

**Universidad de las Ciencias Informáticas**

**Facultad 7**



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

**Sistema para la Gestión Administrativa de la Facultad 7.  
Módulo para la Planificación y Seguimiento de Actividades**

**Autores:** Alejandro Núñez Maldonado  
Yuniel Peraza Suárez

**Tutoras:** Ing. Arianne Méndez Mederos  
Ing. Lourdes Escalona Peral

Ciudad de la Habana, Julio de 2010  
“Año 52 de la Revolución”

## **DATOS CONTACTO**

**TUTORA: Ing. Lourdes Escalona Peral.** Graduada de Ingeniería Informática curso 2003-2004, Holguín-CUJAE. Ha impartido las asignaturas de Ingeniería de Software 1 y 2, Metodologías de la Investigación Científica y Seminario de Tesis. Fue líder del proyecto Atención Primaria de Salud y del proyecto de Calidad de la Facultad 7.

Por cuatro años consecutivos desempeña la función de Jefa de Departamento de asignaturas de la especialidad. El trabajo sobre la estrategia para evaluar la Práctica profesional en la Facultad, en el cual fue coautora, obtuvo premio del rector en el año 2007. Presidenta de la comité técnico de la Facultad, así como responsable del comité de tesis por 3 años consecutivos. Resultó cuadro destacado durante el año 2008.

**Correo-Electrónico:** [lescalonap@uci.cu](mailto:lescalonap@uci.cu)

**TUTORA: Ing. Arianne Méndez Medero.**

**Correo-Electrónico:** [arianne@uci.cu](mailto:arianne@uci.cu)

## RESUMEN

El presente trabajo surge ante la necesidad de viabilizar el proceso de gestión de la planificación y seguimiento de las actividades de los profesores en la Facultad 7 de la Universidad de las Ciencias Informáticas.

Es por ello que se propone en esta investigación la implementación de un sistema que permita viabilizar la gestión de la información que se genera en los procesos relacionados con la gestión de la planificación y seguimiento de las actividades de los profesores en la Facultad 7 de la Universidad de las Ciencias Informáticas

Debido a que la solución propuesta formará parte del Sistema Integral de Gestión de Administrativa de la Facultad, la dirección de la misma definió un conjunto de tecnologías y herramientas a emplear para la implementación de los mismos. Como Metodología de desarrollo se define Rational Unified Process (RUP). Como lenguaje de Modelado Lenguaje de Modelado Unificado (UML 2.0). Como herramienta de modelado Enterprise Architect (EA). Como gestor de bases de datos se tiene PostgreSQL. Se usa Python como lenguaje de programación, Django como marco de desarrollo web y WingIDE como entorno de desarrollo. Además se implementa el patrón de arquitectura Model View Template.

Una vez desarrollado el sistema se espera facilitar el trabajo de los directivos y de los profesores, referente a la forma en que se dará seguimiento a las actividades que realiza durante su vida laboral; y posibilite llegar a una evaluación final basada en el cumplimiento o incumplimiento de las diferentes acciones planificadas en la institución donde radica, en este caso, la Facultad 7 en la Universidad de las Ciencias Informáticas (UCI).

## TABLA DE CONTENIDOS

INTRODUCCIÓN.....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	4
1.1. Sistemas automatizados existentes vinculados al campo de acción.....	8
1.2. Metodología de desarrollo de software, lenguaje y herramienta para el modelado.....	12
1.2.1. RUP.....	13
1.2.2. UML.....	15
1.2.3. Enterprise Architect (EA) .....	17
1.3. Sistema Gestor de Base de Datos .....	18
1.3.1. PostgreSQL.....	19
1.4. Lenguaje de programación y entorno de desarrollo.....	20
1.4.1. Python .....	20
1.4.2. Framework Django .....	22
1.4.3. Wing IDE .....	23
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA .....	25
2.1. Objeto de Estudio.....	25
2.1.1. Flujo actual de los procesos involucrados en el campo de acción.....	25
2.1.2. Análisis crítico de la ejecución de los procesos.....	25
2.2. Información que se maneja. ....	26
2.3. Objeto de automatización.....	26
2.4. Modelo de Dominio. ....	26
2.5. Reglas del Negocio. ....	28
2.6. Especificación de los Requisitos del Software.....	28
2.6.1. Requisitos funcionales.....	28
2.6.2. Requisitos no funcionales .....	29
2.7. Modelo de Casos de Uso .....	31
2.7.1. Actores del Sistema.....	31
2.7.2. Diagrama de Caso de Uso del Sistema .....	31
2.7.3. Especificación de Casos de Uso del Sistema .....	32
2.7.4. Casos de Uso expandidos .....	33
CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA.....	42
3.1. Descripción de la arquitectura. Fundamentación.....	42

3.2. Análisis.....	44
3.2.1. Diagramas de Clases de Análisis.....	44
3.3. Diseño.....	47
3.3.1. Principios de Diseño .....	47
3.3.2. Diagrama de clases del diseño .....	47
3.3.3. Modelado mediante estereotipos Web.....	47
3.3.4. Diagramas de interacción .....	51
3.3.5. Descripción de las clases del diseño. ....	56
CAPÍTULO 4. IMPLEMENTACIÓN. ....	60
4.1. Modelo de datos .....	60
4.2. Diagrama de componentes .....	63
4.3 Diagrama de despliegue .....	66
4.5 Seguridad. ....	68
CONCLUSIONES .....	70
RECOMENDACIONES .....	71
REFERENCIAS BIBLIOGRÁFICAS.....	72
BIBLIOGRAFÍA.....	73
GLOSARIO .....	75



## INTRODUCCIÓN

Con el paso del tiempo la Informática ha tomado auge y continúa desarrollándose, de manera que actualmente se lleva a cabo una etapa de informatización masiva, en la que la mayoría de los países buscan soluciones rentables a sus problemas apoyándose principalmente en esta ciencia.

Planificar bien las tareas en una institución como la Universidad de las Ciencias Informáticas (UCI) es sin duda alguna, un problema difícil, más cuando se trata del gran número de trabajadores que, orientados por el Reglamento Docente-Metodológico de la Universidad están comprometidos no solo a impartir clases, sino que realizan además actividades de acuerdo a su categoría docente y otras relacionadas con la Formación, Ciencia y Técnica, Producción, Superación y Extensión Universitaria, las cuales son efectuadas asiduamente. La planificación de las actividades y el seguimiento de la participación de los profesores en estas, se dificulta debido a las diversas responsabilidades que pueden tener los mismos.

Actualmente la Facultad 7 de dicha Universidad presenta dificultades a la hora de llevar a cabo un seguimiento y control de las actividades que se programan a los profesores, ya sean docentes, productivas o extracurriculares en cada una de las estructuras existentes en la misma. Esto trae como consecuencia en la mayoría de los casos se reciban indicaciones de diferentes instancias de la Facultad ya sean de otros departamentos, de la UJC, de la producción e incluso, de la propia Universidad, lo que conlleva a que al mismo profesor le coincidan varias actividades. La falta de control de todas las actividades asignadas a un profesor, provoca en la mayoría de los casos, que algunos estén sobrecargados con respecto a otros.

Dichas situaciones hacen que se vea afectada la toma de decisiones pues no se efectúa la planificación con la organización requerida, lo cual es necesario y de vital importancia. Identificándose las siguientes situaciones:

- Existen ineficiencias a la hora de divulgar las informaciones necesarias tales como: plan de trabajo, avisos, convocatorias, documentos legales.
- Se aprecian dificultades referidas a la divulgación de la planificación de actividades como: distribución de la guardia obrera, trabajos voluntarios, eventos, entre otras. Esto sucede debido a que la vía que se utiliza para la divulgación es el correo electrónico, medio que puede



provocar pérdida de información o arribo de la información fuera del tiempo necesario. No permite llevar un registro histórico con todos los datos de forma organizada.

- Difícil acceso y visualización de los registros.

A raíz de la situación problemática antes expuesta, se formula el siguiente **Problema a resolver**:

¿Cómo viabilizar el proceso de gestión la planificación y seguimiento de las actividades de los profesores de la Facultad 7 de la Universidad de las Ciencias Informáticas?

Se define como **Objeto de estudio** el proceso de gestión administrativa de la facultad, centrando el **Campo de acción** el proceso de gestión de la información relacionada con la gestión de planificación y seguimiento a las actividades de los profesores en la Facultad 7 de la Universidad de las Ciencias Informáticas (UCI)

Generándose como **Objetivo General** de la investigación el desarrollo de una aplicación web para la gestión de la planificación y seguimiento de las actividades de los profesores de la Facultad 7.

Derivándose las siguientes tareas de la investigación:

- Realizar un análisis acerca de los sistemas informáticos existentes a nivel nacional e internacional referentes a la gestión de la información relacionada con la gestión de la planificación y seguimiento de actividades.
- Asimilar las herramientas a usar para el análisis, diseño e implementación del sistema de gestión de la planificación y seguimiento de las actividades de los profesores.
- Caracterizar la metodología de desarrollo de software a usar en el proceso de desarrollo del sistema.
- Diseñar el sistema informático utilizando la arquitectura definida por el desarrollo del Sistema Integral de Gestión Administrativa.
- Implementar el sistema informático aplicando las pautas de diseño y siguiendo las necesidades de funcionamiento establecidas en la Especificación de Requisitos de Software.



---

A continuación se presenta un breve resumen de los capítulos en que fue distribuido el desarrollo de la investigación.

**Capítulo 1: Fundamentación Teórica:** Se exponen un conjunto de conceptos asociados al dominio del problema planteado que ayudarán a familiarizarse con el entorno en el que se manifiesta la investigación, y con los que se espera la comprensión de la solución propuesta y la necesidad de la misma. Además se describen en detalles elementos relacionados al objeto de estudio que dio origen a este trabajo.

**Capítulo 2: Presentación de la Solución Propuesta:** Se realiza la descripción de la solución propuesta del trabajo. Se elabora el análisis y diseño del software generando todos los artefactos fundamentales a partir de la modelación de los diferentes diagramas realizados, así como la identificación y descripción del flujo de trabajo de negocio y la definición de los requisitos funcionales y no funcionales que debe cumplir y tener el sistema respectivamente.

**Capítulo 3: Construcción de la Solución Propuesta:** Incluye la elaboración de los principales artefactos necesarios para el mejor entendimiento del sistema a implementar, estos artefactos están conformados por los diagramas de clases del análisis y el diseño que propone la metodología empleada. Se muestra también el modelo de datos a utilizar, la descripción de las tablas de la base de datos y algunos patrones de diseño a tener en cuenta.

**Capítulo 4: Implementación:** Se realiza toda la implementación del sistema propuesto, mostrando los principales diagramas de implementación, como los diagramas de despliegue y componentes. Se realiza la descripción de los casos de pruebas de integración que se llevan a cabo sobre el sistema y además se efectúa el cálculo de costos y beneficios para tener una idea de cuanto tiempo durará el proyecto y cuanto le costará a la Universidad el desarrollo del software que se desea implementar.





## **Introducción**

Este capítulo es el resultado de la búsqueda y análisis de la información relacionada con el objeto de estudio, se realiza el análisis del estado del arte del tema tratado, de las tendencias, técnicas, tecnologías, metodologías y herramientas usadas en la actualidad o de las que sirven de apoyo para la solución del problema a resolver.

### **¿Qué es la gestión?**

La gestión es el conjunto de tareas que se realizan para desarrollar un proceso o para lograr un producto determinado a través de la dirección o administración de una empresa o de un negocio. Otra forma de considerar la gestión es plantearla como "Una función institucional global e integradora de todas las fuerzas que conforman una organización". [1]

En ese sentido la gestión hace énfasis en la dirección y en el ejercicio del liderazgo. Existen diferentes tipos de gestión, tales como: la social, de proyectos, del conocimiento, gestión ambiental, etcétera. La gestión enfatiza la realización de diligencias conducentes al logro de un negocio o un deseo cualquiera. Involucra el conjunto de trámites que se ejecutan para resolver un asunto o concretar un proyecto. Se puede resumir la gestión como el proceso de planeación y manejo de tareas y recursos.

### **La Gestión de la Información.**

El estudio de los Sistemas de Información se originó como una subdisciplina de las ciencias de la computación en un intento por entender y racionalizar la administración de la tecnología dentro de las organizaciones. Estos sistemas tratan el desarrollo, empleo y administración de la infraestructura de la tecnología de la información en una organización. Además de ser un conjunto organizado de elementos originados por:

- Personas.
- Datos.
- Actividades o técnicas de trabajo.



- Recursos materiales en general (típicamente recursos informáticos y de comunicación, aunque no tienen por qué ser de este tipo obligatoriamente).

Todos estos elementos interactúan entre sí para procesar los datos y la información (incluyendo procesos manuales y automáticos) y distribuirla de la manera más adecuada posible en una determinada organización en función de sus objetivos.

Los Sistemas de Información han ido madurando hasta convertirse en un campo de estudios superiores dentro de la administración. Adicionalmente, cada día se enfatiza más como un área importante dentro de la investigación en los estudios de administración. Durante los próximos años, según se ha evidenciado a través de estudios realizados, estos sistemas cumplirán tres objetivos básicos dentro de las organizaciones:

- Automatización de procesos operativos.
- Proporcionar información relevante al proceso de toma de decisiones.
- Lograr ventajas competitivas a través de su implantación y uso.

Un Sistema de Información realiza cuatro actividades básicas: entrada, almacenamiento, procesamiento y salida de información, las cuales se detallan a continuación:

Entrada de Información: es el proceso que toma los datos requeridos para procesar la información y las entradas pueden ser manuales o automáticas. Las manuales son aquellas que se proporcionan en forma directa por el usuario, mientras que las automáticas son datos o información que provienen o son tomados de otros sistemas o módulos. Estos últimos se denominan interfaces automáticas.

Almacenamiento de información: el almacenamiento es una de las actividades o capacidades más importantes que tiene una computadora, ya que a través de esta propiedad el sistema puede recordar la información guardada en la sección o proceso anterior. Esta información suele ser almacenada en estructuras de información denominadas archivos.

Procesamiento de Información: es la capacidad para efectuar cálculos de acuerdo con una secuencia de operaciones preestablecidas. Estos cálculos pueden efectuarse con datos introducidos recientemente en el sistema o bien con datos que están almacenados. Esta



característica de los sistemas permite la transformación de datos fuente en información que puede ser utilizada para la toma de decisiones.

Salida de Información: la salida es la capacidad para extraer la información procesada o datos de entrada al exterior. Es importante aclarar que la salida de un Sistema de Información puede constituir la entrada a otro Sistema de Información o módulo.

## **La Planificación.**

El mundo de los negocios es un mundo volátil, complejo y sin reglas, pero sobre todo es un mundo incierto. Ante estas características se necesitan herramientas para minimizar el riesgo y la planificación es una de ellas.

Planificar supone reducir el riesgo de una determinada acción al permitir una anticipación a sus consecuencias. Es un proceso continuo que refleja los cambios del ambiente en torno a cada organización y busca adaptarse a ellos. Una de las cuestiones que toda organización debe hacer, es la de ordenar el trabajo que se va a desarrollar, el cual siempre debe obedecer a unos objetivos establecidos, previamente estipulados en la planificación. De esta manera la planificación es considerada como el ejercicio a ejecutar con el fin de reducir el riesgo en la empresa y organizar los medios productivos en correspondencia con los objetivos buscados.

## **Propósitos principales de la planificación.**

En el entorno organizacional, la planificación cumple dos propósitos principales en las organizaciones: el protector y el afirmativo. El propósito protector consiste en minimizar el riesgo, reduciendo la incertidumbre que rodea al mundo de los negocios y definiendo las consecuencias de una acción administrativa determinada. El propósito afirmativo de la planificación consiste en elevar el nivel de éxito organizacional. [2]

Un propósito adicional de la planificación consiste en coordinar los esfuerzos y los recursos dentro de las organizaciones. Se ha dicho que la planificación es como una locomotora que arrastra el tren de las actividades de la organización, la dirección y el control.



Por otro lado, se puede considerar a la planificación como el tronco fundamental de un árbol imponente, del que crecen las ramas de la organización, la dirección y el control. Sin embargo, el propósito fundamental es facilitar el logro de los objetivos de la empresa. Implica tomar en cuenta la naturaleza del ámbito futuro en el cual deberán ejecutarse las acciones planificadas.

La planificación es un proceso continuo que refleja los cambios del ambiente en torno a cada organización y busca adaptarse a ellos. Uno de los resultados más significativos del proceso de planificación es una estrategia para la organización.

### **Sistema de Gestión.**

En el siglo XXI, para poder convertir las presiones de la competencia en ventajas comparativas, las empresas se ven en la necesidad de aumentar el rendimiento operativo en forma sistemática, por lo que los sistemas de gestión pueden ayudar a centrar, organizar y sistematizar los procesos para la gestión y mejora. Algunos de los retos por los que transitan las empresas son:

- Rentabilidad.
- Competitividad.
- Globalización.
- Velocidad de los cambios.
- Capacidad de adaptación.
- Crecimiento.
- Tecnología.

Equilibrar éstos y otros requisitos empresariales puede constituir un proceso difícil y desalentador. Es aquí donde entran en juego los sistemas de gestión, al permitir aprovechar y desarrollar el potencial existente en la organización. El uso de uno de estos sistemas permite renovar constantemente los objetivos, las estrategias, las operaciones y los niveles de servicio de las empresas; además de que la implementación eficaz de los mismos puede ayudar a:

- Gestionar los riesgos sociales, medioambientales y financieros.
- Mejorar la efectividad operativa.
- Reducir costos.
- Aumentar la satisfacción de clientes y partes interesadas.
- Proteger la marca y la reputación.



- Lograr mejoras continuas.
- Potenciar la innovación.
- Eliminar las barreras al comercio.
- Aportar claridad al mercado. [3]

## 1.1. Sistemas automatizados existentes vinculados al campo de acción

### Antecedentes Internacionales.

#### Redmine.

Es un código abierto, multiplataforma, basado en la web de gestión de proyectos y seguimiento de errores. Incluye el calendario y los Diagramas de Gantt para ayudar a la representación visual de los proyectos y sus plazos. Es compatible con múltiples proyectos. Escrito utilizando el framework Ruby on Rails, Redmine brinda la posibilidad de dar seguimiento de problemas y soporte para múltiples opciones de control de versión. Asimismo, con una solución de software libre, puede encontrar un scrum útil y ágil alternativa. [4]

#### Quartz.

Es un sistema de planificación de trabajos de fuente abierta que puede ser integrado con aplicaciones Java, ya sean de escritorio o empresariales (web). Está lleno de características que le permiten crear planificadores simples o complejos, de forma sencilla, para ejecutar decenas, cientos o incluso, miles de trabajos, los cuales son definidos como componente Java estándares. Además, Quartz incluye características de nivel empresarial como transacciones JTA y clusterización, lo cual le brinda una gran escalabilidad. [5]

#### TimeGT.

Permite clasificar tareas y citas en carpetas temáticas, tanto para tus obligaciones laborales como proyectos personales. Además, el usuario dispone de etiquetas y de un buscador para no perder ninguna tarea. Añadir una nueva tarea en TimeGT es tan fácil como acudir a la bandeja de sistema o ir a la carpeta indicada. Basta con una frase que resuma lo que tiene que hacer el usuario, aunque puede incluir notas y una fecha de inicio y de finalización. Con TimeGT puede importar sus tareas de Remember The Milk, así como sincronizar tareas con Google Calendar.



## **Dates Reminder.**

Es una agenda diseñada para recordar eventos y fechas importantes como cumpleaños, aniversarios, etcétera. Se ejecuta directamente desde la bandeja de sistema. Su interfaz está dividida en dos apartados. En el superior se listan los eventos ya existentes y en el inferior se añaden los nuevos. El manejo de Dates Reminder es muy intuitivo, apenas se necesitan indicaciones para entender su funcionamiento y prescinde únicamente de un sistema de alertas. [6]

## **Sistema de Gestión de la Actividad Docente en la Universidad de León, España.**

Este Sistema es una Aplicación Desktop que permite la gestión de información sobre la carga docente en esta Universidad, por ejemplo, controlar las altas y bajas de los profesores y de las asignaturas que imparten, controlar las modificaciones del plan docente existente, conocer disponibilidad de profesores, listar las asignaturas existentes por áreas y brindar información a los profesores sobre todo lo que ocurre en función de la docencia.

## **Universidad de Antioquia (España).**

En esta Universidad se utiliza un plan de trabajo para profesores donde se recogen todas las actividades realizadas por un profesor durante todo un curso académico, basándose para ello en un archivo en formato de MS Excel (Ver anexo 1), en el cual quedan recogidas todas las actividades relacionadas con este proceso. En dicha institución aún no se ha logrado informatizar completamente este proceso, por lo que resulta un tanto tediosa la manera en que se maneja la información.

## **Universidad Javeriana (Bogotá).**

La Universidad Javeriana ha optado por una gestión planificada y esto implica, entre otras cosas, planear las actividades del cuerpo profesoral. También implica disponer de información oportuna y veraz para la planeación de la Universidad en sus distintos niveles: Unidad, Facultad, Universidad como un todo.



Corresponde a las facultades distribuir las actividades académicas y de gestión entre los miembros del cuerpo profesoral existente con los objetivos de planeación, seguimiento y evaluación. También es importante consignar esta distribución en los sistemas de información de la Universidad con el propósito de tener información institucional consolidada, válida y oportuna (Ver anexo 2). Las directrices pueden facilitar a las facultades, la distribución equitativa de las actividades de sus profesores.

El sistema que se aplica en esta Universidad es una buena opción para la solución del problema a resolver pero tiene algunos inconvenientes:

- La forma en que se maneja el Plan de Trabajo de Profesores tiene algunas funcionalidades que en la universidad no se tiene en cuenta para la evaluación profesoral.
- La aplicación no es de libre acceso, por lo que solo se pudo acceder al manual de usuario. Obligatoriamente hay que tener privilegios previos y pertenecer a la Universidad para poder acceder a dicho sistema.
- Utiliza además, para su implantación el sistema de gestor de Base de Datos: Oracle, el cual es un sistema propietario.
- Otra dificultad que se encontró en el sistema, es que constituye una aplicación desktop, lo que implica su instalación y utilización sólo en los ordenadores identificados.

### **Antecedentes Nacionales.**

El **Sistema de Gestión de la Información de un departamento docente en la Universidad de Matanzas**, es otro ejemplo de lo que se quiere lograr. Este Sistema se ha diseñado y elaborado para realizar la gestión de la información en un departamento docente, teniendo en cuenta la necesidad del intercambio de información, rápido acceso y nivel de actualización. Se desarrolla en ambiente Web, utilizando PHP e Interbase. Está constituido por cuatro módulos:

1. Módulo de Capital Humano.
2. Módulo de Formación del Profesional.
3. Módulo de Educación de Postgrado.



#### 4. Módulo de Ciencia y Técnica.

Estos módulos facilitan la gestión de la información del capital humano de un departamento docente, dentro de ellos quedan registrados los datos personales de los trabajadores del Departamento, cumplimiento del plan de trabajo, los objetivos del profesor y su evaluación. Se controlan los datos referentes a sus asesoramientos a otros profesores. También se desarrolla todo el trabajo de planificación de la carga docente de los profesores. Por otro lado, se gestiona la documentación de la educación, posgraduados, cursos, maestrías, diplomados, doctorados que se ofertan o son recibidos por profesores del departamento, obteniendo de reportes como el plan de postgrados del departamento, el estado de la superación del claustro, su planificación y control.

Permite también, la inscripción en las diferentes opciones del sistema de postgrado. Además de lo anterior, de igual modo, se trabaja con la documentación de ciencia y técnica a través de proyectos, líneas de investigación, publicaciones, de los profesores de un departamento docente. Este sistema está en fase de implantación y ha sido probado parcialmente en algunos departamentos de la Universidad de Matanzas "Camilo Cienfuegos", Cuba.

#### **Sistemas automatizados existentes en la Universidad de las Ciencias Informáticas.**

En la Universidad de Ciencias Informáticas (UCI) se han desarrollado también sistemas de planificación. Durante el curso 2006-2007 surgió una propuesta titulada: “**Sistema de Gestión de Información de la Facultad 8**” (SGIF). Para dar solución a los problemas existentes en dicha Facultad; el mismo está compuesto por varios módulos, cada uno con responsabilidades específicas. Un módulo para la gestión de la residencia estudiantil de dicha facultad, otro para el control de las Investigaciones Científicas que se desarrollen a cualquier nivel, con sus estudiantes y profesores participantes. Así como el encargado del control de la Producción.

Conjuntamente con estos, se encuentran los módulos referentes a los términos docencia, cursos optativos, planificación docente, sindicato. Dentro de los módulos que implementa no existe ninguno que gestione la información referente al plan de trabajo y de desempeño entre otras actividades para el control de la información de los profesores. Este sistema está desarrollado con el framework Django y el Sistema gestor de Base de Datos utilizado fue PostgreSQL.





En la UCI existe también el Sistema de Gestión para la planificación y control de las actividades de los profesores desarrollado en la Facultad 4. Este sistema posibilita gestionar datos del profesor, gestionar el plan de trabajo del profesor, gestionar las actividades extracurriculares del profesor, registrar incidencias de los profesores, gestionar el plan de trabajo del profesor en la producción, además de mostrar un reporte de las evaluaciones del profesor. Este carece de funcionalidades que la Facultad 7 necesita, como la gestión de las actividades de superación y de desempeño (con todos los aspectos que este proceso requiere). Este sistema está desarrollado en PHP versión 5.1.6 y el Sistema gestor de Base de Datos utilizado fue PostgreSQL.

Todos los sistemas internacionales ya expuestos no cumplen con las expectativas de los requerimientos de la Facultad 7 para la planificación de actividades laborales, ya que muchos de ellos requieren de una conexión constante a internet y no permiten una buena administración del sistema, además las propuestas cubanas están hechas para dar soluciones a problemas propios de la institución y no a las necesidades de la facultad. Por su parte los sistemas de la Universidad se identificaron un grupo de actividades que no se ponen de manifiesto, aunque se podría adaptar como posible solución al problema a resolver con esta investigación. También se identificaron otras que si se realizan, pero aún no son suficientes de acuerdo con la necesidad que existe.

Por tanto, estas soluciones tampoco pueden ser aplicadas para resolver el problema de esta investigación. Debido a todos los inconvenientes de los sistemas encontrados, se percibió la necesidad del desarrollo de un sistema que permita informatizar todos los procesos que están vinculados a la gestión de la planificación de los profesores de la Facultad 7 de la UCI.

## **1.2. Metodología de desarrollo de software, lenguaje y herramienta para el modelado**

La creciente informatización de los procesos productivos y sociales, ha traído consigo que las organizaciones y empresas requieran cada vez más de software confiable y de alta calidad, tanto en su desarrollo como en su mantenimiento. Es por ello que en los últimos años se han venido publicando estándares, notaciones y metodologías que establecen buenas prácticas para los procesos de desarrollo de software. La metodología de desarrollo es un conjunto de procedimientos, técnicas, herramientas, y un soporte documental que ayuda a los desarrolladores a realizar nuevo software. En otras palabras, una metodología “representa el camino para desarrollar software de una manera sistemática”. Las metodologías persiguen tres necesidades principales:

- Mejores aplicaciones, conducentes a una mejor calidad.



- Un proceso de desarrollo controlado.
- Un proceso normalizado en una organización, no dependiente del personal.

Algunas de las metodologías que se pueden utilizar para guiar un proceso de desarrollo de software son:

- La Metodología Rational Unified Process (RUP) es más adaptable para proyectos de largo plazo.
- La Metodología Extreme Programming (XP) en cambio, se recomienda para proyectos de corto plazo.
- La Microsoft Solution Framework (MSF) se adapta a proyectos de cualquier dimensión y de cualquier tecnología.

En ocasiones, el diseño de un software se realiza de manera rígida con los requerimientos que el cliente solicitó, provocando un difícil mantenimiento si el usuario final solicita un cambio durante la fase de prueba. Esto es consecuencia de no usar o aplicar una metodología adecuada, debido a la falta de conocimiento sobre este tema.

## **1.2.1. Metodología Proceso Unificado de Desarrollo (RUP)**

Acrónimo de Rational Unified Process (Proceso Unificado de Rational) es un proceso de ingeniería de software, una forma disciplinada de asignar tareas y responsabilidades (quién hace qué, cuándo y cómo) en una organización o equipo de desarrollo de software. Es también un producto desarrollado y mantenido por Rational, actualizado constantemente para tener en cuenta las mejores prácticas de acuerdo con la experiencia. Junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

La metodología RUP divide en 4 fases el proceso de desarrollo de software:

Inicio: El objetivo o hito de esta fase es determinar la visión del proyecto.

Elaboración: En esta fase el hito es determinar la línea base de arquitectura.

Construcción: En esta fase el hito es llegar a obtener la capacidad operacional inicial.

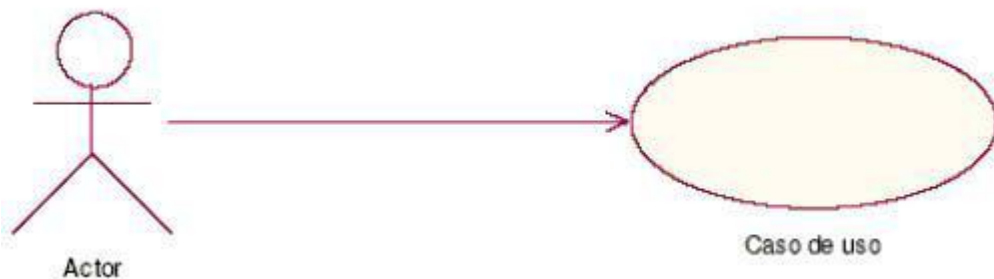
Transición: El hito de esta fase es llegar a obtener una primera versión del proyecto.



Cada una de estas fases es desarrollada mediante ciclos de iteraciones. Una particularidad de esta metodología es que, en cada ciclo de iteración, se hace exigente el uso de artefactos, siendo por este motivo, una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software.

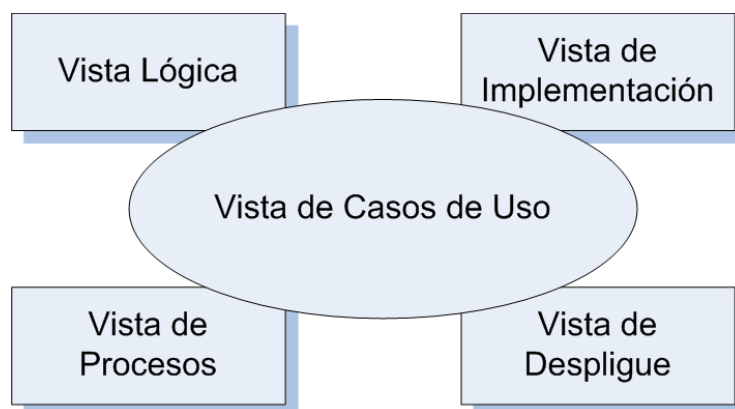
**Características principales de RUP.**

Dirigido por casos de uso: Los casos de uso capturan requerimientos funcionales y representan piezas de funcionalidad que brindan un resultado de valor al usuario.



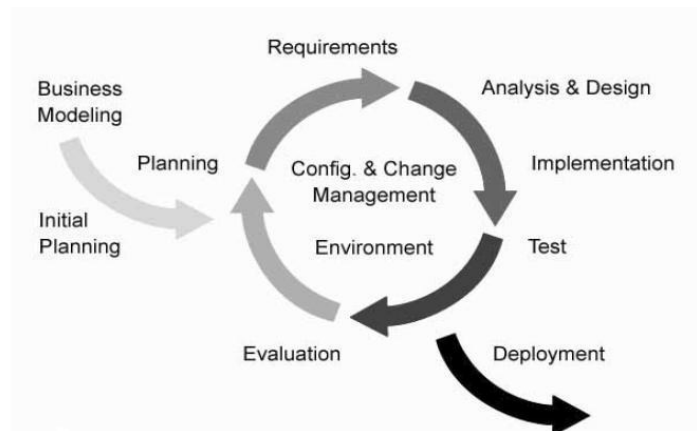
**Figura 1 Diagrama de casos de uso.**

Centrado en una arquitectura: Comprende los aspectos estáticos y dinámicos más importantes del sistema.



**Figura 2. Vistas de la arquitectura.**

Iterativo e incremental: El trabajo se divide en piezas pequeñas o mini proyectos; cada uno proveyendo un subproducto incremental.



**Figura 3. Flujos de trabajo de RUP.**

Para desarrollar este trabajo se decidió utilizar la metodología RUP, la cual es una de las metodologías pesadas, pues está basada en normas provenientes de estándares seguidos por el entorno de desarrollo y proceso mucho más controlado, con numerosas políticas. Está pensada para adaptarse a cualquier proyecto, divide el proceso de desarrollo, en ciclos de iteración, teniendo un producto final, al concluir cada ciclo y es una de las metodologías más importante para alcanzar un grado de certificación en el desarrollo del software. La misma forma parte del Documento de Arquitectura de la Facultad 7 y además fue la seleccionada por la Dirección de dicha entidad para el desarrollo del software que se desea obtener.

### **1.2.2. Lenguaje unificado de modelado (UML)**

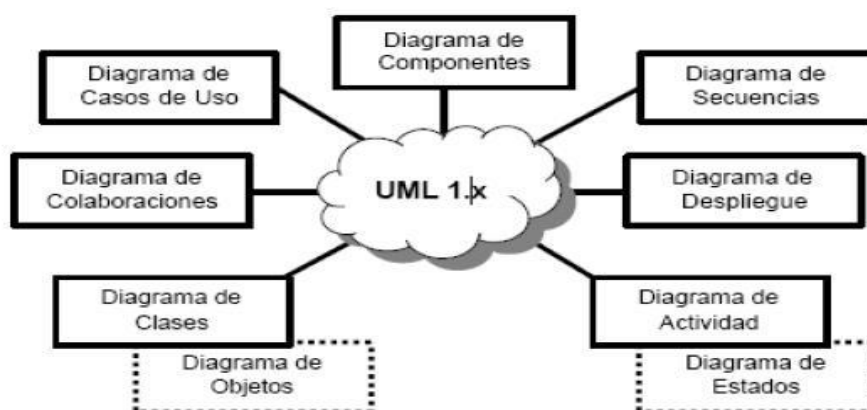
Acrónimo de Unified Modeling Language (Lenguaje Unificado de Modelado) es el lenguaje de modelado más conocido y utilizado en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. Las propiedades que han hecho de UML un estándar son: la concurrencia, es un lenguaje distribuido y adecuado a las necesidades de conectividad actual y futura. Reemplaza a decenas de notaciones empleadas con otros lenguajes. Modela estructuras complejas. Las estructuras que soporta tienen sus fundamentos en las tecnologías orientadas a objetos, tales como clases, componentes y nodos. Representa el comportamiento del sistema a través de casos de uso, diagramas de secuencia y de colaboración.



Uno de los objetivos de este modelado visual es que sea independiente del lenguaje de implementación, de tal forma que los diseños realizados se puedan implementar en cualquier lenguaje que soporte las posibilidades de UML (principalmente lenguajes orientados a objetos). Con este método formal de modelado se pueden automatizar determinados procesos y permite generar código a partir de los modelos y a la inversa.

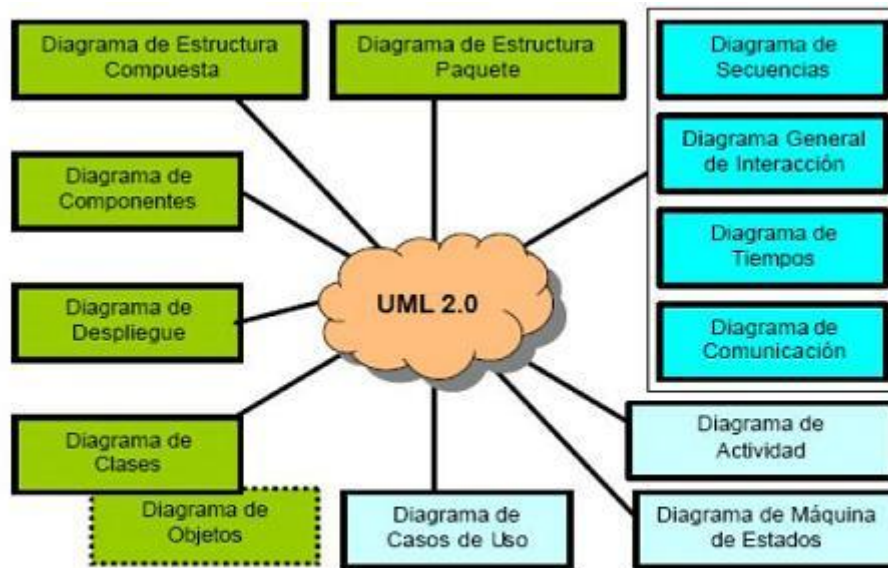
**UML 2.0** es la mayor revisión que se le ha hecho a UML desde la versión 1.0. El modelo conceptual ha sido reestructurado completamente y nuevos diagramas han sido incorporados. Los diagramas tradicionales también han sido mejorados. La nueva versión permitirá a los fabricantes de herramientas CASE, proporcionar a los analistas, arquitectos y desarrolladores, herramientas cada vez más potentes, que les permitan aprovechar mejor los modelos y como consecuencia, generar una mayor cantidad de código, reduciendo significativamente el ciclo de desarrollo de sus aplicaciones. [7]

En UML 2.0 se definen una serie de diagramas adicionales a los establecidos en su versión anterior. Compárese las figuras que se muestran a continuación:



**Figura 4. Diagramas de UML 1.0**

UML 1.x está constituido por siete diagramas básicos y dos diagramas que constituyen variaciones de dos de los anteriores.



**Figura 5. Diagramas de UML 2.0**

En el UML 2.0 se definen una serie de diagramas adicionales a los establecidos UML 1.x. El conjunto de diagramas se encuentra organizado en torno a dos categorías: diagramas estructurales (representados en verde) y diagramas dinámicos o de comportamiento (representados en celeste).

La versión que usa este trabajo es la 2.0 debido a las ventajas que ofrece con respecto a su versión anterior 1.0.

### 1.2.3. Enterprise Architect (EA)

Enterprise Architect, por sus siglas en inglés, es una herramienta flexible, completa y potente de modelado en UML bajo plataforma Windows. Provee lo más nuevo en desarrollo de sistemas, administración de proyectos y análisis de negocio. EA es una herramienta que abarca integralmente el ciclo de vida, cubre el desarrollo de software desde el levantamiento de los requerimientos, a través de las etapas de análisis, modelos de diseño, testing y finalmente el mantenimiento y re-uso. Es utilizada para el desarrollo de varios tipos de software para un amplio rango de industrias, incluyen: bancos, desarrollo web, ingeniería, finanzas, medicina, investigación, educación, transporte, ventas, energía, ingeniería electrónica y muchas más. También es utilizado con efectividad para el



entrenamiento en UML y arquitecturas de negocio en empresas de entrenamiento y universidades alrededor del mundo. [8]

Enterprise Architect se construye en base, al excepcional éxito de las versiones previas con un completo soporte para el estándar UML 2.0 como lo define el Grupo de Gestión de Objetos (OMG por sus siglas en inglés Object Management Group). Con EA, los modeladores tienen todo el poder y la expresividad. EA ayuda a administrar la complejidad con herramientas para rastrear las dependencias, soporte para modelos muy grandes, control de versiones, líneas base por cada punto del tiempo, la utilidad de comparar, para seguir los cambios del modelo, interfaz intuitiva y de alto rendimiento con vista de proyecto como un "explorador".

Por las potencialidades ofrecidas y además por su alto rendimiento combinado a la última especificación de UML 2.0 se escoge la herramienta Enterprise Architect como herramienta de modelado para llevar la documentación del software.

### **1.3. Sistema Gestor de Base de Datos**

El software de base de datos ha experimentado un auge extraordinario a raíz de la progresiva informatización de casi la totalidad de las empresas de hoy día; rapidez, efectividad en los procesos y los grandes flujos de información son las necesidades más apremiantes a la hora de optimizar servicios y productos. Ante esta notable demanda de soluciones informáticas ha surgido un gran cúmulo de gestores de base de datos.

El propósito general de los sistemas de gestión de base de datos es el de manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante, para un buen manejo de datos. Un sistema gestor de base de datos se define como el conjunto de programas que administran y gestionan la información contenida en una base de datos.

Un Sistema Gestor de Base de Datos (SGBD) puede definirse como un paquete generalizado de software, que se ejecuta en un sistema computacional anfitrión, centralizando los accesos a los datos y actuando de interfaz entre los datos físicos y el usuario. Las principales funciones que debe cumplir un SGBD se relacionan con la creación y mantenimiento de la base de datos, el control de accesos, la manipulación de datos de acuerdo con las necesidades del usuario, el cumplimiento de las normas de tratamiento de datos, evitar redundancias e inconsistencias y mantener la integridad.



Existen distintos objetivos que deben cumplir los SGBD:

Abstracción de la información, independencia del tratamiento de información, redundancia mínima, consistencia, seguridad, integridad, respaldo y recuperación de la información, control de la concurrencia y tiempo de respuesta mínimo.

Las principales funciones que debe realizar un SGBD son:

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

Entre los SGBD más utilizados actualmente se encuentran: SQLServer, MySQL, PostgreSQL y Oracle.

### 1.3.1. PostgreSQL

PostgreSQL (llamado también Postgres95) fue derivado del proyecto Postgres. A sus espaldas, este proyecto lleva más de una década de desarrollo, siendo hoy en día, el sistema libre más avanzado con diferencia, soportando la gran mayoría de las transacciones SQL, control concurrente, teniendo a su disposición varios "language bindings" como por ejemplo C, C++, Java, Python, PHP y muchos más.

El Sistema Gestor de bases de datos que almacena los datos de la aplicación es PostgreSQL v8.2.3, PostgreSQL es un sistema de base de datos relacional perteneciente al ámbito del software libre que destaca por su robustez, escalabilidad y cumplimiento de los estándares SQL. Cuenta con versiones para una amplia gama de sistemas operativos, entre ellos: Linux, Windows, Mac OS X, Solaris, BSD, Tru64 y otros más. PostgreSQL soporta ACID, o lo que es lo mismo, la realización de transacciones seguras; también, vistas, uniones, claves extranjeras, procedimientos almacenados, triggers, entre otros. Incluye la mayor parte de los tipos de datos especificados en los estándares SQL92 y SQL99, como: entero, numérico, booleano, char, varchar, fecha, interval o timestamp.





Otras características interesantes de PostgreSQL son las siguientes:

- Alta concurrencia, que evita tener que bloquear una tabla cuando se está escribiendo en ella.
- Copias de seguridad en línea.
- Replicación asíncrona.
- Transacciones anidadas.
- Optimizador de consultas.

Si de cifras se trata, es importante saber que en PostgreSQL el tamaño máximo de la base de datos es ilimitado; el de una tabla asciende a 32 TB, el de una fila a 1.6 TB y el de un campo de datos a 1 GB; el número de filas en una tabla es ilimitado, pero no el de columnas, que oscila entre 250 y 1600 columnas por tabla.

#### **1.4. Lenguaje de programación y entorno de desarrollo**

La programación Web, parte de las siglas WWW, que significan World Wide Web o red mundial. Para realizar una página con la programación Web, se deben tener claros, uno de los conceptos fundamentales el cual es, el URL (Uniform Resource Locators), sistema con el cual se localiza un recurso dentro de la red, este recurso puede ser una página web, un servicio u otro medio accesible.

##### **1.4.1. Python**

Python es un lenguaje de scripting independiente de plataforma y orientado a objetos, preparado para realizar cualquier tipo de programa, desde aplicaciones Windows a servidores de red o incluso, páginas web. Es un lenguaje interpretado, lo que significa que no se necesita compilar el código fuente para poder ejecutarlo, lo que ofrece ventajas como la rapidez de desarrollo e inconvenientes como una menor velocidad. En los últimos años el lenguaje se ha hecho muy popular, gracias a varias razones como:

- La cantidad de librerías que contiene, tipos de datos y funciones incorporadas en el propio lenguaje, que ayudan a realizar muchas tareas habituales sin necesidad de tener que programarlas desde cero.
- La sencillez y velocidad con la que se crean los programas. Un programa en Python puede tener de 3 a 5 líneas de código menos que su equivalente en Java o C.



- La cantidad de plataformas en las que podemos desarrollar, como Unix, Windows, OS/2, Mac y otros.
- Además, es gratuito, incluso para propósitos empresariales. [9]

### **Características:**

- Lenguaje interpretado o de script: Tiene muchas de las características de los lenguajes compilados, por lo que se podría decir que es semi-interpretado.
- Tipado dinámico: no es necesario declarar el tipo de dato que va a contener una determinada variable, sino que su tipo se determinará en tiempo de ejecución según el tipo de valor que se le asigne, y el tipo de esta variable puede cambiar si se le asigna un valor de otro tipo.
- Fuertemente tipado: No se permite tratar a una variable como si fuera de un tipo distinto al que tiene, es necesario convertir de forma explícita dicha variable al nuevo tipo previamente.
- Multiplataforma: El intérprete de Python está disponible en multitud de plataformas (UNIX, Solaris, Linux, DOS, Windows, OS/2, Mac OS.)
- Orientado a objetos: permite la programación imperativa, programación funcional y programación orientada a aspectos.

Aspecto que disminuiría aún más el tiempo de respuesta de los desarrolladores para cada una de las soluciones personalizadas de la distribución GNU/Linux Nova.

Este lenguaje presenta una serie de ventajas que lo hacen muy atractivo, tanto para su uso profesional como para el aprendizaje de la programación, entre las que se destacan:

- Es un lenguaje muy “expresivo”, es decir, los programas son muy compactos, un programa en Python suele ser bastante más corto que su equivalente en lenguajes como C, por muchos es considerado un lenguaje de programación de muy alto nivel.
- Es muy legible, la sintaxis es muy elegante y permite la escritura de programas cuya lectura resulta fácil, en comparación con otros lenguajes.
- Puede usarse como lenguaje imperativo procedimental o como lenguaje orientado a objeto.
- Una ventaja fundamental es la gratuidad de su intérprete.



Por todas estas características y ser un lenguaje de fácil aprendizaje, se convierte en el ideal para la implementación de la aplicación que se quiere desarrollar en el marco de la Facultad, pues se dispone de muy poco tiempo para la implementación de la misma. Python facilita llevar a cabo este proceso sin perder el foco en el desarrollo. Se utilizará la versión 2.6 pues es la versión más estable y compatible para trabajar con Django.

## 1.4.2. Framework Django

Un framework, en el desarrollo del software es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto. Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio. Provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio.

Django es un framework (marco de trabajo) de alto nivel basado en Python, facilita el desarrollo de Aplicaciones web dinámicas, te abstrae de los problemas comunes del desarrollo web y acelera las tareas más frecuentes en la programación. Proporciona un método de mapear las urls, básicamente ejecuta un código en especial para cada url, o sea, urls que permite a los usuarios que buscan, encuentren de una manera más rápida. [10]

### Ventajas de Django

- Permite mostrar y validar formularios de manera muy simple, fácilmente puede manipular el código del formulario adaptándolo a las necesidades de la aplicación y a una respuesta simple para el usuario.
- Convierte los datos enviados por los usuarios (a través de formularios, por ejemplo) en estructuras de datos que pueden ser manipuladas fácilmente.
- A través de plantillas ayuda a separar el contenido de la presentación, evitándo tener que manipular la lógica de negocio cuando tengas que realizar cambios de apariencia en tu website.



- Se integra fácilmente con la capa de base de datos, y permite que el trabajo sea más productivo a un nivel de abstracción más alta.
- Aparte de las ventajas de Python, Django permite un desarrollo rápido.
- Django mantiene de una manera estricta un diseño limpio en su código y te permite entender más fácilmente las aplicaciones.
- Django promueve el acoplamiento débil, diferentes módulos de la aplicación deberían ser intercambiables y se comunicarse con otros módulos a través de APIs limpias y concisas.
- Django utiliza la arquitectura MVC, es el patrón que dice que el código para definir y acceder a los datos (modelo) está separado de la lógica de negocio (el controlador) que está separado de la interface de usuario (la vista).
- Django tiene un único lugar donde guardar la configuración y la capa de acceso a la BD tiene un nivel alto de abstracción para poder cambiar el servidor de BBDD (de mysql a postgresQL) de una manera rápida y sencilla.

Por la gran facilidad y el ahorro de tiempo que brinda Django cuando se utiliza asociado con el lenguaje de programación Python, es que se decide trabajar con este entorno de desarrollo en su versión 1.1.

### 1.4.3. Wing IDE

Wing IDE es un entorno integrado de desarrollo especialmente diseñado para el lenguaje de programación Python, disponible en Linux y en Windows. Permite el desarrollo rápido de aplicaciones de plataforma cruzada para escritorio, web y empresariales. Wing IDE se enfoca en incrementar la productividad y la calidad del código, especialmente en proyectos complejos con requerimientos cambiantes. Proporciona a los desarrolladores un completo editor de código fuente, depurador, explorador de código y muchas otras herramientas diseñadas específicamente para su uso con Python.

Entre las principales características del Wing IDE se encuentran:

- Auto completado de código.
- Asistente de codificación.



- Identación automática.
- Búsquedas en múltiples archivos.
- Búsquedas en todo el disco.
- Búsquedas con expresiones regulares.
- Soporte de CVS, Subversión.
- Sintaxis coloreada.
- Evaluación de archivos o selecciones de archivo.
- Gratuitos para desarrollos Open Source.

Estas características son las que definen el uso de Wing IDE como entorno de desarrollo en la implementación del software en su versión 3.2.

En el presente capítulo se identificaron un grupo de resultados de otros sistemas como son Redmine, Groobix, Quartz, Sistema de Gestión de la Información de un departamento docente en la Universidad de Matanzas y el Sistema de Gestión de Información de la Facultad 8” (SGIF), que persiguen el mismo objetivo de informatización, cada uno de ellos con características comunes pero con marcadas diferencias con la universidad debido al complejo nivel organizativo que presenta cada facultad y con funcionalidades que no entran dentro del alcance del presente trabajo.

Por lo que sólo se utilizarán estos sistemas como ejemplo o guía para el desarrollo de la solución propuesta. Además se asimilaron las principales tendencias, tecnologías y herramientas definidas por la Facultad tomando como Metodología de desarrollo se definió Rational Unified Process (RUP). Como lenguaje de Modelado Lenguaje de Modelado Unificado (UML 2.0). Como herramienta de modelado Enterprise Architect (EA). Como gestor de bases de datos se tiene PostgreSQL. Se usa Python como lenguaje de programación, Django como marco de desarrollo web y WingIDE como entorno de desarrollo. Además se implementa el patrón de arquitectura Model View Template.



## **Introducción**

Los temas que se abordarán a continuación hacen referencia al objeto de estudio dada la descripción del mecanismo del negocio: planificación y seguimiento de actividades. Con la realización del modelo del dominio, la especificación de los requisitos y definición de los casos de uso, se forman los cimientos de la aplicación web para la planificación y seguimiento de las actividades de los profesores en la Facultad 7.

Como paso inicial para el desarrollo del software se realizó un estudio detallado de los procesos que se llevan a cabo en la planificación y seguimiento de las actividades de los profesores la Facultad 7. Los mismos ayudarán a obtener los requerimientos necesarios, posibilitando identificar los casos concretos que debe tener el sistema, la relación que debe existir entre la ingeniería de software y el negocio, con el fin de aclarar el enfoque que quiere tener el cliente con el software, más un conjunto de artefactos para iniciar la modelación de dicha propuesta de solución al problema presentado.

### **2.1. Objeto de Estudio**

#### **2.1.1. Descripción de los conceptos asociados al dominio del problema.**

Cada departamento tiene un Jefe de Departamento, quien se encarga de controlar y planificar las actividades docentes que debe realizar cada profesor durante un curso académico. Según el cumplimiento que tengan dichas actividades se obtiene una evaluación al final del curso, que influye mucho en su categoría docente y en su preparación profesoral. Para comprender perfectamente los procesos del negocio que intervienen dentro del campo de acción es fundamental analizar cómo se ejecutan estos procesos actualmente en la Universidad.

#### **2.1.2. Análisis crítico de la ejecución de los procesos**

El proceso de planificación y seguimiento de actividades que se realiza en la actualidad se ve afectado por un grupo de aspectos que entorpecen la obtención de buenos resultados. Todo el trabajo que se realiza con la planificación se hace de forma manual, lo cual ocasiona que ocurran errores en la



realización de la misma, demora en su entrega y aprobación por parte del personal al cual se subordina cada profesor, así como que ocurran coincidencias en la planificación de varias actividades por no llevar el control de las mismas.

### 2.2. Información que se maneja.

Dentro de la información que se maneja se encuentran:

- **Tarea:** Actividades asignadas a cada uno de los profesores o estructuras docentes residentes en la Facultad.
- **Planificación de actividades del profesor:** Plantilla que permite el control de todas las actividades que un profesor debe cumplir en todo un curso académico.
- **Controlar actividades de superación del profesor:** Según la planificación que se realizó, se controla las actividades de superación del profesor.
- **Controlar actividades de producción del profesor:** Según la planificación que se realizó, se controla las actividades de producción del profesor si se encuentra vinculado a alguna actividad productiva.
- **Controlar actividades de formación del profesor:** Según la planificación que se realizó, se controla las actividades de formación del profesor.
- **Controlar actividades científicas del profesor:** Según la planificación que se realizó en, se controla las actividades científicas del profesor.

### 2.3. Objeto de automatización.

Los objetos de automatización de la aplicación son la información de las tareas, la gestión de las mismas, así como la realización de la planificación de las estas para cada profesor o estructura de trabajo, para ser mostrado a cada uno y enviado por correo. La asignación de las tareas se realiza por separado ya que estas son asignadas por cada directivo a cada uno de sus profesores, por lo contrario la gestión de las tareas puede realizarlo el mismo profesor según su nivel de acceso al sistema.

### 2.4. Modelo de Dominio.

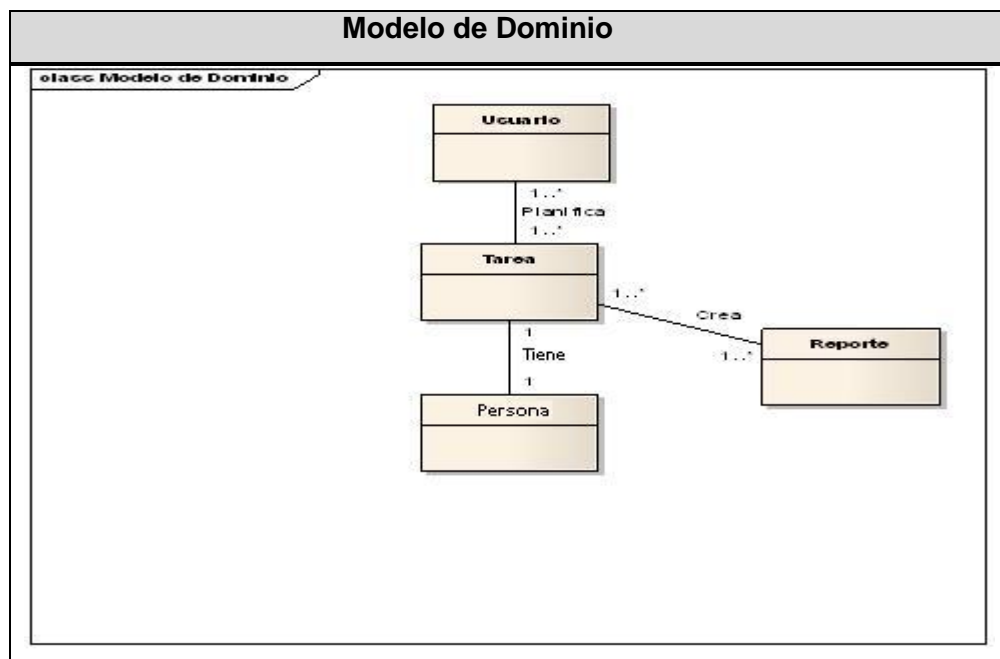
El Modelo de Dominio o Modelo Conceptual es una representación visual de los conceptos u objetos del mundo real significativos para un problema o área de interés. Representa clases conceptuales del



dominio del problema. Una clase conceptual puede ser una idea o un objeto físico (símbolo, definición y extensión). El modelo de dominio se representa en UML con un diagrama de clases en los que se muestra:

- Conceptos u objetos del dominio del problema: clases conceptuales.
- Asociaciones entre las clases conceptuales.
- Atributos de la clase conceptuales.

El modelo del dominio que se representa, describe cómo se desarrolla la planificación y control de las actividades de los profesores docente en la Facultad 7. Este mecanismo no es más, que la Técnica de funcionamiento que es ejercida por cada profesor a la hora de realizar la planificación de una tarea. Tiene como finalidad, lograr una buena programación de las actividades de dichos profesores.



**Figura 6. Modelo de dominio**

- Tarea: Actividad a realizar por una persona para darle cumplimiento a los objetivos de su área.
- Usuario: Persona que trabaja en un área determinada, encargada de planificar sus propias actividades extra docentes y las de los que están a su cargo.





- Aviso: Mensaje de notificación que es enviado al profesor o responsable de área que se le ha planificado una tarea mediante un correo electrónico.
- Reportes: Informes que son generados al ser asignada una tarea.

### **2.5. Reglas del Negocio.**

Durante el desarrollo del negocio se determinó establecer un conjunto de reglas, estas deben tomarse en cuenta al llevar a cabo la informatización de los procesos del negocio, esto permite que el sistema elaborado se ajuste y garantice el cumplimiento de las necesidades del cliente.

- El Jefe de Departamento verifica la situación del profesor y sobre la base de esto deberá planificar las actividades necesarias en cada caso.
- Dentro del colectivo de profesores se pueden identificar dos grupos: profesores en adiestramiento y profesores con categoría docente, en dependencia de esto se realizará la planificación de actividades.
- Para los profesores vinculados a proyectos productivos, se deben reflejar las actividades de esta índole, según la planificación realizada por su Jefe de Proyecto.
- Los Jefes de proyectos realizan la planificación de tareas para sus profesores, en función con los roles que desempeñan, y posteriormente envían notificación al Jefe de Departamento.
- El planificador de una tarea solo puede ser el que modifique la prioridad de la misma.

### **2.6. Especificación de los Requisitos del Software**

Con la identificación de los requisitos de la aplicación se van igualando las funcionalidades, restricciones y condiciones que la aplicación debe tener. De esta forma se llega a un mejor entendimiento entre el usuario y los desarrolladores del sistema, sin dejar de mencionar que una buena elección de los requerimientos hace posible mantener el orden y un seguimiento detallado de la aplicación. Estos pueden ser:

#### **2.6.1. Requisitos funcionales**

Los Requerimientos Funcionales son capacidades o condiciones con las que debe cumplir el producto a elaborar, no alteran la funcionalidad del software, se mantienen invariables sin importar con que cualidades o propiedades se relacionen. [11]



A partir del estudio y las investigaciones que se realizaron de los procesos, se obtuvieron una serie de requerimientos funcionales:

1. Gestionar la planificación de las actividades del profesor.
  - 1.1 Insertar planificación de actividad.
  - 1.2 Modificar planificación de actividad.
  - 1.3 Eliminar planificación de actividad.
  - 1.4 Buscar actividades
2. Controlar actividades planificadas del profesor.
  - 2.1 Asignar evaluación del cumplimiento de una actividad.
3. Generar Reportes.
  - 3.1 Reporte de actividades por prioridad.
  - 3.2 Reporte de actividades por fecha de cumplimiento.
  - 3.3 Reporte de actividades general

### **2.6.2. Requisitos no funcionales**

Los requerimientos no funcionales son capacidades o cualidades que el producto debe tener, o sea, características que hagan al producto atractivo, rápido, usable o confiable. Están estrechamente vinculados a los requisitos funcionales, puesto que una vez que está definido lo que el sistema debe hacer, es necesario especificar como ha de hacerlo. Pueden llegar a marcar la diferencia entre un producto bien aceptado por los clientes y usuarios o uno de poca o ninguna calidad y aceptación. [12]

1. Requerimientos de Software:
  - Del lado del cliente se necesita de un navegador Web principalmente Internet Explorer, Mozilla Firefox y NetsCape Navegador, aunque se puede utilizar otros.
  - Del lado del servidor se necesita el Apache como servidor Web, como Sistema Gestor de Base de Datos (SGBD) se utilizará PostgreSQL y Python como lenguaje de programación.
2. Requerimientos de Apariencia o Interfaz Externa
  - Diseño orientado a llamar la atención del usuario y con una navegación sencilla.
  - El sistema debe tener una interfaz sencilla, debe ser fácil de usar.
  - La navegación dentro del sitio debe ser global para los módulos principales.
3. Requerimientos de Seguridad



- Identificar al usuario antes de que pueda realizar cualquier acción sobre el contenido del sistema.
- Garantizar que la información sea editada únicamente por quien tiene derecho a editarla.
- Garantizar que las funcionalidades del sistema se muestren de acuerdo al nivel de acceso del usuario activo.
- Protección contra acciones no autorizadas o que puedan afectar la integridad de los datos.
- Verificación sobre acciones irreversibles (eliminaciones).

#### 4. Requerimientos de Usabilidad.

- El sistema podrá ser usado por cualquier persona que posea conocimientos básicos en el manejo de la computadora y de un ambiente Web en sentido general.

#### 5. Requerimientos de Soporte.

- El sistema deberá ser fácil de instalar.
- Se permitirá realizar modificaciones posteriores para adaptar mejoras al sistema o en caso que cambien las necesidades de los clientes.

#### 6. Requerimientos de Disponibilidad.

- El sistema deberá estar disponible las 24 horas del día para trabajar en el momento deseado, aunque puede ser a consideración de los clientes.

#### 7. Requerimientos de Confiabilidad.

- La herramienta de implementación a utilizar tiene soporte para recuperación ante fallos y errores.
- Se deben establecer distintos roles que limiten el nivel de accesibilidad de los usuarios.

#### 8. Requerimientos de Portabilidad.

- Necesidad de que el sistema sea multiplataforma.

#### 9. Requerimientos de Rendimiento.

- Tiempos de respuesta no mayor de 2 segundos, al igual que la velocidad de procesamiento de la información.
- Estará implementado con tecnología Web para facilitar el uso a través de la red.

#### 10. Requerimientos de Funcionalidad.

- El sistema debe estar accesible desde la intranet de la facultad.



- Mínima cantidad de páginas para ejecutar todas las funciones posibles (preferiblemente que estén relacionadas).

## 11. Requerimientos de Integridad.

- Los datos serán transmitidos por la red a través de un producto seguro.

## 12. Requerimientos de Implementación.

- Se usará como entorno de desarrollo integrado (IDE) Django, lenguaje de programación Python y como gestor de bases de datos PostgreSQL.

## 2.7. Modelo de Casos de Uso

### 2.7.1. Actores del Sistema

Un actor del sistema es una entidad externa, representando un rol de una o varias personas, un equipo o un sistema automatizado que interactúa con el sistema, por lo que no forma parte del sistema. Puede intercambiar información o ser un recipiente pasivo de información.

Actor	Función
Usuario	Persona que trabaja en un área determinada, encargada de planificar sus propias actividades docentes y extra docentes y darle seguimiento a estas.
Directivo	Persona responsable de un área determinada encargada de planificar las actividades de dicha área y darle seguimiento a las mismas.

Tabla 1. Actores del Sistema

### 2.7.2. Diagrama de Caso de Uso del Sistema

Los diagramas de casos de usos explican gráficamente un conjunto de casos de usos del sistema. Tiene como objeto mostrar una clase de diagrama contextual que permite conocer rápidamente los actores externos de un sistema y las formas básicas en que lo utilizan.

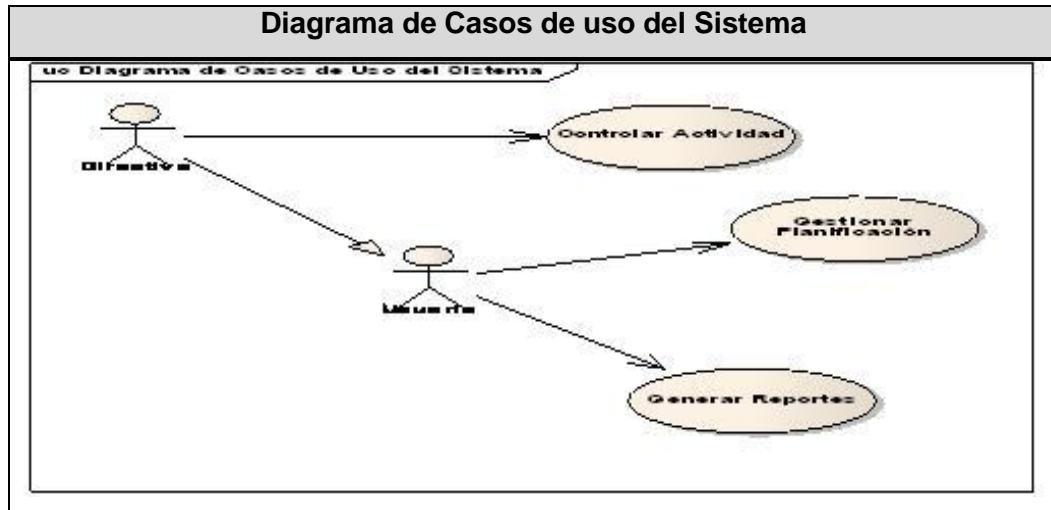


Figura 7. Diagrama de Casos de Uso del Sistema

### 2.7.3. Especificación de Casos de Uso del Sistema

A continuación se describen las especificaciones de los casos de uso que forman parte del sistema para que se comprenda mejor sobre utilización y definición.

CU Gestionar planificación de las actividades del profesor.

<b>Caso de Uso</b>	CU Gestionar planificación de las actividades del profesor.
<b>Usuario</b>	Usuario(profesor)
<b>Descripción</b>	El caso de uso inicia cuando un usuario (Usuario o Directivo) decide realizar acciones como buscar, asignar, modificar y eliminar una actividad.
<b>Referencia</b>	RF 1

Tabla 2.CU Gestionar planificación de las actividades del profesor.

CU Controlar actividades planificadas del profesor.

<b>Caso de Uso</b>	CU Controlar actividades planificadas del profesor.
<b>Usuario</b>	Directivo



<b>Descripción</b>	El caso de uso se inicia cuando un profesor desea controlar sus actividades personales planificadas o cuando un directivo desea controlar las actividades asignadas a un subordinado.
<b>Referencia</b>	RF 2

Tabla 3. CU Controlar actividades planificadas del profesor.

CU Generar Reportes.

<b>Caso de Uso</b>	CU Generar Reportes.
<b>Usuario</b>	Usuario(Profesor)
<b>Descripción</b>	El caso de uso se inicia cuando un usuario desea generar un determinado reporte.
<b>Referencia</b>	RF 3

Tabla 4. CU Generar Reportes

#### 2.7.4. Casos de Uso expandidos

Para conocer en detalle lo que trae asociado cada caso de uso de muestra la descripción general de cada uno de estos.

<b>CASO DE USO:</b>	Gestionar planificación de las actividades del profesor.
<b>Resumen:</b>	El caso de uso inicia cuando un Profesor o directivo decide realizar acciones como buscar, registrar, modificar o eliminar una actividad. El caso de uso concluye cuando se termina la gestión de la actividad.
<b>Actores:</b>	Usuario
<b>Precondiciones:</b>	
<b>REFERENCIAS</b>	
<b>Referencias:</b>	RF 1
<b>FLUJO NORMAL DE EVENTOS</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El caso de uso inicia cuando el	



<p>actor accede a la opción "Gestionar Tarea".</p>	
<p>2. Decide realizar una de las siguientes acciones:</p> <ul style="list-style-type: none"> <li>• Agregar tarea. (Ver sección Agregar tarea.)</li> <li>• Modificar tarea. (Ver sección Modificar tarea.)</li> <li>• Eliminar tarea. (Ver sección Eliminar tarea.)</li> <li>• Buscar tarea. (Ver sección Buscar tarea.)</li> </ul>	
<b>Flujo Alternativo 1</b>	
<b>Sección: "Agregar tarea"</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<p>1. El caso de uso inicia cuando el actor accede a la opción Crear tarea.</p>	<p>2. Muestra un formulario con los campos para el registro de los datos necesarios para crear una tarea:</p> <ul style="list-style-type: none"> <li>• Título.</li> <li>• Descripción.</li> <li>• Fecha_Cumplimiento.</li> <li>• Responsable.</li> <li>• Prioridad.</li> <li>• Porcentaje_cumplimiento</li> <li>• Tipo.</li> <li>• Control.</li> <li>• Fecha Planificación.</li> <li>• Responsables</li> <li>• Participantes</li> </ul>



<p>3. Inserta los datos especificados.</p> <p>4. Decide realizar una de las siguientes acciones:</p> <ul style="list-style-type: none"> <li>▪ Asignar tarea en el sistema. (Botón “Asignar”)</li> <li>▪ Cancelar el registro. (Botón “Cancelar”)</li> </ul>	<p>5. Si el usuario selecciona el botón “Asignar”:</p> <p>5.1 Verifica que los datos insertados están completos y son correctos.</p> <ul style="list-style-type: none"> <li>▪ Detecta que existen campos vacíos que son obligatorios de llenar. (Ver Flujo Alternativo 1)</li> <li>▪ Detecta errores en los datos registrados. (Ver Flujo Alternativo 2)</li> <li>▪ Detecta que ya existe esa tarea. (Ver Flujo Alternativo 3)</li> <li>▪ Se insertan los datos del evento y se muestra el mensaje: <i>“Se ha asignado la tarea exitosamente.”</i></li> </ul> <p>5.2. Si el usuario selecciona el botón “Cancelar” ver Flujo Alternativo 4.</p>
<b>Flujo Alternativo 1</b>	
	<p>6. Muestra el mensaje: <i>“Existen campos vacíos que son obligatorios.”</i></p> <p>Permite volver a registrar los datos.</p>
<b>Flujo Alternativo 2</b>	
	<p>7. Muestra el mensaje: <i>“Existen datos incorrectos.”</i></p> <p>Permite volver a registrar los datos.</p>
<b>Flujo Alternativo 3</b>	
	<p>8. Muestra el mensaje: <i>“Esta tarea ya existe en el sistema.”</i></p> <p>Permite volver a insertar un nuevo evento.</p>
<b>Flujo Alternativo 4</b>	
	<p>9. La tarea no es registrada.</p> <p>Regresa a la vista anterior.</p>
<b>Sección: “Modificar tarea”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>





<p>1. El actor accede a la opción "Controlar Tarea"</p>	<p>2. El sistema muestra un formulario con la opción de "Buscar Tarea" (Ver sección "Buscar Tarea") y a continuación un listado con todas las tareas que el usuario ha creado o asignado.</p>
<p>3. El actor selecciona el icono de modificación de la tarea.</p>	<p>4. Muestra los datos de la tarea seleccionada, permitiendo su modificación:</p> <ul style="list-style-type: none"> <li>• Título.</li> <li>• Descripción.</li> <li>• Fecha_Cumplimiento.</li> <li>• Responsable.</li> <li>• Prioridad.</li> <li>• Porciento_cumplimiento</li> <li>• Tipo.</li> <li>• Control.</li> <li>• Fecha Planificación.</li> <li>• Responsables</li> <li>• Participantes</li> </ul>
<p>5. Modifica los datos existentes.</p> <p>6. Decide realizar una de las siguientes acciones:                  Modificar los datos. (Botón "Aceptar")                  Cancelar la modificación. (Botón "Cancelar")</p>	<p>7. Si el usuario selecciona el botón "Modificar":</p> <p>7.1 Verifica que los datos insertados están completos y son correctos.                  Detecta que existen campos vacíos que son obligatorios de llenar. (Ver Flujo Alternativo 1)                  Detecta errores en los datos registrados. (Ver Flujo Alternativo 2)                  Detecta que ya existe esa tarea. (Ver Flujo Alternativo 3)                  Se modifican los datos de la tarea y se muestra el mensaje: <i>"Se ha modificado la tarea exitosamente."</i></p> <p>7.2 Si el usuario selecciona el botón</p>



	“Cancelar” ver Flujo Alterno 4.
<b>Flujo Alterno 1</b>	
	8. Muestra el mensaje: <i>“Existen campos vacíos que son obligatorios.”</i> . Permite volver a modificar los datos.
<b>Flujo Alterno 2</b>	
	9. Muestra el mensaje: <i>“Existen datos incorrectos.”</i> . Permite volver a modificar los datos.
<b>Flujo Alterno 3</b>	
	10. Muestra el mensaje: <i>“La tarea ya existe en el sistema.”</i> . Permite modificar nuevamente la tarea.
<b>Flujo Alterno4</b>	
	11. La tarea no es modificada. Regresa a la vista anterior.
<b>Sección: “Eliminar Tarea”</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El actor accede a la opción “Todas mis Tareas”	2. El sistema muestra un formulario con la opción de “Buscar Tarea” (Ver sección “Buscar Tarea”) y a continuación un listado con todas las tareas que el usuario ha creado o asignado.
3. El actor selecciona el icono para eliminar la tarea.	4. Se muestra el mensaje: <i>“¿Está seguro que desea realizar esta acción?”</i> . 4.1 Si el usuario selecciona el botón “Aceptar” del mensaje de confirmación, se elimina la tarea y se muestra el mensaje: <i>“La tarea ha sido eliminada exitosamente.”</i> . 4.2 Si el usuario selecciona el botón “Cancelar” del mensaje de confirmación ver Flujo Alterno 1.



Flujo Alternativo 1	
	5. La tarea no es eliminada del sistema.
Sección: "Buscar tarea"	
	6. Se muestra el listado de las tareas que cumplen con el criterio de selección y brinda por cada tarea las opciones: <ul style="list-style-type: none"> <li>▪ Modificar una tarea. (Icono "Modificar").</li> <li>▪ Eliminar una tarea ya existente. (Icono "Eliminar").</li> </ul>
7. Decide realizar una de las siguientes acciones: <ul style="list-style-type: none"> <li>▪ Modificar una tarea ya existente. (Ver Sección "Modificar tarea")</li> <li>Eliminar una tarea ya existente. (Ver Sección "Eliminar tarea")</li> </ul>	

Tabla 5. CU Expandido "Gestionar Planificación."

<b>CASO DE USO:</b>	Controlar actividades
<b>Resumen:</b>	El caso de uso inicia cuando el actor desea realizar un control de las actividades. El caso de uso concluye cuando se termina el control de dichas actividades.
<b>Actores:</b>	Directivo
<b>Precondiciones:</b>	El usuario debe estar autenticado, y tener tareas que haya asignado para controlar.
REFERENCIAS	
<b>Referencias:</b>	RF 2
FLUJO NORMAL DE EVENTOS	
Acción del Actor	Respuesta del Sistema
1. El caso de uso inicia cuando el actor	2. Muestra un formulario con la opción "Buscar", que



<p>accede a la opción "Controlar Tarea".</p>	<p>permite realizar las acciones:</p> <ul style="list-style-type: none"> <li>▪ Buscar por criterios de búsqueda.                             <ul style="list-style-type: none"> <li>• Título</li> <li>• Fecha_Cumplimiento</li> <li>• Prioridad</li> </ul> </li> </ul> <p>Y a continuación un listado de todas las tareas que ha asignado que cumplan con el criterio especificado o con varios.</p>
<p>3. Decide realizar una de las siguientes acciones:</p> <ul style="list-style-type: none"> <li>▪ Buscar una tarea. (Ver Sección "Buscar Tarea")</li> <li>▪ Seleccionar una de las tareas que se encuentra en el listado.</li> <li>▪ Cancelar. (Ver Flujo Alterno 1)</li> </ul> <p>4. El usuario accede a una de las tareas que se encuentra en el listado y selecciona la opción control.</p>	<p>5. Muestra un formulario con las opciones:</p> <ul style="list-style-type: none"> <li>• Bien.</li> <li>• Regular.</li> <li>• Mal</li> </ul> <p>Además de un cuadro de texto para escribir cualquier observación realizada a la misma.</p>
<p>6. Marca cualquiera de estas opciones indicando el control realizado a la tarea.</p>	<p>7. Se modifican los datos de la tarea y se muestra el mensaje: <i>"La tarea ha sido controlada satisfactoriamente."</i></p>
<p><b>Flujo Alterno 1</b></p>	
	<p>8. Cancela la acción y regresa a la página principal, concluyendo así el caso de uso.</p>
<p style="text-align: center;"><b>Sección: "Buscar tarea"</b></p>	
<p><b>Acción del Actor</b></p>	<p><b>Respuesta del Sistema</b></p>



	<p>1. Se muestra el listado de las tareas que cumplen con el criterio de selección y brinda por cada tarea las opciones:</p> <ul style="list-style-type: none"> <li>▪ Modificar una tarea. (Botón “Modificar”)</li> <li>▪ Eliminar una tarea ya existente. (Botón “Eliminar”)</li> <li>▪ Controlar Tarea(Botón “Controlar”))</li> </ul>
<p>2. Decide realizar la siguiente acciones:</p> <ul style="list-style-type: none"> <li>▪ Controlar una tarea ya existente.</li> </ul>	

Tabla 6. CU Expandido “Controlar Actividad.”

<b>CASO DE USO:</b>	Generar Reporte	
<b>Resumen:</b>	El caso de uso inicia cuando el actor desea visualizar un reporte según el criterio de selección. El caso de uso concluye cuando se muestran los resultados.	
<b>Actores:</b>	Usuario	
<b>Precondiciones :</b>	El usuario debe estar autenticado y haber realizado la búsqueda correspondiente.	
<b>REFERENCIAS</b>		
<b>Referencias:</b>		
<b>FLUJO NORMAL DE EVENTOS</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1. El caso de uso inicia cuando el actor accede a la opción “Reportes”.	2. Muestra un cuadro de dialogo para guardar el listado de tareas en formato pdf.	
<p>3. Selecciona según los criterios brindados la opción:</p> <ul style="list-style-type: none"> <li>▪ Guardar. (Botón “Aceptar”)</li> <li>▪ Cancelar (Botón “Cancelar”), ver Flujo Alterno 1</li> </ul>	<p>4. Si el usuario selecciona el botón “Aceptar”</p> <p>Verifica que se haya seleccionado algún criterio de búsqueda.</p> <ul style="list-style-type: none"> <li>▪ Detecta que no se seleccionó ninguna opción (Ver Flujo Alterno 2)</li> <li>▪ Se muestra un reporte en formato HTML y permite exportarlo a formato PDF (Botón “Exportar a PDF”).</li> </ul>	



Flujo Alternativo 1	
	5. Se cancela el reporte. 6. Regresa a la página principal.
Flujo Alternativo 2	
	7. Muestra un reporte general.

**Tabla 7. CU Expandido “Generar Reporte.”**

En el presente capítulo se ha realizado un análisis de cómo ocurre el flujo de la información entre las distintas estructuras que existen y los profesores de la facultad. De la información obtenida, se definieron las necesidades de funcionamiento de la aplicación a implementar entre las que se encuentra el control y seguimiento de las actividades asignadas.

A partir de los requerimientos funcionales y no funcionales se definieron los Casos de Usos del Sistema entre los que se encuentran Gestionar Planificación, Controlar Tarea y Generar Reporte, que son las funcionalidades que se van a implementar y se describió además cada uno de los Casos de Usos del Sistema.



## Introducción

En el presente capítulo se abordará el flujo de trabajo de Análisis y Diseño del sistema que propone RUP, donde a través de los artefactos que se generan, se modelan los casos de usos seleccionados para esta iteración del producto que se ha propuesto. Se profundiza en los casos de usos detallándolos de manera que permitan reflejar una vista interna del sistema descrito con el lenguaje de los desarrolladores. El análisis es toda parte física que debe representar dicho sistema mientras que el diseño es la parte del proceso de desarrollo de software cuyo propósito primario es decidir cómo el sistema se llevará a cabo.

### 3.1. Descripción de la arquitectura. Fundamentación

Para el desarrollo del Modelo de Diseño es necesario establecer una descomposición del mismo en subsistemas, con sus interfaces y las dependencias. Esta representación es de gran importancia para la arquitectura, ya que los subsistemas y sus interfaces constituyen la estructura fundamental del producto de software. Se utilizó el patrón de arquitectura Modelo-Vista-Controlador (MVC) que separa los datos de la interfaz de usuario y la lógica del negocio.

El patrón MVC es un patrón de arquitectura de software en el cual todo el proceso está dividido en tres capas, típicamente estas capas son el Modelo, la Vista y el Controlador. El Modelo incorpora la capa de dominio y persistencia, es el encargado de guardar los datos en un medio persistente (ya sea una base de datos o un archivo de texto.). Se refiere a la lógica del negocio o servicio y los datos asociados con la aplicación.

La Vista se encarga de presentar la interfaz al usuario, en sistemas Web, esto es típicamente HTML, aunque pueden existir otros tipos de vistas. En la vista, sólo se deben realizar operaciones simples. Es la encargada de la presentación de los datos.



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

Django es un framework de desarrollo web que sigue el modelo MVC, aunque los desarrolladores del proyecto prefieren llamarlo Modelo-Plantilla-Vista (*Model-Template-View* "MVT"). El controlador en este caso sería el propio *framework*. Separa la programación del diseño, con lo que se puede mantener trabajando en paralelo a programadores y diseñadores, ahorrando mucho tiempo. Esto también permite cambiar en cualquier momento el diseño del sitio, sin preocuparse porque la aplicación pueda resultar dañada.

La principal ventaja en lo referente al diseño de usar Django, es que directamente obliga a diseñar el servidor de una determinada manera forzando a cumplir con el patrón MVC. Una aplicación MVC en Django estará distribuida de la siguiente forma:

- El archivo *models.py* contiene una descripción de la tabla de la base de datos, la declaración de la base de datos se hace de la misma forma que si se declarase una clase Python. El contenido de este fichero se corresponderá con el modelo descrito en el patrón MVC. Usando esta clase se pueden crear, buscar, actualizar y borrar entradas de la base de datos usando código Python sencillo en lugar de escribir declaraciones SQL. Será el propio framework el que se encargará de realizar la asociación entre los objetos declarados y sus tablas correspondientes en la base de datos.
- El archivo *urls.py* define patrones de Urls asociados a un manejador. Cuando recibe una petición comparará la URL solicitada con el patrón definido y decide a qué manejador redirigir la petición. Se corresponderá con el controlador del patrón MVC.
- El archivo *views.py* contiene los manejadores a los que redirige la petición el fichero "urls.py". Cuando el manejador es llamado, atiende la petición y devuelve la respuesta en el formato esperado por el cliente. Se corresponde con la vista del patrón MVC.

Tomadas en su conjunto, estas piezas se aproximan al patrón de diseño MVC. Dicho de manera más fácil, MVC precisa una forma de desarrollar software en la que el código para definir y acceder a los datos (el modelo) está separado de la redirección lógica de asignación de ruta (el controlador), que a su vez está separado de la interfaz del usuario o datos devueltos (la vista).





### 3.2. Análisis.


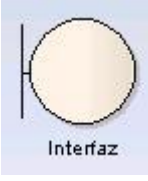
El análisis consiste en obtener una visión un poco más detallada del sistema, de modo que sólo se interesa por los requisitos funcionales, para refinarlos y estructurarlos. Esto proporciona una estructura centrada en el mantenimiento en aspectos como la flexibilidad ante los cambios y la reutilización de componentes, útil además para el comienzo de las actividades de diseño e implementación. Para describir un modelo de análisis se utiliza el lenguaje de los desarrolladores al razonar sobre los funcionamientos internos del sistema.

El objetivo del análisis, se centra en comprender perfectamente los requisitos del software y no precisar cómo se implementa el producto. En el modelo del análisis se refinan los requisitos, no se toman en cuenta el lenguaje de programación a usar en la construcción, la plataforma en la que se ejecuta la aplicación, los componentes reutilizables de otras aplicaciones, entre otras características que afectan al sistema.

#### 3.2.1. Diagramas de Clases de Análisis

El diagrama de clases del análisis se realiza para cada caso de uso del sistema y muestra las clases participantes en dicho caso de uso y sus relaciones. En los diagramas de clases se identifican tres tipos de clases: Interfaz, Controladora y Entidad (Ver Tabla 3).

A continuación sólo se muestran los diagramas de los CU más significativos para la arquitectura.

Representación	Nombre	Características
 <p>Entidad</p>	Entidad	Modelan información que posee larga vida y que es a menudo persistente.
 <p>Interfaz</p>	Interfaz	Modelan la interacción entre el sistema y sus actores.



	<p>Control</p>	<p>Coordinan la realización de uno o unos pocos casos de uso, regulando las actividades de los objetos que implementan la funcionalidad del caso de uso.</p>
--	----------------	--

Tabla 8. Representación de los estereotipos a usar en el Diagrama de Clases del Análisis.

**Diagrama de clases de análisis del caso de uso: Gestionar Planificación**

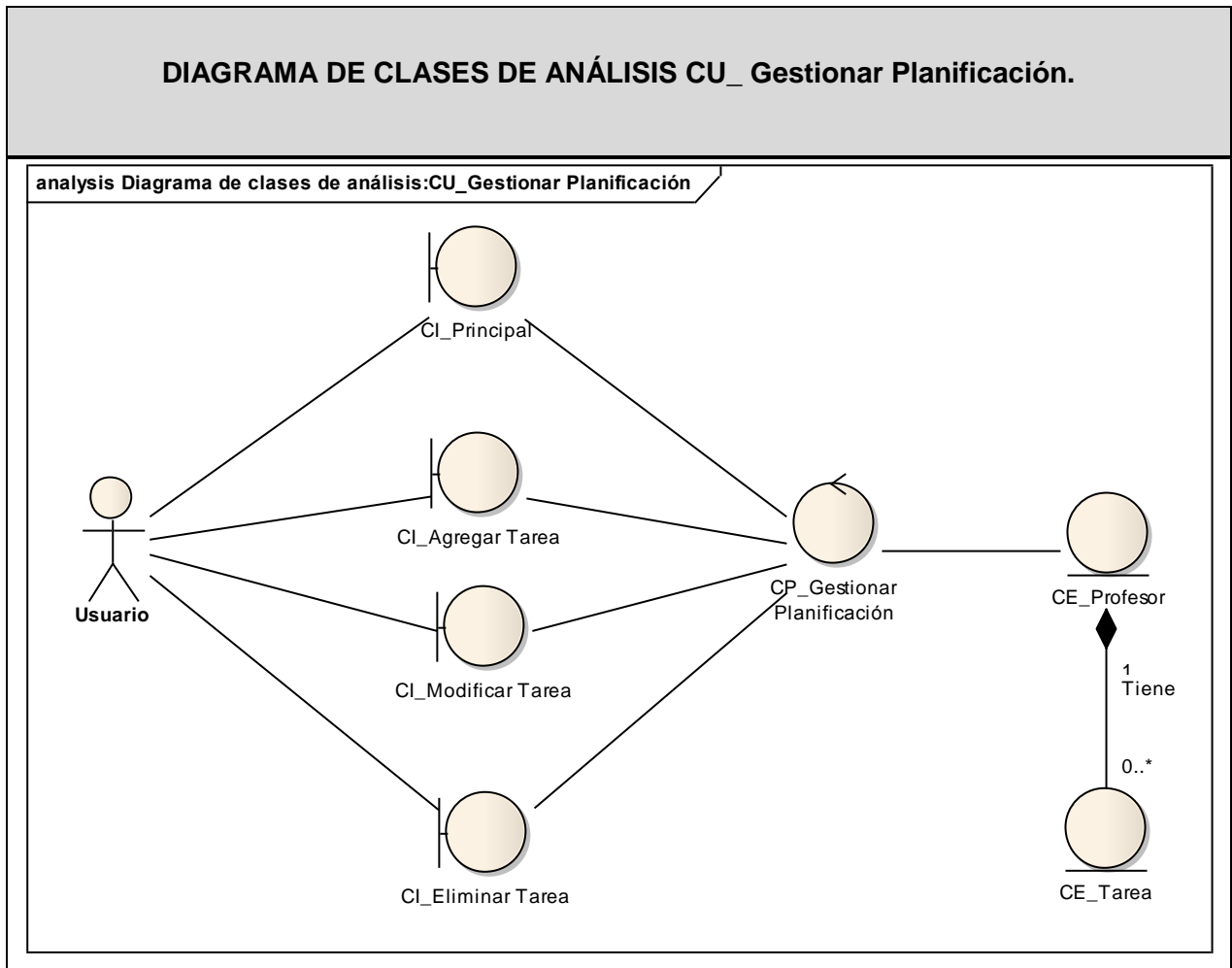


Figura 8. Diagrama de clases de análisis del caso de uso: Controlar Actividad.

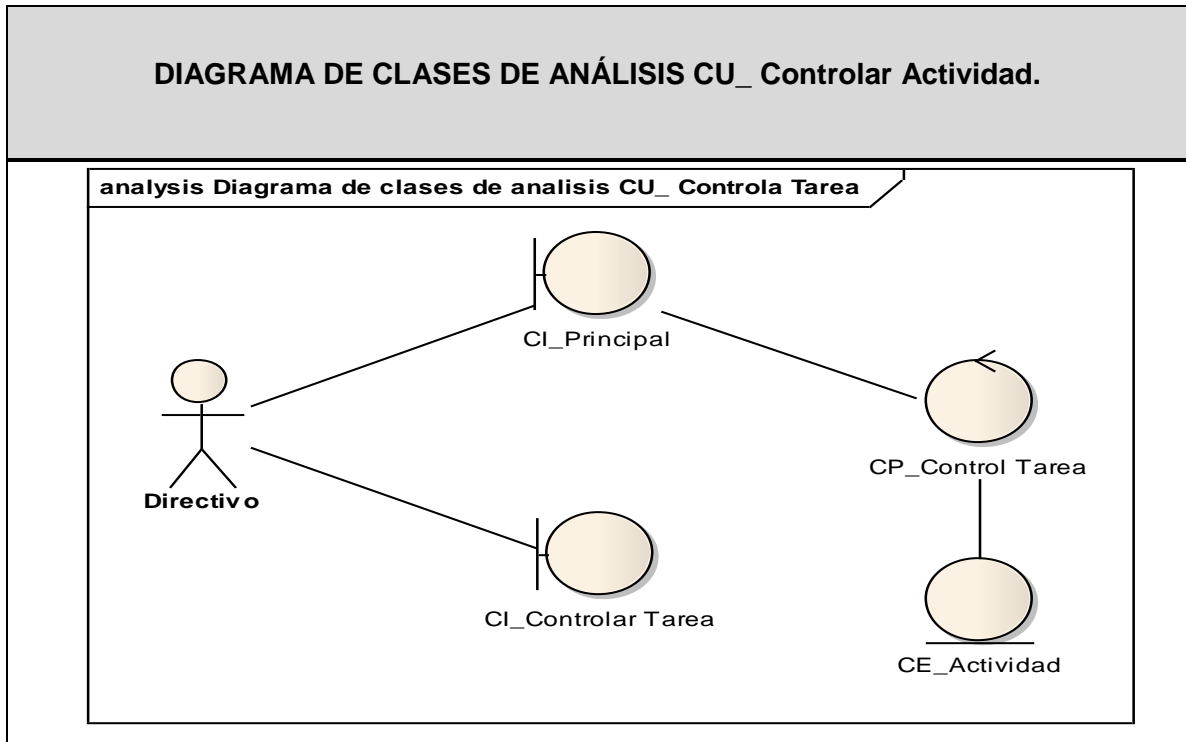


Figura 9. Diagrama de clases de análisis del caso de uso: Controlar Actividad.

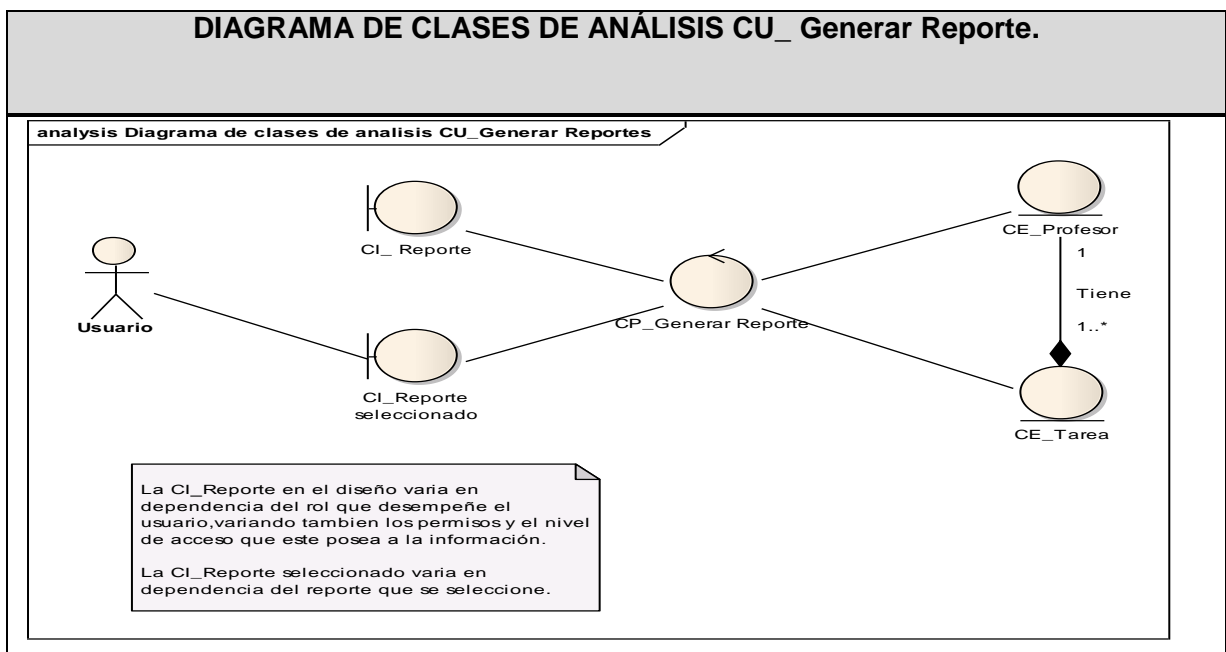


Figura 10. Diagrama de clases de análisis del caso de uso: Generar Reporte



## **3.3. Diseño**

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

### **3.3.1. Principios de Diseño**

El Diseño es la antesala a la implementación del sistema, mediante el Modelo de Diseño se puede verificar el cumplimiento de los requerimientos del sistema sin tener un prototipo funcional de este, además define e identifica las consecuencias del ambiente de implementación; es un modelo o guía para realizar la implementación del sistema que se va a desarrollar.

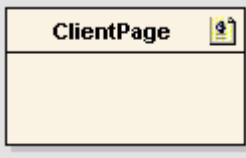
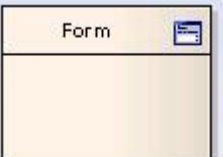

### **3.3.2. Diagrama de clases del diseño**

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

### **3.3.3. Modelado mediante estereotipos Web**

Para la realización de los Diagramas de Clases del Diseño se utiliza la extensión de UML para el modelado de aplicaciones Web. Esta extensión presenta como elementos significativos a tres clases UML: Server Page, Client Page y Form empleadas para el código servidor, código cliente y formularios respectivamente, permite además representar ficheros contenedores de sentencias script. A continuación se muestra una tabla con las posibles relaciones entre clase, a la hora de ser desarrollados los diagramas del diseño de la aplicación.



Estereotipos	Hasta Desde	Client Page	Form	Server Page
	Client Page	<b>&lt;&lt;link&gt;&gt;</b> , <b>&lt;&lt;redirect&gt;&gt;</b>	<b>aggregation</b>	<b>&lt;&lt;link&gt;&gt;</b> , <b>&lt;&lt;redirect&gt;&gt;</b>
Representan páginas que son mostradas por el navegador web y pueden ser una combinación de algún o algunos lenguajes de marcado.				
	Form	<b>aggregated by</b>	<b>&lt;&lt;submit&gt;&gt;</b>	Form
Representa una colección de campos de entrada que forman parte con una página del lado cliente (Client Page).				
	Server Page	<b>&lt;&lt;build&gt;&gt;</b> , <b>&lt;&lt;redirect&gt;&gt;</b>	<b>&lt;&lt;redirect&gt;&gt;</b> , <b>&lt;&lt;include&gt;&gt;</b>	Server Page
Representa una página Web que tiene scripts ejecutados por el servidor. Estos scripts interactúan con los recursos que se encuentran al alcance del servidor.				

**Tabla 9. Representación de los estereotipos a usar en el Diagrama de Clases del Diseño.**

La ServerPage es la responsable de construir o generar el resultado XHTML que conforma a la ClientPage mediante la relación **<<build>>**. Esta página cliente está compuesta por Form (formularios) de ahí que la relación entre ambas sea de agregación. Los formularios se encargan de enviar los datos a la ServerPage para ser procesados mediante la relación **<<submit>>**. Entre páginas clientes pueden existir vínculos (**<<link>>**) o redireccionamientos (**<<redirect>>**). Una página cliente es construida por una sola página servidora.

La servidora a su vez, puede completar su funcionamiento si incluye código existente en otra página de este mismo tipo, utilizando la relación de inclusión (**<<include>>**), que a pesar de no ser propia de la extensión de UML, es considerada para representar las relaciones existentes en el modelo. En este



modelo del diseño, los casos de uso son realizados por las clases del diseño y sus objetos. A continuación se muestran los diagramas de clases del diseño y diagramas de interacción para la realización de los casos de usos descritos en el capítulo anterior.

## Diagrama de clases del diseño del caso de uso: Gestionar Planificación.

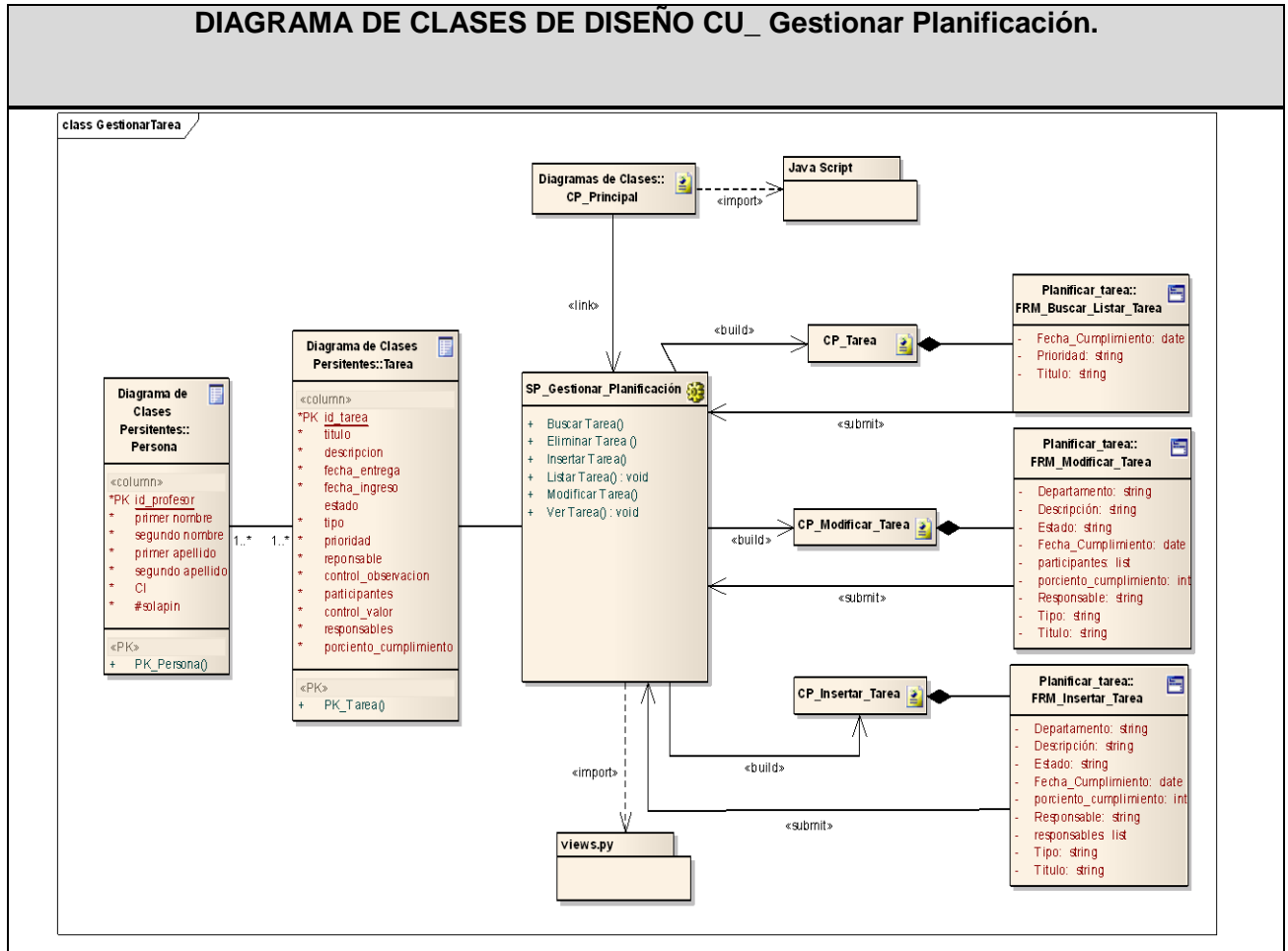


Figura 11. Diagrama de Clases del Diseño del Caso de Uso, Gestionar Planificación.



Diagrama de clases del diseño del caso de uso: Controlar Actividad.

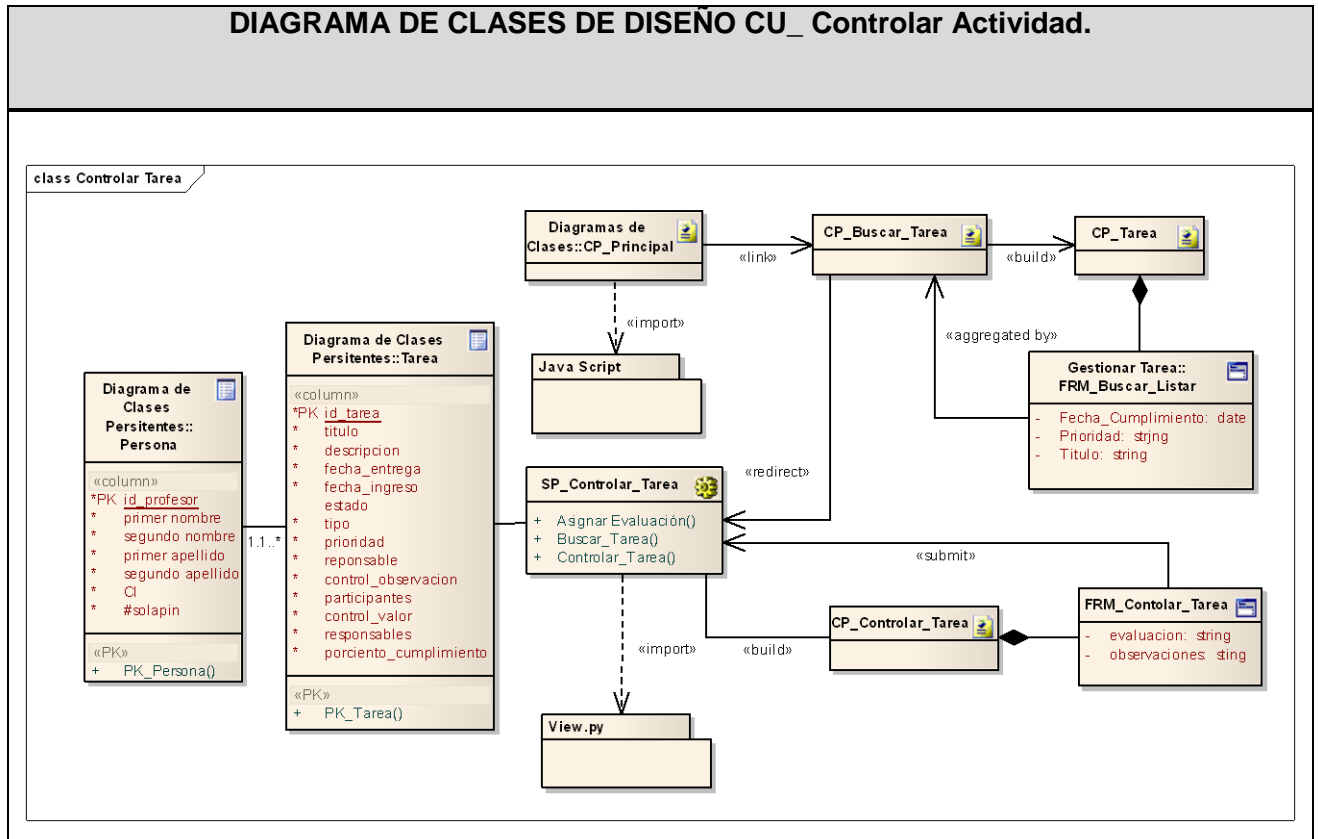


Figura 12. Diagrama de Clases del Diseño del Caso de Uso, Controlar Actividad.



Diagrama de clases del diseño del caso de uso: Generar Reporte.

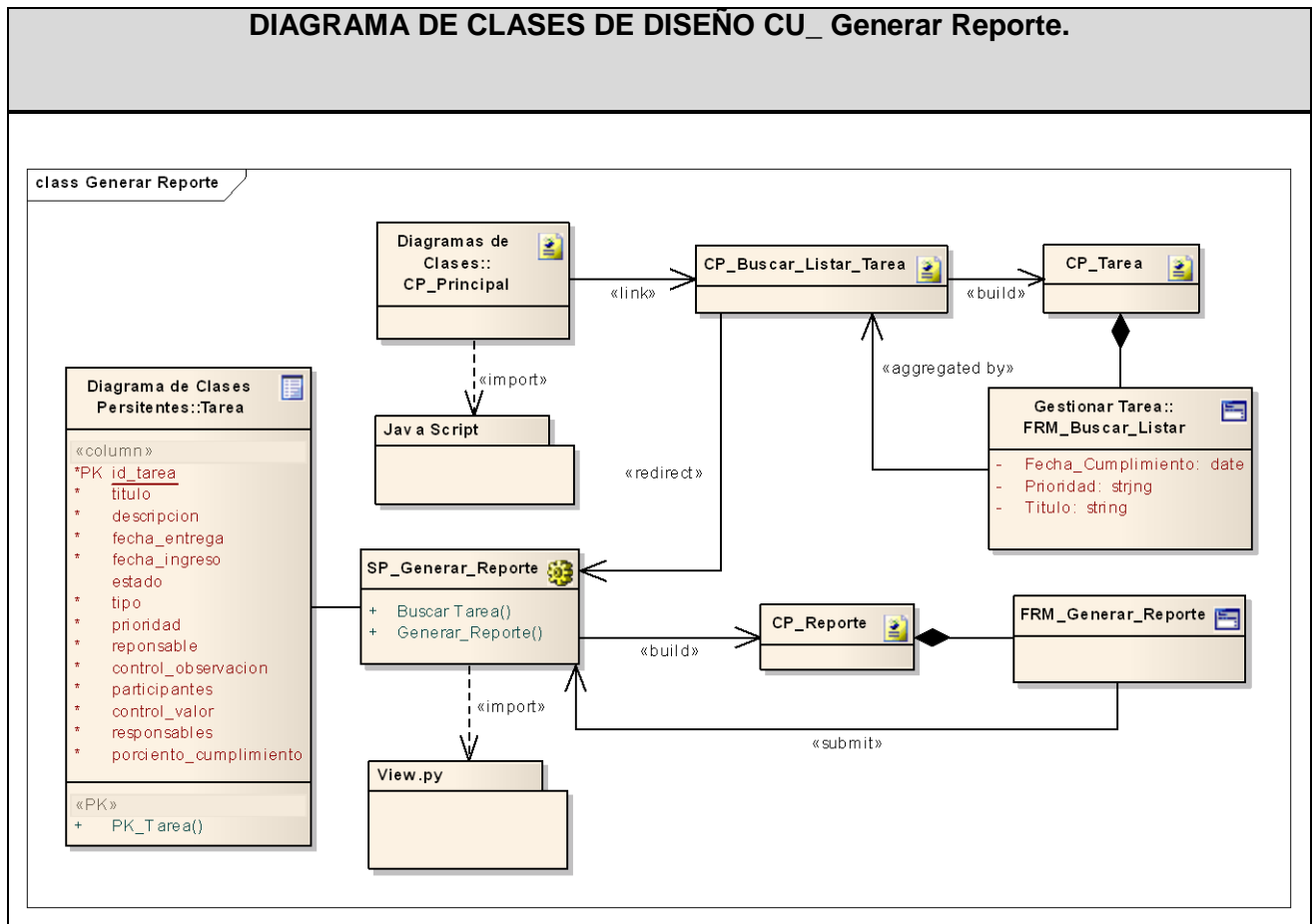


Figura 13. Diagrama de Clases del Diseño del Caso de Uso, Generar Reporte.

3.3.4. Diagramas de interacción

Entre los artefactos que brinda UML para expresar las iteraciones entre los objetos dándole cumplimiento a los requerimientos del sistema, se encuentran los Diagramas de Interacción. Estos se clasifican en dos tipos: Diagramas de Secuencia y Diagramas de Colaboración. Permiten modelar aspectos dinámicos del sistema. Se utilizan para realizar una traza de la ejecución de un escenario y a cada uno de estos le corresponde un diagrama de interacción. Una interacción es un conjunto de objetos y sus relaciones, incluye los mensajes ordenados temporalmente, mediante los cuales pueden establecer comunicación. Un diagrama de colaboración destaca la organización estructural de los objetos que envían y reciben mensajes.





## Diagrama de Secuencia

El diagrama de secuencia es uno de los diagramas más efectivos para modelar interacción entre objetos en un sistema. Un diagrama de secuencia muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada método de la clase. Mientras que el diagrama de casos de uso permite el modelado de una vista del escenario, el diagrama de secuencia contiene detalles de implementación del escenario, incluye los objetos y clases que se usan para implementar el escenario, y mensajes pasados entre los objetos.

Los diagramas de secuencias se usan continuamente como modelos explicativos para los escenarios de los casos de uso. Muestra los objetos que intervienen en el escenario con líneas discontinuas verticales, y los mensajes pasados entre los objetos como flechas horizontales. Existen dos tipos de mensajes: síncronos y asíncronos.

- Los mensajes síncronos se corresponden con llamadas a métodos del objeto que recibe el mensaje. El objeto que envía el mensaje queda bloqueado hasta que termina la llamada.
- Los mensajes asíncronos terminan inmediatamente, y crean un nuevo hilo de ejecución dentro de la secuencia.



DIAGRAMA DE INTERACCIÓN DEL DISEÑO CU\_Asignar Tarea.

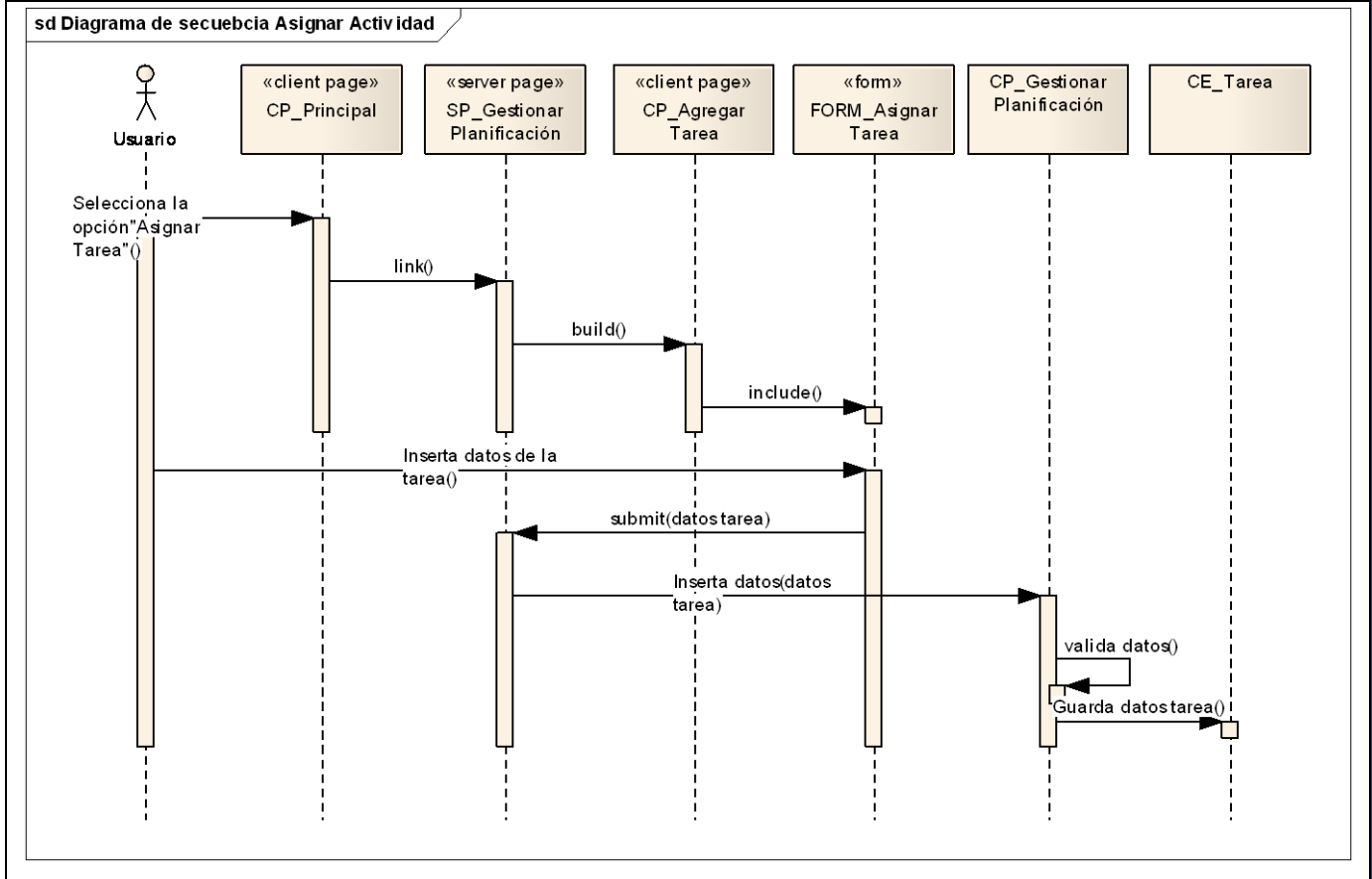


Figura 14. Diagrama de Interacción del Diseño Asignar Tarea.

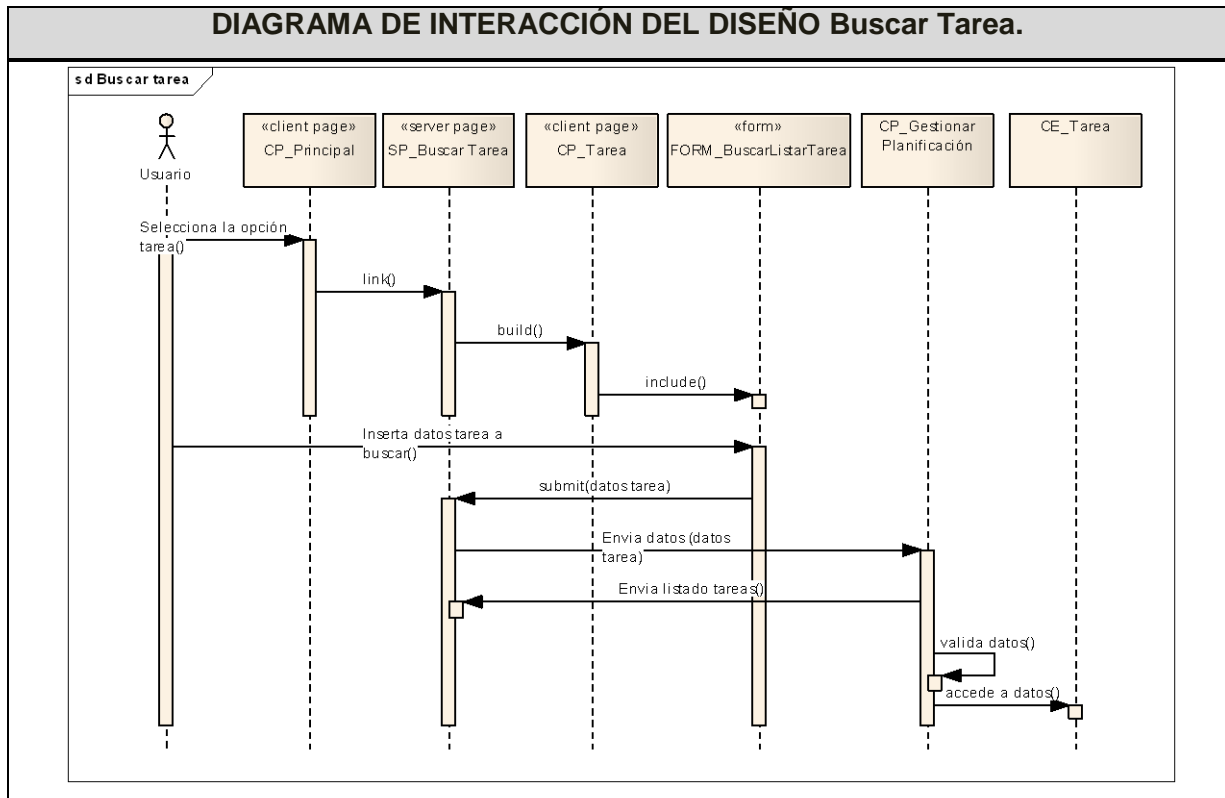


Figura 15. Diagrama de Interacción del Diseño Buscar Tarea.

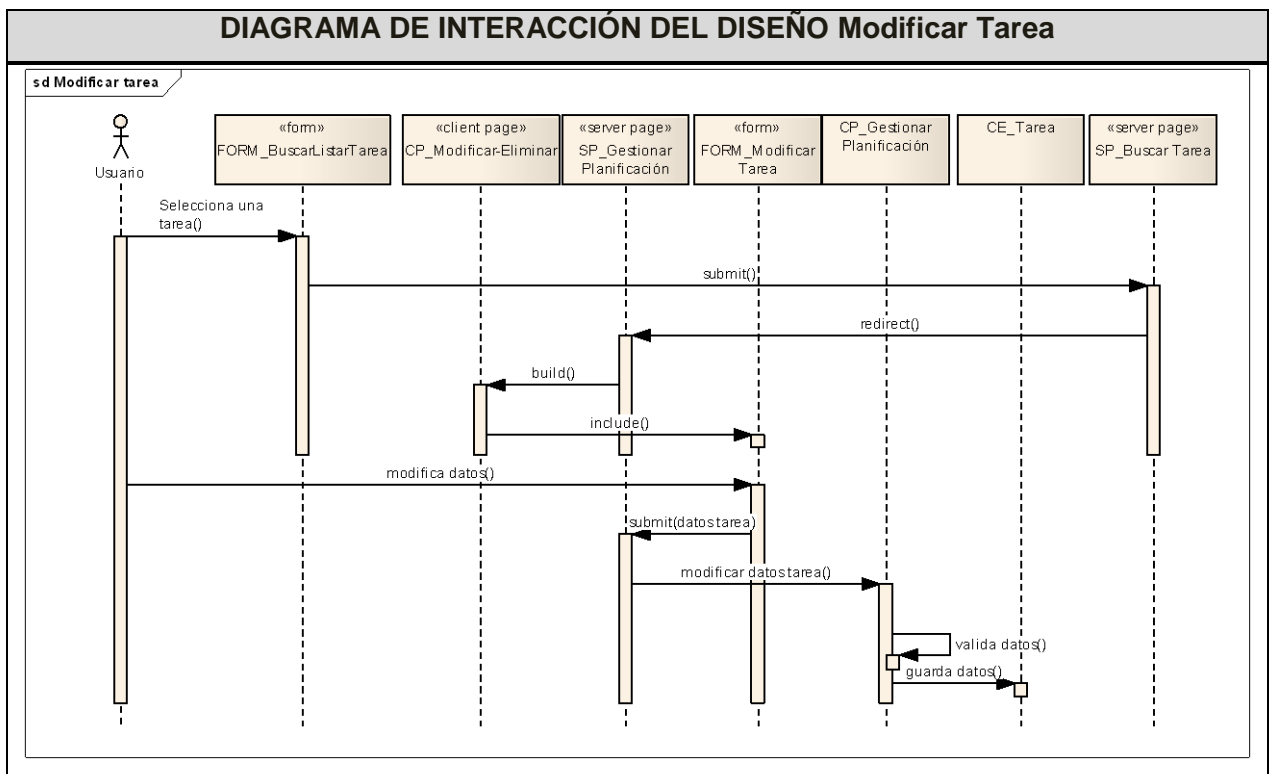


Figura 16. Diagrama de Interacción del Diseño Modificar Tarea

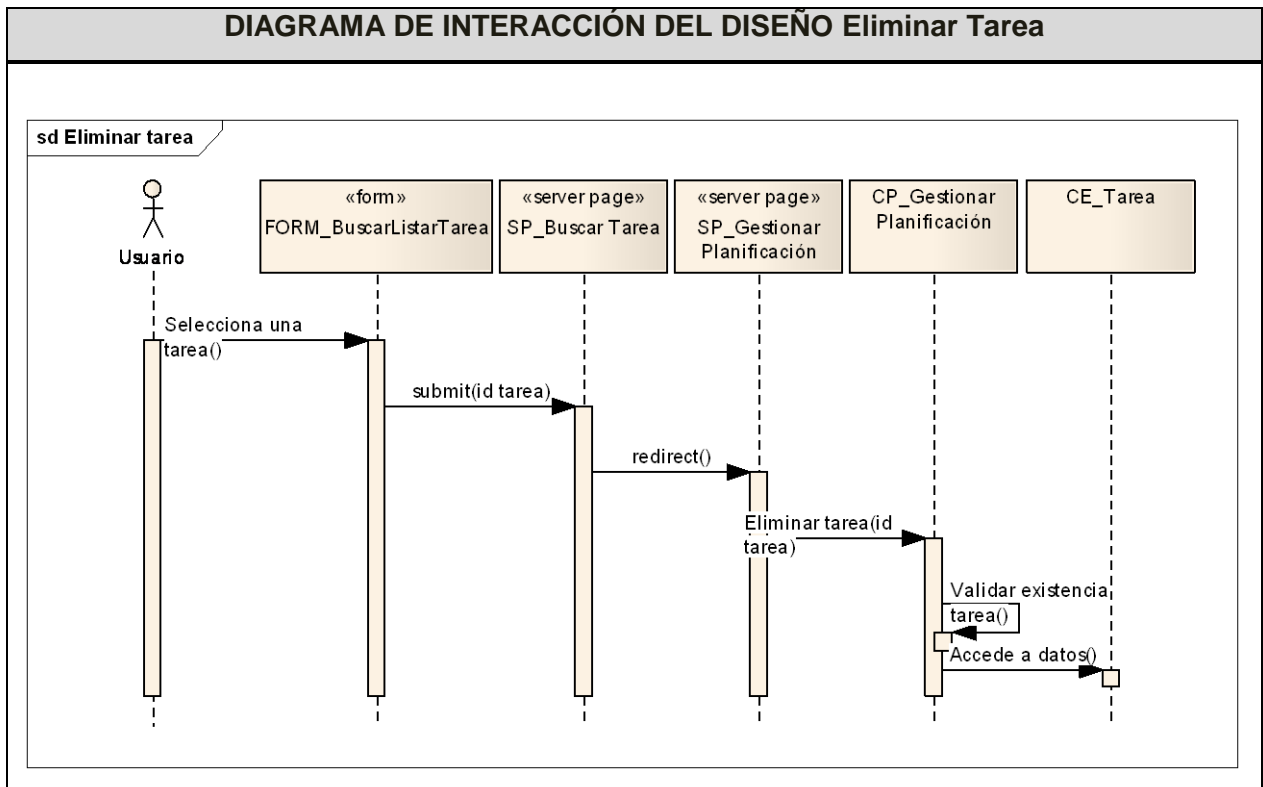


Figura 17. Diagrama de Interacción del Diseño Eliminar Tarea

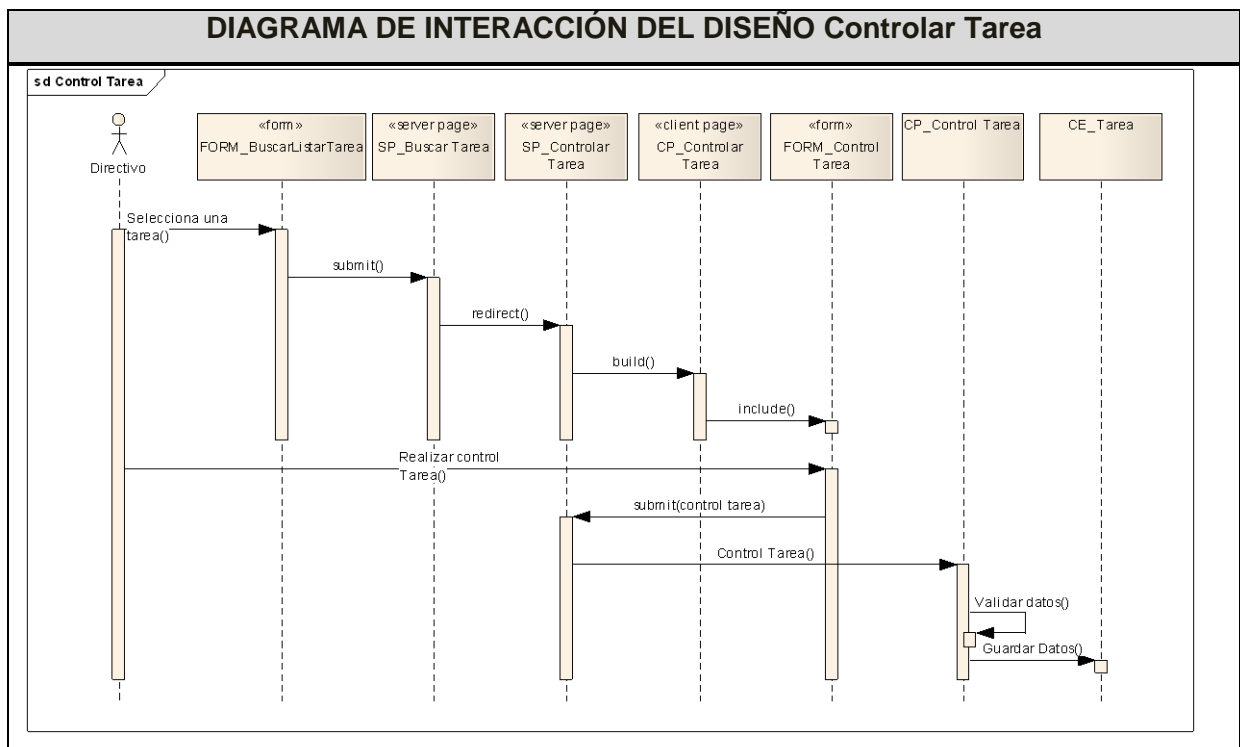


Figura 18. Diagrama de Interacción del Diseño Controlar Tarea



### 3.3.5. Descripción de las clases del diseño.

La descripción de las clases del diseño permite conocer los atributos que conforman las mismas así como las operaciones más importantes de las que cuenta cada una de estas.

#### Descripción de las clases para Gestionar Planificación.

<b>NOMBRE:</b>	Buscar Listar Tarea	
<b>Tipo de Clase:</b>	Interfaz	
<b>Atributo</b>	<b>Tipo</b>	
Título	Campo de texto	
Responsable	Combo box	
Ingresada por:	Combo box	
Objetivo(desde, hasta)	Calendar	
Buscar	Botón	
Cancelar	Botón	
Modificar	Botón	
Eliminar	Botón	
Listado de tareas	Tabla	
<b>Descripción:</b>	Esta clase muestra una interfaz para Buscar una tarea. Además muestra el resultado de la búsqueda de las mismas.	

<b>NOMBRE:</b>	Modificar	
<b>Tipo de Clase:</b>	Interfaz	
<b>Atributo</b>	<b>Tipo</b>	
Título	Campo de texto	
Descripción	Campo de texto	
Fecha_Cumplimiento	Calendar	
Responsable	Combo box	
Participantes	Combo box	
Prioridad	Combo box	
Porcentaje_Cumplimiento	Combo box	
Tipo	Combo box	



Estado	Combo box
Fecha_Planificación	DateTime
Eliminar	Botón
Modificar	Botón
Cancelar	Botón
<b>Descripción:</b>	Esta clase muestra una interfaz para Modificar todos los datos relacionados con una Tarea.

<b>NOMBRE:</b>	Tarea
<b>Tipo de Clase:</b>	Interfaz
<b>Atributo</b>	<b>Tipo</b>
Título	Campo de texto
Descripción	Campo de texto
Fecha_Cumplimiento	Calendar
Responsable	Combo box
Participantes	Combo box
Prioridad	Combo box
Porcentaje_Cumplimiento	Combo box
Tipo	Combo box
Estado	Combo box
Fecha_Planificación	DateTime
Eliminar	Botón
Modificar	Botón
Cancelar	Botón
<b>Descripción:</b>	Esta clase muestra una interfaz para Insertar todos los datos relacionados con una Tarea.

<b>NOMBRE:</b>	Gestionar Planificación
<b>Tipo de Clase:</b>	Controladora



Para cada responsabilidad:	
<b>Nombre:</b>	Buscar Tarea() Eliminar Tarea() Insertar Tarea() Modificar Tarea() Listar Tarea() Ver Tarea() Controlar Tarea()
<b>Descripción:</b>	Esta clase permite gestionar todos los datos relacionados con una tarea.

#### Descripción de las clases para Controlar Actividad

<b>NOMBRE:</b>	Buscar Listar Tarea	
<b>Tipo de Clase:</b>	Interfaz	
<b>Atributo</b>	<b>Tipo</b>	
Titulo	Campo de texto	
Fecha_Cumplimiento	DateTime	
Prioridad	Combo box	
Buscar	Botón	
Cancelar	Botón	
Controlar	Botón	
Listado de tareas	Tabla	
<b>Descripción:</b>	Esta clase muestra una interfaz para Buscar una tarea asignada a otra persona. Además muestra el resultado de la búsqueda de las mismas.	

<b>NOMBRE:</b>	Control Tarea	
<b>Tipo de Clase:</b>	Interfaz	
<b>Atributo</b>	<b>Tipo</b>	
Bien	Checkbox	
Regular	Checkbox	
Mal	Checkbox	
Observaciones	Campo de texto	



Buscar	Botón
Cancelar	Botón
Aceptar	Botón
Listado de tareas	Tabla
<b>Descripción:</b>	Esta clase muestra una interfaz para Controlar una tarea asignada a otra persona.

En este capítulo se obtuvo el Modelo de Análisis del sistema propuesto. Se realizaron diferentes artefactos mediante los cuales es posible adquirir una visión detallada de las responsabilidades de cada Clase del Análisis para el correcto funcionamiento del sistema a desarrollar. Los artefactos desarrollados fueron los Diagramas de Clases del Análisis y la Descripción de las Clases.

Además se obtuvo el Modelo de Diseño. Los artefactos generados durante la modelación del diseño, definen la entrada, a las actividades que se realizan en el Flujo de Trabajo de Implementación, estos son los Diagramas de Clases del Diseño, la Descripción de las Clases y los Diagramas de Secuencia.





## Introducción

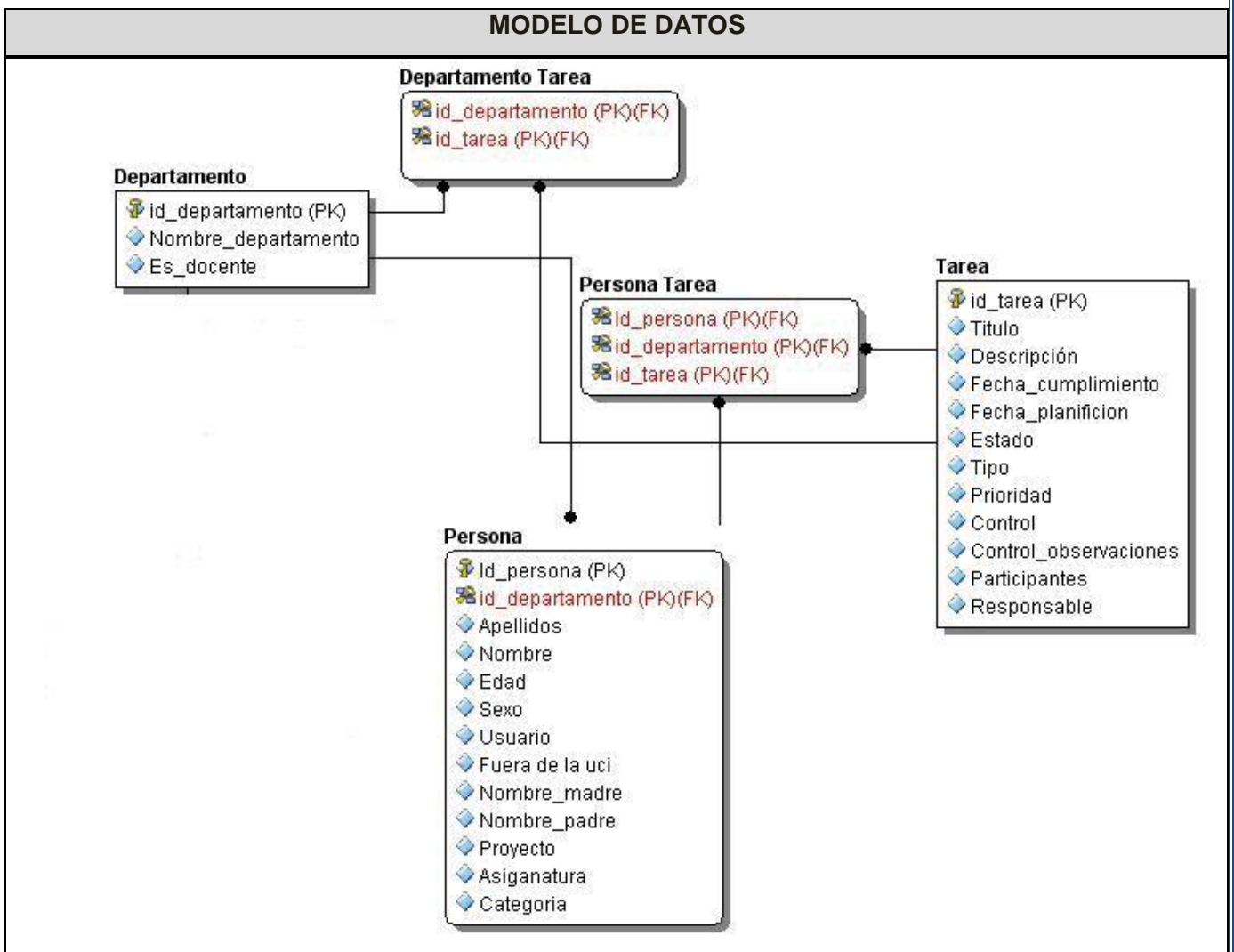
El presente capítulo constituye la secuencia lógica del Diseño del Sistema, a través del mismo se implementan las clases y subsistemas encontrados durante el Flujo de Trabajo anterior en términos de componentes. Se describen las tablas del modelo de datos que se realizó específicamente para posibilitar la planificación y control de las actividades. Como parte del Modelo de Implementación, se obtiene además el Diagrama de Despliegue, que indica cómo la solución implementada está distribuida físicamente.

### 4.1. Modelo de datos

Los modelos de datos aportan la base conceptual para diseñar aplicaciones que hacen un uso intensivo de datos, así como la base formal para las herramientas y técnicas empleadas en el desarrollo y uso de sistemas de información. Un modelo de datos no es más que un conjunto de conceptos, reglas y convenciones que permiten describir y manipular determinados datos que se desean almacenar en una base de datos. Al producto del modelo de datos se le llama esquema (descripción de la estructura de la base de datos) y a los datos almacenados e ese momento, ocurrencia del esquema.



**Modelo de datos.**



**Figura 19. Modelo de datos.**

A continuación se describen las tablas que se utilizan para el desarrollo del sistema propuesto.

<b>NOMBRE:</b> Tarea		
<b>Descripción:</b> Almacena los datos correspondientes a las tareas que tienen los profesores de la Facultad.		
Atributo	Tipo	Descripción
id_tarea	IntegerField	Llave primaria: representa el identificador de una tarea. Atributo que



		no cambia su valor y mantiene la integridad y relación entre los datos.
Id_departamento	IntegerField	Llave foránea: contiene el id de el departamentos a que pertenece cada persona.
Titulo	TextField	Contiene el titulo de la tarea
Descripción	TextField	Contiene una breve descripción de la tarea
Fecha_Cumplimiento	DateField	Contiene la fecha en que debe ser entregada la tarea.
Fecha Planificación	DateField	Contiene la fecha en que la tarea fue ingresada en el sistema.
Estado	CharField	Contiene el estado en que se encuentra una tarea, inicialmente esta en "No empezado"
Tipo	CharField	Contiene el tipo de una tarea.
Prioridad	CharField	Contiene la prioridad de una tarea.
Porciento_Cumplimiento	IntegerField	Contiene el porciento de realización de una tarea.
Control	CharField	Contiene control asignado a una tarea.
Control_observaciones	TextField	Contiene las observaciones realizados durante el control de la tarea
Participante	TextField	Contiene el nombre de los participantes para la tarea.
Responsable	TextField	Contiene el nombre del responsable para la tarea.

<b>NOMBRE:</b> Departamento_Tarea		
<b>Descripción:</b> Almacena los datos correspondientes las tareas que posee un departamento		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id_departamento	IntegerField	Llave primaria: representa el identificador de un departamento. Atributo que no cambia su valor y



		mantiene la integridad y relación entre los datos.
id_tarea	IntegerField	Llave foránea: contiene los id de las tareas de los departamentos

<b>NOMBRE:</b> Persona_Tarea		
<b>Descripción:</b> Almacena los datos correspondientes a las Tareas de las personas.		
Atributo	Tipo	Descripción
id_persona	IntegerField	Llave primaria: representa el identificador de una persona. Atributo que no cambia su valor y mantiene la integridad y relación entre los datos.
id_departamento	IntegerField	Llave foránea: contiene los id de los departamentos en que se encuentran las personas.
Id_tarea	IntegerField	Llave foránea: contiene los id de las tareas que posee la persona.

## 4.2. Diagrama de componentes

Un diagrama de componentes representa la separación de un sistema de software en componentes físicos (archivos, cabeceras, módulos, paquetes) y muestra las dependencias entre estos componentes. Estos son utilizados para modelar la vista estática de un sistema. Muestra la organización y las dependencias entre un conjunto de componentes. No es necesario que un diagrama incluya todos los componentes del sistema, normalmente se realizan por partes. Cada diagrama describe un apartado del sistema.

En estos apartados se sitúan librerías, tablas archivos, ejecutables y documentos que forman parte del sistema. Uno de los usos principales es que puede servir para ver cuales componentes pueden compartirse entre sistemas o entre diferentes partes de un sistema. Lo que distingue a un diagrama de componentes de otros tipos de diagramas es su contenido. Normalmente contienen componentes, interfaces y relaciones entre ellos. Y como todos los diagramas, también puede contener paquetes utilizados para agrupar elementos del modelo.



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

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

Un paquete en un diagrama de componentes representa una división física del sistema. Los paquetes se organizan en una jerarquía de capas donde cada capa tiene una interfaz bien definida. Un ejemplo típico de una jerarquía en capas de este tipo es: Interfaz de usuario; Paquetes específicos de la aplicación; Paquetes reusables; Mecanismos claves; y Paquetes hardware y del sistema operativo. Un diagrama de componentes se representa como un grafo de componentes software unidos por medio de relaciones de dependencia (generalmente de compilación). Puede mostrar también que un componente software contiene una interfaz, es decir, la soporta.

La estructura de paquetes del diagrama de componentes de la Aplicación Web para el Módulo de planificación y seguimiento de actividades para los profesores en la Facultad 7 queda de la siguiente manera:(Ver Fig.16)

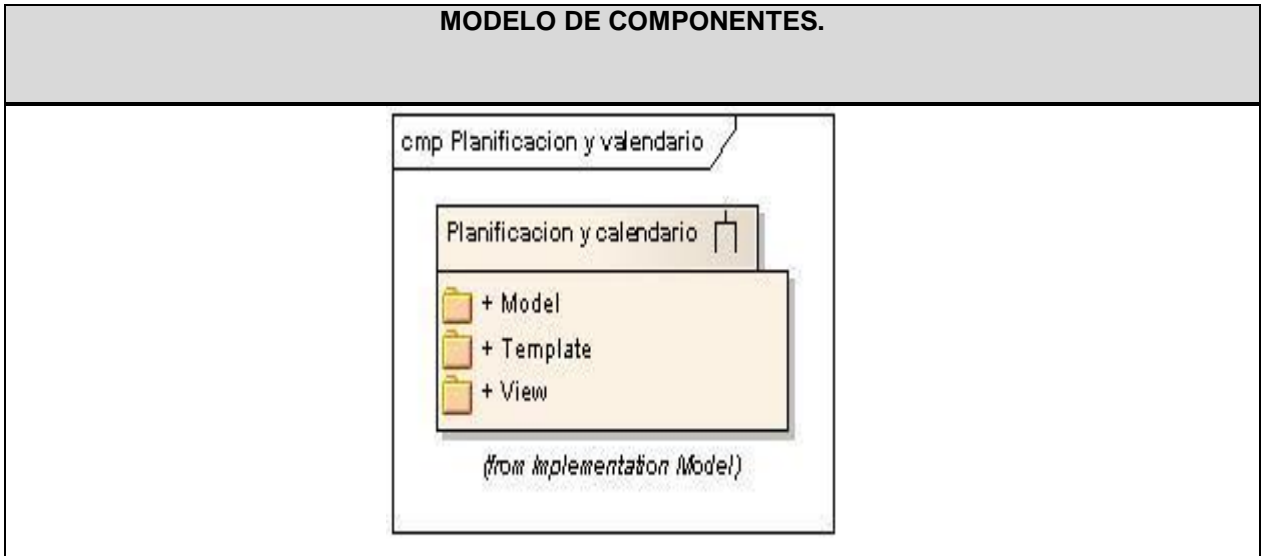


Figura 20. Diagrama de Componentes General.

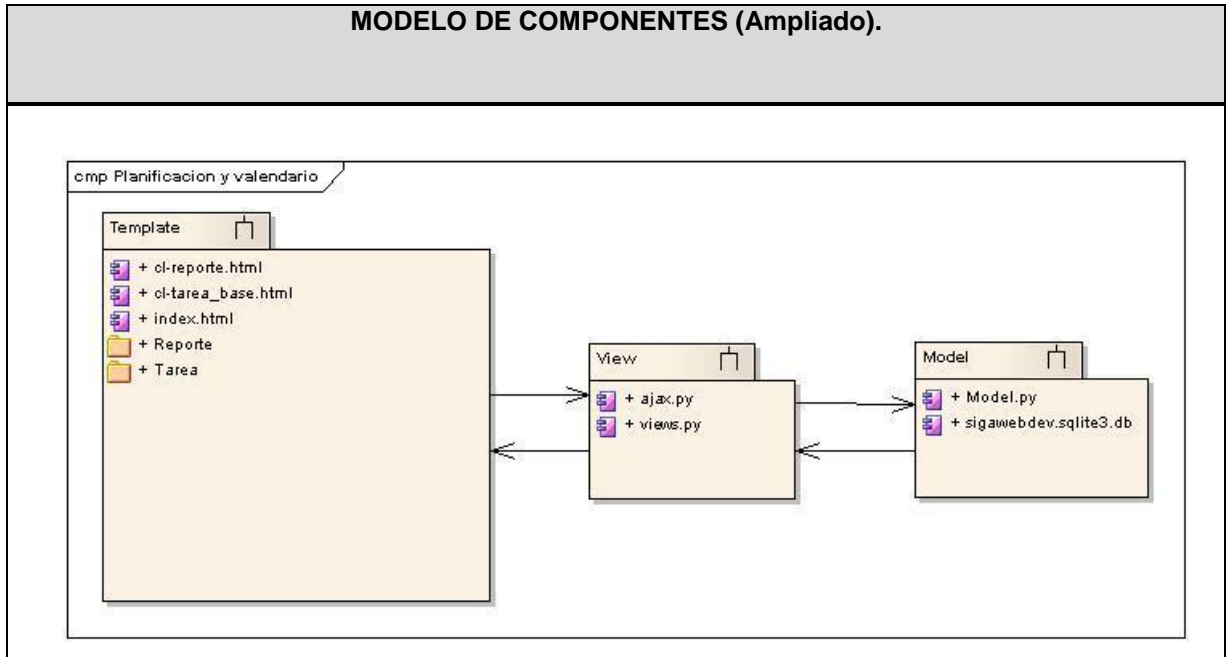


Figura 21. Modelo de componentes

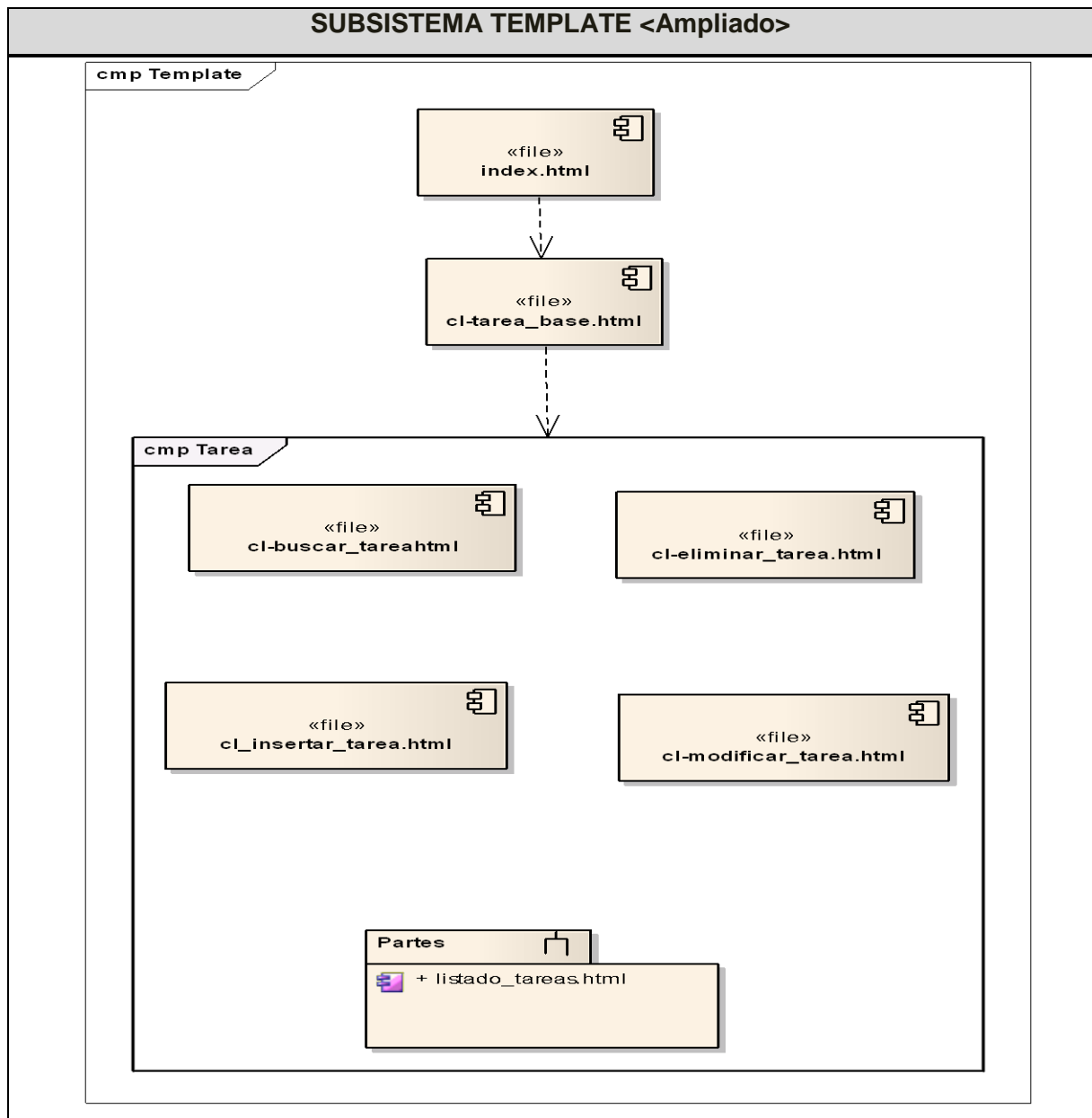


Figura 22. Subsistema Template.

### 4.3 Diagrama de despliegue

El modelo físico de despliegue, provee un modelo detallado de la forma en la que los componentes se despliegan a lo largo de la infraestructura del sistema. En este se detallan las capacidades de red, las especificaciones del servidor, los requisitos de hardware y otra información relacionada al despliegue



del sistema propuesto. Un diagrama de despliegue, muestra las relaciones físicas entre los componentes hardware y software en el sistema final, es decir, la configuración de los elementos de procesamiento en tiempo de ejecución y los componentes software (procesos y objetos que se ejecutan entre ellos).

Además, se puede ver, como es un grafo de nodos unidos por conexiones de comunicación. Un nodo, puede contener instancias de componentes software, objetos y procesos. Esta es una unidad de computación de algún tipo. Las instancias de componentes software pueden estar unidas por relaciones de dependencia.

Para la estructura de la posible distribución del sistema se utilizó un nodo PC\_Cliente que representa la computadora del usuario. Esta se conecta mediante el protocolo de transferencia de hipertexto (http por sus siglas en inglés, HyperText Transfer Protocol) al nodo Servidor WEB y de Aplicación. Desde el servidor se puede acceder al nodo Servidor de BD mediante el protocolo diseñado para facilitar la reutilización de código de bases de datos, la capa de abstracción de bases de datos (TCP/IP, Transmission Control Protocol / Internet Protocol, por sus siglas en inglés).

Además se tiene una impresora a la que se accede mediante un puerto bus universal en serie (USB por sus siglas en inglés, Universal Serial Bus) o Conductor Universal en Serie (CUS), abreviado comúnmente USB, es un puerto que sirve para conectar periféricos a una computadora. (Ver. Fig. 19)



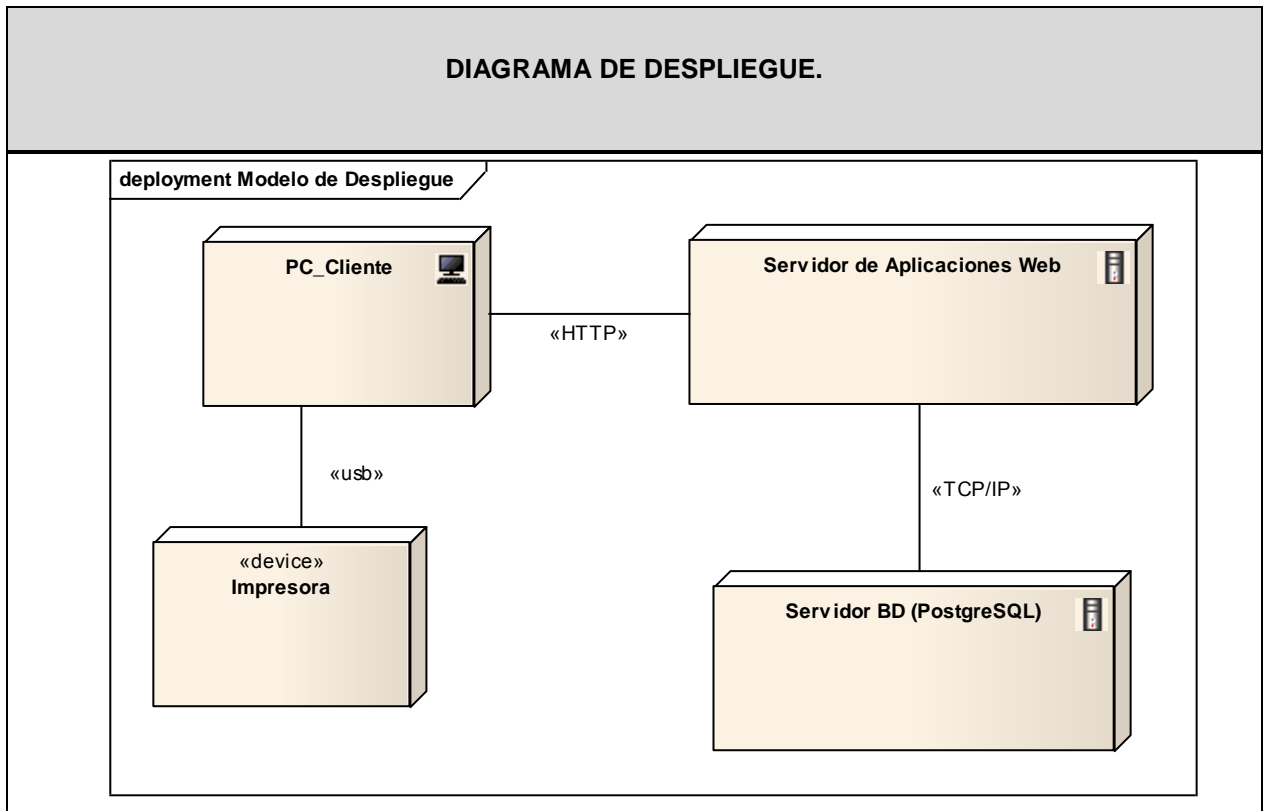


Figura 23. Diagrama de Despliegue del Sistema Control Docente.

#### 4.5 Seguridad.

La seguridad como elemento esencial para preservar la integridad, autenticidad y confiabilidad de la información debe estar presente en cada sistema de software que se desarrolle, por ello y debido a que el sistema implementado está basado en la web, la seguridad juega un papel fundamental para el correcto funcionamiento.

El uso de Django garantizará la protección automática de muchos de los errores comunes de seguridad, brindando facilidades y herramientas para combatir vulnerabilidades como inyecciones SQL, inyección de cabeceras de email, exposición de mensajes de error, entre otras facilidades.



El sistema provee además un sistema de autenticación donde se solicita nombre de usuario y contraseña verificando que los datos introducidos sean válidos y garantizando que usuarios no autorizados accedan a los datos que se manejan en el sistema.

En el capítulo se completó el ciclo de vida del software, la etapa de Implementación. Esta es el centro durante las iteraciones de construcción. También se lleva a cabo el trabajo de implementación durante la fase de Elaboración, para crear la línea base ejecutable de la arquitectura.

Durante la fase de Transición, para tratar defectos tardíos; debido a que el modelo de implementación denota la implementación actual del sistema en términos de componentes y subsistemas de implementación.

Por lo que es significativo, mantener el modelo de implementación a lo largo de todo el ciclo de vida del software.



## CONCLUSIONES

Para la confección del presente trabajo de diploma se realizaron estudios exhaustivos de los procesos que intervienen en la planificación y el control de las actividades de los profesores, llegando de esta manera a la conclusión de que la principal causa de los problemas que existen en la Facultad 7 de la UCI con respecto a la planificación y control de las actividades de los profesores, es la carencia de una herramienta informatizada y especializada que facilite la actualización, procesamiento y obtención de la información vinculada a dichos procesos.

Se analizaron los distintos sistemas informatizados existentes a nivel internacional y nacional (Redmine, Groobix, SGIF, entre otros) además de un análisis de las tendencias y tecnologías escogidas que se relacionaban con el sistema que se iba a desarrollar para lograr informatizar todos los procesos que se relacionan con la confección de la planificación y control de las actividades.

Se definieron todos los artefactos que se encuentran relacionados con cada flujo de trabajo de la metodología RUP, trayendo consigo el cumplimiento estricto de las reglas del negocio y los requerimientos escogidos.

Se obtuvo como resultado una aplicación web definida bajo una arquitectura Modelo Vista Controlador, específicamente el Modelo Vista Template. Implementada y diseñada mediante Rational Unified Process (RUP) como Metodología de desarrollo. Como lenguaje de Modelado Lenguaje de Modelado Unificado (UML 2.0). Como herramienta de modelado Enterprise Architect (EA). Como gestor de bases de datos se tuvo PostgreSQL. Se usó Python como lenguaje de programación, Django como marco de desarrollo web y WingIDE como entorno de desarrollo.

Por lo anteriormente expuesto se concluye que se han cumplido satisfactoriamente los objetivos trazados para la confección del presente trabajo de diploma; obteniendo como resultado un producto informático que favorecerá el mejoramiento de todos los procesos relacionados con la planificación y el control de las actividades de los profesores durante un curso académico.



## RECOMENDACIONES

Con el cumplimiento del objetivo propuesto para la solución de la investigación y en aras de mejorar el funcionamiento del sistema se recomienda:

- Elaborar un plan de pruebas con el objetivo de verificar el buen funcionamiento de esta aplicación antes de su puesta en marcha.
- Documentar las pruebas que se le hagan a la aplicación para su reparación posterior.
- Extender la Planificación de Actividades, por esta vía, a todos los sectores de la Facultad, para así llevar un mejor control de las tareas organizadas por dicha entidad aumentando el alcance en el almacenamiento de datos e incorporando a los estudiantes en estas.
- Posibilitar la exportación de los reportes a otros formatos como documentos Word y Excel.
- Hacer un uso posterior de este sistema en el resto de las facultades de la Universidad de las Ciencias Informáticas.
- Mejorar el diseño actual del sistema para ganar una buena aceptación del usuario.

**REFERENCIAS BIBLIOGRÁFICAS.**

1. ¿Qué es la gestión? Autor Yibetza Romero [En línea] [Citado el: 9 de marzo de 2010.] <http://www.monografias.com/trabajos53/tecnologia-comunicacion/tecnologia-comunicacion.shtml>
2. Propósitos de la planificación, autor Eduardo, Fernández. Monografías. Monografías [En línea] [Citado el: 9 de marzo de 2010.] <http://www.monografias.com/trabajos34/planificacion/planificacion.shtml>.
3. Sistemas de gestión [En línea] [Citado el: 9 de marzo de 2010.] <http://www.bsigroup.com.mx/es-mx/Auditoria-y-Certificacion/Sistemas-de-Gestion/De-un-vistazo/Que-son-los-sistemas-de-gestion/>
4. Redmine [En línea] [Citado el: 9 de marzo de 2010.] <http://en.wikipedia.org/wiki/Redmine>
5. Quartz [En línea] [Citado el: 9 de marzo de 2010.] <http://www.turincon.net/timegt-1-1-1-gestor-de-tareas-con-etiquetas-y-google-calendar>
6. Dates Reminder. Luis Ponce de León [En línea] [Citado el: 9 de marzo de 2010.] <http://dates-reminder.softonic.com/>
7. UML2.0. Fowler, Martín. UML Gota a Gota. 1999 [En línea] [Citado el: 9 de marzo de 2010.] <http://www.docshare.com/doc/195995/UML-Gota-a-Gota-Martin-Fowler-Kendall-Scott-P.>
8. Enterprise Architect - Herramienta de diseño UML. [Citado el: 9 de marzo de 2010.] <http://www.sparxsystems.com.ar/products/ea.html> ].
9. Python. Miguel Ángel Álvarez [En línea] [Citado el: 9 de marzo de 2010.] <http://www.desarrolloweb.com/articulos/1325.php>.
10. Tutorial de Django. [En línea] [Citado el: 9 de febrero de 2010.] <http://davidasorey.net/static/django-tutorial/>
11. Requisitos Funcionales y no Funcionales. [En línea] [Citado el: 30 de febrero de 2010.] <http://synergix.wordpress.com/2008/07/07/requisito-funcional-y-no-funcional/>

**BIBLIOGRAFÍA.**

1. Área de Enfermería. [En línea] [Citado el: 9 de febrero de 2010.]  
[http://www.madrid.org/cs/Satellite?cid=1142436193437&language=es&pagename=HospitalRamonCajal%2FPage%2FHRYC\\_contenidoFinal](http://www.madrid.org/cs/Satellite?cid=1142436193437&language=es&pagename=HospitalRamonCajal%2FPage%2FHRYC_contenidoFinal) .
2. E-Mergencia. [En línea] [Citado el: 12 de febrero de 2010.]  
<http://www.e-mergencia.com/foro/showthread.php?t=13543>.
3. Enterprise Architect. [En línea] [Citado el: 9 de febrero de 2010.]  
<http://www.sparxsystems.com.ar/products/ea.html>
4. Ingeniería de Software II."Continuación del FT Análisis de Diseño." Conferencia # 1.s.I.: UCI, 2009 - 2010.
5. Ministerio del Poder Popular para las Telecomunicaciones y la Informática. MeRinde. Modelo de Diseño [En línea] [Citado el: 30 de febrero de 2010.]  
[http://merinde.rinde.gob.ve/index.php?option=com\\_content&task=view&id=92&Itemid=296](http://merinde.rinde.gob.ve/index.php?option=com_content&task=view&id=92&Itemid=296)
6. Patrón MVC. [En línea] [Citado el: 30 de febrero de 2010.]  
<http://www.programacionweb.net/articulos/articulo/?num=505>
7. PostgreSQL. [En línea] [Citado el: 9 de febrero de 2010.]  
<http://es.wikipedia.org/wiki/PostgreSQL>.
8. Leguaje Unificado de Modelado. [En línea] [Citado el: 9 de febrero de 2010.]  
[http://es.wikipedia.org/wiki/Lenguaje\\_Unificado\\_de\\_Modelado](http://es.wikipedia.org/wiki/Lenguaje_Unificado_de_Modelado).
9. Requisitos Funcionales y no Funcionales. [En línea] [Citado el: 30 de febrero de 2010.]  
<http://synergix.wordpress.com/2008/07/07/requisito-funcional-y-no-funcional/>
10. RUP. [En línea] [Citado el: 9 de febrero de 2010.] [www.slideshare.net/dersteppenwolf/la-ingeniera-de-software-y-rup](http://www.slideshare.net/dersteppenwolf/la-ingeniera-de-software-y-rup)
11. Tutorial de Django. [En línea] [Citado el: 9 de febrero de 2010.]  
<http://davidasorey.net/static/django-tutorial/>
12. Tutorial de Python. [En línea] [Citado el: 9 de febrero de 2010.]  
<http://lateral.netmanagers.com.ar/static/tutorial-5.pdf>
13. Universidad de Alcalá. [En línea] [Citado el: 9 de febrero de 2010.]  
[http://www.uah.es/empresariales/facultad/documentos/sistema\\_calidad/procesos\\_apoyo/PA-04.pdf](http://www.uah.es/empresariales/facultad/documentos/sistema_calidad/procesos_apoyo/PA-04.pdf).
14. Universidad de Jaén. [En línea] [Citado el: 9 de febrero de 2010.]  
<http://www.ujaen.es/centros/facsoc/SGIC/PROC.%20DE%20APOYO/PA06grm.pdf>.



- 
15. Universidad de Murcia. [En línea] [Citado el: 9 de febrero de 2010.] [http://www.um.es/f-bellasartes/sgic/pa/pa\\_bbaa\\_06\\_grm.pdf](http://www.um.es/f-bellasartes/sgic/pa/pa_bbaa_06_grm.pdf)



## GLOSARIO

**Aplicación o Sistema Informático:** Programas con los cuales el usuario final interactúa a través de una interfaz y que realizan tareas útiles para éste.

**Informática:** Disciplina que estudia el tratamiento automático de la información utilizando dispositivos electrónicos y sistemas computacionales.

**Informatizar:** Proceso de aplicar sistemas o equipos informáticos al tratamiento de la información.

**Software:** Conjunto de programas y procedimientos necesarios para hacer posible la realización de una tarea específica, en contraposición a los componentes físicos del sistema.

**Caso de Uso:** Fragmentos de funcionalidad que el sistema ofrece para aportar un resultado de valor para sus actores.

.