>- Si selecciona “Vincular Profesor Asignatura Grupo” ver sección de Vincular Profesor Asignatura Grupo.</p> <p>- Si selecciona “Eliminar Vínculo Profesor Asignatura” ver sección de Eliminar Vínculo Profesor Asignatura.</p> <p>- Si selecciona “Eliminar Vínculo Profesor Asignatura Grupo” ver sección de Eliminar Vínculo Profesor Asignatura Grupo.</p>
Sección “Adicionar Profesor”	
Acciones del Actor	Respuesta del Sistema
3. El Jefe de Departamento despliega el campo Adicionar profesor.	3.1 El sistema muestra una interfaz para adicionar un nuevo profesor.
4. El Jefe de Departamento entra los datos del profesor (Usuario, Nombre y Apellidos, Departamento) para adicionar un nuevo profesor al sistema.	<p>4.1 El sistema verifica que todos los campos obligatorios estén llenos.</p> <p>4.2 El sistema verifica que el usuario no aparezca en la base de datos</p> <p>4.3 Se insertan los datos del profesor en el sistema.</p> <p>4.4 Se muestra un mensaje informando que fue adicionado un nuevo Profesor y termina el Caso de Uso.</p>
Curso alternativo	
Acciones del Actor	Respuesta del Sistema
	<p>4.1 El sistema muestra un mensaje informando que no fueron llenados todos los campos.</p> <p>4.2 El sistema emite un mensaje informando que ese usuario existe en el sistema.</p>
Sección “Eliminar Profesor”	
Acciones del Actor	Respuesta del Sistema
3. El Jefe de Departamento despliega el campo	3.1 El sistema muestra una interfaz para eliminar un





Eliminar profesor.	profesor.
4. El Jefe de Departamento selecciona al profesor y oprime el botón eliminar.	4.1 El sistema muestra un mensaje informando que el profesor fue eliminado y termina el caso de uso.
Curso alternativo	
Acciones del Actor	Respuesta del Sistema
	4.1 El sistema muestra un mensaje informando que no fue seleccionado ningún profesor.
Sección “Modificar Profesor”	
Acciones del Actor	Respuesta del Sistema
3. El Jefe de Departamento despliega el campo Modificar profesor.	3.1 El sistema muestra una interfaz para modificar un profesor.
4. El Jefe de Departamento llena los datos del profesor (Usuario, Nombre y Apellidos, Departamento) para modificar un profesor en el sistema.	4.1 El sistema verifica que todos los campos obligatorios estén llenos. 4.2 El sistema verifica que el usuario no aparezca en la base de datos. 4.3 Se insertan los datos del profesor en el sistema. 4.4 Se muestra un mensaje informando que fue modificado el Profesor y termina el Caso de Uso.
Curso alternativo	
Acciones del Actor	Respuesta del Sistema
	4.1 El sistema muestra un mensaje informando que no fueron llenados todos los campos. 4.2 El sistema emite un mensaje informando que ese usuario existe en el sistema.
Sección “Vincular Profesor Asignatura”	
Acciones del Actor	Respuesta del Sistema
3. El Jefe de Departamento despliega el campo Vincular Profesor Asignatura.	3.1 El sistema muestra una interfaz para Vincular Profesor Asignatura.
4. El Jefe de Departamento selecciona al profesor y la asignatura a vincular y oprime el botón aceptar.	4.1 El sistema muestra un mensaje informando que el profesor y la asignatura fueron vinculados y termina el caso de uso.
Curso alternativo	
Acciones del Actor	Respuesta del Sistema
	4.1 El sistema muestra un mensaje informando que no fue seleccionado ningún profesor o asignatura.
Sección “Vincular Profesor Asignatura Grupo”	
Acciones del Actor	Respuesta del Sistema
3. El Jefe de Departamento despliega el campo Vincular Profesor Asignatura Grupo.	3.1 El sistema muestra una interfaz para Vincular Profesor Asignatura Grupo.
4. El Jefe de Departamento selecciona al profesor y asignatura a vincular con el grupo y oprime el botón aceptar.	4.1 El sistema muestra un mensaje informando que el profesor y asignatura fue vinculado con el grupo y termina el caso de uso.
Curso alternativo	
Acciones del Actor	Respuesta del Sistema
	4.1 El sistema muestra un mensaje informando que no fue seleccionado ningún profesor-asignatura o grupo.



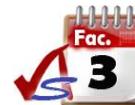


Sección “Eliminar Vínculo Profesor Asignatura”	
Acciones del Actor	Respuesta del Sistema
3. El Jefe de Departamento despliega el campo Eliminar Vínculo Profesor Asignatura.	3.1 El sistema muestra una interfaz para eliminar el Vínculo Profesor Asignatura.
4. El Jefe de Departamento selecciona al profesor y la asignatura y oprime el botón eliminar.	4.1 El sistema muestra un mensaje informando que fue eliminado el vínculo Profesor Asignatura y termina el caso de uso.
Curso alternativo	
Acciones del Actor	Respuesta del Sistema
	4.1 El sistema muestra un mensaje informando que no fue seleccionado ningún profesor o asignatura.
Sección “Eliminar Vínculo Profesor Asignatura Grupo”	
3. El Jefe de Departamento despliega el campo Eliminar Vínculo Profesor Asignatura Grupo.	3.1 El sistema muestra una interfaz para eliminar el Vínculo Profesor Asignatura Grupo.
4. El Jefe de Departamento selecciona al profesor-asignatura y el grupo y oprime el botón eliminar	4.1 El sistema muestra un mensaje informando que fue eliminado el vínculo Profesor-Asignatura Grupo y termina el caso de uso.
	4.2 El sistema muestra un mensaje informando que no fue seleccionado ningún profesor-asignatura o grupo.
Prioridad:	Secundario.

Tabla 4: Descripción del Caso de Uso del Sistema - Gestionar Profesor.

Nombre del Caso de Uso	Gestionar Actividad Docente
Actores	Jefe de Departamento.
Propósito	Este caso de uso tiene como propósito gestionar las Actividad Docente para las asignaturas.
Resumen	El caso de uso comienza cuando el Jefe Departamento accede a adicionar, modificar y/o eliminar una Actividad Docente.
Referencias	RF 2.
Precondiciones	---
Poscondiciones	Se adiciona, modifica o se elimina una Actividad Docente.
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema
1. El Jefe de Departamento dirige al menú “Gestionar Datos, Actividad Docente”.	1.1 El sistema le brinda la posibilidad de “Adicionar Actividad Docente” o “Eliminar Actividad Docente” o “Modificar Actividad Docente”.
2. El Jefe de Departamento selecciona una de las opciones presentadas.	2.1 Si selecciona “Adicionar Actividad Docente” ver sección de Adicionar Actividad Docente . - Si selecciona “Eliminar Actividad Docente” ver sección de Eliminar Actividad Docente . - Si selecciona “Modificar Actividad Docente” ver sección de Modificar Actividad Docente .
Sección “Adicionar Actividad Docente”	
Acciones del Actor	Respuesta del Sistema
3. El Jefe de Departamento despliega el campo Adicionar Actividad Docente.	3.1 El sistema muestra una interfaz para adicionar una Actividad Docente.
4. El Jefe de Departamento entra los datos de la	4.1 El sistema verifica que todos los campos obligatorios





Actividad Docente (Nombre y Abreviatura) para adicionar una Actividad Docente al sistema.	estén llenos. 4.2 El sistema verifica que no aparezcan nombres iguales en la base de datos. 4.3 El sistema verifica que no aparezcan abreviaturas iguales en la base de datos. 4.4 Se insertan los datos de la Actividad Docente en el sistema. 4.5 Se muestra un mensaje informando que fue adicionado una nueva Actividad Docente y termina el Caso de Uso.
Curso alternativo	
Acciones del Actor	Respuesta del Sistema
	4.1 El sistema muestra un mensaje informando que no se llenaron todos los campos. 4.2 El Sistema muestra un mensaje informando que el Nombre ya existe. 4.3 El Sistema muestra un mensaje informando que la abreviatura ya existe.
Sección “Eliminar Actividad Docente”	
Acciones del Actor	Respuesta del Sistema
3. El Jefe de Departamento despliega el campo Eliminar Actividad Docente.	3.1 El sistema muestra una interfaz para eliminar una Actividad Docente.
4. El Jefe de Departamento selecciona la Actividad Docente y oprime el botón eliminar.	4.1 El sistema muestra un mensaje informando que la Actividad Docente fue eliminada y termina el caso de uso.
Curso alternativo	
Acciones del Actor	Respuesta del Sistema
	4.1 El sistema muestra un mensaje informando que no ha seleccionada ninguna Actividad Docente para eliminar.
Sección “Modificar Actividad Docente”	
Acciones del Actor	Respuesta del Sistema
3. El Jefe de Departamento despliega el campo Modificar Actividad Docente.	3.1 El sistema muestra una interfaz para modificar una Asignatura.
4. El Jefe de Departamento entra los datos de la Actividad Docente (Nombre y Abreviatura) para modificar una Actividad Docente en el sistema.	4.1 El sistema verifica que todos los campos obligatorios estén llenos. 4.2 El sistema verifica que no aparezcan nombres iguales en la base de datos. 4.3 El sistema verifica que no aparezcan abreviaturas iguales en la base de datos. 4.4 Se actualizan los datos de la Actividad Docente en el





	sistema.
	4.4 Se muestra un mensaje informando que fue modificada la Actividad Docente y termina el Caso de Uso.
Curso alternativo	
Acciones del Actor	Respuesta del Sistema
	4.1 El sistema muestra un mensaje informando que no se llenaron todos los campos.
	4.2 El Sistema muestra un mensaje informando que el Nombre ya existe.
	4.3 El Sistema muestra un mensaje informando que la abreviatura ya existe.
Prioridad:	Secundario.

Tabla 5: Descripción del Caso de Uso del Sistema - Gestionar Actividad Docente.

Nombre del Caso de Uso	Consultar Reporte
Actores	Usuario.
Propósito	Este caso de uso tiene como propósito brindarle al usuario la posibilidad de consultar los reportes deseados.
Resumen	El caso de uso comienza cuando el usuario accede a consultar reporte y escoger el reporte deseado.
Referencias	RF 11.
Precondiciones	Que existan datos del reporte seleccionado.
Poscondiciones	Brinda al usuario los datos del reporte seleccionado.
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema
1. El usuario se dirige al menú “Reporte”.	1.1 El sistema le brinda la posibilidad de consultar un reporte.
2. El usuario selecciona el reporte deseado.	2.1 El sistema muestra el reporte seleccionado.
Prioridad:	Secundario.

Tabla 6: Descripción del Caso de Uso del Sistema - Consultar Reporte.



3.8 Anexo # 2. Diagramas de Clases del Diseño.

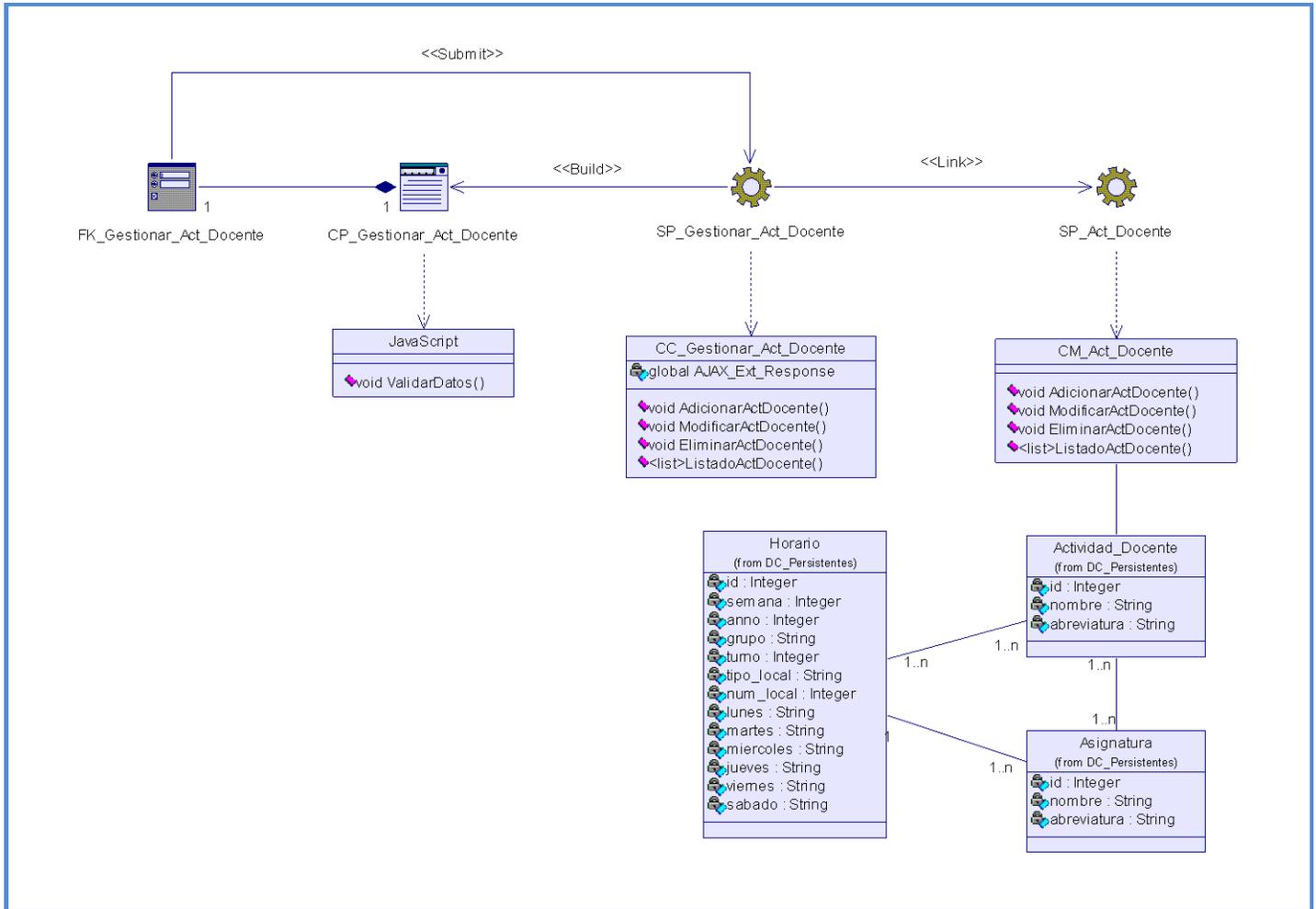
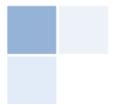


Imagen 1: Diagrama de Clases del Diseño del Caso de Uso del Sistema – Gestionar Actividad Docente.



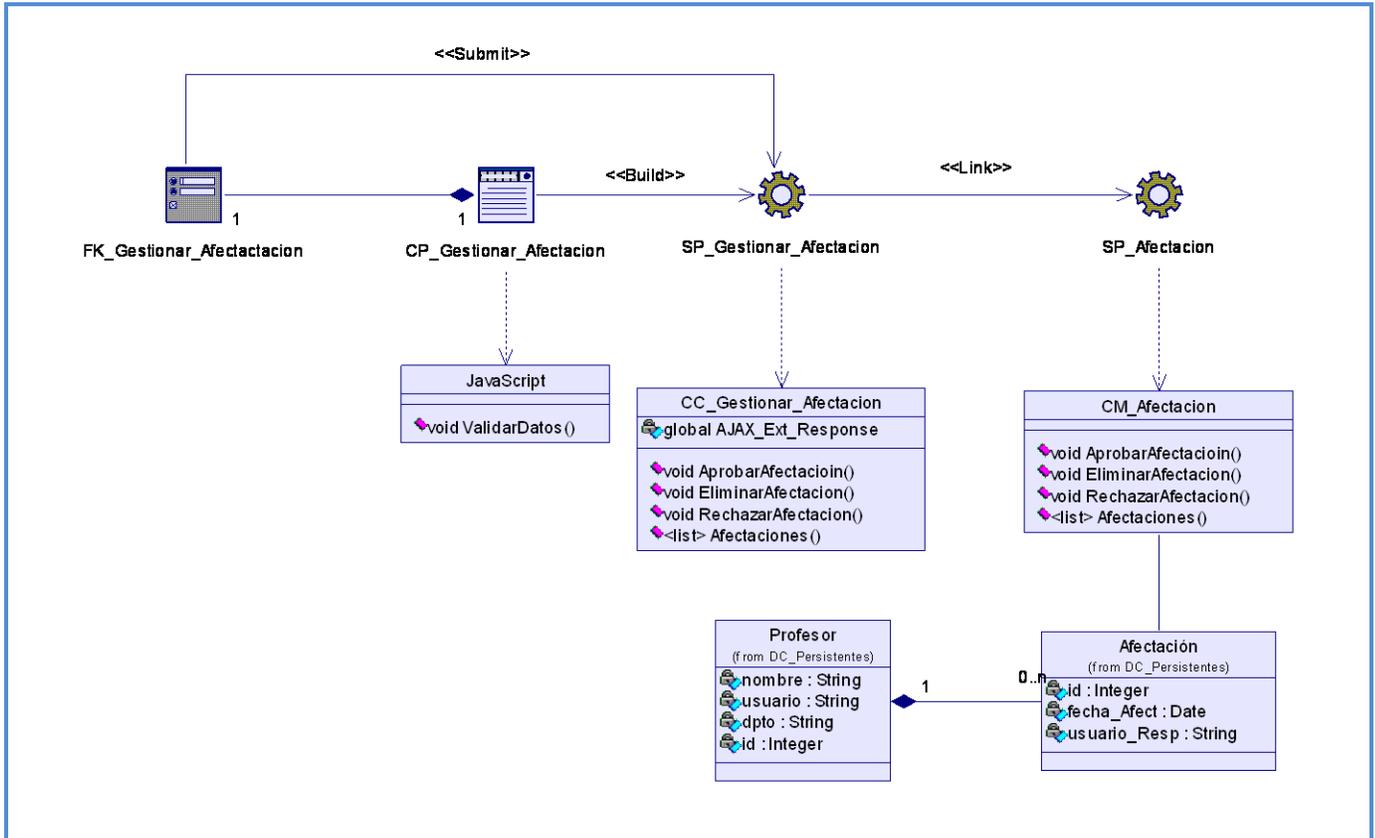
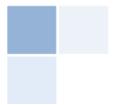


Imagen 2: Diagrama de Clases del Diseño del Caso de Uso del Sistema – Gestionar Afectación.



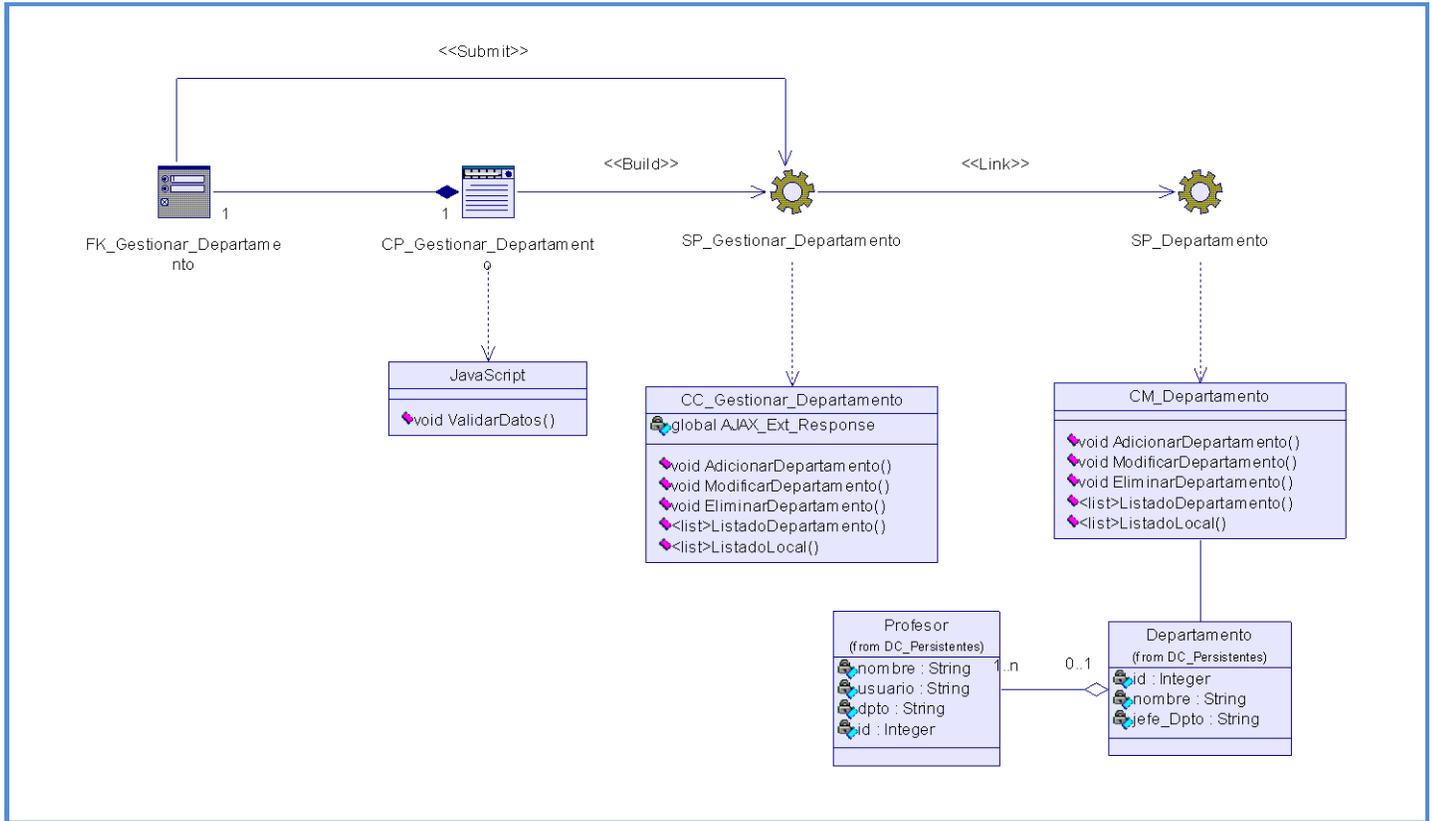
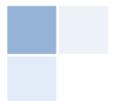


Imagen 3: Diagrama de Clases del Diseño del Caso de Uso del Sistema – Gestionar Departamento.



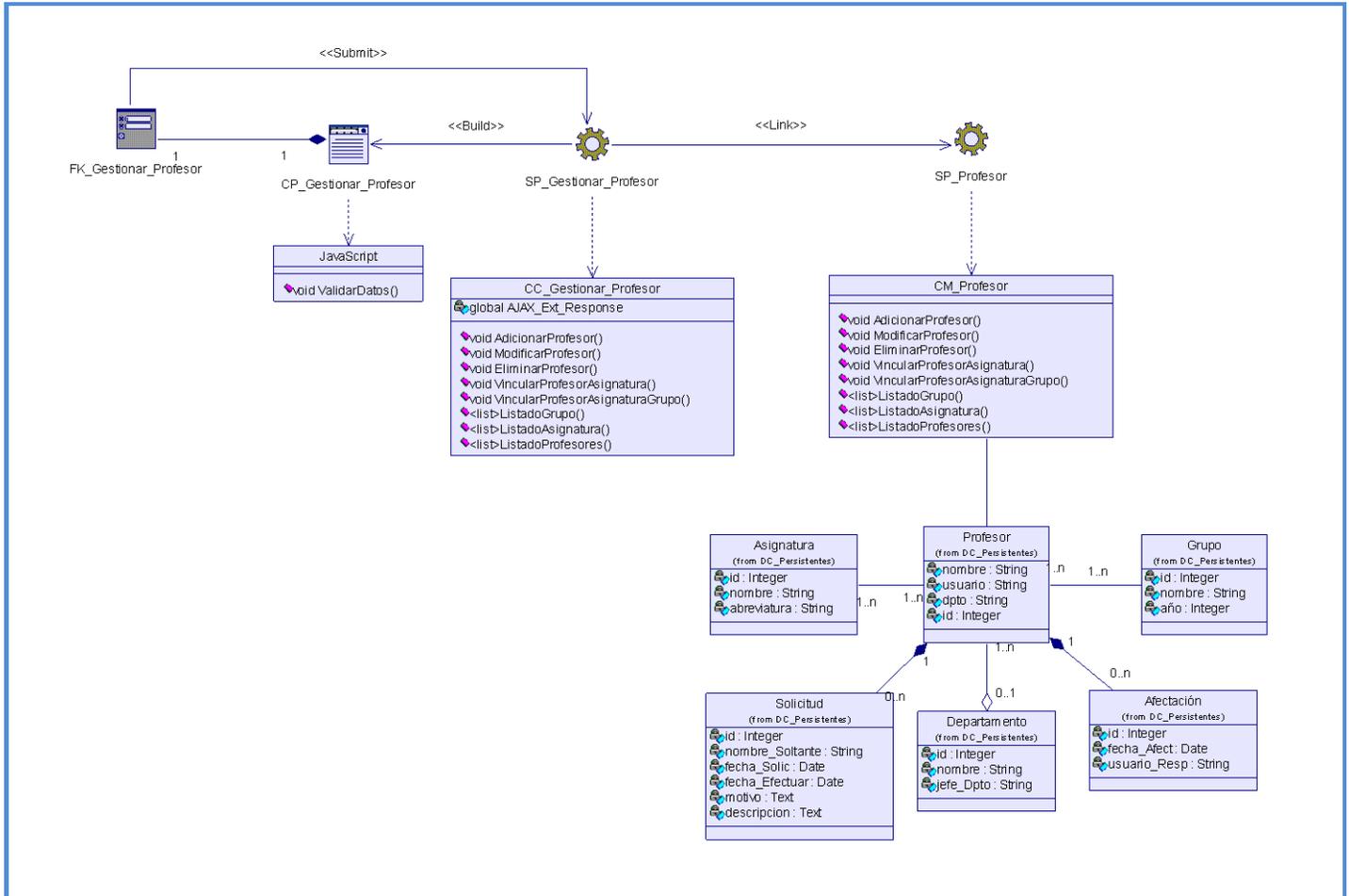
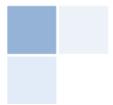


Imagen 4: Diagrama de Clases del Diseño del Caso de Uso del Sistema – Gestionar Profesor.



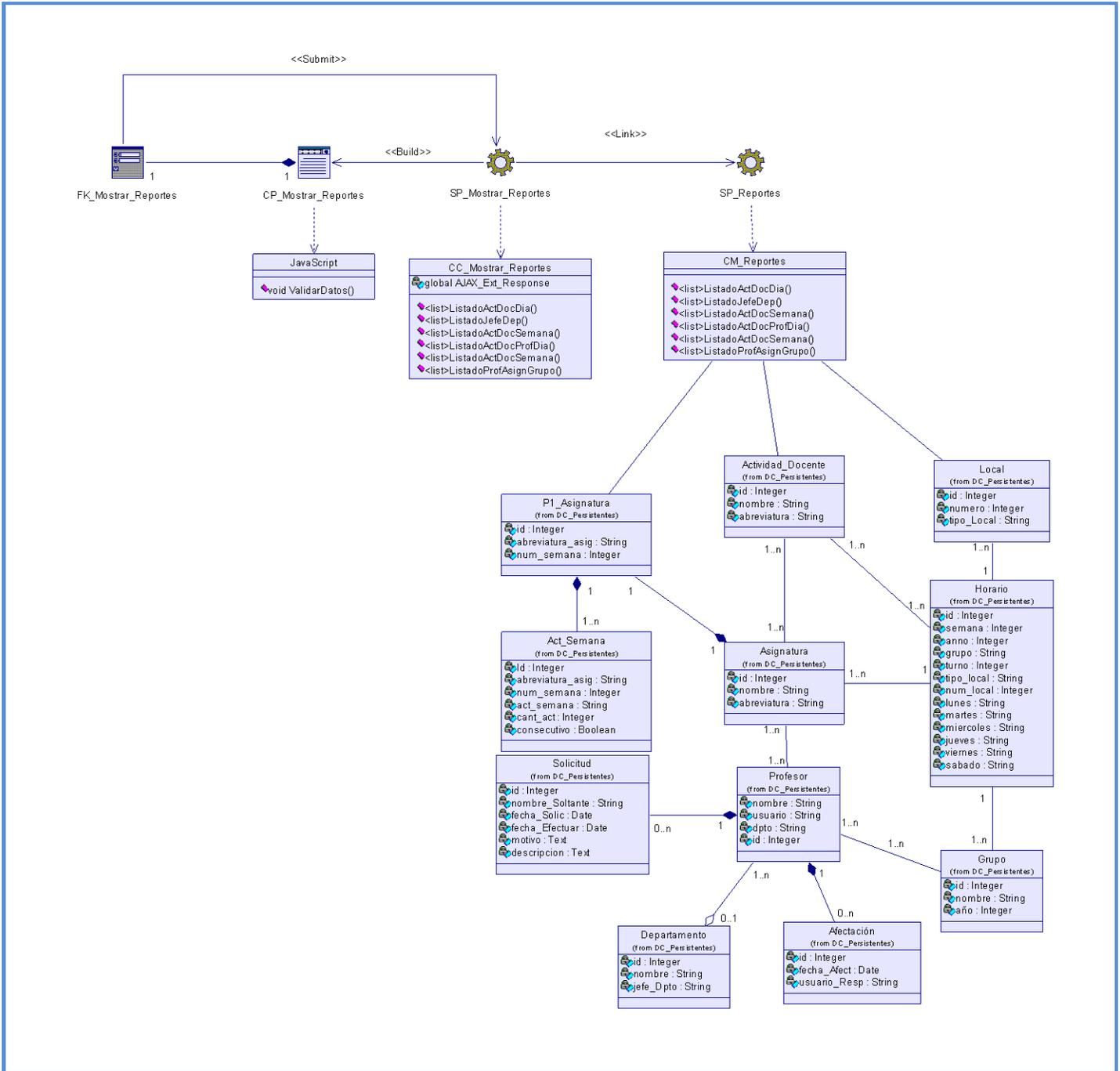
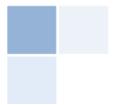


Imagen 5: Diagrama de Clases del Diseño del Caso de Uso del Sistema – Mostrar Reportes.



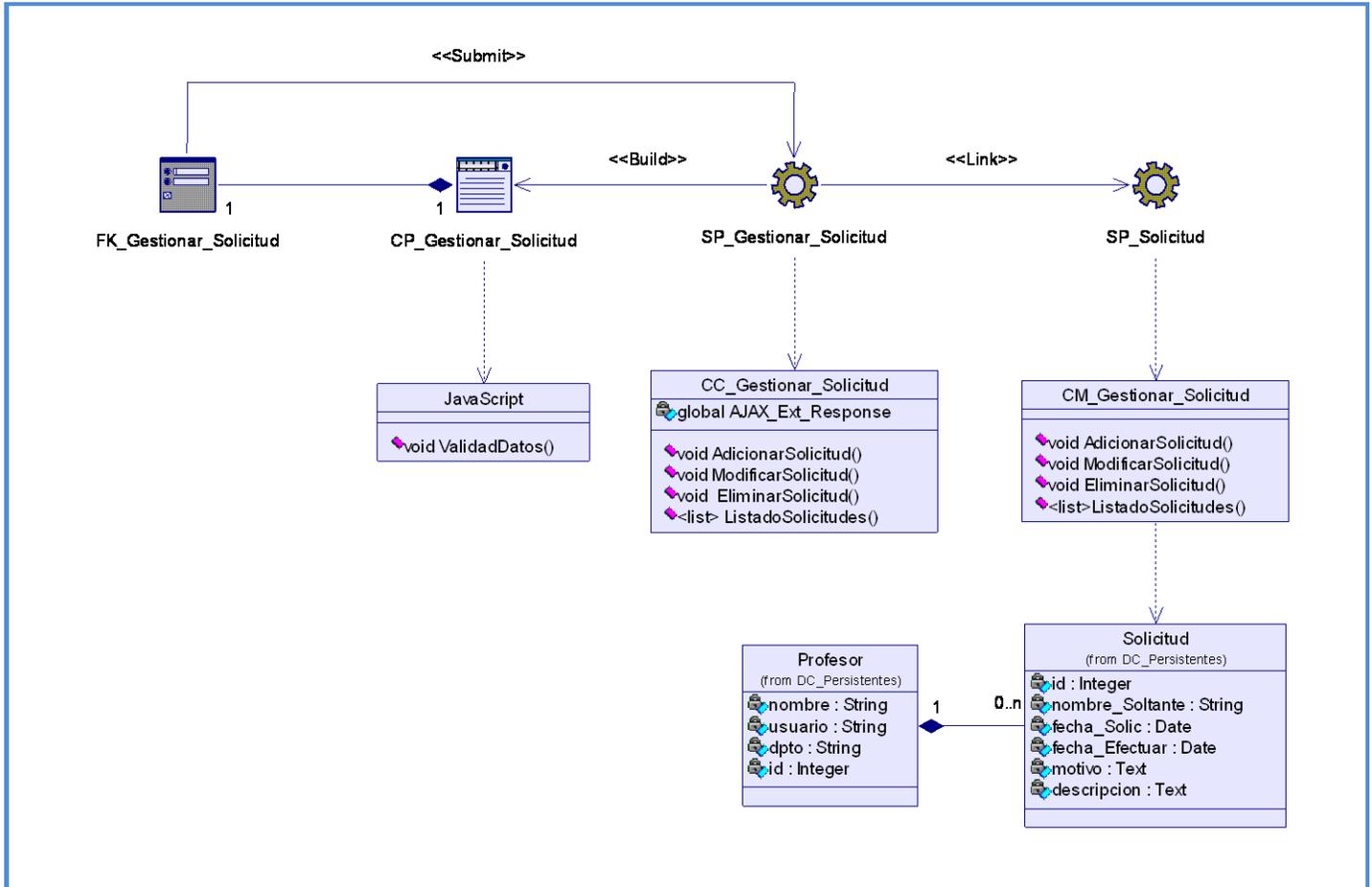
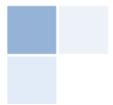


Imagen 6: Diagrama de Clases del Diseño del Caso de Uso del Sistema – Gestionar Solicitud.





3.9 Anexo # 3. Descripción de las Clases a Implementar.

Nombre: Actividad_Docente	
Tipo de Clase: Entidad	
Atributos	Tipo
id	int
nombre	string
abreviatura	string
Responsabilidades	
Nombre	Descripción
Adicionar_Actividad_Docente(nombre: string, abreviatura: string)	Adiciona una nueva actividad docente.
Eliminar_Actividad_Docente (nombre: string)	Eliminar una nueva actividad docente.
Modificar_Actividad_Docente (nombre: string)	Modifica una nueva actividad docente.
Obtener_Actividad_Docente ()	Devuelve una lista de los tipos de actividades (las abreviaturas).
Buscar_Actividad_Docente (nombre: string)	Dado el nombre de una actividad dice si existe o no.

Tabla 7: Descripción de la Clase Entidad - Actividad_Docente.

Nombre: Local	
Tipo de Clase: Entidad	
Atributos	Tipo
id	int
numero	int
tipo_Local	string
Responsabilidades	
Nombre	Descripción
Adicionar_Local(tipo: string, numero: int)	Adiciona un local al sistema.
Eliminar_Local(tipo: string, numero: int)	Elimina un local al sistema.
Bloquear_Local(tipo: string, numero: int)	Bloquea el local, no estando disponibles para hacer ningún tipo de planificación en el.
DesBloquear_Local(tipo: string, numero: int)	Poner el local disponible para la planificación docente.
LocalOcupado(tipo: string, numero: int, turno: int)	Para saber si el local está siendo usado o no.
Buscar_Local(tipo: string, numero: int)	Dado un tipo y un número dice si existe o no el local.
Obtener_Locales(tipo: string)	Dado un tipo de local, devuelve una lista con todos los que existan de ese tipo.

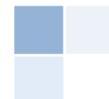




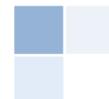
Tabla 8: Descripción de la Clase Entidad - Local.

Nombre: P1_Asignatura	
Tipo de Clase: Entidad	
Atributos	Tipo
id	int
abreviatura_asig	string
num_semana	int
Responsabilidades	
Nombre	Descripción
Adicionar_P1_Asignatura (semana: int , tema: string, contenido: string, observación: string)	Adiciona los datos de la semana del P1 de la asignatura.
Eliminar_P1_Asignatura (semana: int)	Elimina la semana del P1.
Modificar_P1_Asignatura(semana: int , tema: string, contenido: string, observación: string)	Modifica los datos de la semana al P1 de la asignatura.
Obtener_Dato_Semana(semana: int)	Dado el número de la semana devuelve los datos de la misma, en caso de que exista, sino devuelve -1.

Tabla 9: Descripción de la Clase Entidad – P1_Asignatura.

Nombre: Act_Semana	
Tipo de Clase: Entidad	
Atributos	Tipo
id	int
abreviatura_asig	string
num_semana	int
act_semana	string
cant_act	int
Consecutivo	bool
Responsabilidades	
Nombre	Descripción
Adicionar_Act_Semana (tipoAct: string, tipoLocal: string, orden: int, consecutiva: bool)	Adiciona las actividades del P1 de la asignatura que se deben realizar en la semana.
Obtener_Act_Semana ()	Devuelve un listado de las actividades de la semana

Tabla 10: Descripción de la Clase Entidad – Act_Semana.



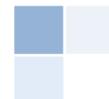


Nombre: Departamento	
Tipo de Clase: Entidad	
Atributos	Tipo
id	int
nombre	string
jefe_Dpto	string
Responsabilidades	
Nombre	Descripción
Adicionar_Departamento(nombre: string, jefeDpto: string, local: string)	Adiciona un nuevo departamento.
Eliminar_Departamento(nombre: string)	Dado un nombre elimina al departamento.
Modificar_Departamento(nombre: string)	Modifica un departamento.
Obtener_Departamento()	Devuelve una lista de los departamentos.
Buscar_Departamento(nombre: string)	Dado el nombre dice si el departamento devuelve el nombre del jefe Dpto., en caso contrario -1.
Buscar_JefeDpto(jefeDpto: string)	Dado el nombre del jefe de departamento, devuelve el departamento que es jefe, en caso contrario -1.

Tabla 11: Descripción de la Clase Entidad - Departamento.

Nombre: Asignatura	
Tipo de Clase: Entidad	
Atributos	Tipo
Id	int
nombre	string
abreviatura	string
Responsabilidades	
Nombre	Descripción
Adicionar_Asignatura(nombre: string, abreviatura: string)	Adiciona una nueva asignatura.
Eliminar_Asignatura(nombre: string)	Dado un nombre, elimina la asignatura.
Modificar_Asignatura(nombre: string)	Modifica una asignatura.
Buscar_Asignatura(nombre: string)	Dado un nombre dice si existe o no la asignatura.
Obtener_Asignatura()	Devuelve un listado de las asignaturas.

Tabla 12: Descripción de la Clase Entidad - Asignatura.





Nombre: Solicitud	
Tipo de Clase: Entidad	
Atributos	Tipo
id	int
nombre_Soltante	string
fecha_Solic	date
fecha_Efectuar	date
motivo	text
descripcion	text
Responsabilidades	
Nombre	Descripción
Adicionar_Solicitud(usuario: string, fechaSolicitud: DateTime, fechaRealizar: DateTime, motivoSolicitud: string)	Adiciona una nueva solicitud.
Eliminar_Solicitud(usuario: string, fechaSolicitud: DateTime)	Dado el usuario y la fecha de solicitud, es eliminada la solicitud.
Modificar_Solicitud(usuario: tring, fechaSolicitud: DateTime)	Modifica una solicitud.
Obtener_Solicitud(usuario: string)	Dado un usuario devuelve una lista con todas las solicitudes de ese usuario, en caso contrario devuelve -1.

Tabla 13: Descripción de la Clase Entidad - Solicitud.

Nombre: Profesor	
Tipo de Clase: Entidad	
Atributos	Tipo
id	int
nombre	string
usuario	string
dpto	string
Responsabilidades	
Nombre	Descripción
Adicionar_Profesor(usuario: string, nombre: string, dpto: string)	Adiciona un nuevo profesor.
Eliminar_Profesor(usuario: string)	Dado un usuario elimina al profesor.
Modificar_Profesor(usuario: string)	Modifica un profesor.
Buscar_Profesor(usuario: string)	Dado un usuario devuelve la información del profesor, encaso contrario -1.
Obtener_Profesor()	Devuelve un listado de profesores.

Tabla 14: Descripción de la Clase Entidad - Profesor.





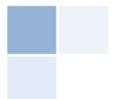
Nombre: Grupo	
Tipo de Clase: Entidad	
Atributos	Tipo
id	int
nombre	string
año	int
Responsabilidades	
Nombre	Descripción
Adicionar_ Grupo(numero: int, año: int, matricula: int)	Adiciona un nuevo grupo.
Eliminar_ Grupo(numero: int, año: int)	Dado un año y un número elimina al grupo.
Buscar_ Grupo(numero: int, año: int)	Dado un año y un número devuelve los datos del grupo, en caso contrario -1.
Obtener_ Grupo(año: int)	Devuelve un listado de los grupos por año.

Tabla 15: Descripción de la Clase Entidad - Grupo.

Nombre: Departamento	
Tipo de Clase: Entidad	
Atributos	Tipo
id	int
nombre	string
año	int
Responsabilidades	
Nombre	Descripción
Adicionar_ Departamento(nombre: string, jefeDpto: string)	Adiciona un nuevo departamento.
Eliminar_ Departamento(nombre: string, jefeDpto: string)	Elimina un departamento.
Modificar_ Departamento(nombre: string, jefeDpto: string)	Modifica un departamento.

Tabla 16: Descripción de la Clase Entidad - Departamento.

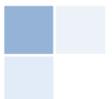
Nombre: Horario	
Tipo de Clase: Entidad	
Atributos	Tipo
id	int
semana	int





anno	Int
turno	Int
num_local	Int
grupo	String
tipo_local	String
lunes	String
martes	String
miercoles	String
jueves	String
viernes	String
sabado	String
Responsabilidades	
Nombre	Descripción
Gestionar_Horario_PorLocal(grupo: string, actdoc: string, asignatura: string)	Gestiona el horario a través de los locales.
Gestionar_Horario_PorGrupo(local: string, actdoc:string, asignatura:string)	Gestiona el horario a través de los grupos.

Tabla 17: Descripción de la Clase Entidad - Horario.



3.10 Anexo # 4. Diagramas de Componentes.

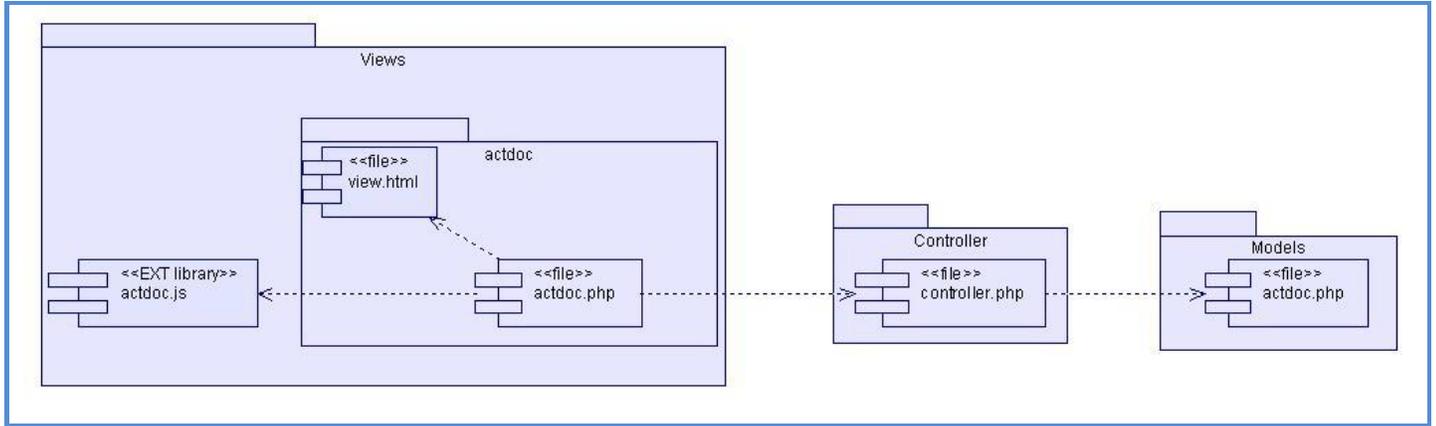


Imagen 7: Diagrama de Componentes del Caso de Uso Gestionar Actividad Docente.

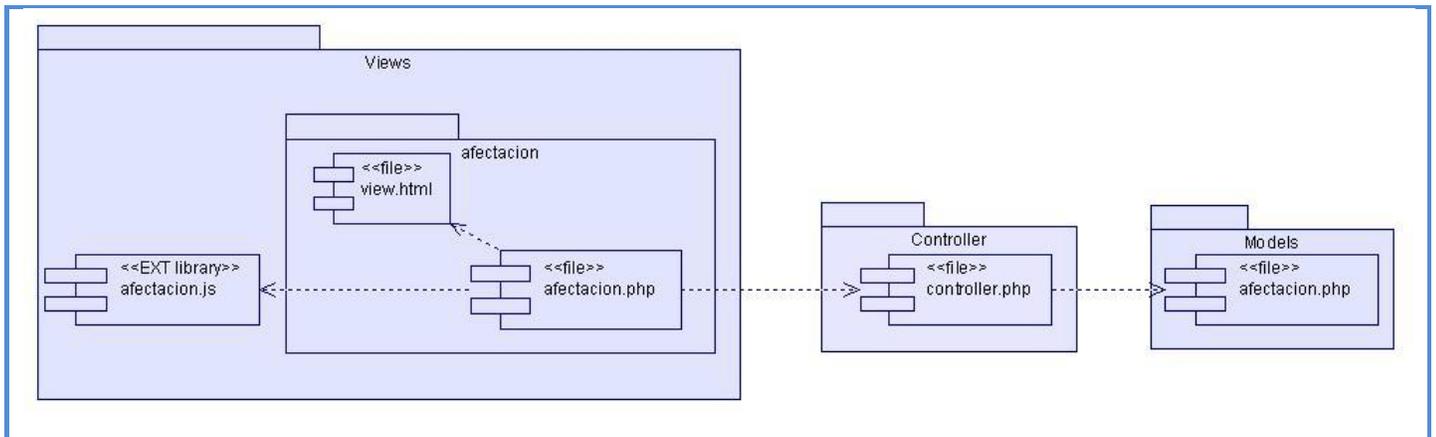
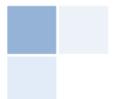


Imagen 8: Diagrama de Componentes del Caso de Uso Gestionar Afectación.



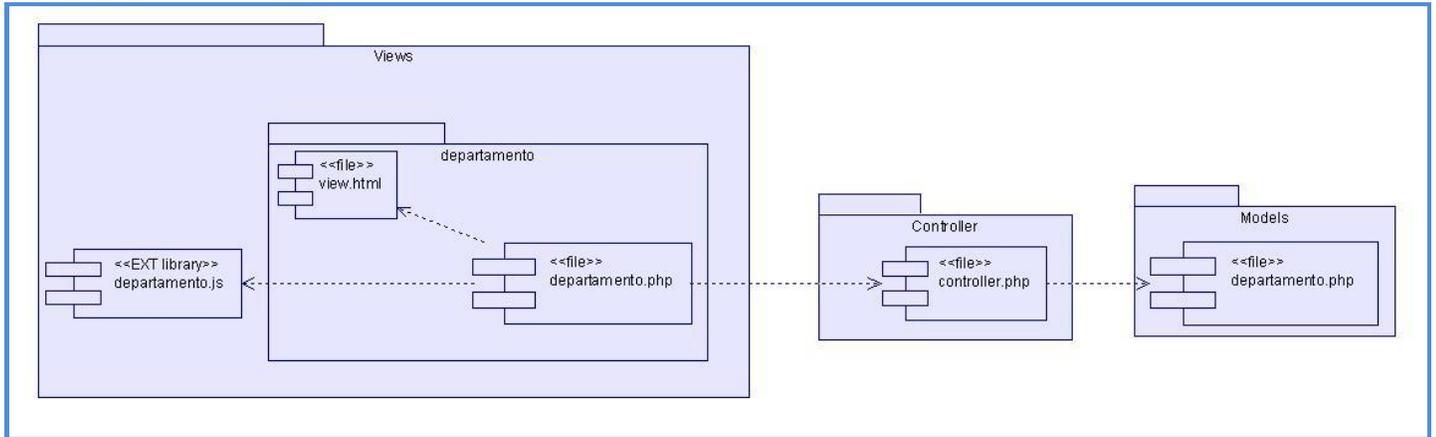


Imagen 9: Diagrama de Componentes del Caso de Uso Gestionar Departamento.

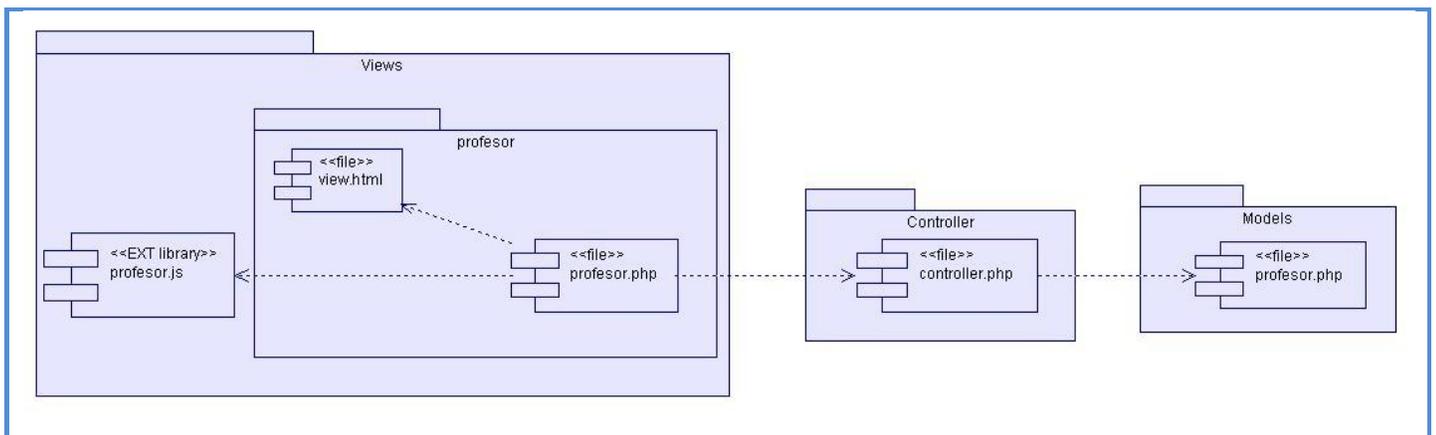


Imagen 10: Diagrama de Componentes del Caso de Uso Gestionar Profesor.

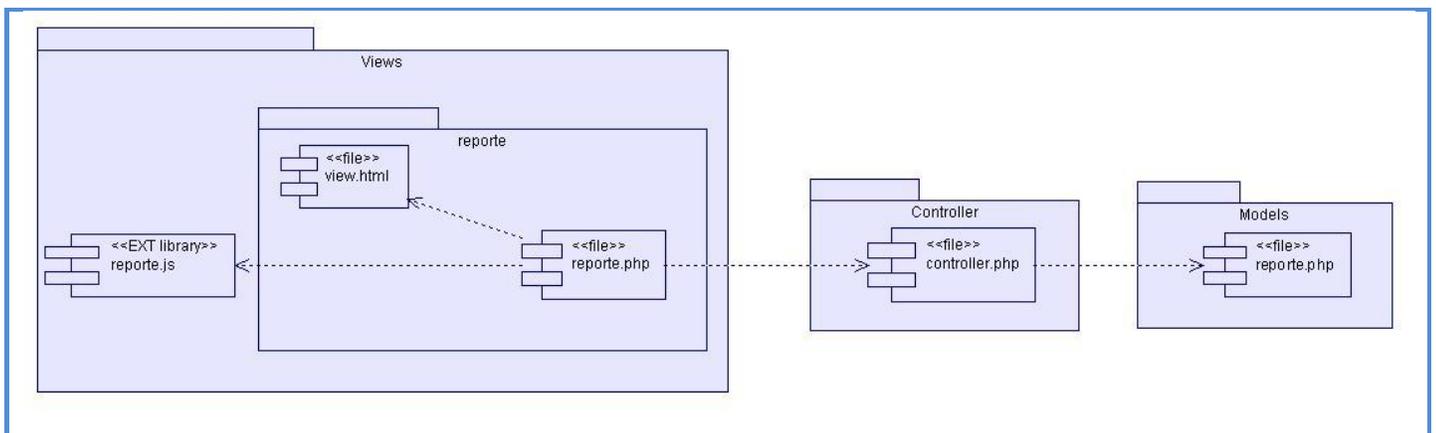
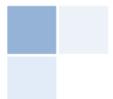


Imagen 11: Diagrama de Componentes del Caso de Uso Gestionar Mostrar Reporte.



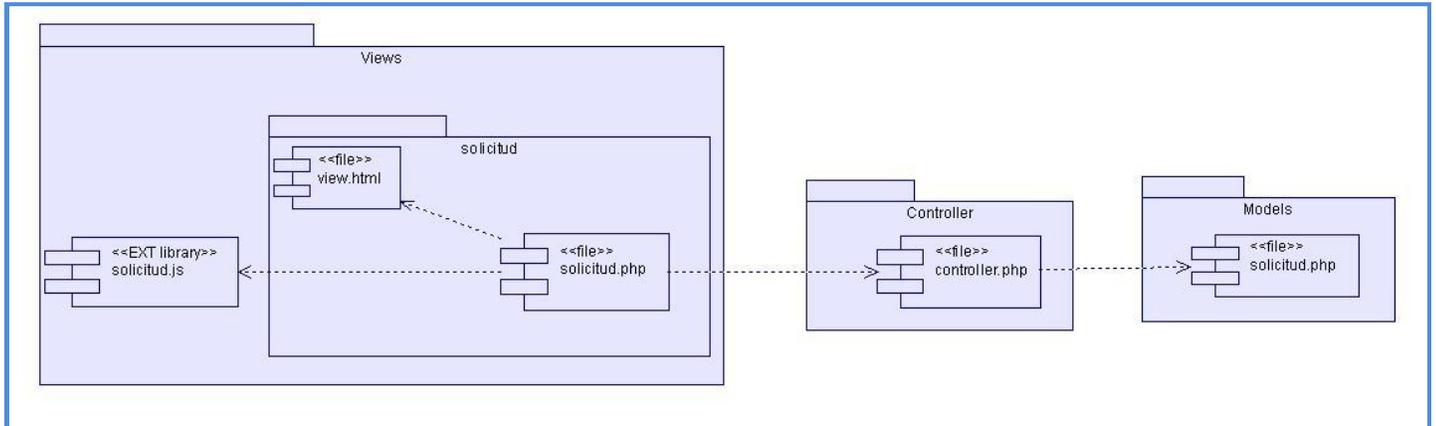
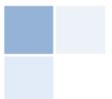
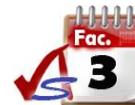


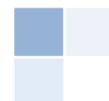
Imagen 12: Diagrama de Componentes del Caso de Uso Gestionar Solicitud.

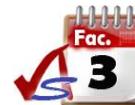




3.11 Anexo # 5: Casos de Prueba.

Escenario	Fecha de solicitud	Fecha a Efectuarse	Motivo de Solicitud	Breve descripción	Solicitudes	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
Adicionar Solicitud	V = 2009-05-26	V = 2009-05-30	V = Reunión de Departamento	I = &\$\$&?&?)?)&==/) T((//%%%TU GGUITGU/ (aaa\$\$%%\$E gdfg4565		El sistema debe mostrar un mensaje diciendo que hay un error en la entrada de los datos o que la cadena esta fuera del rango definido.	Satisfactorio.	1. Gestionar Datos c. Gestionar Solicitud. d. Adicionar Solicitud.
	V = 2009-05-26	V = 2009-05-30	I = 5445454))&==/) T45UGG UITGU/ (aaa445	V = Tengo que atenderme en el hospital por cusas de enfermedad .		El sistema debe mostrar un mensaje diciendo que hay un error en la entrada de los datos o que la cadena esta fuera del rango definido.	Satisfactorio.	
	V = 2009-05-26	I = No se selecciono ninguna fecha	V = Turno medico	V = Tengo que atenderme en el hospital por cusas de enfermedad .		El sistema debe mostrar un mensaje de error informando que se debe llenar el campo.	Satisfactorio.	
	I = No se selecciono ninguna fecha	V = 2009-05-26	V = Turno medico	V = Tengo que atenderme en el hospital por cusas de enfermedad .		El sistema debe mostrar un mensaje de error informando que se debe llenar el campo.	Satisfactorio.	
Eliminar Solicitud					I = No se selecciono ninguna solicitud	El sistema muestra un mensaje indicando que fue eliminada la solicitud	Satisfactorio.	1. Gestionar Datos a. Gestionar Solicitud. b. Eliminar Solicitud.





Escenario	Fecha de solicitud	Fecha a Efectuarse	Motivo de Solicitud	Breve descripción	Solicitudes	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
Estado Solicitudes					I = No se selecciono o ninguna solicitud	El sistema debe mostrar un mensaje de error informando que se debe llenar el campo.	Satisfactorio	1. Gestionar Datos <ul style="list-style-type: none"> a. Gestionar Solicitud. b. Estado Solicitudes.

Tabla 18: Descripción del Caso de Prueba del Caso de Uso-Gestionar Solicitud.

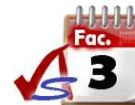
Escenario	Usuario	Nombre y Apellidos	Departamento	Profesor	Asignatura	Profesor-Asignatura	Grupo	Profesor-Asignatura-Grupo	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
Adicionar profesor	Vjcperez	V = Juan Carlos Perez Garcia	I = No selecciona ningun departamento						El sistema debe mostrar un mensaje de error informando que se debe llenar el campo.	Satisfactorio.	1. Gestionar Datos <ul style="list-style-type: none"> a. Gestionar Profesor b. Adicionar Profesor.
	Vjdelgado	I = dfkxlnkfer9848afe98qy3rfg3= dfkxlnkfer9848afe	V = Seleccionar "Humanidades"						El sistema debe mostrar un mensaje diciendo que hay un error en la entrada de los datos o que la cadena esta fuera del	Satisfactorio.	
	I = fbcwieew8hf9wefhbfihwefiwbeew	V = Manuel Tejada	V = Seleccionar "Español"						El sistema debe mostrar un mensaje diciendo que hay un error en la entrada de los datos o que la cadena esta fuera del	Satisfactorio.	
Vincular Profesor-Asignatura.				V = Seleccionar "Juan Carlos Perez Garcia"	I = No selecciona ningun asignatura				El sistema debe mostrar un mensaje de error informando que se debe llenar el	Satisfactorio.	1. Gestionar Datos <ul style="list-style-type: none"> a. Gestionar Profesor. b. Vincular Profesor-





Escenario	Usuario	Nombre y Apellidos	Departamento	Profesor	Asignatura	Profesor-Asignatura	Grupo	Profesor-Asignatura-Grupo	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
				I = No seleccionar ningún profesor	V = Seleccionar "Juan Carlos Perez"				El sistema debe mostrar un mensaje de error informando que se debe	Satisfactorio.	Asignatura.
Vincular Profesor-Asignatura-Grupo						V = Seleccionar "Juan Carlos Perez Garcia"	I = No seleccionar ningún grupo.		El sistema debe mostrar un mensaje de error informando que se debe llenar el	Satisfactorio.	1. Gestionar Datos <ul style="list-style-type: none"> a. Gestionar Profesor. b. Vincular Profesor-Asignatura-Grupo.
						I = No seleccionar ningún vinculo Profesor-Asignatura	V = Seleccionar "3104"		El sistema debe mostrar un mensaje de error informando que se debe llenar el campo.	Satisfactorio.	
Eliminar Vinculo Profesor-Asignatura				V = Seleccionar "Jairol Vázquez Peña"	I = No seleccionar ningun a asignatura.				El sistema debe mostrar un mensaje de error informando que se debe llenar el campo.	Satisfactorio.	1. Gestionar Datos <ul style="list-style-type: none"> a. Gestionar Profesor. b. Eliminar Vinculo Profesor-Asignatura.
Eliminar Vinculo Profesor-Asignatura-Grupo								I = No seleccionar ningún vinculo Prof-Asign-Grupo	El sistema debe mostrar un mensaje de error informando que se debe llenar el campo.	Satisfactorio.	1. Gestionar Datos <ul style="list-style-type: none"> a. Gestionar Profesor. b. Eliminar Vinculo Profesor-Asignatura-Grupo.
Eliminar Profesor				I = No seleccionar ningún profesor					El sistema debe mostrar un mensaje de error informando que se debe llenar el	Satisfactorio.	1. Gestionar Datos <ul style="list-style-type: none"> a. Gestionar Profesor. b. Eliminar Profesor.





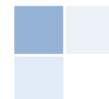
Escenario	Usuario	Nombre y Apellidos	Departamento	Profesor	Asignatura	Profesor-Asignatura	Grupo	Profesor-Asignatura-Grupo	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
Modificar Profesor	Vmsantos	VMaria Santos Romero	V= Selecciona "Humanidades"	I= No selecciona ningún profesor					El sistema debe mostrar un mensaje de error informando que se debe llenar el campo.	Satisfactorio.	1. Gestionar Datos a. Gestionar Profesor. b. Modificar Profesor.
	Vmsantos	VMaria Santos Romero	I= No selecciona ningún departamento	V= Selecciona el profesor "Carla Amara Sosa"					El sistema debe mostrar un mensaje de error informando que se debe llenar el campo.	Satisfactorio.	
	Vcamaran	I= KGfgFGKjdfdljgr8848hvrherehvoehdfvnsehvenrferh	V= Selecciona el departamento "Humanidades"	V= Selecciona el profesor "Carla Amara Sosa"					El sistema debe mostrar un mensaje diciendo que hay un error en la entrada de los datos o que la cadena esta fuera del rango definido.	Satisfactorio.	
	I,mxnrf4errhk bvkbrfbbfvkbvv	V=Raul Vegas Linare	V= Selecciona el departamento "Especialida	V= Selecciona el profesor "Carla Amara Sosa"					El sistema debe mostrar un mensaje diciendo que hay un error en la entrada de los datos o que la cadena esta fuera del rango	Satisfactorio.	

Tabla 19: Descripción del Caso de Prueba del Caso de Uso-Gestionar Profesor.





Escenario	Nombre del Departamento.	Nombre y Apellidos del Jefe Departamento.	Usuario del Jefe Dpto.	Listado Departamentos.	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
Adicionar Departamento.	V Humanidades	= V = Rafael Martínez García	I hfkdhf482 FHF\$\$\$r	=	El sistema debe mostrar un mensaje diciendo que hay un error en la entrada de los datos o que la cadena esta fuera del rango definido.	Satisfactorio.	1. Gestionar Datos <ul style="list-style-type: none"> a. Gestionar Departamento. b. Adicionar Departamento
	V Humanidades	= I dfkbc3872c23 2znxn	= V yromero	=	El sistema debe mostrar un mensaje diciendo que hay un error en la entrada de los datos o que la cadena esta fuera del rango definido.	Satisfactorio.	
	I msbdkbawie3i2 3i3eukdfk6565	= V = Karina Rodríguez Santana	V ksantana	=	El sistema debe mostrar un mensaje diciendo que hay un error en la entrada de los datos o que la cadena esta fuera del rango definido.	Satisfactorio.	
Eliminar Departamento.				I = No seleccionar ningún departamento.	El sistema debe mostrar un mensaje de error informando que se debe llenar el campo.	Satisfactorio.	1. Gestionar Datos <ul style="list-style-type: none"> a. Gestionar Departamento. b. Eliminar Departamento
Modificar Departamento.	V = Especialidad	V = Juan Pérez Romero.	V = jperez	I = No seleccionar ningún departamento.	El sistema debe mostrar un mensaje de error informando que se debe llenar el campo.	Satisfactorio.	1. Gestionar Datos <ul style="list-style-type: none"> a. Gestionar Departamento. b. Modificar Departamento





Escenario	Nombre del Departamento.	Nombre y Apellidos del Jefe Departamento.	Usuario del Jefe Dpto.	Listado Departamentos.	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
V	Humanidades	= Teresa Junquera Valdés	I = dfkjnesiw32393923238939	V = Seleccionar “Humanidades”	El sistema debe mostrar un mensaje diciendo que hay un error en la entrada de los datos o que la cadena esta fuera del rango definido.	Satisfactorio.	
V	Humanidades	= I = zdfkw472398283nekhd2393h#zcvzkckzxkxbzxbkzx	V = jcperez	V = Seleccionar “Humanidades”	El sistema debe mostrar un mensaje diciendo que hay un error en la entrada de los datos o que la cadena esta fuera del rango definido.	Satisfactorio.	
I	= mdkeiu823u23i238ndslnxc,nncn,zxxxzncnznzx,nz,nxznc,znxc,znx,nzxcnznzzxczxczxc	V = Juan Carlos Perez Garcia.	V = jcperez	V = Seleccionar “Especialidad”	El sistema debe mostrar un mensaje diciendo que hay un error en la entrada de los datos o que la cadena esta fuera del rango definido.	Satisfactorio.	

Tabla 20: Descripción del Caso de Prueba del Caso de Uso-Gestionar Departamento.

Escenario	Nombre	Abreviatura	Actividad Doc	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
Adicionar Actividad Docente	I= Dedicada a la liberación de nuestros Cinco Héroes Prisioneros del Imperio Inscribete	V= DL		El sistema debe mostrar un mensaje diciendo que hay un error en la entrada de los datos o que la cadena esta fuera del rango definido.	Satisfactorio	1. Gestionar Datos <ul style="list-style-type: none"> a. Gestionar Actividad Docente. b. Adicionar Actividad Docente.





Escenario	Nombre	Abreviatura	Actividad Doc	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
	V = Conferencia	I = Horario Facultad 3		El sistema debe mostrar un mensaje diciendo que hay un error en la entrada de los datos o que la cadena esta fuera del rango definido.	Satisfactorio	
Eliminar Actividad Docente			I = No seleccionar ninguna actividad docente.	El sistema debe mostrar un mensaje de error informando que se debe llenar el campo.	Satisfactorio	1. Gestionar Datos <ul style="list-style-type: none"> a. Gestionar Actividad Docente. b. Eliminar Actividad Docente.
Modificar Actividad Docente	V = Clase Práctica	V = CP	I = Introducir en el campo “Conferencia”	El sistema no debe permitir introducir ningún texto en el campo Actividad Doc.	Satisfactorio	1. Gestionar Datos <ul style="list-style-type: none"> a. Gestionar Actividad Docente. b. Modificar Actividad Docente.
	V = Laboratorio	I = Laboratorio	V = Seleccionar “Conferencia”	El sistema debe mostrar un mensaje diciendo que hay un error en la entrada de los datos o que la cadena esta fuera del rango definido.	Satisfactorio	
	I = Nuestra dirección no solo pone a disposición información general de la labor de nuestro claustro de profesores sino	V = S	V = Seleccionar “Seminario”	El sistema debe mostrar un mensaje diciendo que hay un error en la entrada de los datos o que la cadena esta fuera del rango definido.	Satisfactorio	

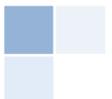
Tabla 21: Descripción del Caso de Prueba del Caso de Uso-Gestionar Actividad Docente.





Escenario	Solicitudes	Afectación	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
Aprobar o Rechazar Solicitud.	≡ No se seleccione ninguna solicitud		El sistema debe mostrar un mensaje de error informando que se debe seleccionar algún campo.	Satisfactorio.	1. Gestionar Datos <ul style="list-style-type: none"> a. Gestionar Afectación. b. Aprobar o Rechazar Solicitud.
Eliminar Afectación		≡ No se seleccione ninguna afectación	El sistema debe mostrar un mensaje de error informando que se debe seleccionar algún campo.	Satisfactorio.	1. Gestionar Datos <ul style="list-style-type: none"> a. Gestionar Afectación. b. Eliminar Afectación.

Tabla 22: Descripción del Caso de Prueba del Caso de Uso-Gestionar Afectación.





3.12 Anexo # 6. Carta de aceptación del cliente.

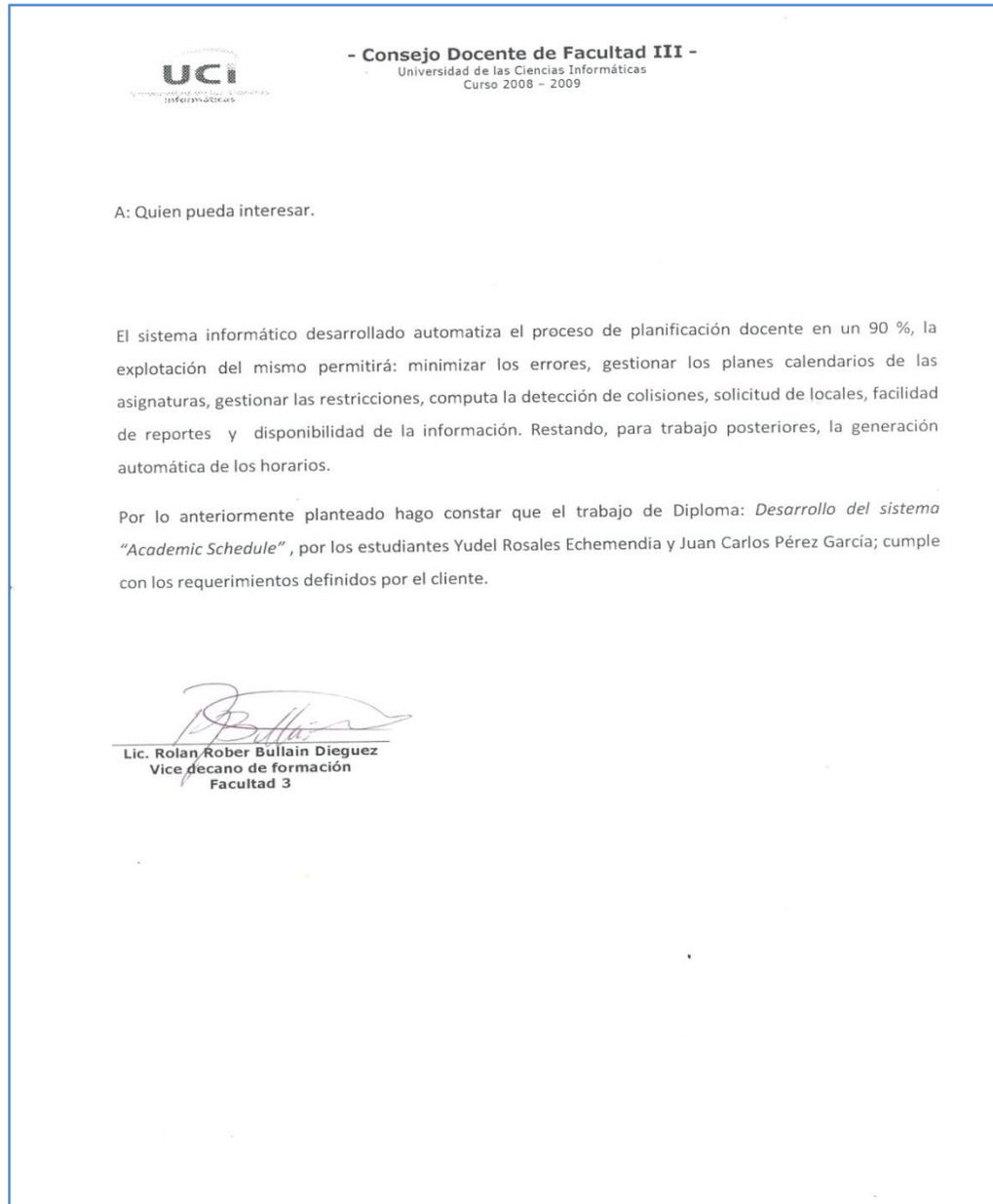
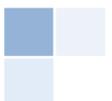


Imagen 13: Carta de aceptación del cliente.





Glosario de Términos

A

ACM: acrónimo de Association for Computing Machinery. Fue fundada en 1947 como la primera sociedad científica y educativa acerca de la Computación.

Actividad Cognoscitiva: es la acción mental mediante la cual el individuo, asimila ideas, crea y se recrea hasta llegar a la construcción del conocimiento.

B

Backtracking: “vuelta atrás” es una estrategia para encontrar soluciones a problemas que satisfacen restricciones.

Booch: es un lenguaje de modelado de objetos y una metodología ampliamente usada en el diseño de software orientado a objetos.

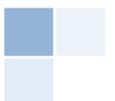
B – Prolog: es un sistema Prolog con extensiones para programar concurrencia, limitaciones, y gráficos interactivos.

C

Clase: una clase es un contenedor de uno o más datos (variables o propiedades) junto a las operaciones de manipulación de dichos datos (funciones/métodos).

Código: el código fuente de un programa informático (o software) es un conjunto de líneas de texto que son las instrucciones que debe seguir la computadora para ejecutar dicho programa.

Ciencia: conocimiento, que es un conjunto de información adquirida a través de la experiencia.





CASE: ayuda por Computadora a la Ingeniería de Software.

D

Didáctica: es una disciplina científico - pedagógico que vinculada a la organización escolar y a la orientación educativa la cual busca fundamentar y regular los procesos de enseñanza y aprendizaje.

E

Excepciones: el manejo de excepciones es una estructura de control de los lenguajes de programación diseñada para manejar condiciones anormales que pueden ser tratadas por el mismo programa que se desarrolla.

G

GNU: es una licencia de software libre que está diseñada para garantizar su libertad de compartir y modificar el software.

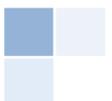
J

JavaScript: es un lenguaje de programación interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas web, con una sintaxis semejante a la del lenguaje Java y el lenguaje C.

L

Lenguaje: 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.

LGPL: es una licencia diseñada para asegurar que usted tiene la libertad de distribuir copias de Software Libre.





M

Método: utilizado principalmente en programación orientada a objetos, el término se refiere a las porciones de código asociadas exclusivamente con una clase (se les denomina entonces métodos de clase o métodos estáticos) o con un objeto (en este caso métodos de instancia).

O

OOSE: Ingeniería del software Orientada a Objeto es un idioma y metodología de modelado de objeto para manejar diseño de software.

OTM: Organización Mundial de Turismo.

P

Paradigma: un paradigma de programación representa un enfoque particular o filosofía para la construcción del software.

Programación: en informática, la programación es un proceso por el cual se escribe (en un lenguaje de programación), se prueba, se depura y se mantiene el código fuente de un programa informático.

Proceso: se refiere al cambio de estado desde un estado inicial hasta un estado final.

S

SWI: es una ejecución de fuente abierta del PROLOG de lenguaje de programación, comúnmente usado para las aplicaciones de red semánticas.

Software: es el conjunto de los programas de cómputo, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un sistema de computación.

