Universidad de las Ciencias Informáticas
Facultad 8

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

Título: Sistema para la Ingeniería y Gestión de Software (SIGS).

Autores: Raidel Berrillo González
Danyer Fidel Arias Acosta

Tutor: Lic. Surelys Veunes Pérez

Ciudad de la Habana
Junio/2008

Declaración de Autoría

Declaramos que Raidel Berrillo Gonzalez y Danyer Fidel Arias Acosta somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas (UCI) y a la Facultad (8) para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmamos la presente a los ____ días del mes de junio del 2008.

_____	_____
Firma del Autor	**Firma del Autor**
Raidel Berrillo Gonzalez	Danyer Fidel Arias Acosta

_____

Firma del Tutor

Lic. Surelys Veunes Pérez

Opinión del Tutor del Trabajo de Diploma

Titulo: Sistema para la Ingeniería y Gestión de Software.

Autores: Danyer Fidel Arias Acosta

Raidel Berrillo González

La tutora del presente Trabajo de Diploma considera que durante su ejecución los diplomantes mostraron las cualidades que a continuación se detallan:

Los diplomantes demostraron muy alta independencia y laboriosidad en el desarrollo de su trabajo. A pesar de trazarse objetivos ambiciosos lograron un producto con un elevado nivel de terminación y detalle, solo alcanzable mediante un arduo trabajo investigativo y un nivel de conocimientos consolidado en la labor sistemática y responsable que llevaron a cabo. Demostraron su originalidad y creatividad en el desarrollo de su trabajo. Así consta en su documento de tesis, en el que se abordan temas novedosos, con una adecuada organización y uso correcto del vocabulario técnico. El documento presenta además un lenguaje claro y sencillo que aborda las principales temáticas relacionadas con el tema de investigación. La bibliografía consultada fue abundante y actualizada.

Este trabajo además tiene un aporte importante para el desarrollo de proyectos productivos en la facultad 8 pues permite llevar a cabo la planificación de los mismos, la elaboración de su expediente y la documentación correspondiente a todas sus fases de forma integrada. Podría hacerse extensible además a otras facultades de la UCI, a otras universidades y centros productivos del país.

Por todo lo anteriormente expresado considero que los estudiantes están aptos para ejercer como Ingeniero en Ciencias Informáticas; y propongo que se le otorgue al Trabajo de Diploma la calificación de _____.

_____ _____

Lic. Surelys Veunes Pérez Fecha

Pensamiento

Todos queremos desarrollar un software que haga bien las cosas, evitando que esas cosas malas merodeen por las sombras de los esfuerzos fracasados. Para tener éxito al diseñar y construir un software necesitaremos disciplina.

Roger S. Pressman

Agradecimientos

A la Revolución cubana, a la Universidad de las Ciencias Informáticas y principalmente a nuestro Comandante Fidel por darnos la oportunidad de formarnos como profesionales.

A la tutora de este trabajo, la Lic. Surelys Veunes Pérez por sus orientaciones y su ayuda para que fuera posible el desarrollo de este trabajo.

A nuestros padres por la confianza que siempre han depositado.

A nuestros hermanos y hermanas por todo el cariño que nos han brindado.

A toda la familia, tíos, primos, abuelos, a los familiares presentes y no presentes, por su apoyo.

A todas las amistades con la que hemos compartido estos intensos 5 años de carrera por su compañerismo.

A Daymi y Susel por habernos ayudado incondicionalmente y haber tenido la paciencia por las tantas horas de ausencia.

Dedicatoria

Raidel:

Para aquellas personas que ocuparon su lugar en mi corazón.

A mis padres, por confiar siempre en mí y ofrecerme todo el apoyo, cariño y amor que necesitaba en todo momento.

A mis hermanos por dejarme ser su guía, por ser lo que son hoy en día y por el hecho de que los quiero mucho.

A toda mi familia por darme ese estimulo de continuar siempre adelante y sobrepasar todo obstáculo que se me presente.

A mi futura esposa Susel por haberme ayudado desde que nos conocimos, y haber soportado mi ausencia por el desarrollo de este trabajo

Danyer:

A quienes me enseñaron las cosas más importantes de la vida.

A mis padres, por brindarme todo lo que estuvo a su alcance y mucho más, por todo su amor y cariño, por haber hecho este sueño realidad.

A mi hermana por estar siempre a mi lado y apoyándome.

A mi abuela y bisabuela por permanecer siempre a mi lado y brindarme mucho apoyo en todos estos años de mi vida.

A mi amor Daymi por su apoyo incondicional y estar siempre a mi lado en todos estos año de mi carrera.

A todas las personas que han influido de una forma u otra a que lograra este sueño de joven universitario.

Resumen

En este trabajo se aborda uno de los problemas que existe en el proceso de desarrollo y gestión de software en la Universidad de Ciencias Informáticas, así como su posible solución mediante la creación de una herramienta de integración.

Durante el proceso investigativo se llega a la conclusión de la posible solución mediante una herramienta que integre los procesos de Ingeniería y Gestión de Software, la cual fue desarrollada sobre tecnología web, utilizando como lenguaje de programación del lado del servidor PHP 5, como Sistema Gestor de Bases de Datos PostgreSQL 8.3 y la técnica de AJAX; todo bajo los estándares de la W3C.

El resultado del presente trabajo proporciona un grupo de mejoras, fundamentalmente en cuanto a la calidad del desarrollo de un software, así como una mejor organización de todos los procesos inmersos en el mismo

Índice

Introducción

Con la Era Digital y el auge alcanzado por el mercado de productos informáticos a nivel mundial, así como la dependencia creada por las diferentes ramas de la sociedad hacia esos productos y servicios, se ha hecho necesaria la creación de herramientas y metodologías que faciliten la gestión y el desarrollo de soluciones de software a la medida. De manera general, las herramientas existentes se especializan solo en uno de los campos del desarrollo del software (*ingeniería de software (ISW)* o *gestión de software (GS))*; como ejemplos de estas se pueden ver el DotProject, utilizado para la planificación de proyectos, las herramientas CASE, empleadas para la realización de diagramas pertenecientes a las diferentes actividades dentro de la ISW, entre otras.

En los procesos de desarrollo y gestión de software existen algunas fallas debido, entre otros factores, a la falta de un análisis en profundidad de los mismos, así como la omisión de ciertos pasos, que son subestimados sin tener en cuenta la magnitud de su importancia. Aún así la falta de integración de ambos procesos en una única herramienta que permita a un tiempo desarrollar las diferentes metodologías dentro de la ISW y gestionar el tiempo, el alcance y el costo de un producto, afecta considerablemente la fluidez del proceso.

La Universidad de las Ciencias Informáticas (UCI), surgida al fragor de la Batalla de Ideas, está a la vanguardia de la producción de software en Cuba. En este centro se emplean las metodologías y herramientas existentes en el mundo para el desarrollo del software; de ahí que se presenten los fallos mencionados anteriormente.

A partir de esta situación surge el **problema científico**:

Los procesos de desarrollo ISW y GS carecen de una herramienta que permita la integración de ambos, lo que conlleva a que la planificación y organización del ciclo de vida del software no presente en muchas ocasiones la calidad requerida.

El **objeto** de esta investigación es:

- Las herramientas relacionadas al proceso de desarrollo y gestión de software.

Dicho objeto enmarca el **campo de acción** de esta investigación:

- Las herramientas relacionadas a los procesos de desarrollo, gestión de configuración y gestión de proyecto en la UCI.

El **objetivo** principal de esta investigación es:

- Desarrollar una herramienta que permita la integración de los procesos de ISW, *gestión de configuración (GC)* y *gestión de proyecto (GP)*.

Los **objetivos específicos** que se persiguen son:

1. Elaborar el Marco teórico conceptual.
2. Realizar un diagnóstico sobre los procesos de gestión de software en la UCI.
3. Realizar análisis, diseño, implementación y prueba del sistema para lograr la integración de ambos procesos.

Idea a defender:

- Si se crea una herramienta que integre los procesos de ISW, GC y GP, mejora la calidad de la planificación, del control de versiones y organización del ciclo de vida del software.

Para lograr el objetivo propuesto se han trazado una serie de **tareas** a realizar:

- Investigar las distintas vías de integración de los procesos de ISW, GC y GP.
- Indagar cómo se realizan los procesos de gestión de configuración y gestión de proyecto en la UCI.
- Estudiar las distintas fases, flujos y artefactos que genera la metodología Proceso Unificado Racional (RUP) en la UCI.
- Estudiar las vías de organización y administración del ciclo de vida del software.
- Realizar un análisis sobre las utilidades de las herramientas existentes para la ISW y GS en pos de

una posible integración de las mismas.

- Estudiar las condiciones tecnológicas disponibles en la UCI y en el mundo para el desarrollo de la aplicación.
- Seleccionar las tecnologías a utilizar de acuerdo a la tarea anterior.
- Realizar el modelo del negocio y levantamiento de requisitos.
- Elaborar el modelo de análisis y diseño del sistema.
- Efectuar la implementación y prueba de la aplicación.

Esta investigación se lleva a cabo empleando métodos científicos de investigación, el primero de ellos, el histórico; donde se analiza la trayectoria completa del fenómeno, su condicionamiento a las diferentes prioridades de la historia, revela las etapas fundamentales de su desenvolvimiento y las conexiones históricas fundamentales; además se emplea como otro método la entrevista; donde se realiza una conversación planificada con el fin de obtener información individual o colectiva.

Este trabajo cuenta con 4 capítulos:

En el capítulo 1 se hace referencia a la fundamentación teórica, donde se plantea la situación existente con la integración de los procesos de ISW, GC y GP en una única herramienta, así como las tecnologías, herramientas y lenguajes de programación a emplear durante la construcción de la aplicación. Se fundamenta la metodología de desarrollo de software utilizada para guiar la investigación.

En el capítulo 2 se explican los procesos del negocio a través de un modelo de dominio, y a partir de esto se comienza a hacer el análisis del sistema a desarrollar. Además se identifican y refinan los requisitos funcionales definidos, los cuales están implícitos en los casos de uso del sistema, y se describen detalladamente.

En el capítulo 3 se muestran los diagramas de clases del análisis, del diseño web para cada realización de caso de uso y el diagrama de clases persistente con su modelo de datos. Además se describen las clases utilizadas en el diseño.

En el capítulo 4 se muestran los diagramas de componentes que representan las dependencias entre los ficheros que integran el subsistema en su conjunto y por paquetes de componentes, así como un diagrama de despliegue que identifica la estructura física con la que contará el subsistema una vez llegado el proceso de instalación del producto. También, se describen las pruebas realizadas al software, como las pruebas de caja negra.

Capítulo 1

Fundamentación Teórica

1.1 Introducción

Hoy en día la informática va evolucionando considerablemente y junto con ella la producción de los diversos software. El siguiente capítulo es el resultado de una detallada y profunda investigación acerca de los conceptos esenciales asociados al entorno de nuestro problema, de las herramientas existentes relacionadas al campo de acción, las diversas y nuevas tecnologías de desarrollo utilizado para darle solución al problema planteado.

1.2 Conceptos Fundamentales

La *ingeniería de software (ISW)* afecta a la economía y las sociedades de muchas maneras. Dicho término se refiere al área de las ciencias de la computación que trata con la construcción de sistemas de software, los cuales son tan grandes y complejos que se construyen con equipos de ingenieros. Es un proceso definido paso a paso, que facilita la especificación, el diseño, la implementación y las pruebas de una solución de software, para un conjunto de requisitos explícitos, de modo eficiente y eficaz. Esto requiere que antes de empezar se tenga: objetivos claros, planes para lograr los objetivos, procedimientos que implementan los planes, un ambiente conducente al logro de los objetivos. [1]

La *gestión de software(GS)* es la disciplina de organizar y administrar recursos de manera tal que se pueda culminar todo el trabajo requerido en el proyecto dentro del alcance, el tiempo, y coste definidos; garantizando la calidad del producto. La GS comprende un conjunto de gestiones, en los cuales se encuentra la gestión de configuración y la gestión de proyecto.

La *gestión de configuración (GC)* es la administración de los recursos del desarrollo del *sistema de software (SS)*, y se realiza durante todas las etapas del proyecto. Sirve para controlar la evolución de

dicho SS. El objetivo de las actividades de gestión de configuración del software es establecer y mantener la integridad de los productos generados durante un proyecto de desarrollo de software y a lo largo de todo el ciclo de vida del producto. [2]

La *gestión de proyecto (GP)* es el proceso por el cual se planifica, dirige y controla el desarrollo de un sistema aceptable con un costo mínimo y dentro de un período de tiempo especifico. [3]

Herramientas CASE

CASE son las siglas que corresponden a: Computer Aided Software Engineering; y en su traducción al Español significa Ingeniería de Software Asistida por Computadora. El concepto de CASE es muy amplio; y una buena definición genérica, que pueda abarcar esa amplitud de conceptos, sería la de considerar a la Ingeniería de Software Asistida por Computadora (CASE), como la aplicación de métodos y técnicas, las cuales son útiles a las personas para comprender las capacidades de las computadoras, por medio de programas de procedimientos y su respectiva documentación.

Tipos de herramientas CASE

No existe una única clasificación de herramientas CASE y, en ocasiones, es difícil incluirlas en una clase determinada. Podrían clasificarse atendiendo a:

1. Las plataformas que soportan.

2. Las fases del ciclo de vida del desarrollo de sistemas que cubren.

3. La arquitectura de las aplicaciones que producen.

4. Su funcionalidad.

Las herramientas CASE en función de las fases del ciclo de vida que abarcan, se pueden agrupar de la forma siguiente:

- Herramientas integradas, I-CASE (Integrated CASE, CASE integrado): abarcan todas las fases del

ciclo de vida del desarrollo de sistemas. Son llamadas también CASE workbench.

Herramienta(s) que comprende(n) alguna(s) fase(s) del ciclo de vida de desarrollo de software:

- Herramientas de alto nivel, U-CASE (Upper CASE - CASE superior o front-end) orientadas a la automatización y soporte de las actividades desarrolladas durante las primeras fases del desarrollo: análisis y diseño.

- Herramientas de bajo nivel, L-CASE (Lower CASE - CASE inferior o back-end) dirigidas a las últimas fases del desarrollo: desarrollo e implantación.

- Juegos de herramientas o toolkits: son el tipo más simple de herramientas CASE. Automatizan una fase dentro del ciclo de vida. Dentro de este grupo se encontrarían las herramientas de reingeniería, orientadas a la fase de mantenimiento.

Las herramientas I-CASE se basan generalmente en una metodología. Tienen un repositorio (BD del proyecto) y aportan técnicas para todas las fases del ciclo de vida. Sin embargo, no todas ellas son modernas en el sentido de aprovechar la potencia de las estaciones de trabajo, la utilización de lenguajes de alto nivel o técnicas de construcción de prototipos.

Una alternativa posible a los I-CASE es utilizar una U-CASE para análisis y diseño, combinada con otras herramientas más modernas para las fases de desarrollo y pruebas. En este caso, habría que vigilar cuidadosamente la integración entre las distintas herramientas.

Otra posible clasificación, utilizando la funcionalidad como criterio principal, es la siguiente:

- Herramientas de planificación de sistemas de gestión: sirven para modelar los requisitos de información estratégica de una organización. Proporcionan un "metamodelo" del cual se pueden obtener sistemas de información específicos. Su objetivo principal es ayudar a comprender mejor cómo se mueve la información entre las distintas unidades organizativas. Estas herramientas proporcionan una ayuda importante cuando se diseñan nuevas estrategias para los sistemas de información y cuando los métodos y sistemas actuales no satisfacen las necesidades de la

organización.

- Herramientas de Análisis y Diseño: permiten al desarrollador crear un modelo del sistema que se va a construir y también la evaluación de la validez y consistencia de este modelo. Proporcionan un grado de confianza en la representación del análisis y ayudan a eliminar errores con anticipación. Entre ellas podemos encontrar:

    1. Herramientas de análisis y diseño (Modelado).

    2. Herramientas de creación de prototipos y de simulación.

    3. Herramientas para el diseño y desarrollo de interfaces.

- Herramientas de programación: se engloban aquí los compiladores, los editores y los depuradores de los lenguajes de programación convencionales. Ejemplos de estas herramientas son:

    1. Herramientas de codificación convencionales.

    2. Herramientas de codificación de cuarta generación (asociadas a SGBD)

    3. Herramientas de programación orientadas a objetos.

- Herramientas de integración y prueba: sirven de ayuda a la adquisición, medición, simulación y prueba de los equipos lógicos desarrollados. Entre las más utilizadas están:

    1. Herramientas de análisis estático.

    2. Herramientas de generación de casos de prueba.

- Herramientas de gestión de prototipos: los prototipos son utilizados ampliamente en el desarrollo de aplicaciones, para la evaluación de especificaciones de un sistema de información, o para un mejor entendimiento de cómo los requisitos de un sistema de información se ajustan a los objetivos perseguidos.

- Herramientas de mantenimiento: la categoría de herramientas de mantenimiento se puede subdividir en:

    1. Herramientas de Ingeniería Inversa.

    2. Herramientas de reestructuración y análisis de código.

    3. Herramientas de reingeniería.

- Herramientas de gestión de proyectos: la mayoría de las herramientas CASE de gestión de proyectos se centran en un elemento específico de la gestión del proyecto, en lugar de proporcionar un soporte global para la actividad de gestión. Utilizando un conjunto seleccionado de las mismas se puede: realizar estimaciones de esfuerzo, coste y duración, hacer un seguimiento continuo del proyecto, estimar la productividad y la calidad, etc. Existen también herramientas que permiten realizar un seguimiento que va desde los requisitos de condiciones técnicas inicial, hasta el trabajo de desarrollo que convierte estos requisitos en un producto final. Se incluyen dentro de las herramientas de control de proyectos las siguientes:

    1. Herramientas de planificación de proyectos.

    2. Herramientas de seguimiento de requisitos.

    3. Herramientas de gestión y medida.

- Herramientas de soporte: se engloban en esta categoría las herramientas que recogen las actividades aplicables en todo el proceso de desarrollo, como las que se relacionan a continuación:

    1. Herramientas de documentación.

    2. Herramientas para software de sistemas.

    3. Herramientas de control de calidad.

4. Herramientas de bases de datos.

1.2.1 Antecedentes

Actualmente existen en el mundo diversas aplicaciones que ayudan a desarrollar productos de software mediante procesos de desarrollo, algunos están más bien vinculados a los procesos de GS o a los procesos de ISW. De ellos podemos citar a:

Dotproject: Fue creado con el fin de de construir una herramienta para la Gestión de Proyectos. Aplicación web desarrollada en PHP y MySQL, permite la administración de usuarios, sistemas de tickets por email, administración de clientes y empresas, listado de proyectos, listado de tareas, lista de contactos. Presenta una interfaz de usuario simple, clara y consistente. Es un software distribuido y desarrollado libremente.[4]

Gforge: Es una herramienta para el desarrollo de software en forma comunitaria que permite organizar y administrar grandes cantidades de proyectos, proporciona un conjunto integrado de herramientas que facilitan el trabajo en colaboración, y en concreto, la gestión de proyectos de software libre. Este contiene:

- Foros.
- Herramientas de gestión y monitorización de tareas.
- Gestión de errores y mejoras.
- Lista de correos.
- Herramientas de control de versiones de software.
- Manejo de proyectos.
- Manejo de usuarios.

Es una plataforma para el desarrollo de aplicaciones orientada a grupos de trabajo; es el mayor sitio web de desarrollo de software, en el cual desarrolladores de software libre pueden acceder a diversos servicios:

- Proporciona un entorno configurable para el control de versiones.
- Herramientas para comunicación entre desarrollos y servidor web por proyecto.
- Permite desarrollar una base de conocimiento compartida del proyecto.[5]

Open Workbench: Es una aplicación de código abierto que proporciona un robusto sistema de planificación y funcionalidades de gestión. Es una alternativa gratuita a Microsoft Project. Ofrece un avanzado sistema de planificación de proyectos, con tareas y con una eficaz estructura de trabajo. Una vez creado el proyecto, se podrá, asignar recursos y utilizar herramientas de planificación, ejecución, control y seguimiento.[6]

Gantt Project: Software de escritorio, desarrollado en Java, divide el proyecto en un árbol de tareas y asigna recursos humanos a cada una de ellas, establece dependencias entre tareas, genera informes PDF y HTML, intercambia datos con MS Project y hojas de cálculo. Es una herramienta gratuita para crear una completa planificación de un proyecto de forma visual. Establece dependencias entre las tareas, de esta forma, una tarea no podrá empezar hasta que la anterior culmine. Permite exportar el trabajo a una imagen (JPG, PNG), PDF y HTML.[7]

MS Project: Microsoft Project es un potente programa propietario de gestión de proyectos que se utiliza y demanda cada vez más por parte de las empresas para crear planes de proyecto, introducir datos reales de evolución y realizar un completo seguimiento, así como contabilizar la variación que se produce en el transcurso de un proyecto respecto a lo que inicialmente se había programado (línea base). Con este programa se gestionan y controlan tanto las tareas que componen un proyecto, como los recursos que se utilizan para su desarrollo y las asignaciones recurso-tarea.[8]

Visual Paradigm: Es una herramienta CASE que utiliza “UML”: como lenguaje de modelado. Se integra con las siguientes herramientas Java:

- Eclipse/IBM WebSphere
- JBuilder

- NetBeans IDE
- Oracle JDeveloper
- BEA Weblogic

Permite realizar ingeniería tanto directa como inversa, es capaz de desplegar todas las clases asociadas a las tablas (siguiendo el patrón de diseño Una Clase-Una Tabla). Para gestionar la persistencia y el mapeo de estas clases con la base de datos utiliza Hibernate para Java y NHibernate para .Net. Además, la herramienta es colaborativa, es decir, soporta múltiples usuarios trabajando sobre el mismo proyecto; genera la documentación del proyecto automáticamente en varios formatos como Web o PDF, y permite realizar control de versiones. Cabe destacar igualmente su robustez, usabilidad y portabilidad.

Trac

Trac es una aplicación para la gestión de proyectos de software. Está desarrollada principalmente en Python, por tanto funciona en cualquier plataforma con interprete python, por ejemplo GNU/Linux, MacOS X, BSD o Windows.

Ofrece una visión muy abierta de la gestión de proyectos, no obliga a utilizar una metodología determinada. Para ello proporciona tres elementos básicos: un gestor de hitos, un navegador de svn (control de versiones) y un sistema de control de tareas/bugs. Además, permite una visión en línea de tiempo de los eventos que van ocurriendo en el desarrollo. Todo ello a través de una interfaz web con edición tipo wiki, aumentando la ubicuidad del acceso.

En trac, a diferencia de otras plataformas de gestión de proyectos, no se lleva una contabilidad tan fina del tiempo y el progreso sino que se persigue una mayor naturalidad para el desarrollador. Su función podría resumirse simplemente como una aplicación de gestión de tareas que permite cruzar los datos de las tareas con las inserciones en el sistema de control de versiones.

1.3 Lenguajes de Programación para la Web.

Con la aparición de Internet, la programación web ha adquirido disímiles avances, producto al desarrollo

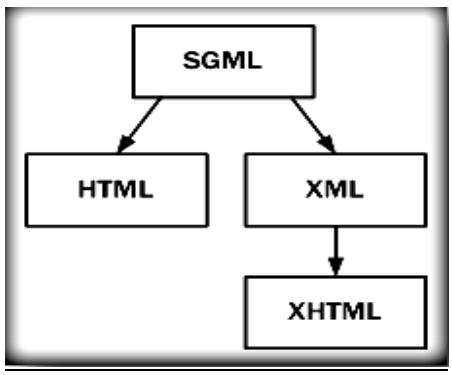
de los lenguajes de programación para la web que facilitaron y propiciaron un desarrollo vertiginoso hasta nuestros días. Dichos lenguajes se clasifican en dos partes fundamentales que reconocen la propia arquitectura Cliente/Servidor de esta plataforma de desarrollo: los lenguajes del lado del Servidor y los lenguajes del lado del Cliente.

Entre los lenguajes del lado del servidor se encuentran PHP, ASP, Perl, JSP, C++, Python, entre otros. Estos se caracterizan por desarrollar la lógica de negocio dentro del Servidor, además de ser los encargados del acceso a bases de datos, tratamiento de la información etc. Del lado del cliente se encuentran XHTML, AJAX, CSS, JScript que son totalmente independientes del servidor, son aquellos que pueden ser directamente "ejecutados" por el navegador y no necesitan un pre-tratamiento.

Esta distinción en los lenguajes ha sido necesaria debido a que la Web funciona en modo “desconectado”, o sea, un usuario a través de un navegador hace una petición de una página web a un servidor web (request), el servidor recepciona la petición, la procesa y le envía la respuesta al cliente (response), este la recepciona y se desconecta.

XHTML (e**X**tensible **H**yper**T**ext **M**arkup **L**anguage**)**

El lenguaje XHTML es muy similar al lenguaje HTML. De hecho, XHTML no es más que una adaptación de HTML al lenguaje XML. Técnicamente, HTML es descendiente directo del lenguaje SGML, mientras que XHTML lo es del XML (que a su vez, también es descendiente de SGML). [9]



CSS (Cascading Style Sheets**)**

CSS es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas. Mientras que el lenguaje HTML/XHTML se utiliza para marcar los contenidos, es decir, para designar lo que es un párrafo, lo que es un título o lo que es una lista de elementos, el lenguaje CSS se utiliza para definir el aspecto de todos los contenidos, es decir, el color, tamaño y tipo de letra de los párrafos de texto, la separación entre titulares y párrafos, la tabulación con la que se muestran los elementos de una lista, etc.[10]

Javascript

Javascript es un lenguaje con muchas posibilidades, utilizado para crear pequeños programas que luego son insertados en una página web y en programas más grandes, orientados a objetos mucho más complejos. Con Javascript podemos crear diferentes efectos e interactuar con los usuarios. Este lenguaje posee varias características, entre ellas se puede mencionar que es un lenguaje basado en acciones que posee menos restricciones. Además, utiliza Windows y sistemas X-Windows, gran parte de la programación en este lenguaje está centrada en describir objetos, escribir funciones que respondan a movimientos del mouse, aperturas, utilización de teclas, cargas de páginas entre otros. Es necesario resaltar que hay dos tipos de JavaScript: por un lado está el que se ejecuta en el cliente, este es el Javascript propiamente dicho, aunque técnicamente se denomina Navigator JavaScript. Pero también existe un Javascript que se ejecuta en el servidor, es más reciente y se denomina LiveWire Javascript. [11]

ASP (Pagina Activa del Servidor**)**

El ASP es una tecnología dinámica funcionando del lado del servidor, lo que significa que cuando el usuario solicita un documento ASP, las instrucciones de programación dentro del script son ejecutadas para enviar al navegador únicamente el código HTML resultante. La ventaja principal de las tecnologías dependientes del servidor radica en la seguridad que tiene el programador sobre su código, ya que éste se encuentra únicamente en los archivos del servidor que al ser solicitado a través de la web, es ejecutado, por lo que los usuarios no tienen acceso más que a la página resultante en su navegador.[12]

PERL

Es un lenguaje de programación muy utilizado para construir aplicaciones CGI para la web. Perl es un

acrónimo de Practical Extracting and Reporting Languaje, que viene a indicar que se trata de un lenguaje de programación muy práctico para extraer información de archivos de texto y generar informes a partir del contendido de los ficheros. Es un lenguaje de uso libre. Antes estaba muy asociado a la plataforma Uníx, pero en la actualidad está disponible en otros sistemas operativos como Windows.[13]

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

JSP (JavaServer Pages)

Es una tecnología orientada a crear páginas web con programación en Java. Con JSP se pueden crear aplicaciones web que se ejecuten en variados servidores web de múltiples plataformas, ya que Java es en esencia un lenguaje multiplataforma. Las páginas JSP están compuestas de código HTML/XML mezclado con etiquetas especiales para programar scripts de servidor en sintaxis Java. El motor de las páginas JSP está basado en los servlets de Java, programas en Java destinados a ejecutarse en el servidor, aunque el número de desarrolladores que pueden afrontar la programación de JSP es mucho mayor, dado que resulta mucho más sencillo aprender que los servlets.[14]

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, esto ofrece ventajas como la rapidez de desarrollo e inconvenientes como una menor velocidad.[15]

1.4 Sistemas de Gestión de Bases de Datos (SGBD).

Los sistemas de bases de datos están diseñados para gestionar grandes volúmenes de información. Generalmente, las bases de datos requieren gran cantidad de espacio de almacenamiento, por lo que las bases de datos de las organizaciones se miden en términos de gigabytes o terabytes de datos. Un gigabyte equivale a 1000 megabytes (un billón de bytes), y un terabyte equivale a un millón de megabytes (un trillón de bytes). Un sistema de bases de datos tiene como objetivo simplificar y facilitar el acceso a los datos y hacer que los tiempos de respuesta a las solicitudes de los usuarios sean muy reducidos.

De forma sencilla, un sistema de gestión de bases de datos se puede definir como una colección de datos interrelacionados y un conjunto de programas para acceder a esos datos. Adoración de Miguel lo define como conjunto coordinado de programas, procedimientos, lenguajes, etc. que suministra, tanto a los usuarios no informáticos como a los analistas, programadores o al administrador, los medios necesarios para describir, recuperar y manipular los datos almacenados en la base, manteniendo su integridad, confidencialidad y seguridad.[16]

Microsoft SQL Server

Microsoft SQL Server es un sistema de gestión de bases de datos relacionales (SGBD) basado en el lenguaje Transact-SQL, capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea. Constituye la alternativa de Microsoft a otros potentes sistemas gestores de bases de datos como son Oracle, Sybase ASE, PostgreSQL o MySQL.[17]

Oracle

Es un sistema de gestión de base de datos relacional (o RDBMS por el acrónimo en inglés de Relational Data Base Management System), fabricado por Oracle Corporation. Se considera a Oracle como uno de los sistemas de bases de datos más completos, destacando su:

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

Ha sido criticada por algunos especialistas la seguridad de la plataforma, y las políticas de suministro de parches de seguridad, modificadas a comienzos de 2005 y que incrementan el nivel de exposición de los usuarios. En los parches de actualización provistos durante el primer semestre de 2005 fueron corregidas 22 vulnerabilidades públicamente conocidas, algunas de ellas con una antigüedad de más de 2 años.

Aunque su dominio en el mercado de servidores empresariales ha sido casi total hasta hace poco, recientemente sufre la competencia del Microsoft SQL Server de Microsoft y de la oferta de otros RDBMS con licencia libre como PostgreSQL, MySQL o Firebird. Las últimas versiones de Oracle han sido certificadas para poder trabajar bajo Linux.[18]

MySQL

Es un sistema de gestión de base de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. MySQL AB desarrolla MySQL como software libre en un esquema de licenciamiento dual. MySQL AB pertenece a Sun Microsystems desde enero de 2008.

Por un lado se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia, pero las empresas que quieran incorporarlo en productos privativos pueden comprar a la empresa una licencia específica que les permita este uso. Está desarrollado en su mayor parte en ANSI C.

Al contrario de proyectos como Apache, donde el software es desarrollado por una comunidad pública y el copyright del código está en poder del autor individual, MySQL es propiedad y está patrocinado por una empresa privada, que posee el copyright de la mayor parte del código.[19]

1.5 Servidores para aplicaciones web

Internet Information Services (IIS**)**

Es una serie de servicios para los ordenadores que funcionan con Windows. Originalmente era parte del Option Pack para Windows NT. Luego fue integrado en otros sistemas operativos de Microsoft destinados a ofrecer servicios, como Windows 2000 o Windows Server 2003. Windows XP Profesional incluye una versión limitada de IIS. Los servicios que ofrece son: FTP, SMTP, NNTP y HTTP/HTTPS.

Este servicio convierte a un ordenador en un servidor de Internet o Intranet es decir que en las computadoras que tienen este servicio instalado se pueden publicar páginas web tanto local como remotamente (servidor web).[20]

Zope

Es un servidor de aplicaciones web de código abierto escrito en el lenguaje de programación Python. Puede ser manejado casi totalmente usando una interfaz de usuario basada en páginas web.

Un sitio web de Zope está compuesto de objetos en lugar de archivos, como es usual con la mayoría de los otros sistemas de servidores web. Las ventajas de usar objetos en lugar de archivos son:

- Combinan el comportamiento y los datos en una forma más natural que los archivos de texto plano.

- Alientan el uso de componentes estándares que se ocupan de una parte particular de las que forman

una aplicación Web, permitiendo flexibilidad y buena descomposición.

- Posibilitan procesos automáticos de gestión de información.

Lo más característico de Zope es su base de datos orientada a objetos, llamada ZODB o Zope Object Database. Esta base de datos almacena objetos ordenados en un sistema similar a un sistema de ficheros, pero cada objeto tiene propiedades, métodos u otros objetos. Esta aproximación es muy diferente de las bases de datos relacionales habituales. Sin embargo, Zope dispone de múltiples conectores para las diferentes bases de datos relacionales y ofrece sistemas básicos de conexión y consulta abstrayéndolos como objetos.[21]

1.6 Tecnologías a utilizar

Como ***propuesta de solución*** se tiene, desarrollar una aplicación web que gestione lo referente a la IS y GS, así como las fases, flujos y actividades de un proyecto y los artefactos que dentro de estos se encuentran. Las plantillas se podrán generar en formato PDF, para su uso fuera de la aplicación. Para el desarrollo del sistema se investigaron varias tecnologías que se utilizan en el mundo, de las cuales fueron elegidas algunas según sus ventajas y facilidades para dar solución al problema.

Servidor HTTP Apache

El **servidor HTTP Apache** es un servidor de páginas web que permite acceder a las páginas web que están alojadas en una computadora. Es de código abierto y actualmente es un servidor web muy utilizado en el mundo, encontrándose por encima de sus competidores, tanto gratuitos como comerciales. Está diseñado para ser un servidor web potente y flexible que pueda funcionar en la más amplia variedad de plataformas y entornos. Las diferentes plataformas y los diferentes entornos, hacen que a menudo sean necesarias diferentes características o funcionalidades, o que una misma característica o funcionalidad sea implementada de diferente manera para obtener una mayor eficiencia. Apache se ha adaptado siempre a una gran variedad de entornos a través de su diseño modular. Este diseño permite a los

administradores de sitios Web elegir qué funcionalidades van a ser incluidas en el servidor seleccionando que módulos se van a usar, ya sea al compilar o al ejecutar el servidor.

Este servidor tiene capacidad para servir páginas estáticas como dinámicas a través de otras herramientas soportadas que facilitan la actualización de los contenidos usando bases de datos, ficheros u otras fuentes de información.

Ventajas frente a otros servidores:

- Modular
- Multiplataforma
- Software Libre
- Extensible
- Popular
- Gratuito

PHP v5.0 o superior

La iníciales PHP significan "PHP Hypertext Pre-processor" y se trata de un lenguaje de programación que es usado para la creación de aplicaciones para servidores, o creación de contenido dinámico para sitios web. Es un lenguaje de programación usado generalmente para la creación de contenido para sitios o aplicaciones web. La versión 5 de PHP presenta un magnífico trabajo con el Paradigma Orientado a Objetos que permite la reutilización de código entre otras facilidades.

Ventajas de trabajar con PHP comparado con otros lenguajes similares:

- ✓ Es un lenguaje multiplataforma.
- ✓ Rapidez de ejecución.
- ✓ Mantiene un bajo consumo de recursos de máquina.

- ✓ Gran seguridad, muy poca probabilidad de corromper los datos.
- ✓ Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.
- ✓ Posee una amplia documentación en internet, incluyendo una gran variedad de ejemplos y de ayudas.
- ✓ Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- ✓ Permite las técnicas de Programación Orientada a Objetos.
- ✓ Permite crear los formularios para la web.
- ✓ No requiere definición de tipos de variables ni manejo detallado de bajo nivel.

FPDF

FPDF es una clase escrita en PHP que permite generar documentos PDF directamente desde PHP. Sus principales características son:

- ✓ Elección de la unidad de medida, formato de página y márgenes
- ✓ Gestión de cabeceras y pies de página
- ✓ Salto de página automático
- ✓ Salto de línea y justificación del texto automáticos
- ✓ Admisión de imágenes (JPEG y PNG)
- ✓ Mantiene los colores dados por código CSS
- ✓ Enlaces
- ✓ Admisión de fuentes TrueType, Type1 y codificación
- ✓ Compresión de página [22]

JpGraph

Es una librería que incluye una serie de clases código orientado a objetos que sirven para crear imágenes con todo tipo de gráficas, dinámicamente desde páginas PHP.
El sistema está muy depurado y soporta multitud de funcionalidades. Además, la mayoría de las configuraciones de las gráficas vienen con opciones por defecto, así que resulta bastante sencillo obtener resultados rápidamente.

Características

JpGraph es una librería muy completa y entre las características más destacadas se pueden listar:

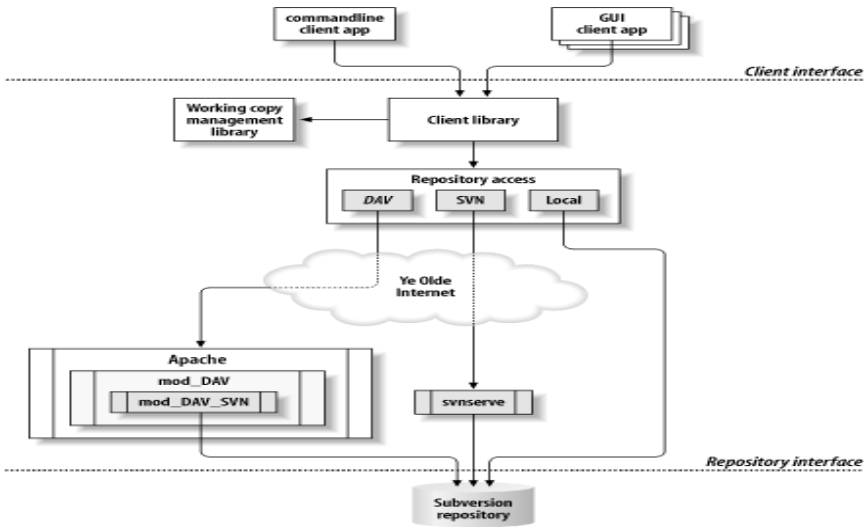
- ✓ Soporte para GD1 y GD2.
- ✓ Soporte para incluir texto a las imágenes y soporte para tipos de letra.
- ✓ Soporte para niveles de transparencia.
- ✓ Soporte para gráficos complejos de Gantt.
- ✓ Manejo de las escalas para los ejes del gráfico.
- ✓ Soporta formatos PNG, GIF y JPG.
- ✓ Soporte para gráficas de barras horizontales.
- ✓ Soporte para gráficas de tipo científico.
- ✓ Soporte para generación de la escala automática dependiendo de los datos.
- ✓ Soporta varios tipos de relleno para las gráficas.
- ✓ Documentación muy amplia y con una referencia completa de las funciones[23]

Subversion

Es un software de sistema de control de versiones diseñado específicamente para reemplazar al popular

software CVS, el cual posee varias deficiencias en cuanto al control de versiones sobre un producto. Es software libre bajo una licencia de tipo Apache/BSD y se le conoce también como svn por ser ese el nombre de la herramienta de línea de comandos. Una característica importante de Subversion es que, a diferencia de CVS, los archivos versionados no tienen cada uno un número de revisión independiente. En cambio, todo el repositorio tiene un único número de versión que identifica un estado común de todos los archivos del repositorio en cierto punto del tiempo. Maneja ficheros y directorios a través del tiempo. Hay un árbol de ficheros en un repositorio central. El repositorio es como un servidor de ficheros ordinario, excepto porque recuerda todos los cambios hechos a sus ficheros y directorios. Esto le permite recuperar versiones antiguas de sus datos, o examinar el historial de cambios de los mismos.

Arquitectura de Subversion



En un extremo se encuentra un repositorio de Subversion que conserva todos los datos versionados. Al otro lado, hay un programa cliente Subversion que administra réplicas parciales de esos datos versionados (llamadas “copias de trabajo”). Entre estos extremos hay múltiples rutas a través de varias capas de acceso al repositorio (AR). Algunas de estas rutas incluyen redes de ordenadores y servidores de red que después acceden al repositorio. Otras pasan por alto la red y acceden al repositorio directamente.[24]

AJAX

AJAX, acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Estas se ejecutan en el cliente, es decir, en el navegador de los usuarios y mantiene comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre la misma página sin necesidad de recargarla. Esto significa aumentar la interactividad, velocidad y usabilidad en la misma.

AJAX es una combinación de tres tecnologías ya existentes:

- XHTML (o HTML) y hojas de estilos en cascada (CSS) para el diseño que acompaña a la información.
- Document Object Model (DOM) accedido con un lenguaje script por parte del usuario, especialmente implementaciones ECMAScript como JavaScript y JScript, para mostrar e interactuar dinámicamente con la información presentada.
- El objeto XMLHttpRequest para intercambiar datos asincrónicamente con el servidor web. En algunos frameworks y en algunas situaciones concretas, se usa un objeto iframe en lugar del XMLHttpRequest para realizar dichos intercambios.
- XML es el formato usado comúnmente para la transferencia de vuelta al servidor, aunque cualquier formato puede funcionar, incluyendo HTML preformateado, texto plano, JSON(JavaScript Object Notation) y hasta EBML(Meta Lenguaje Binario Extendible).

Como el DHTML (HTML Dinámico), LAMP (se refiere a un conjunto de subsistemas software necesarios para alcanzar una solución global, en este caso configurar sitios web o servidores dinámicos con un esfuerzo reducido), AJAX no constituye una tecnología en sí, sino que es un término que engloba a un grupo de éstas que trabajan conjuntamente.

Ventajas:

- Recuperación asíncrona de datos, el usuario no tiene que esperar después de una petición.
- Acercamiento del ambiente de escritorio a la web.

- No requiere plugins(es una aplicación informática que interactúa con otra aplicación para aportarle una función o utilidad específica).
- Se reduce el tamaño de la información intercambiada.

ExtJS

ExtJS empezó siendo un conjunto de librerías y extensiones para YUI (Yahoo User Inteface). Con el tiempo se convirtió en un FrameWork independiente y a principios de 2007 se creo una compañía para comercializar y dar soporte del FrameWork Ext. De esta forma Ext tiene dos tipos de licencias, LGPL y comercial

ExtJS es una librería Javascript para construir aplicaciones ricas en internet. Incluye:

- Alto rendimiento, componentes personalizables en entorno de usuario (UI)
- Modelo de Componentes extensibles
- API fácil de utilizar
- Licencias Comerciales y Código Abierto disponibles[25]

PostgreSQL

Es un Sistema Gestor de Bases de Datos, liberado bajo la licencia BSD o Berkeley Software Distribution. Permite el uso del código fuente en software no libre, posee una serie de características positivas respecto a otros gestores.

1. Gran escalabilidad. Es ajustable al número de procesadores y a la cantidad de memoria que posee el sistema de forma eficiente, por este motivo es capaz de soportar una mayor cantidad de peticiones simultáneas. Teniendo en cuenta esto, es vital en la universidad, ya que no se requiere de un avanzado sistema de cómputo para trabajar con él.
2. Tiene la capacidad de almacenar procedimientos almacenados en la propia base de datos.
3. Multiusuario, con arquitectura cliente-servidor y control de privilegios de acceso.

4. Los tipos internos han sido mejorados, incluyendo nuevos tipos de fecha/hora de rango amplio y soporte para tipos geométricos adicionales.

5. La velocidad del código del motor de datos ha sido incrementada aproximadamente en un 20% - 40%, y su tiempo de arranque ha bajado el 80% desde que la versión 6.0 fue lanzada.

1.7 Arquitectura utilizada

Arquitectura

Una arquitectura es el conjunto de decisiones significativas sobre la organización del SS, la selección de los elementos estructurales y sus interfaces, con los que se compone el sistema, junto con su comportamiento tal como se especifica en las colaboraciones entre esos elementos, la composición de esos elementos estructurales y de comportamiento en subsistemas progresivamente más amplios, y el estilo de arquitectura que guía esta organización, estos elementos y sus interfaces, sus colaboraciones, y su composición. Ej.: los Patrones de diseño relacionados con el diseño de los objetos y frameworks de pequeña y mediana escala, que son aplicables al diseño de una solución para conectar los elementos de gran escala que se definen mediante los patrones de arquitectura, y durante el trabajo de diseño detallado para cualquier aspecto del diseño local. También se conocen como patrones de micro-arquitectura. El patrón Fachada, que se puede utilizar para proporcionar la interfaz de una capa a la siguiente [26].

Patrón de Arquitectura MVC (Modelo Vista Controlador)

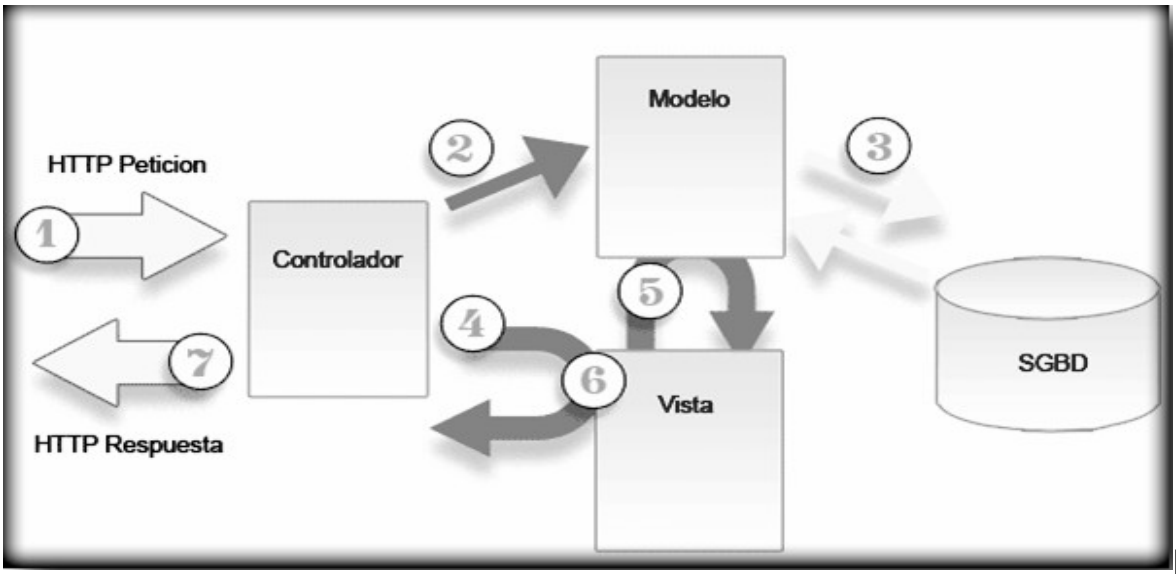
Para el diseño de aplicaciones con sofisticadas interfaces se utiliza el patrón de diseño Modelo-Vista-Controlador. La lógica de una interfaz de usuario cambia con más frecuencia que los almacenes de datos y la lógica de negocio. Si se realiza un diseño ofuscado, es decir, una forma de mezclar los componentes de interfaz y de negocio, entonces la consecuencia será que, cuando se necesite cambiar la interfaz, se tendrá que modificar trabajosamente los componentes de negocio. Mayor trabajo y más riesgo de error.

Se trata de realizar un diseño que desacople la vista del modelo, con la finalidad de mejorar la reusabilidad. De esta forma las modificaciones en las vistas impactan en menor medida en la lógica de

negocio o de datos.

Elementos del patrón:

- Modelo: datos y reglas de negocio
- Vista: muestra la información del modelo al usuario
- Controlador: gestiona las entradas del usuario



[27]

Un modelo puede tener diversas vistas, cada una con su correspondiente controlador. Analizan cada componente:

1. El modelo es el responsable de:

- Acceder a la capa de almacenamiento de datos. Lo ideal es que el modelo sea independiente del sistema de almacenamiento.
- Definir las reglas de negocio (la funcionalidad del sistema).
- Llevar un registro de las vistas y controladores del sistema.

- Si el modelo es activo, notificará a las vistas los cambios que en los datos pueda producir un agente externo (por ejemplo, un fichero .bat que actualiza los datos, un temporizador que desencadena una inserción, etc.).

2. El controlador es responsable de:

- Recibir los eventos de entrada (un clic, un cambio en un campo de texto, etc.).
- Contiene reglas de gestión de eventos, del tipo "SI Evento Z, entonces Acción W". Estas acciones pueden suponer peticiones al modelo o a las vistas. Una de estas peticiones a las vistas puede ser una llamada a actualizar.

3. Las vistas son responsables de:

- Recibir datos del modelo y mostrarlo al usuario.
- Tienen un registro de su controlador asociado (normalmente porque además lo instancia).
- Pueden dar el servicio de actualizar, para que sea invocado por el controlador o por el modelo (cuando es un modelo activo que informa de los cambios en los datos producidos por otros agentes).

1.8 Herramientas utilizadas en la propuesta de solución.

Zend Studio v5.5

Zend Studio 5.5 es un Entorno de Desarrollo Integrado (IDE). Incluye un poderoso editor de código con soporte para todos los formatos de la web. Es un entorno de desarrollo para PHP, que integrando el uso y completado de código personalizado de Zend Framework y vista de la lista de las funciones del framework desde la Visualización de funciones PHP; utiliza una fuerte herramienta para la documentación de código llamada PHPDocumentor , lo cual proporciona que el código resulte más entendible.[28]

Zend Platform

Zend Platform servidor de aplicación para PHP; administración, integración, utilizando PHP para sus aplicaciones de negocios críticas. Mediante la inclusión de capacidades que hacen más eficientes el desarrollo y despliegue, Zend Platform mejora la experiencia del usuario, la respuesta de la aplicación, integración a su infraestructura existente y aumenta la fiabilidad y escalabilidad de la aplicación.[29]

EMS PostgreSQL

EMS SQL Manager para PostgreSQL es una poderosa herramienta gráfica para la administración y desarrollo de PostgreSQL Database Server (servidor de bases de datos PostgreSQL). PostgreSQL Manager funciona con cualquier versión de PostgreSQL, hasta la 8.1, y soporta todas las nuevas características de PostgreSQL incluyendo espacios de tablas (tablespaces), argumentos nombrados (named arguments) en funciones. Ofrece una gran variedad de herramientas poderosas para usuarios avanzados, tales como Visual Database Designer (diseñador visual de base de datos), Visual Query Builder (constructor visual de consultas), y un poderoso editor de objetos binarios (BLOB) para satisfacer todas sus necesidades. [30]

Dreanweaver 8

Dreamweaver 8 es la opción profesional para la creación de sitios y aplicaciones web. Proporciona una combinación potente de herramientas visuales de disposición, permite desarrollo de aplicaciones y soporte para la edición de código. Gracias a las robustas características para la integración y diseño basado en CSS. Incluye potentes controles basados en normas para asegurar un diseño de alta calidad. Un entorno de diseño construido en torno a las hojas de estilo en cascada (CSS) hace posible un desarrollo más rápido y más eficiente de sitios profesionales creados con código limpio. Es un entorno de desarrollo integrado para desarrollar sitios web de HTML, XHTML, XML, ASP, ASP.NET, JSP, PHP y Macromedia ColdFusion.[31]

Fireworks 8

Fireworks es un software práctico, flexible e innovador que incorpora todo tipo de herramientas para el tratamiento de imágenes de mapa de bits y gráficos vectoriales.
Incluye nuevas funciones y mejoras de diseño y desarrollo, máxima integración con la Suite de

Macromedia 8 y otros programas de diseño y editores HTML. Destacan la creación y exportación de menús emergentes utilizando CSS, dando completo control sobre el código sin necesidad de volver a programarlo, mejoras en el flujo de trabajo con los nuevos paneles de acceso rápido de edición de imágenes e inserción de caracteres y nuevos formatos de imagen admitidos para guardar e importar.[32]

Rational Rose Enterprise

Rational Rose Enterprise proporciona un lenguaje de modelado común para permitir la creación más rápida de la calidad de software. Incluye Unified Modeling Language (UML) de apoyo. Proporciona la base de datos para el modelado UML diseños, con la capacidad de representar a la integración de los datos y los requisitos de las aplicaciones a través de diseños físicos y lógicos. Crea definiciones XML de tipo de documento (DTD) para uso en su aplicación. Los sistemas operativos soportados: HP Unix, Linux, Windows. Rational Rose ayuda a las empresas a superar la paradoja de software por la unificación de las comunidades que desarrollan software con un lenguaje visual que acelera la creación de aplicaciones robustas.[33]

1.9 Metodologías de desarrollo de software

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos software. Las técnicas indican cómo debe ser realizada una actividad técnica determinada identificada en la metodología. Se debe tener en consideración que una técnica determinada puede ser utilizada en una o más actividades de la metodología de desarrollo de software. Todo desarrollo de software es riesgoso y difícil de controlar, pero si no lleva una metodología de por medio, lo que se obtiene es clientes insatisfechos con el resultado y desarrolladores aún más insatisfechos .Una metodología puede seguir uno o varios modelos de ciclo de vida, es decir, el ciclo de vida indica qué es lo que hay que obtener a lo largo del desarrollo del proyecto pero no cómo hacerlo. Indica cómo hay que obtener los distintos productos parciales y finales.

A continuación se describen las principales características de tres de las más famosas y conocidas metodologías de desarrollo de software: Proceso Unificado de Racional (Rational Unified Process, RUP), Programación Extrema (Extreme Programming, XP) y Microsoft Solution Framework (MSF).

Rational Unified Process (RUP)

La metodología RUP, llamada así por sus siglas en inglés Rational Unified Process, divide en 4 fases el desarrollo del software:

- Inicio: El Objetivo en esta etapa es determinar la visión del proyecto.
- Elaboración: En esta etapa el objetivo es determinar la arquitectura óptima.
- Construcción: En esta etapa el objetivo es obtener la capacidad operacional inicial.
- Transmisión: El objetivo es llegar a obtener la versión lista para su instalación en las condiciones reales.

Cada una de estas etapas es desarrollada mediante el ciclo de iteraciones, la cual consiste en reproducir el ciclo de vida en cascada a menor escala. Los objetivos de una iteración se establecen en función de la evaluación de las iteraciones precedentes. Vale mencionar que el ciclo de vida que se desarrolla por cada iteración, es llevada bajo dos disciplinas:

Disciplina de Desarrollo

- Ingeniería de Negocios: Entendiendo las necesidades del negocio.
- Requerimientos: Trasladando las necesidades del negocio a un sistema automatizado.
- Análisis y Diseño: Trasladando los requerimientos dentro de la arquitectura de software.
- Implementación: Creando software que se ajuste a la arquitectura y que tenga el comportamiento deseado.
- Pruebas: Asegurándose que el comportamiento requerido es el correcto y que todo lo solicitado está presente.
- Disciplina de Soporte
- Configuración y administración del cambio: Guardando todas las versiones del proyecto.

- Administrando el proyecto: Administrando horarios y recursos.
- Ambiente: Administrando el ambiente de desarrollo.

Distribución: Hacer todo lo necesario para la salida del proyecto

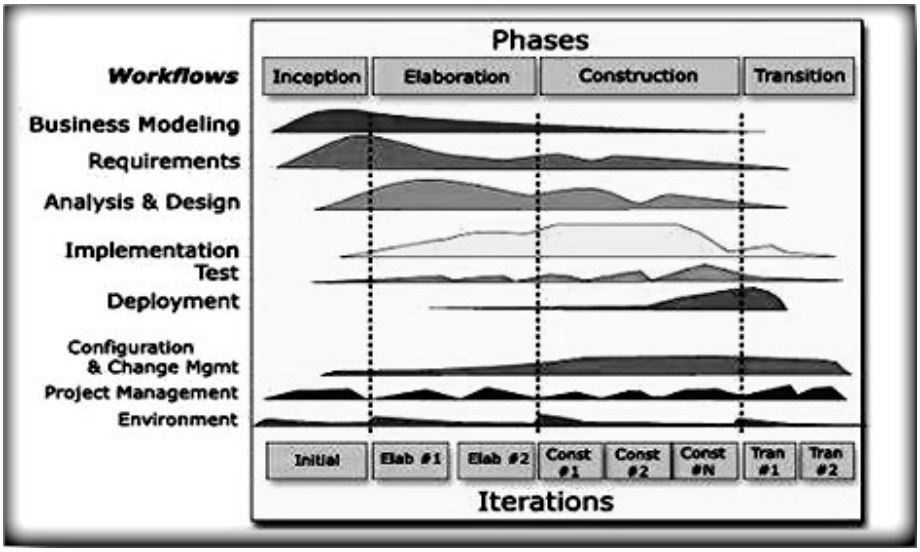


Figura 1: Fases e Iteraciones de la Metodología RUP

Es recomendable que a cada una de estas iteraciones se les clasifique y ordene según su prioridad, y que cada una se convierta luego en un entregable al cliente. Esto trae como beneficio la retroalimentación que se tendrá en cada entregable o en cada iteración.

Los elementos del RUP son:

- Actividades, Son los procesos que se llegan a determinar en cada iteración.
- Trabajadores, Son las personas o entes involucrados en cada proceso.
- Artefactos, Un artefacto puede ser un documento, un modelo, o un elemento de modelo.

Una particularidad de esta metodología es que, en cada ciclo de iteración, se hace necesario 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.

Extreme Programing (XP)

Es una de las metodologías de desarrollo de software más exitosas en la actualidad, utilizadas para proyectos de corto plazo y equipos pequeños. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto.

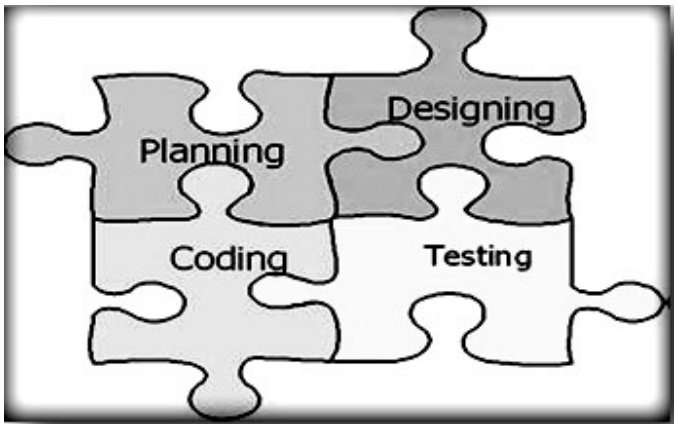


Figura 2: Metodología Extreme Programming

La metodología XP se basa en:

- Pruebas Unitarias: se basa en las pruebas realizadas a los principales procesos, realizando pruebas de las fallas que pudieran ocurrir. Es como si se realizara un adelanto de los posibles errores a obtener.
- Refabricación: se basa en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.
- Programación en pares: una particularidad de esta metodología es que propone la programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento.

¿Qué es lo que propone XP?

- Añade funcionalidad con retroalimentación continua
- El manejo del cambio se convierte en parte sustantiva del proceso
- El costo del cambio no depende de la fase o etapa
- No introduce funcionalidades antes que sean necesarias
- El cliente o el usuario se convierte en miembro del equipo Derechos del Cliente
- Saber el estado real y el progreso del proyecto
- Añadir, cambiar o quitar requerimientos en cualquier momento
- Obtener lo máximo de cada semana de trabajo
- Decidir como se implementan los procesos
- Crear el sistema con la mejor calidad posible
- Pedir al cliente en cualquier momento aclaraciones de los requerimientos
- Estimar el esfuerzo para implementar el sistema
- Cambiar los requerimientos en base a nuevos descubrimientos

Lo fundamental en este tipo de metodología es:

- La comunicación entre los usuarios y los desarrolladores
- La simplicidad al desarrollar y codificar los módulos del sistema
- La retroalimentación concreta y frecuente del equipo de desarrollo, el cliente y los usuarios finales.

Microsoft Solution Framework (MSF)

Esta es una metodología flexible e interrelacionada con una serie de conceptos, modelos y prácticas de uso, que controlan la planificación, el desarrollo y la gestión de proyectos tecnológicos. MSF se centra en los modelos de proceso y de equipo dejando en un segundo plano las elecciones tecnológicas.

Figura 3: Metodología MSF

MSF tiene las siguientes características:

- Adaptable: es altamente adaptable, lo que permite entregar soluciones de alta calidad y de manera rápida mientras que minimiza el riesgo asociado a todo proyecto ya que éste considera las causas más comunes que hacen fracasar los proyectos de tecnología.
- Escalable: puede organizar equipos tan pequeños entre 3 o 4 personas, así como también, proyectos que requieren aproximadamente 50 personas.
- Flexible: es utilizada en el ambiente de desarrollo de cualquier cliente.
- Tecnología Agnóstica: puede ser usada para desarrollar soluciones basadas sobre cualquier tecnología.

MSF se compone de varios modelos encargados de planificar las diferentes partes implicadas en el desarrollo de un proyecto: Modelo de Arquitectura del Proyecto, Modelo de Equipo, Modelo de Proceso, Modelo de Gestión del Riesgo, Modelo de Diseño de Proceso y finalmente el modelo de Aplicación.

Metodología utilizada para la propuesta de solución

Después de realizar un análisis e investigación de estas tres metodologías, se elige utilizar RUP para el

desarrollo del software, ya que es la más completa y abarcadora, pues como señalan algunos autores, las otras metodologías son casos particulares de esta. Además, XP y MSF presentan algunas debilidades, lo que representa riesgos considerables, como dificultades a la hora de una buena obtención de requisitos para el sistema. RUP es adaptable, según el tipo de proyecto, así serán las características del mismo, haciéndose énfasis en aquellos flujos de trabajo durante la vida del software, que reporten más importancia y sean indispensables. RUP permite trazarse planes de riesgos y pruebas durante el ciclo de vida. Contiene artefactos que son diseñados durante las diferentes fases, los cuales describen detalladamente las características del software desde que se realiza el análisis del problema hasta la entrega final del producto.

UML

El Proceso Unificado RUP (Racional Unified Process) utiliza el Lenguaje de Modelado UML (Unified Modeling Languaje) para preparar todos los esquemas de un sistema software. UML es una parte esencial del Proceso Unificado, está consolidado como el lenguaje estándar en el análisis y diseño de sistemas de cómputo. Mediante UML es posible establecer la serie de requerimientos y estructuras necesarias para plasmar un sistema de software previo al proceso intensivo de escribir código.

UML recomienda 9 diagramas que son modelados durante las diferentes fases de desarrollo del software, es decir, representan las siguientes vistas del sistema:

- Diagrama de Casos de Uso: modela la funcionalidad del sistema agrupándola en descripciones de acciones ejecutadas por un sistema para obtener un resultado.
- Diagrama de Clases: muestra las clases que componen el sistema y cómo se relacionan entre sí.
- Diagrama de Objetos: muestra una serie de objetos y sus relaciones.
- Diagrama de Secuencia: enfatiza la interacción entre los objetos y los mensajes que intercambian entre sí junto con el orden temporal de los mismos.
- Diagrama de Colaboración: igualmente, muestra la interacción entre los objetos resaltando la organización estructural de los objetos en lugar del orden de los mensajes intercambiados.

- Diagrama de Estados: modela el comportamiento de acuerdo con eventos.
- Diagrama de Actividades: simplifica el Diagrama de Estados modelando el comportamiento mediante flujos de actividades.
- Diagrama de Componentes: muestra la organización y las dependencias entre un conjunto de componentes.

Diagrama de Despliegue: muestra los dispositivos que se encuentran en un sistema y su distribución en el mismo. [8]

1.10 Conclusiones

En este capítulo quedaron reflejados los principales conceptos relacionados con el problema, ampliando así los conocimientos para comprender mejor los procesos de negocio que ocurren en la entidad. Se hizo un análisis del estado del arte, las tecnologías y herramientas informáticas, procesos de desarrollo de software, lenguajes de programación y tendencias que fundamentan la solución propuesta.

Capítulo 2
Características del Sistema

2.1 Introducción.

En el presente capítulo se hace la descripción de la propuesta que trae este trabajo, para ello se describen los procesos del negocio que tienen que ver con el objeto de estudio, de acuerdo a esto se llega a la conclusión que debido a la poca estructuración de esos procesos, para poder entender el contexto en que se emplaza el sistema necesitamos definir conceptos que podemos agrupar en un Modelo de Dominio, para capturar correctamente los requisitos y poder construir un sistema correcto.

2.2 Descripción de los procesos del negocio propuestos.

Para describir los procesos del negocio que se relacionan con el campo de acción de este trabajo, es necesario centrar la atención en los procesos de desarrollo, gestión de configuración y gestión de proyectos en la UCI.

Para entender mejor cómo se lleva a cabo los procesos antes mencionados dentro de la universidad se impone el conocimiento del uso de las diversas metodologías de desarrollo de software, en el caso que compete, la metodología RUP.

El centro está compuesto por 10 facultades, las cuales tienen entre sus tareas el desarrollo de software a través de proyectos productivos vinculados a los perfiles correspondientes a cada una de ellas.

Para el surgimiento de un proyecto productivo es necesaria su aprobación por la Vicerrectoría de Producción. Este proceso de puesta en práctica viene dado por la necesidad de darle solución a problemas que surgen tanto en el ámbito universitario como fuera de él. En la iniciación de un proyecto se realiza la estimación de tiempos y costos con el fin de constatar la viabilidad del mismo. El próximo paso consiste en la selección del personal cualificado para el desarrollo de la aplicación pertinente, al cual se le

asigna un rol determinado teniendo en cuenta las competencias necesarias para llevarlo a cabo. Para iniciar el proceso de desarrollo se realiza la selección de una metodología de desarrollo de software que cumpla con las necesidades del proyecto en cuestión, así como un modelo de calidad a seguir, a partir de aquí se necesita un conocimiento profundo de ambos elementos para llevar a cabo los procesos de ingeniería y gestión incluidos en los mismos con el objetivo de desarrollar un software que cumpla con los estándares de calidad seleccionados previamente.

2.3 Modelo del Dominio.

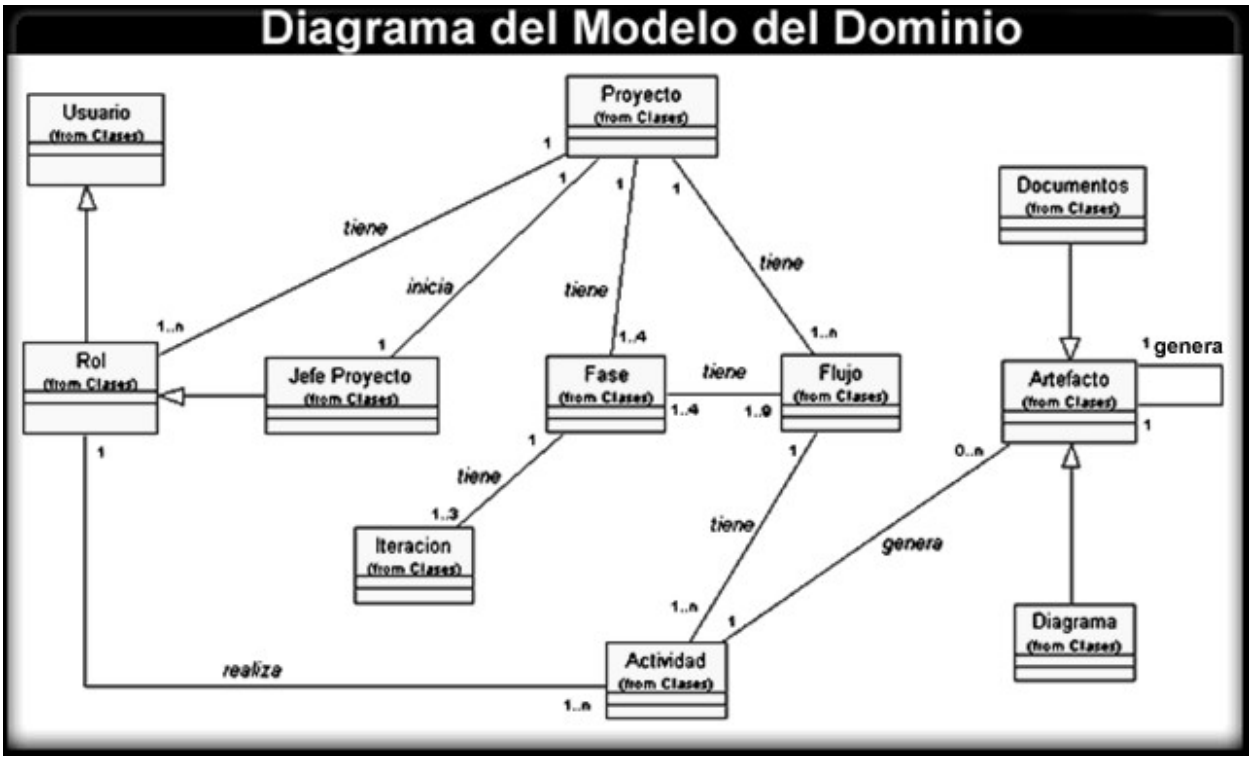
Proponemos un modelo del dominio, ya que nos permite de manera visual mostrar al usuario los principales conceptos que se manejan en el dominio del sistema en desarrollo. Esto ayuda a los usuarios, clientes y desarrolladores e interesados, a utilizar un vocabulario común para poder entender el contexto en que se emplaza el sistema. Para capturar correctamente los requisitos y poder construir un sistema correcto se necesita tener un firme conocimiento del funcionamiento del objeto de estudio. Este modelo va a contribuir posteriormente a identificar algunas clases que se utilizarán en el sistema. Primeramente vamos a identificar todos los conceptos que se utilizarán en el diagrama, mediante un glosario de términos sobre los nombres:

- Se le denominará **usuario** a cualquier persona que trabaje o estudie en la Universidad y que interactúe con el sistema.
- Se le denominará **rol** a una responsabilidad otorgada a un usuario.
- Se le denominará **Jefe de Proyecto** a un rol que tiene la responsabilidad de asignar a los usuarios uno o varios roles, dirige y organizar un proyecto.
- Se le denominará **Proyecto** a un conjunto de actividades interrelacionadas, con un inicio y una finalización definida, que utiliza recursos limitados para lograr un objetivo deseado, es un elemento organizativo a través del cual se gestiona el desarrollo del software.
- Se denominará **Fase** al tiempo entre dos grandes hitos del proyecto, durante el cual un conjunto bien definido de objetivos se cumple, los artefactos están terminados, y las decisiones se toman al

pasar o no pasar a la siguiente fase.

- Se denominará **iteración** a una secuencia de las distintas actividades con un plan de líneas base y los criterios de valoración.
- Se denominará **Flujo** a una secuencia de actividades realizadas por un rol y que produce un resultado de valor observable.
- Se denominará **Actividad** a una tarea que tiene un propósito claro, es realizado por un rol y manipula elementos.
- Se denominará **Artefacto** a un producto tangible del proyecto que son producidos, modificados y usados por las actividades, pueden ser modelos, elementos dentro de un modelo, código fuente y ejecutables.
- Se denominará **Diagrama** a una representación gráfica de una colección de elementos de modelado, a menudo dibujada como un grafo conexo de arcos (relaciones) y vértices (otros elementos del modelo).
- Se denominará **Documentos** a un conjunto de plantillas que a su vez son artefactos generados por la realización de una actividad.

El modelo del dominio se describe mediante diagramas UML, específicamente con un diagrama de clases conceptuales significativas en el dominio del problema.



2.4 Requerimientos Funcionales.

Una vez conocidos los conceptos que rodean al objeto de estudio, se puede analizar ¿Qué debe hacer el sistema para que se cumplan los objetivos planteados al inicio de este trabajo?, para ello serán enumerados a través de requerimientos funcionales las actividades que el sistema deberá ser capaz de realizar. Dentro de ellos se incluyen las acciones que podrán ser ejecutadas por el usuario, las acciones ocultas que debe realizar el sistema, y las condiciones extremas a determinar por el sistema. De acuerdo con los objetivos planteados el sistema debe ser capaz de:

R1 Autenticar Usuario

R2 Gestionar Rol

2.1 Adicionar Rol

2.2 Editar Rol

2.3 Eliminar Rol

R3 Gestionar Actividad

3.1 Inicializar Actividad

3.2 Editar Actividad

3.3 Desactivar Actividad

R4 Gestionar Flujo de Trabajo (FT)

4.1 Adicionar FT

4.2 Editar FT

4.3 Eliminar FT

R5 Gestionar Fase

5.1 Adicionar Fase

5.2 Editar Fase

5.3 Eliminar Fase

R6 Mostrar Artefacto por Categoría

6.1 Filtrar por nombre

6.2 Filtrar por tamaño

6.3 Filtrar por última modificación

6.4 Filtrar por fase

6.5 Filtrar por flujo

R7 Gestionar Usuario

7.1 Adicionar Usuario

7.2 Editar Usuario

7.3 Eliminar Usuario

R8 Generar Documento

8.1 Generar documentos por fecha de revisión a formato pdf

8.2 Generar documentos por nombre a formato pdf

R9 Gestionar Galería de Imágenes de Diagramas

9.1 Mostrar Imagen

9.2 Adicionar Imagen

9.3 Eliminar Imagen

9.4 Filtrar por nombre

9.5 Filtrar por tamaño

9.6 Filtrar por elemento modificado

R10 Gestionar Repositorio de Ficheros

10.1 Adicionar Fichero

10.3 Mostrar revisión por fecha

10.4 Filtrar por nombre

10.5 Filtrar por tamaño

10.6 Filtrar por elemento modificado

R11 Generar Diagrama de Gantt

11.1 Mostar diagrama por rango de fecha

R12 Gestionar Tareas

12.1 Adicionar Tarea

12.2 Eliminar Tarea

12.3 Editar Datos de Tarea

R13 Realizar Estimación de Proyecto

13.1 Mostrar el esfuerzo total en cada flujo de trabajo

13.2 Obtener estimación final del proyecto en horas-hombres

13.3 Mostrar la última estimación del proyecto

R14 Informar Estado de Desarrollo

R15 Crear Expediente de Proyecto

R16 Gestionar Expediente de Proyecto

16.1 Adicionar directorios de plantillas

16.2 Eliminar directorios de plantillas

16.3 Editar directorios de plantillas.

R17: Mantener reglas del negocio

R18: Determinar y ajustar objetivos

R19: Capturar vocabulario común del negocio

R20: Estructurar modelo de CU negocio

R21: Capturar vocabulario común

R22: Plan de gestión de requisitos

R23: Desarrollar documento visión

R24: Estructurar Modelo de CUS

R25: Detallar CUS

R26: Priorizar CUS

R27: Análisis de la arquitectura

R28: Análisis de CU

R29: Estructurar modelo de implementación

R30: Elaborar plan de sistema y subsistema de integración

R31: Definir misión de evaluación

R32: Validar estabilidad de prueba

R33: Elaborar casos de prueba

R34: Identificar objetivo de prueba

R35: Identificar y evaluar riesgo

R36: Iniciación de proyecto

R37: Perfil de Usuario

37.1 Cambiar perfil de Usuario Autenticado.

R38: Ver documentos

38.1 Mostrar documentos por fecha de revisión

38.2 Mostrar documentos por nombre

R39: Conf. Control de Versiones

39.1 Configurar el control de versiones por svn

39.2 Configurar el control de versiones de forma local

R40: Conf. Modo de Autenticación

40.1 Configurar modo de autenticación por ldap

41.2 Configurar modo de autenticación de forma local

2.5 Requerimientos No Funcionales.

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable.

RNF 1 Seguridad

1.1 Solo tiene permiso de gestionar un artefacto, el rol asignado a la actividad a la que pertenece dicho artefacto.

RNF 2 Software

2.1 Necesidad de que el sistema sea multiplataforma.

2.2 Desarrollado para Navegador Firefox (W3C)

RNF 3 Restricciones de diseño

3.1 Posibilidad de cambio de interfaz.

3.2 Realizar la aplicación por módulos.

2.6 Modelo de casos de uso del sistema

Utilizando las facilidades que brinda el UML, se han capturado los requisitos funcionales del sistema y se representan mediante un diagrama de casos de uso. Para ello tenemos que definir cuáles serían los actores que van a interactuar con el sistema, y los casos de uso que van a representar las funcionalidades.

ACTORES	JUSTIFICACIÓN
Usuario	Cualquier persona que esté autorizada a interactuar con el sistema.
Jefe de Proyecto	Rol encargado de administrar y asignar los recursos en forma de prioridades, coordina las interacciones con los clientes, usuario, mantiene su equipo de proyecto. También establece un conjunto de prácticas que garantizan la integridad y la calidad de los artefactos del proyecto.
Analista de Proceso de Negocio	Rol encargado de definir la arquitectura, la definición de de casos de uso y actores, y cómo interactúan.
Analista de Sistema	Rol encargado de dirigir y coordinar las necesidades y funcionalidades del sistema, identificando los actores y las interacciones de los caso de usos sistema.
Especificador de Requerimientos	Rol encargado de especificar y detallar una o más partes de las funcionalidades del sistema mediante la descripción de los requerimientos.
Arquitecto	Rol responsable de la arquitectura del software que incluye las principales decisiones técnicas que limitan el diseño global y la ejecución del proyecto.
Diseñador	Rol encargado de diseñar una parte del sistema dentro de las limitaciones de los requisitos, la arquitectura, y el proceso de desarrollo para el proyecto.

Integrador	Rol encargado de planificaren e integración de los elementos en términos de sistemas y subsistemas de implementación.
Administrador de Prueba	Rol encargado de que la prueba de esfuerzo tenga éxito, así como las actividades de promoción de la calidad y la prueba de planificación y gestión de recursos
Analista de Prueba	Rol responsable de identificar y definir las pruebas necesarias, realizando un seguimiento detallado de los resultados obtenidos en cada ciclo de pruebas y la evaluación de la calidad general experimentada como resultado de las actividades de prueba.
Diseñador de Prueba	Rol encargado de definir el método de prueba y asegurar el éxito de su aplicación, identificando las técnicas apropiadas, herramientas y directrices para la aplicación de las pruebas necesarias, y dar orientación sobre los recursos correspondientes de los requisitos de la prueba de esfuerzo.
Subversion	Sistema externo utilizado para el control de versiones de los artefactos generados por la aplicación.

Producto a que el modelo de casos de uso del sistema es grande y las responsabilidades son distribuidas, se llegó a la conclusión de dividirlo y agruparlos en paquetes, teniendo en cuenta que las referencias cruzadas entre los paquetes deben reducirse o eliminarse para prevenir conflictos entre los elementos del modelo y así se torna más comprensible y separado en partes mas manejables.

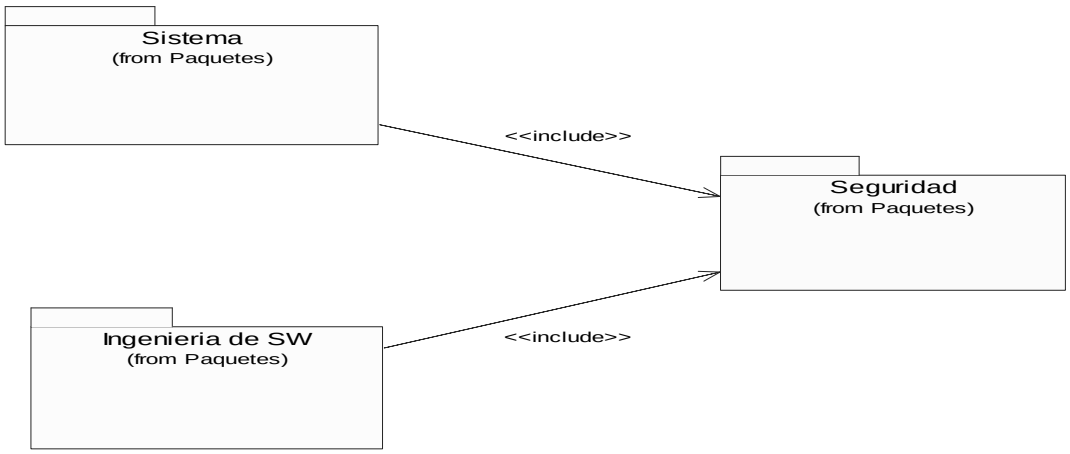
Su distribución esta formada de la siguiente forma:

Paquete Sistema: Engloba las funcionalidades del sistema.

Paquete Seguridad: Engloba las funcionalidades entorno a la seguridad del sistema.

Paquete Ingeniería de SW: Está compuesto por subpaquetes que engloban las funcionalidades de la metodología RUP.

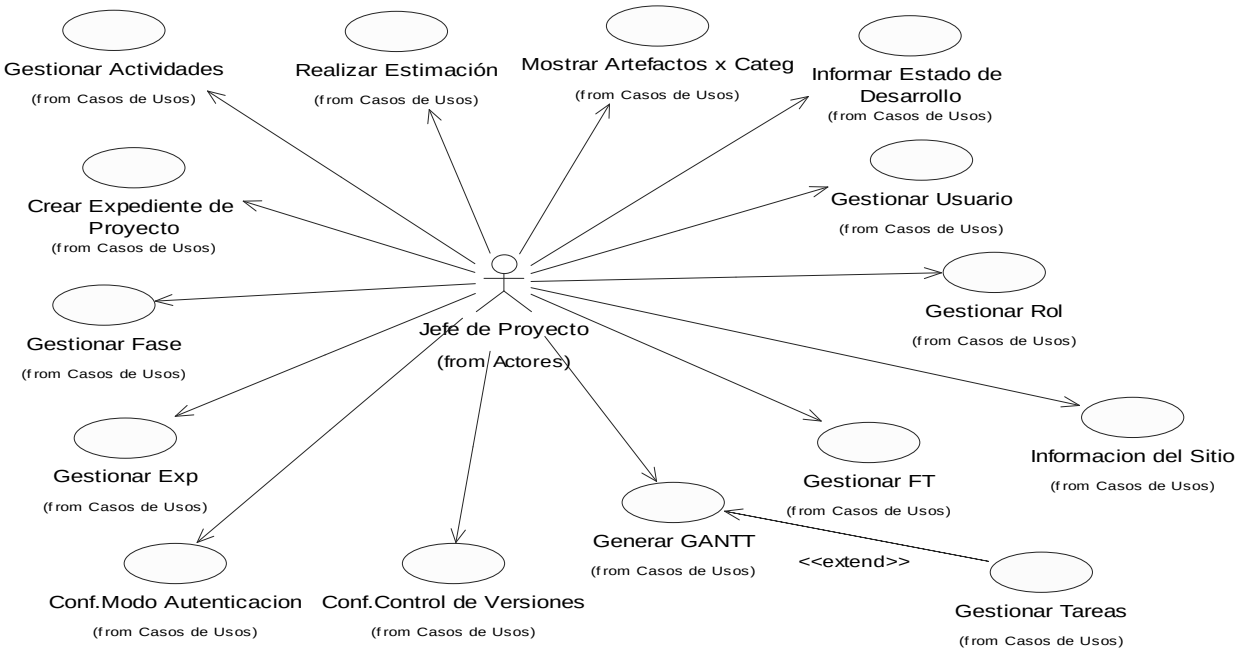
Paquetes



Paquete de Ingeniería de SW

Diagrama de CU Paquete Sistema

El diagrama donde se representa la relación existente entre los actores y los casos de uso se representa a continuación:


Gestionar Actividades
(from Casos de Usos)
Realizar Estimación
(from Casos de Usos)
Mostrar Artefactos x Categ
(from Casos de Usos)
Informar Estado de Desarrollo
(from Casos de Usos)
Crear Expediente de Proyecto
(from Casos de Usos)
Gestionar Usuario
(from Casos de Usos)
Gestionar Rol
(from Casos de Usos)
Gestionar Fase
(from Casos de Usos)
Jefe de Proyecto
(from Actores)
Gestionar Exp
(from Casos de Usos)
Informacion del Sitio
(from Casos de Usos)
Gestionar FT
(from Casos de Usos)
Generar GANTT
(from Casos de Usos)
<<extend>>
Gestionar Tareas
(from Casos de Usos)
Conf.Modo Autenticacion
(from Casos de Usos)
Conf.Control de Versiones
(from Casos de Usos)

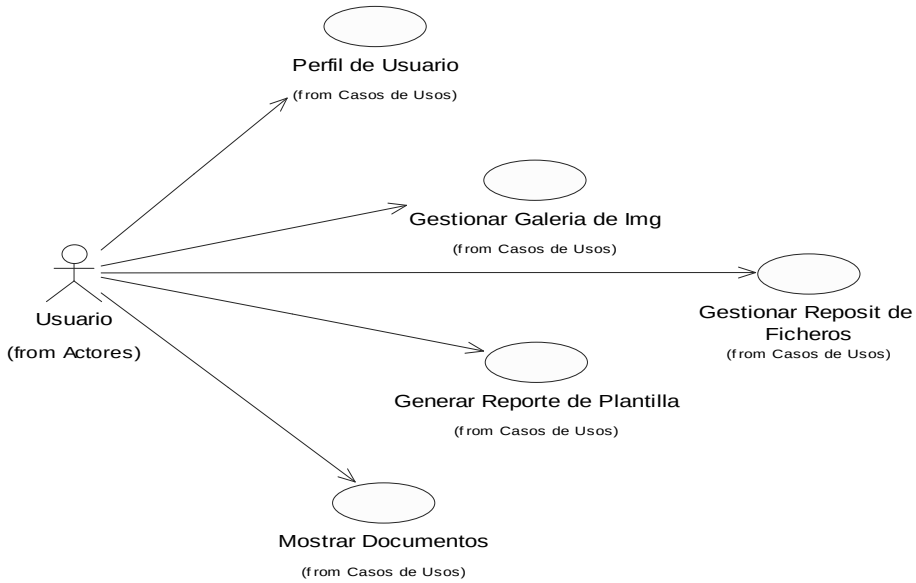

Perfil de Usuario
(from Casos de Usos)
Gestionar Galeria de Img
(from Casos de Usos)
Usuario
(from Actores)
Gestionar Reposit de Ficheros
(from Casos de Usos)
Generar Reporte de Plantilla
(from Casos de Usos)
Mostrar Documentos
(from Casos de Usos)

Diagrama de CU Paquete FT Modelo Negocio

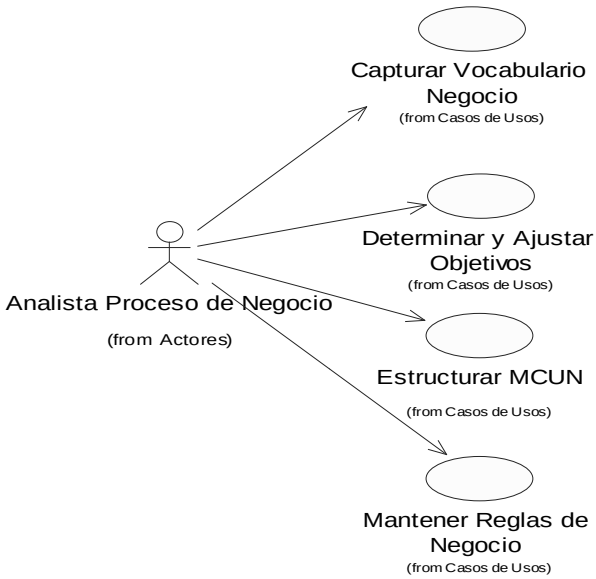


Diagrama de CU Paquete FT Requerimientos

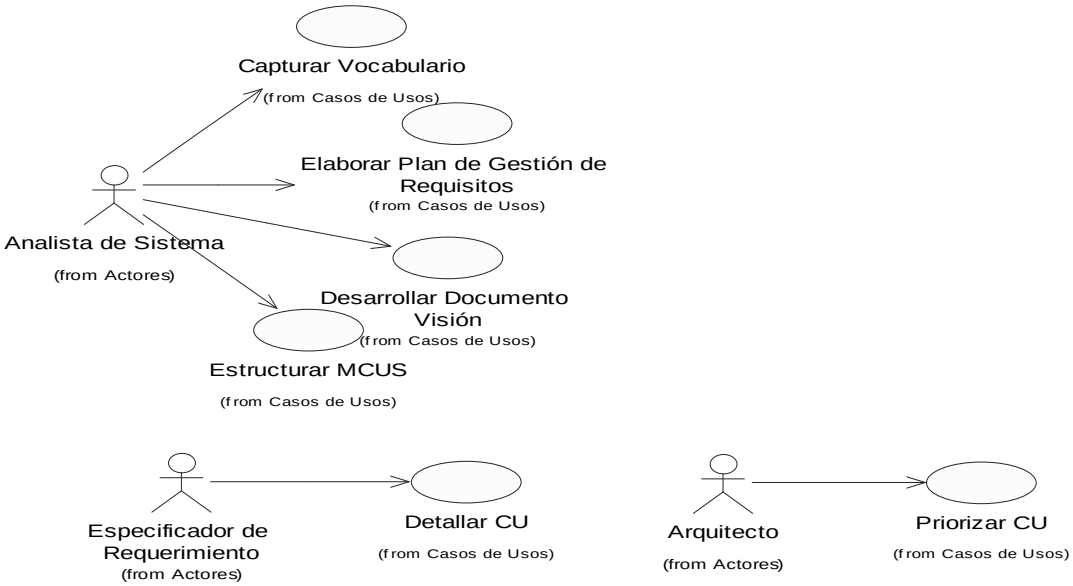


Diagrama de CU Paquete Seguridad

Diagrama de CU Paquete Análisis-Diseño

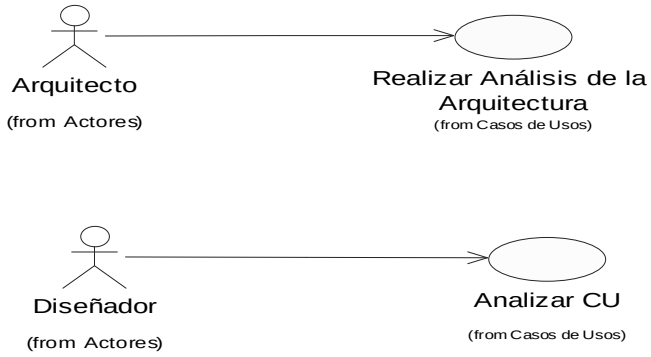


Diagrama de CU Paquete Implementación

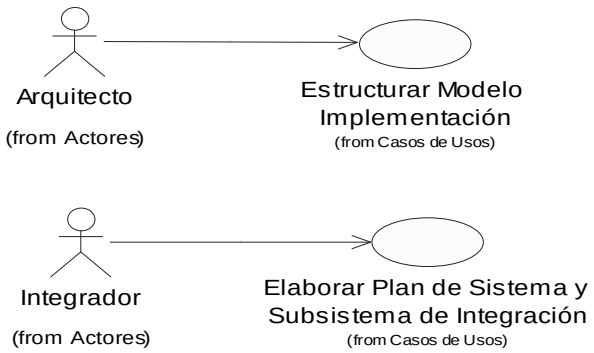


Diagrama de CU Paquete Gestión de Proyecto

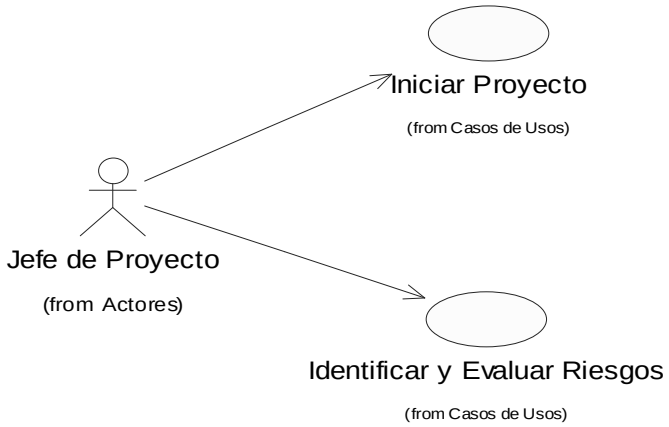
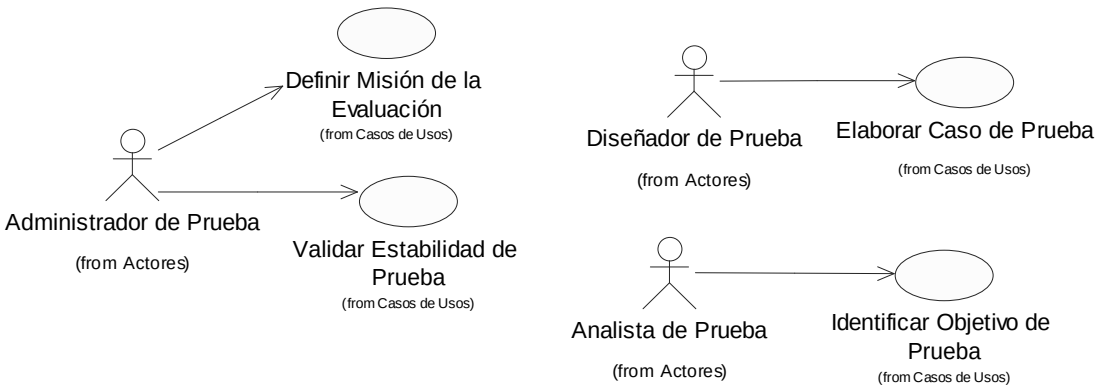
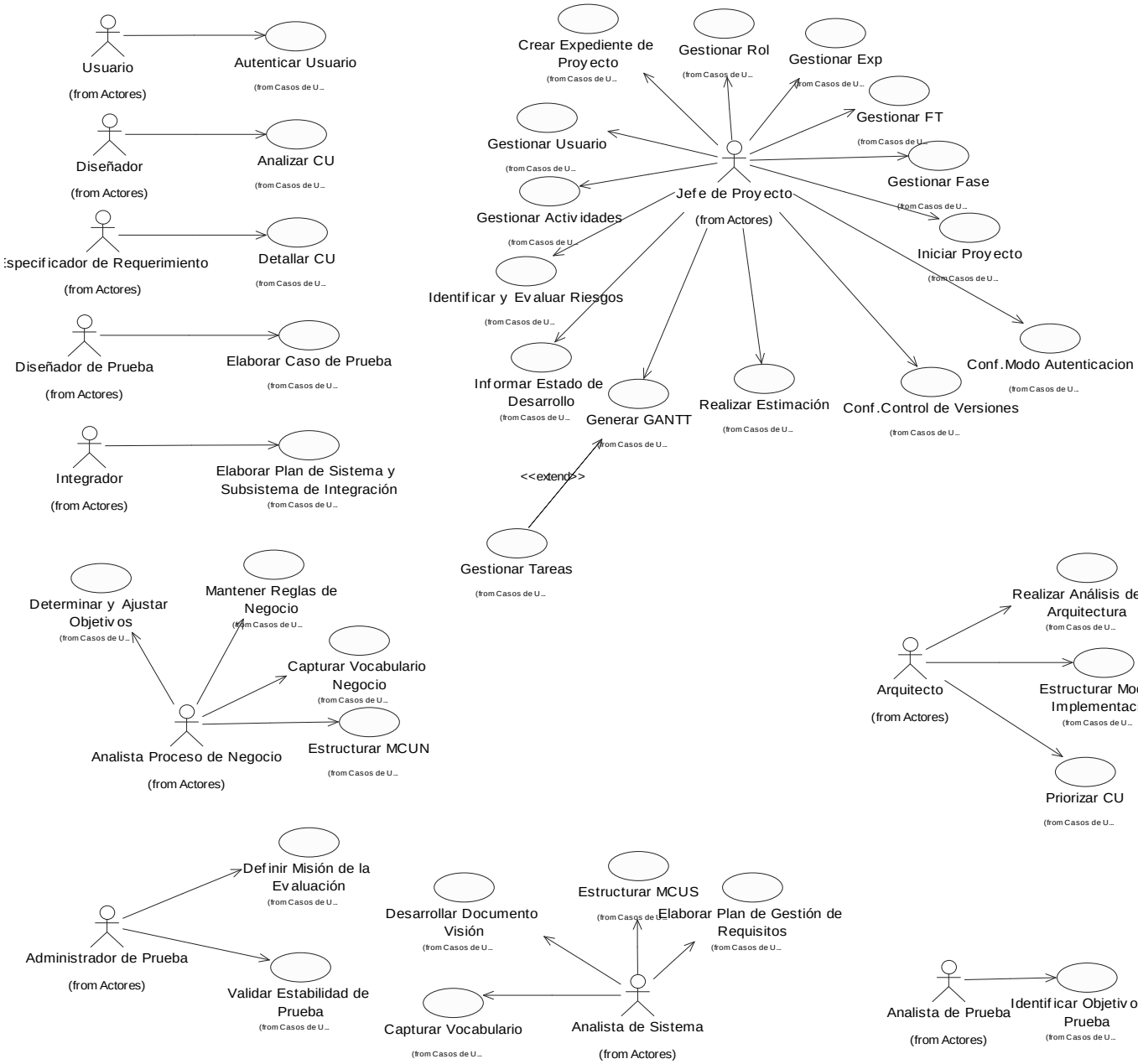


Diagrama de CU Paquete Prueba



Vista de CU de la Arquitectura


Usuario
(from Actores)
Autenticar Usuario
(from Casos de U...
Diseñador
(from Actores)
Analizar CU
(from Casos de U...
Especificador de Requerimiento
(from Actores)
Detallar CU
(from Casos de U...
Diseñador de Prueba
(from Actores)
Elaborar Caso de Prueba
(from Casos de U...
Integrador
(from Actores)
Elaborar Plan de Sistema y Subsistema de Integración
(from Casos de U...
Crear Expediente de Proyecto
(from Casos de U...
Gestionar Rol
(from Casos de U...
Gestionar Exp
(from Casos de U...
Gestionar FT
(from Casos de U...
Gestionar Usuario
(from Casos de U...
Gestionar Fase
(from Casos de U...
Gestionar Actividades
(from Casos de U...
Jefe de Proyecto
(from Actores)
Iniciar Proyecto
(from Casos de U...
Identificar y Evaluar Riesgos
(from Casos de U...
Informar Estado de Desarrollo
(from Casos de U...
Generar GANTT
(from Casos de U...
Realizar Estimación
(from Casos de U...
Conf.Control de Versiones
(from Casos de U...
Conf.Modo Autenticacion
(from Casos de U...
<<extend>>
Gestionar Tareas
(from Casos de U...
Determinar y Ajustar Objetivos
(from Casos de U...
Mantener Reglas de Negocio
(from Casos de U...
Capturar Vocabulario Negocio
(from Casos de U...
Analista Proceso de Negocio
(from Actores)
Estructurar MCUN
(from Casos de U...
Realizar Análisis de la Arquitectura
(from Casos de U...
Arquitecto
(from Actores)
Estructurar Modelo Implementación
(from Casos de U...
Priorizar CU
(from Casos de U...
Definir Misión de la Evaluación
(from Casos de U...
Administrador de Prueba
(from Actores)
Validar Estabilidad de Prueba
(from Casos de U...
Desarrollar Documento Visión
(from Casos de U...
Estructurar MCUS
(from Casos de U...
Elaborar Plan de Gestión de Requisitos
(from Casos de U...
Capturar Vocabulario
(from Casos de U...
Analista de Sistema
(from Actores)
Analista de Prueba
(from Actores)
Identificar Objetivo de Prueba
(from Casos de U...

2.7 Descripción de los casos de uso del sistema

Caso de Uso:	**Autenticar Usuario**
Actor(es):	Usuario (inicia)
Propósito:	Autenticación del usuario en el sistema.
Resumen:	El caso de uso inicia cuando el usuario necesita autenticarse para poder acceder al sistema, entra su usuario y contraseña y accede al sistema.
Referencias:	R1.
Precondiciones:	El Usuario levanta el sistema y se encuentra en la página inicial de autenticación.

Pantalla 1

Flujo Normal de Eventos

Acción del Actor	**Respuesta del Sistema**
1. El usuario necesita entrar al sistema.	2. El sistema muestra la página inicial de autenticación.
3. El usuario introduce su usuario y la contraseña	4. El sistema comprueba que no existan campos vacíos.

	5. El sistema comprueba que el usuario y la contraseña son válidos
	6. El sistema le da acceso a las funcionalidades correspondientes a este usuario.
Flujos Alternos de los Eventos	
	4.1 En caso de que existan campos vacíos el sistema muestra un mensaje informando que deben llenarse los campos.
	5.1 En caso que no esté correcto el usuario o la contraseña, el sistema emitirá un mensaje informando sobre la invalidez de los datos entrados. Retorna a la acción 3.
Poscondiciones:	El usuario entra al sistema.
Prioridad:	Crítico

Caso de Uso:	**Gestionar Actividad**
Actor(es):	Jefe de Proyecto (inicia)
Propósito:	Inicializar, Desactivar y Modificar actividades de RUP en el sistema.
Resumen:	Este caso de uso permite inicializar una actividad de RUP, asignarle un flujo de trabajo, una fase y el rol que puede desarrollar esta actividad.; así como eliminar estas asignaciones.
Referencias:	R3.1, R3.2, R3.3
Precondiciones:	El jefe de proyecto ya autenticado.

	Deben existir actividades en el sistema.

Pantalla 1

Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
1. El jefe de proyecto selecciona la opción “Gestionar actividad” de la sección administrativa del sistema.	2. El sistema muestra (pantalla 1): la lista de las actividades, las acciones de inicializar, Desactivar y la opción de Editar.
3. El Jefe de Proyecto selecciona la opción deseada.	4. El sistema ejecuta una de las siguientes acciones: a) Si el jefe de proyecto decide inicializar actividad, ir a la sección “Inicializar Actividad”. b) Si el jefe de proyecto decide desactivar actividad, ir a la sección “Desactivar Actividad”. c) Si el jefe de proyecto decide editar una actividad, ir a la opción “Editar Actividad”.

Sección: “Inicializar Actividad”	
1. El jefe de proyecto habilita el estado de la actividad(es) que desee inicializar.	2. El sistema verifica que las actividades que generen los artefactos de entrada de la actividad a habilitar estén también habilitadas.
	3. El sistema crea el o los artefactos de la actividad
	4. El sistema muestra un mensaje de confirmación de que se ha habilitado la actividad.
Flujos Alternos de los Eventos	
	2.1 Si la actividad tiene como entrada artefactos de otra actividad que no esté activada.
Sección: “Desactivar Actividad”	
1. El jefe de proyecto deshabilita el estado de la actividad(es) que desee desactivar.	2. El sistema muestra un mensaje informando que se ha desactivado la actividad.
Sección: “Editar Actividad”	
1. El jefe de proyecto selecciona la opción de “Editar”	2. El sistema muestra (pantalla 3) una ventana, con tres pestañas: Flujo de trabajo, Roles y Fases; brindando la opción de asignarle o quitarle un flujo de trabajo, fase o rol a la actividad.
3. El jefe de proyecto realiza las asignaciones o cambios deseados.	4. El sistema muestra un mensaje de confirmación para cada cambio efectuado.

Pantalla 3

Poscondiciones:	Se activa, desactiva o se realizan las asignaciones a la actividad.
Prioridad:	Crítica

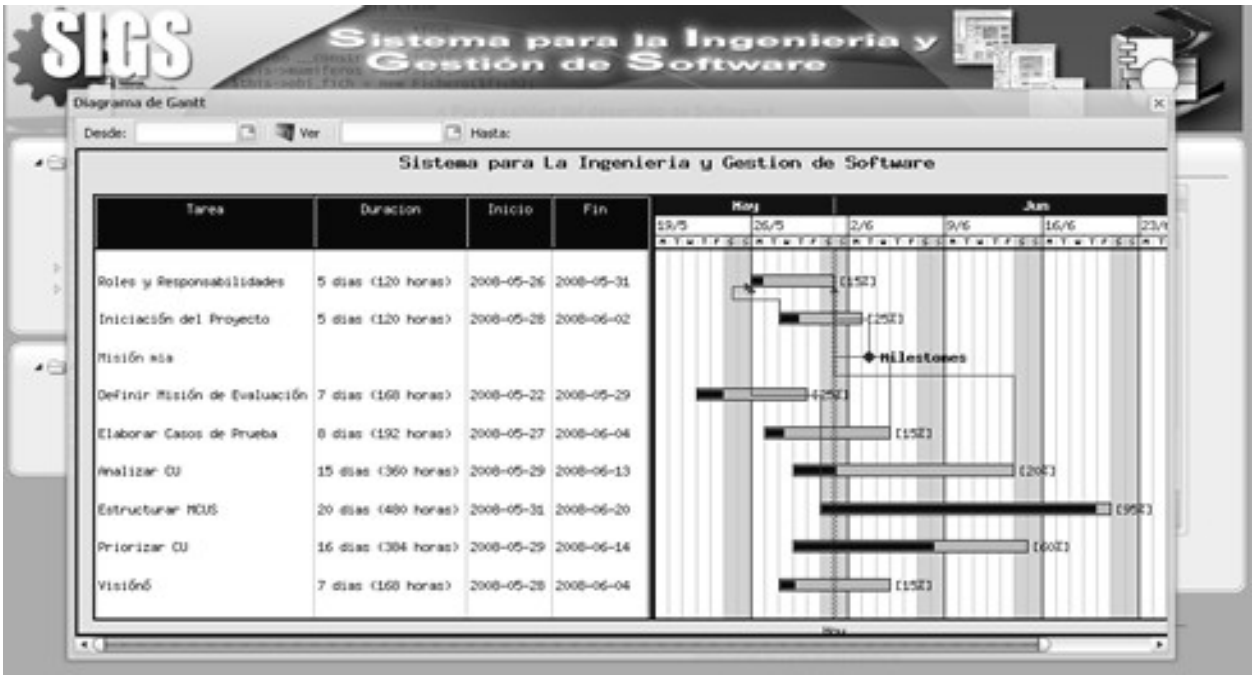
Caso de Uso:	**Generar Diagrama Gantt**
Actor(es):	Jefe de Proyecto (inicia).
Propósito:	Generar el diagrama de Gantt.
Resumen:	El caso de uso inicia cuando el jefe de proyecto decide generar el diagrama de Gantt; una ves adicionadas las tareas el jefe de proyecto selecciona la opción generar diagrama Gantt y finaliza el caso de uso mostrando el diagrama de Gantt generado.

Referencias:	R11.1
Precondiciones:	El Jefe de Proyecto ya autenticado en el sistema. Deben existir tareas adicionadas en el sistema.

Pantalla 1

Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
1. El jefe de proyecto selecciona a la opción “Generar Diagrama de Gantt”.	2. El sistema muestra (pantalla 1) en pantalla una ventana donde se van a mostrar la lista de las tareas existentes en el sistema con sus características.
3. El jefe de proyecto selecciona la opción “Generar Gantt”.	4. El sistema genera el diagrama de Gantt y lo muestra en (pantalla 2).



Pantalla 2

Flujos Alternos de los Eventos	
	4.1 En caso de no existir tareas el sistema muestra un mensaje informando que no existen tareas.
Poscondiciones:	Visualizado Diagrama de Gantt.
Prioridad:	Critico

Caso de Uso:	**Realizar Estimación del Proyecto**
Actor(es):	Jefe de Proyecto (inicia)

Propósito:	Realizar la estimación de horas/hombres necesarias y tiempo que va a durar el proyecto.
Resumen:	El caso de uso inicia cuando el jefe de proyecto decide realizar la estimación de duración del tiempo de vida del proyecto. Mediante el caso de uso se realiza la estimación basada en Puntos de Casos de Usos, la cual consta de tres pasos fundamentales, por los cuales se debe guiar el jefe de proyecto para introducir correctamente los datos en el sistema. Además el caso de uso brinda la opción de ver la última estimación en caso de existir alguna.
Referencias:	R13.1 , R13.2 , R13.3
Precondiciones:	El Jefe de Proyecto ya autenticado en el sistema.

Pantalla 1

Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
1. El jefe de proyecto selecciona la opción “Cálculo	2. El sistema muestra (pantalla 1) el formulario con tres pestañas para introducir los datos necesarios para la

Estimación”.	estimación. El sistema muestra por defecto la pestaña “P1:Puntos de CU sin ajustar
3. El jefe de proyecto introduce los datos correspondientes a esta pestaña.	
4. El jefe de proyecto selecciona la segunda pestaña : “P2: Puntos de CU Ajustados”	5. El sistema muestra la pestaña, que a la vez tiene 3 subpestañas.
6. El jefe de proyecto introduce los datos correspondientes a estas subpestañas.	
7. El jefe de proyecto selecciona la tercera pestaña: “P3: Cálculo Estimación”.	8. El sistema verifica que no existan campos vacíos, ni inválidos y calcula la cantidad de horas/hombres estimadas y el esfuerzo total del proyecto por flujos de trabajo.
	9. El sistema muestra la pestaña, que a la vez tiene 3 subpestañas. La primara con la estimación de horas/hombres necesarias en el proyecto, la segunda con el esfuerzo total del proyecto por flujos de trabajo.
10. jefe de proyecto ve la información de las dos primeras subpestañas y selecciona la tercera subpestaña: “Estimación de Tiempo”	11. El sistema muestra la subpestaña.
12. El jefe de proyecto introduce los datos correspondientes a esta subpestaña.	13. El sistema muestra la estimación de tiempo para el proyecto.

Pantalla 2

Flujos Alternos de los Eventos	
	8.1 En caso de existir campos vacíos el sistema muestra un mensaje informando que existen campos vacíos.
	8.2 En caso de que existan datos que no sean correctos el sistema muestra un mensaje informando que hay datos incorrectos.
Poscondiciones:	Estimación realizada y guardada en la Base de Datos.
Prioridad:	Secundario

Caso de uso	**Crear Expediente de Proyecto**.
Actor(es):	Jefe de Proyecto (inicia).
Propósito:	Crear el Expediente de Proyecto.

Resumen:	El caso de uso inicia cuando el jefe de proyecto necesita crear el Expediente de Proyecto para ir organizando todos los artefactos que constituyen este expediente.
Referencias:	RF 15.
Precondiciones:	Debe estar activado el módulo Expediente de Proyecto.

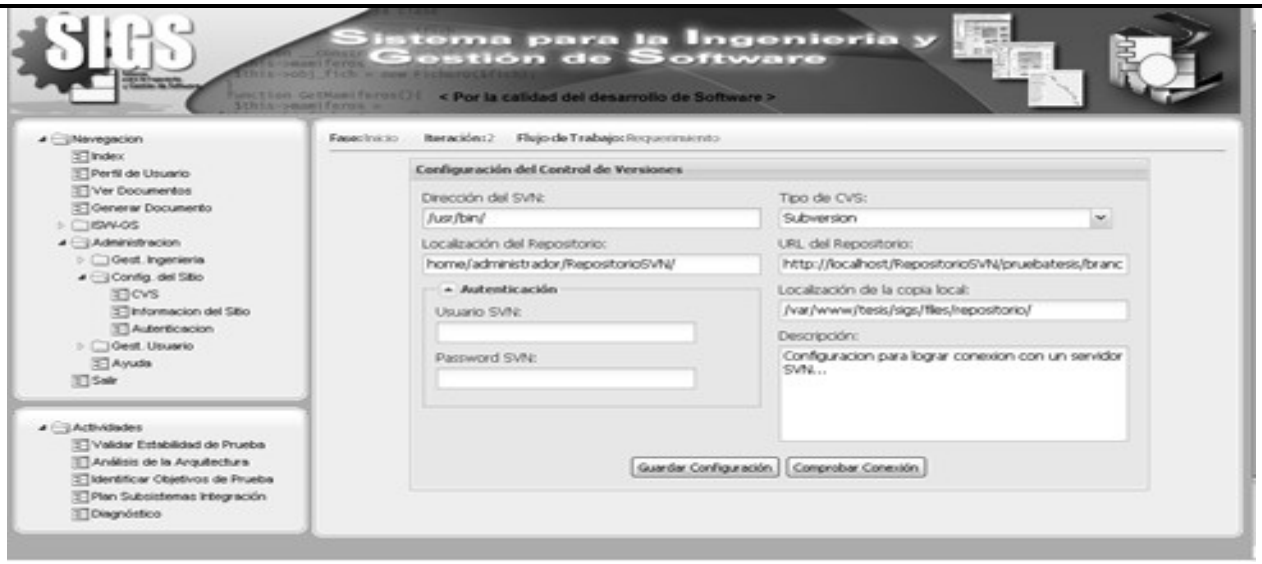
Pantalla 1

Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
1. El jefe de proyecto selecciona la opción Expediente de Proyecto.	2. El sistema muestra (pantalla 1) el expediente de proyecto en forma de árbol, organizados en tres categorías ISW, Soporte y Gestión.
3. El jefe de proyecto selecciona la plantilla a mostrar	4. El sistema muestra en el campo Descripción la descripción de la plantilla seleccionada.

Flujos Alternos de los Eventos	
3.1 El jefe de proyecto selecciona una carpeta del árbol y selecciona mostrar.	3.2 El sistema muestra un mensaje, indicando que debe seleccionar un artefacto.
Poscondiciones:	Artefacto “Expediente de Proyecto” creado
Prioridad	Critico

Caso de Uso:	**Conf. Control de Versiones**
Actor(es):	Jefe de Proyecto (inicia)
Propósito:	Configurar el control de versiones en el sistema
Resumen:	El caso de uso inicia cuando el Jefe de Proyecto necesita configurar el control de versiones del sistema, para ello selecciona el modo local o subversión
Referencias:	R39.1 , R39.2
Precondiciones:	El Jefe de Proyecto ya autenticado en el sistema.



Pantalla 1

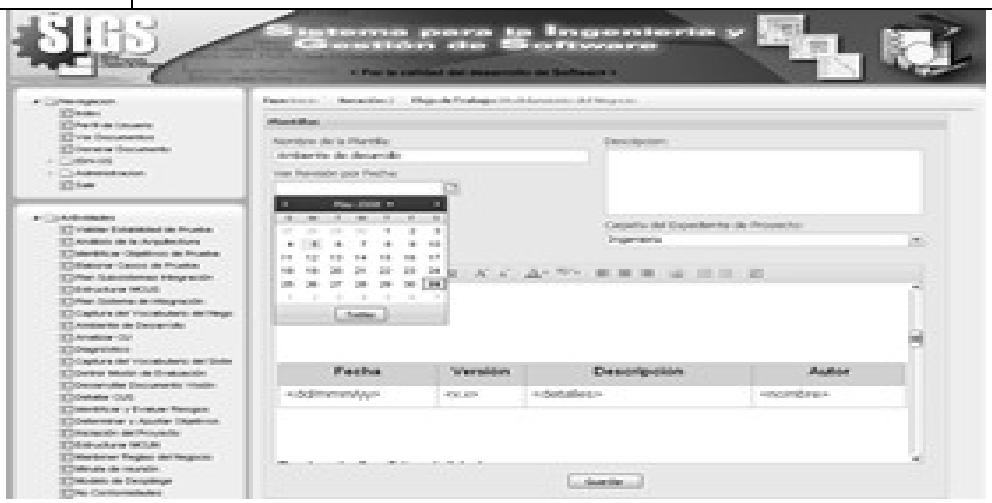
Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
1. El jefe de proyecto selecciona la opción “CVS”.	2. El sistema muestra (pantalla 1)
3. El jefe de proyecto selecciona el modo de configuración para el control de versiones “Local o Subversión” y entra los datos.	
4. En caso de optar por Subversión el Jefe de Proyecto comprueba la conexión.	5. El sistema muestra (pantalla 2) una ventana informándole el estado de la conexión.
6. jefe de proyecto guarda la configuración seleccionada	7. El sistema el muestra un mensaje de confirmación de que la configuración del control de versiones ha sido guardado.

Pantalla 2

Flujos Alternos de los Eventos	
	4.1 En caso de que la conexión haya sido fallida el sistema de muestra un mensaje informándole el estado de la conexión.
	6.1 En caso de que el sistema detecte que le faltan campos a llenar, le muestra un mensaje informándole que debe llenar los campos vacíos para poder guardar satisfactoriamente los cambios.
Poscondiciones:	Sistema configurado con al menos un modo de control de versiones.
Prioridad:	Critico.

Caso de uso	**Mantener Reglas de Negocio**
Actor(es):	Analista Proceso de Negocio.

Propósito:	Registrar las reglas del negocio a ser consideradas en el proyecto.
Resumen:	El caso de uso inicia cuando el analista de procesos de negocio necesita registrar las reglas del negocio. Actualiza la plantilla “Documento Reglas del Negocio” y guarda los cambios efectuados como una nueva versión.
Referencias:	R 17.
Precondiciones:	La actividad “Mantener Reglas del Negocio” debe estar activada.



Pantalla 1

Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
1. El analista de procesos de negocio selecciona la actividad “Mantener Reglas de Negocio”.	2. Si el sistema está configurado para guardar los artefactos en SVN, entonces busca la última revisión de la plantilla “Documento Reglas del Negocio” en SVN y la muestra en pantalla. En caso de que el sistema esté configurado para guardar las versiones localmente, busca la última versión de la plantilla en la BD y la muestra en pantalla. Muestra además las opciones de buscar

	versiones por fecha y crear versión.
3. El analista de procesos de negocio selecciona la opción deseada.	4. El sistema ejecuta una de las siguientes acciones: a) Si el analista de procesos de negocio decide crear una nueva versión, ir a la sección: “Crear Versión”. b) Si el analista de procesos de negocio decide eliminar una versión, ir a la sección: “Buscar Versión por Fecha”.
Flujos Alternos de los Eventos	
	2.2 En caso de no existir ninguna versión de la plantilla el sistema busca en la BD la plantilla original y la muestra en pantalla.
Sección: “Crear Versión”	
1. El analista de procesos de negocio procede a actualizar la plantilla.	
2. El analista de procesos de negocio guarda los cambios realizados.	3. El sistema muestra un mensaje informando que se han guardado los cambios.
Sección: “Buscar Versión por Fecha”	
1. El analista de procesos de negocio selecciona la fecha de la versión que desea buscar.	2. El sistema busca la versión existente para la fecha seleccionada y la muestra en una nueva ventana.
Flujos Alternos de los Eventos	
1.1 El analista de procesos de negocio no selecciona ninguna fecha.	1.2 El sistema muestra un mensaje informando que debe seleccionar una fecha.
	2.2 En caso de no existir ninguna versión en la fecha seleccionada el sistema muestra un mensaje informando

	que no existe versiones en esa fecha.
Poscondiciones:	Se ha creado una versión de la plantilla “Documento Reglas del Negocio”.
Prioridad	Critico

Para no extender en demasía el documento las restantes descripciones se encuentran en el **Anexo 1**.

2.8 Conclusiones

En este capítulo se comenzó a desarrollar la propuesta de solución, obteniéndose a partir del análisis de los procesos del negocio, un listado con las principales funcionalidades que debe tener el sistema y los requisitos adicionales. Se presentaron los diagramas de casos de uso del sistema divididos en paquetes y finalmente se describieron las acciones de los actores del mismo con los casos de uso con los que interactúan.

Capítulo 3
Análisis y Diseño

3.1 Introducción.

Este capítulo tiene el objetivo de plantear la concepción general del análisis y diseño del sistema propuesto y cómo se desarrollará. Para ello se modelan los artefactos que ayudan a manejar las complicaciones que implican la construcción de aplicaciones Web, donde los componentes de la aplicación se tratan como clases, y utilizando las extensiones del UML, se representan a través de diagramas de clases Web. Además se muestra el modelo de datos que es la base para construir finalmente la base de datos que soportará el sistema.

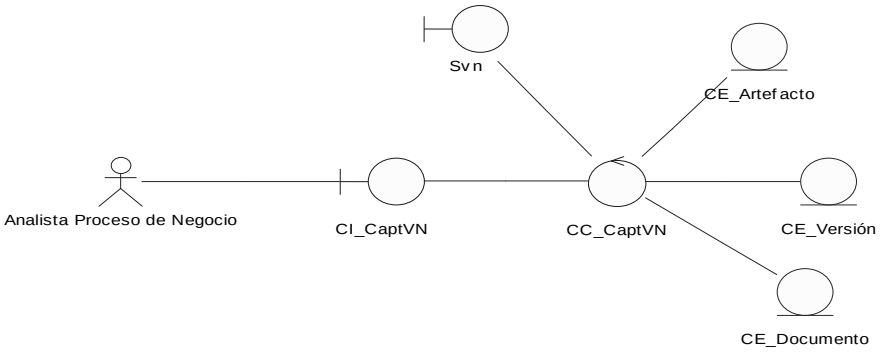
3.2 Análisis.

3.2.1 Diagrama de clases del análisis.

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

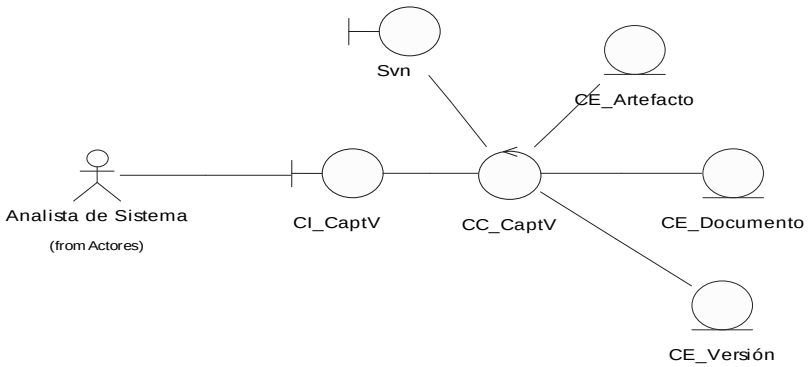
3.2.2 Paquete FT Modelo del Negocio.

CU Capturar Vocabulario Negocio



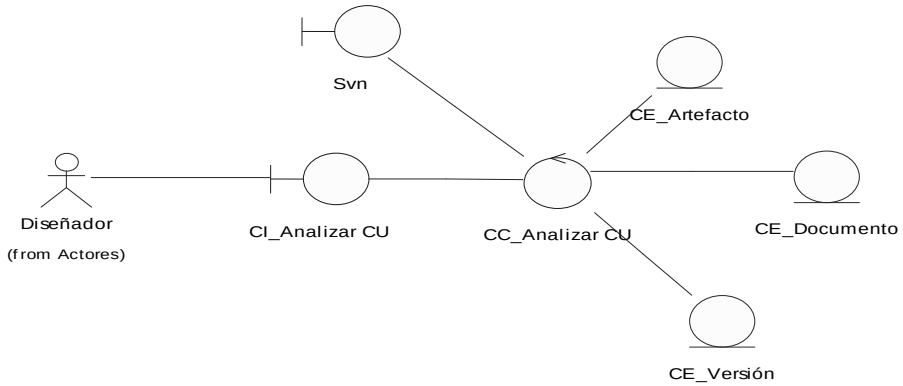
3.2.3 Paquete FT Requerimiento

CU Capturar Vocabulario



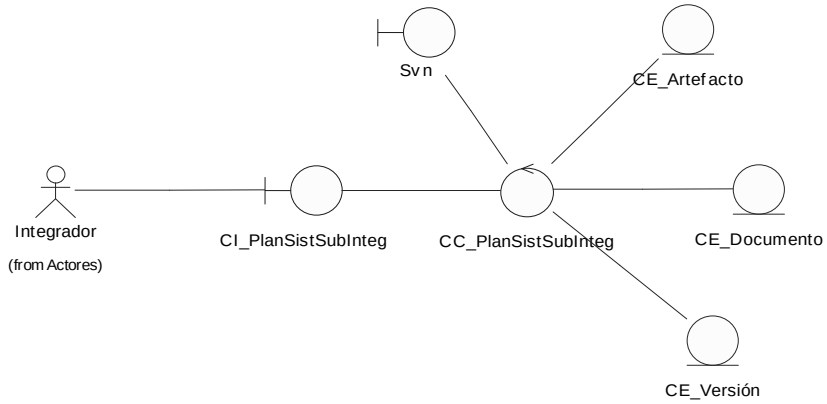
3.2.4 Paquete FT Análisis-Diseño

CU Analizar Caso de Uso



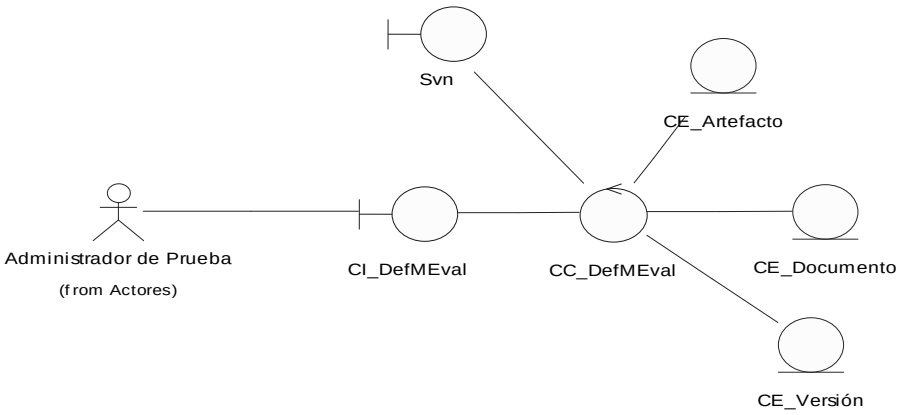
3.2.5 Paquete FT Implementación

CU Elaborar Plan de Sistema y Subsistema de Integración



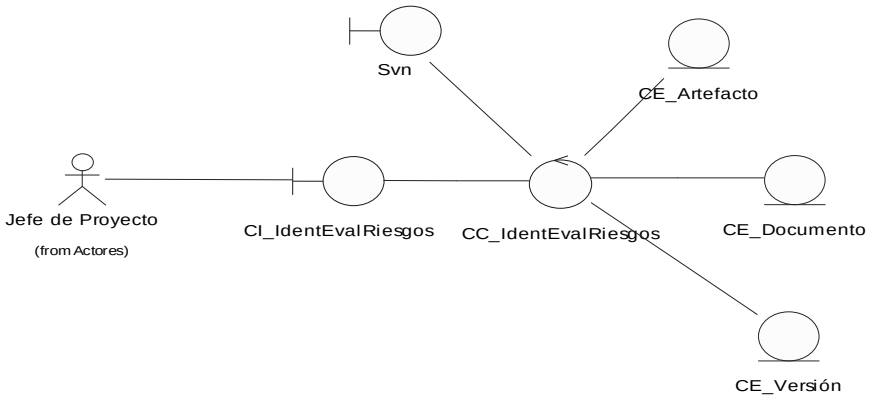
3.2.6 Paquete FT Prueba

CU Definir Misión de la Evaluación



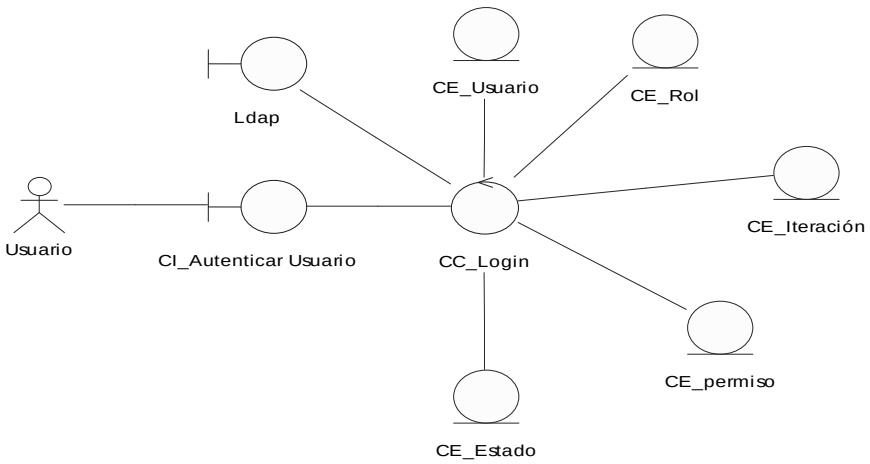
3.2.7 Paquete FT Gestión de Proyecto

CU Identificar y Evaluar Riesgos



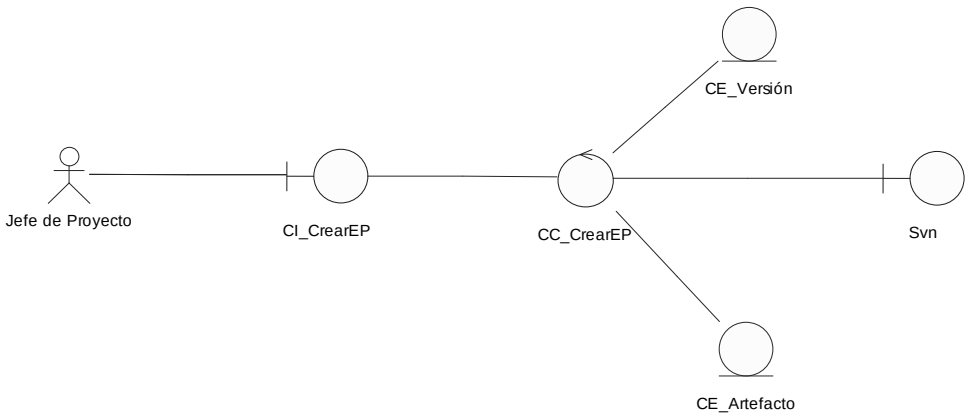
3.2.8 Paquete Seguridad

CU Autenticar Usuario

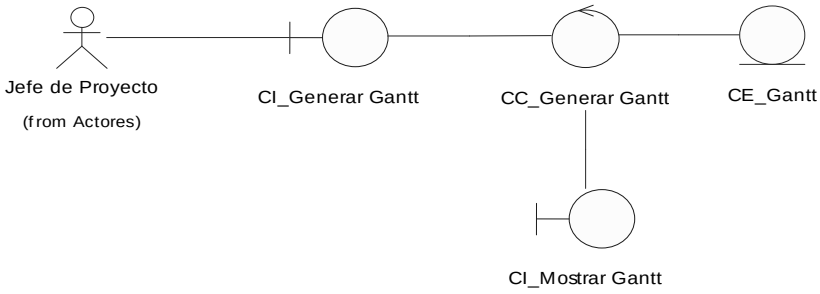


3.2.9 Paquete Sistema

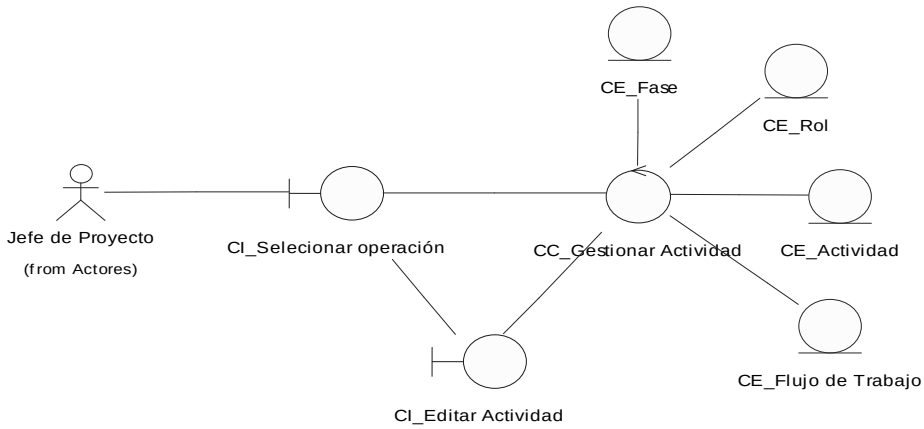
CU Crear Expediente de Proyecto



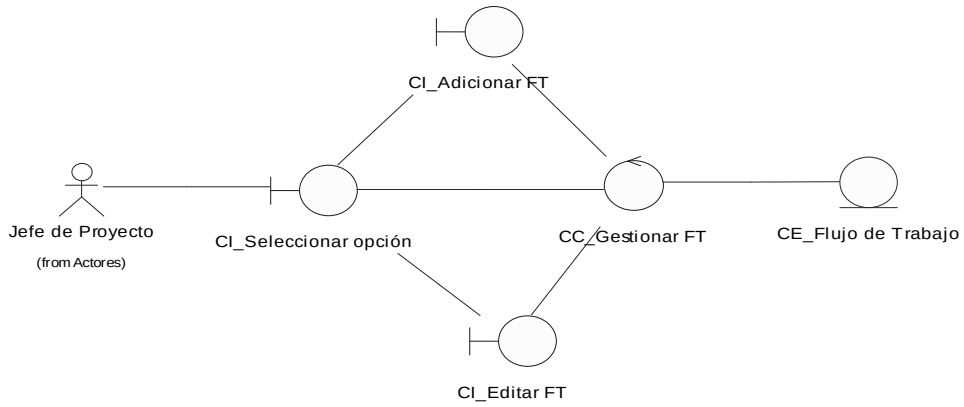
CU Generar Diagrama Gantt



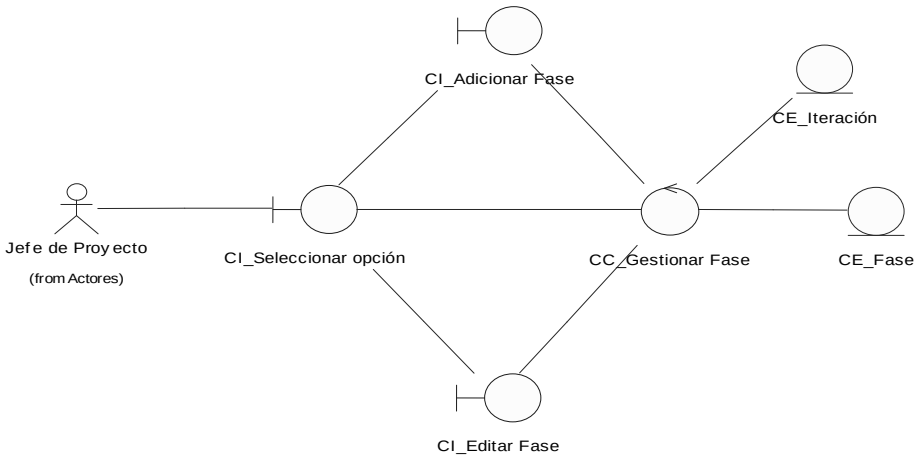
CU Gestionar Actividad



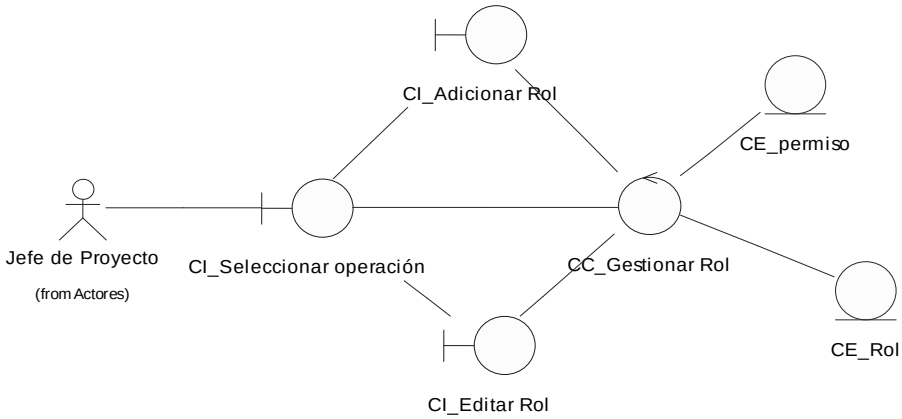
CU Gestionar Flujo de Trabajo



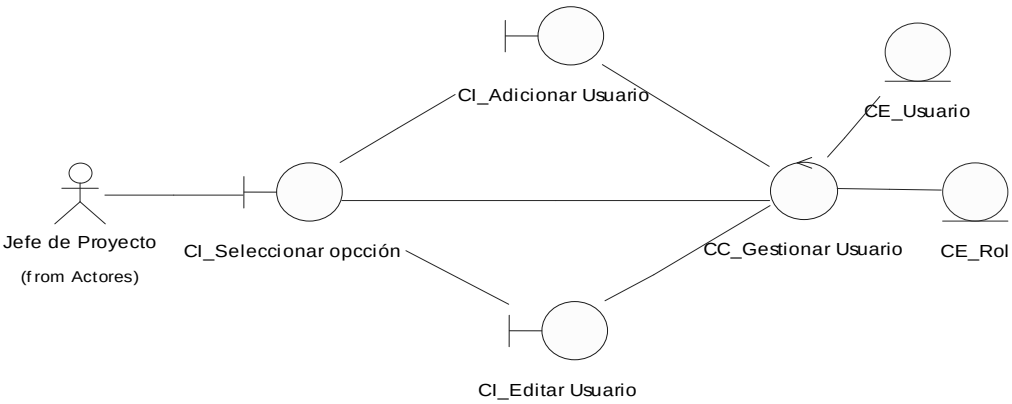
CU Gestionar Fase



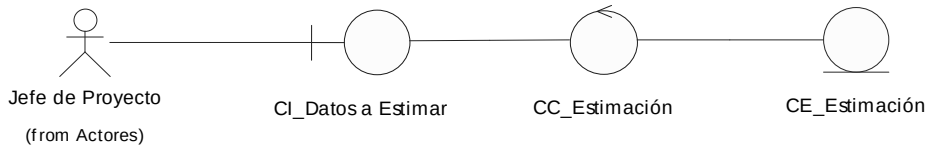
CU Gestionar Rol



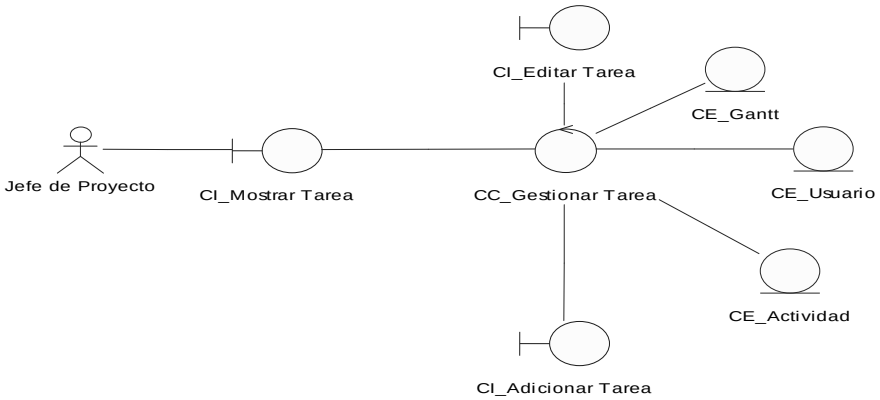
CU Gestionar Usuario



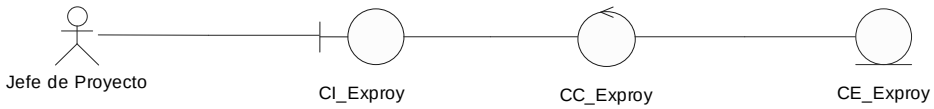
CU Realizar Estimación del Proyecto



CU Gestionar Tarea



CU Gestionar Expediente de Proyecto



Para no extender en demasía el documento las restantes diagramas se encuentran en el **Anexo 2**.

3.3 Diseño

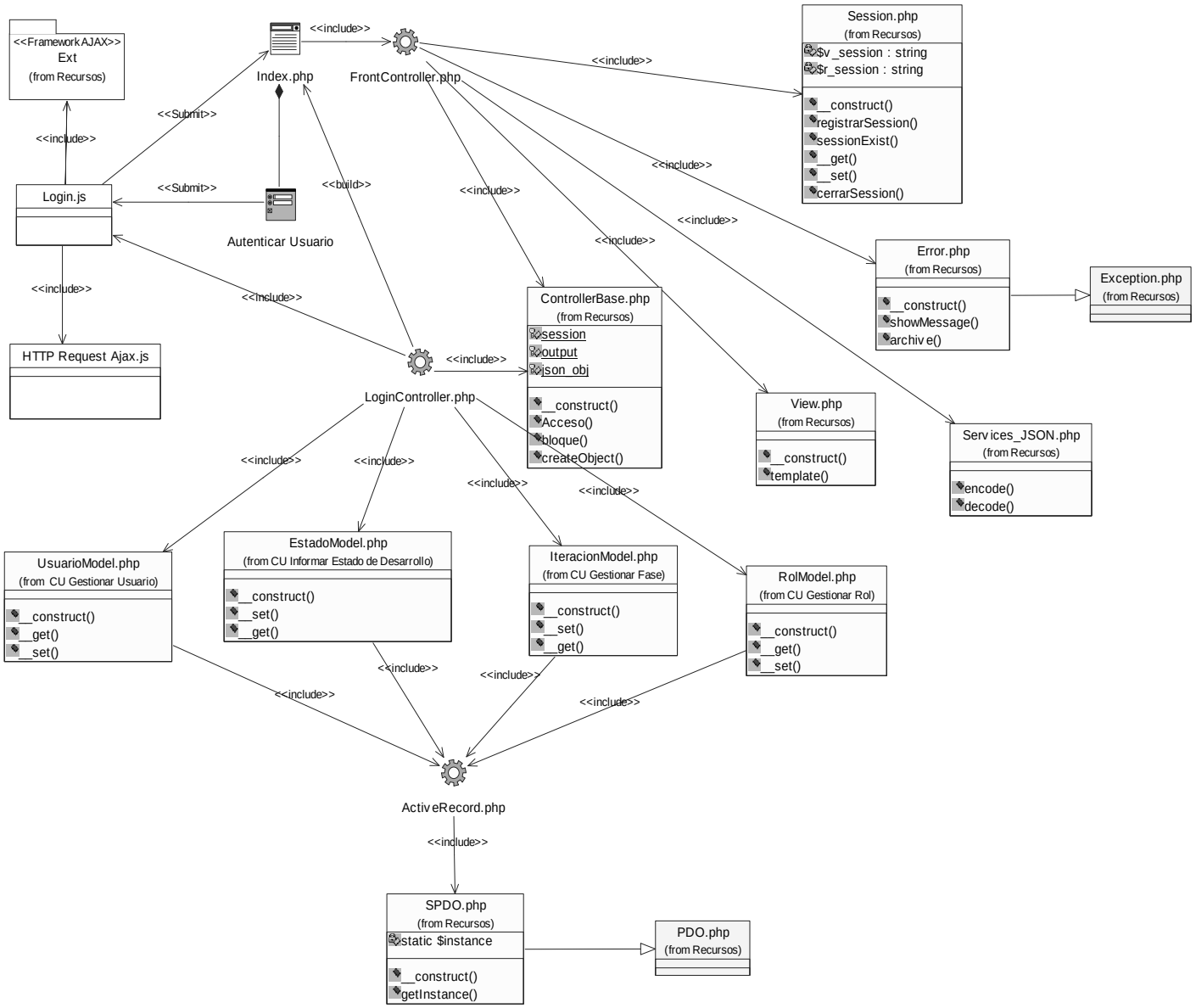
3.3.1 Diagrama de clases del diseño

El diseño es el centro de atención al final de la fase de elaboración y el comienzo de las iteraciones de construcción. Esto contribuye a una arquitectura estable y sólida, y a crear un plano del modelo de implementación. Durante la fase de construcción, cuando la arquitectura es estable y los requisitos están bien entendidos, el centro de atención se desplaza a la implementación.

El modelo de diseño está muy cercano al de implementación, lo que es natural para guardar y mantener el modelo de diseño a través del ciclo de vida completo del software.

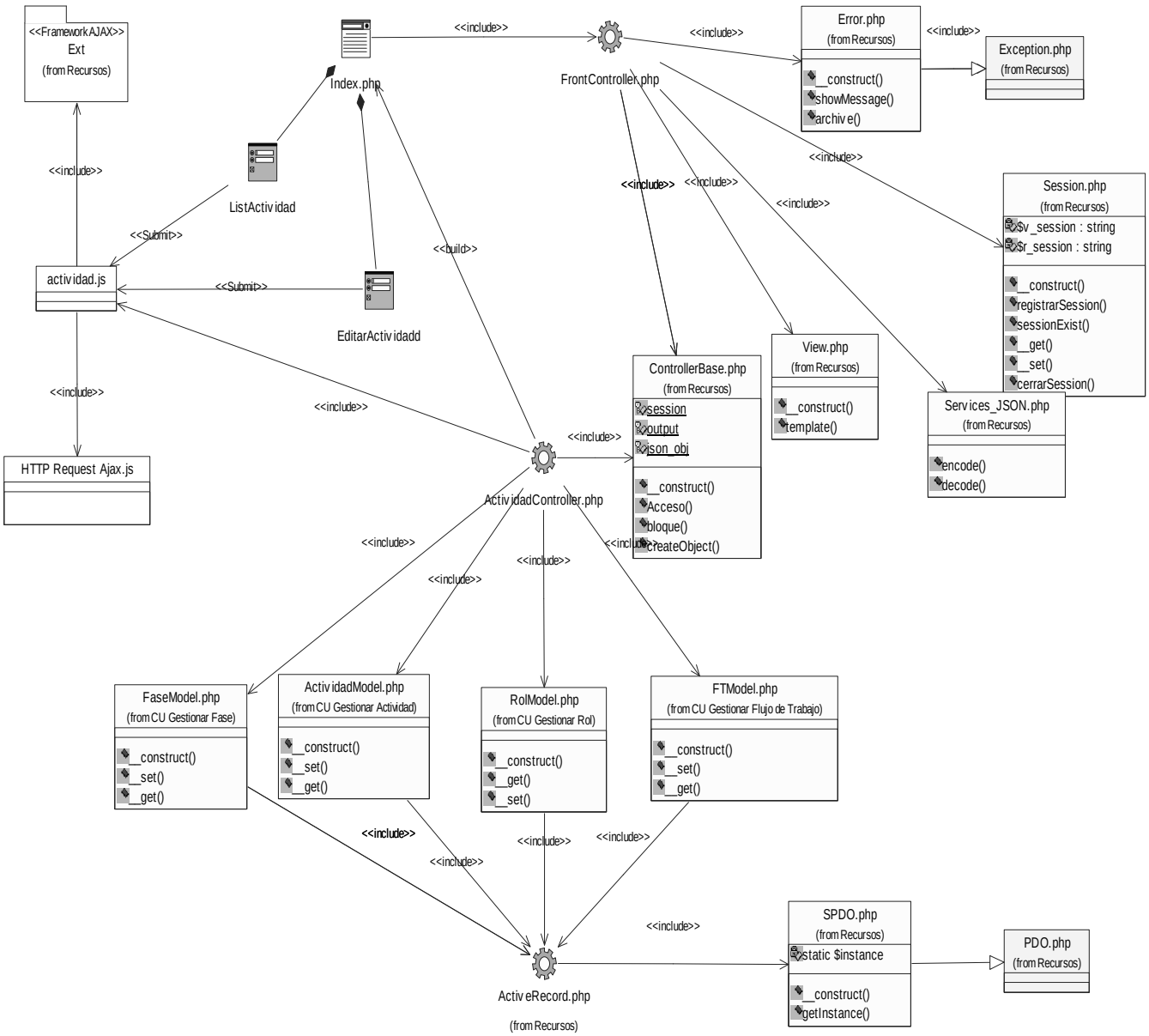
3.3.2 Paquete Seguridad

CU Autenticar Usuario

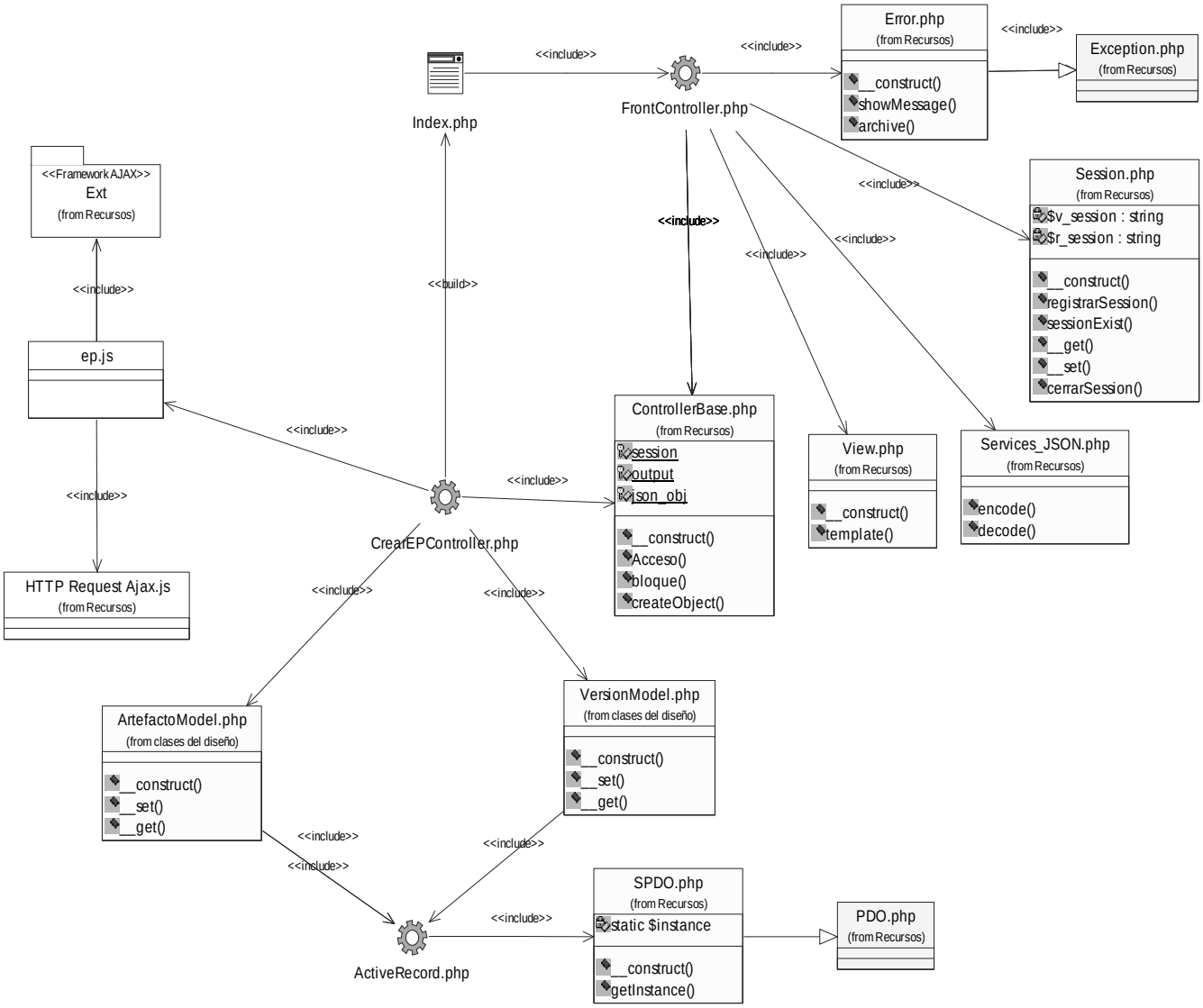


3.3.3 Paquete Sistema

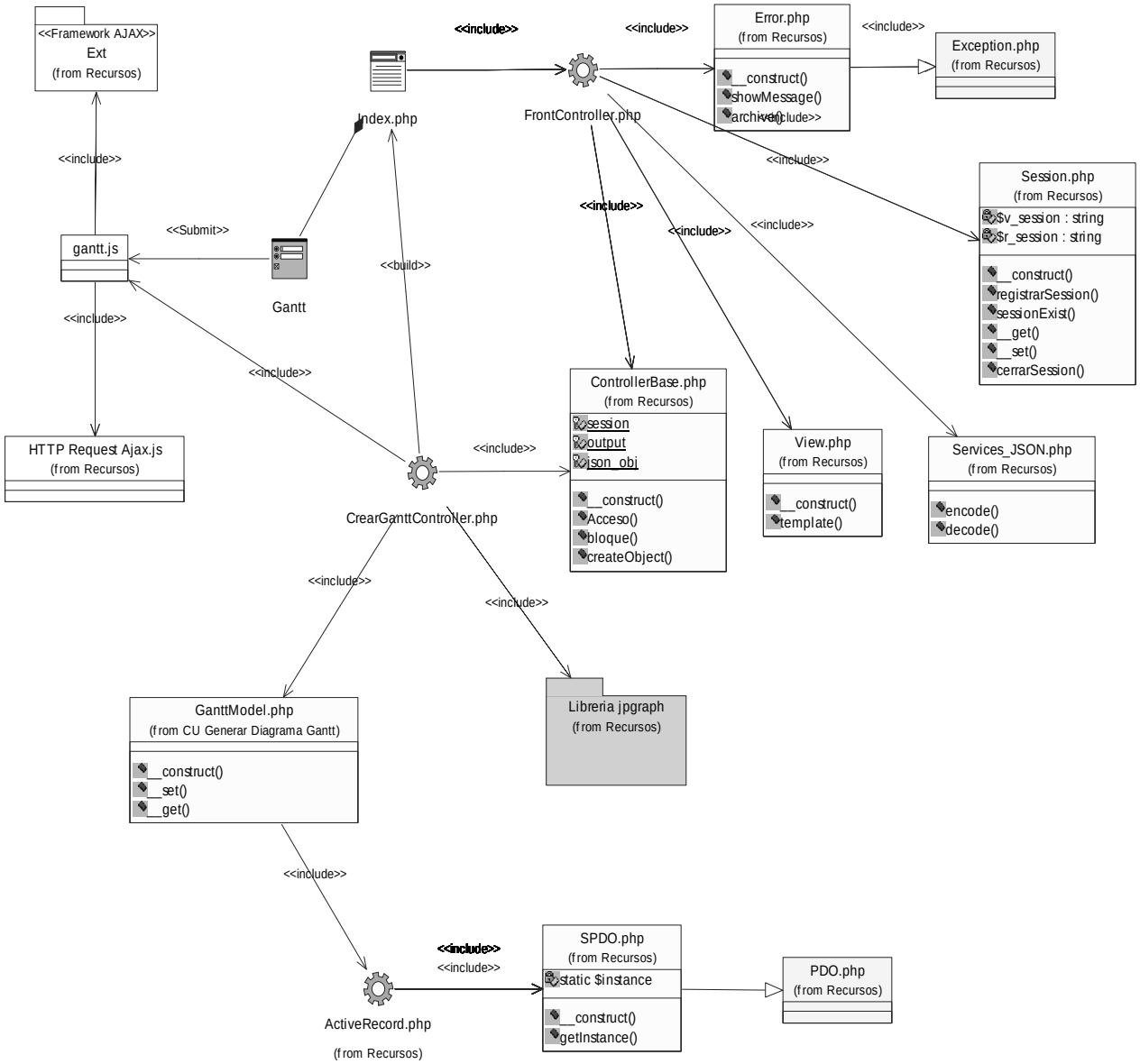
CU Gestionar Actividad



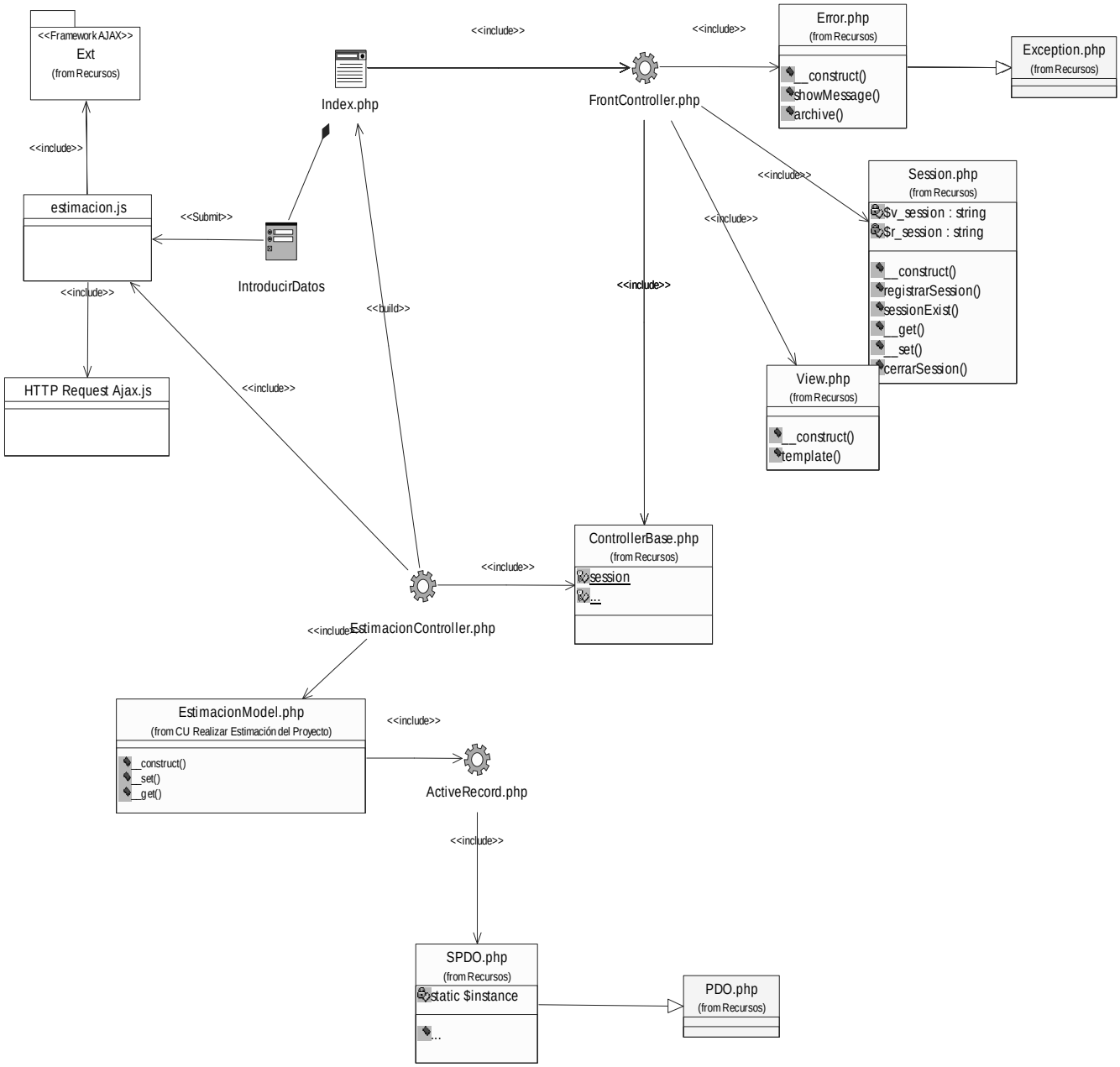
CU Crear Expediente de Proyecto



CU Generar Diagrama Gantt

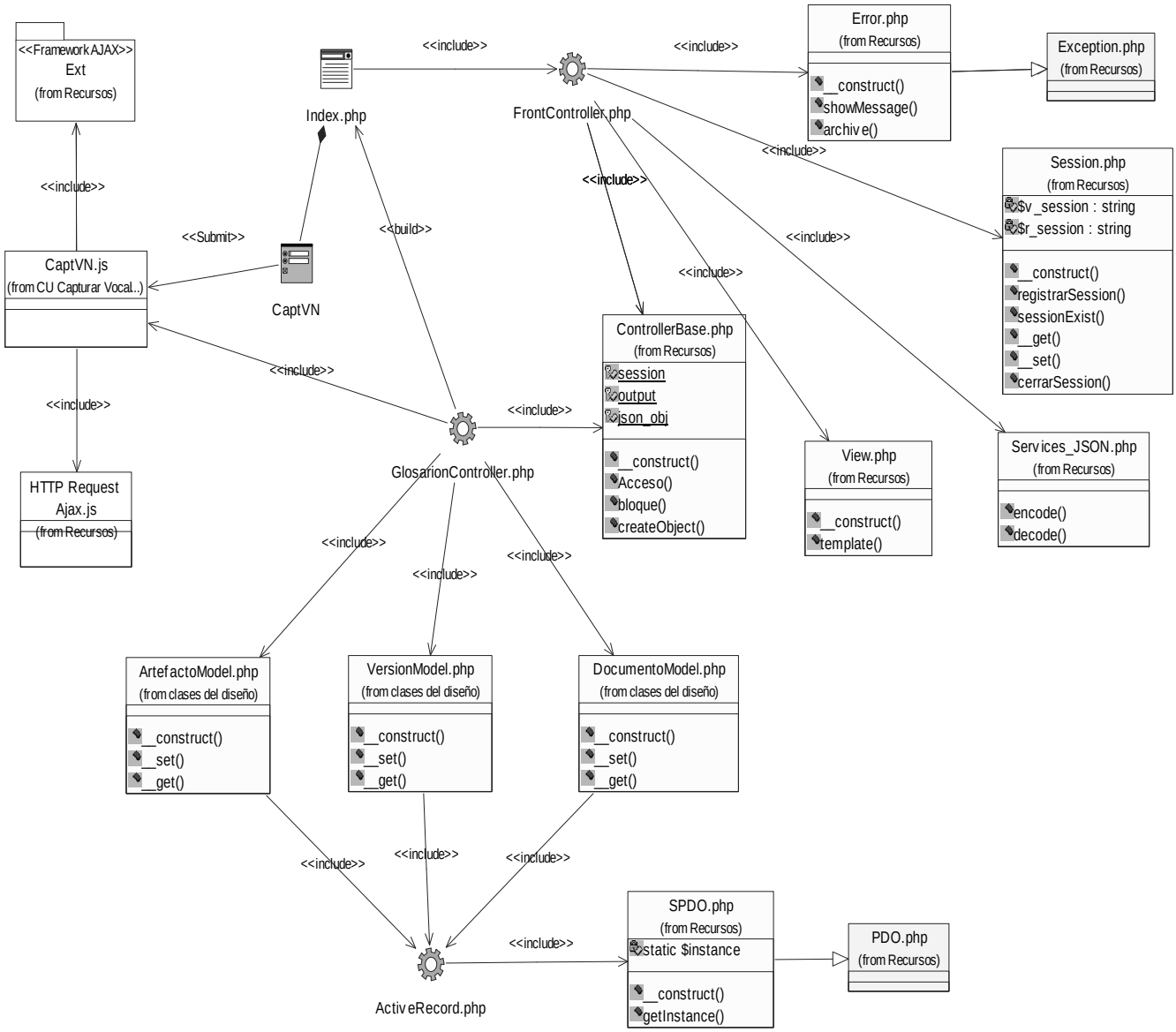


CU Realizar Estimación del Proyecto


<<FrameworkAJAX>>
Ext
(from Recursos)
<<include>>
Index.php
<<include>>
FrontController.php
<<include>>
Error.php
(from Recursos)
__construct()
showMessage()
archive()
Exception.php
(from Recursos)
<<include>>
estimacion.js
<<Submit>>
IntroducirDatos
<<include>>
Session.php
(from Recursos)
$v_session : string
$r_session : string
__construct()
registrarSession()
sessionExist()
__get()
__set()
cerrarSession()
<<include>>
<<include>>
<<include>>
<<build>>
HTTP Request Ajax.js
<<include>>
View.php
(from Recursos)
__construct()
template()
ControllerBase.php
(from Recursos)
session
...
<<include>>
<<include>>
EstimacionController.php
EstimacionModel.php
(from CU Realizar Estimación del Proyecto)
__construct()
__set()
__get()
<<include>>
ActiveRecord.php
<<include>>
SPDO.php
(from Recursos)
static $instance
...
PDO.php
(from Recursos)

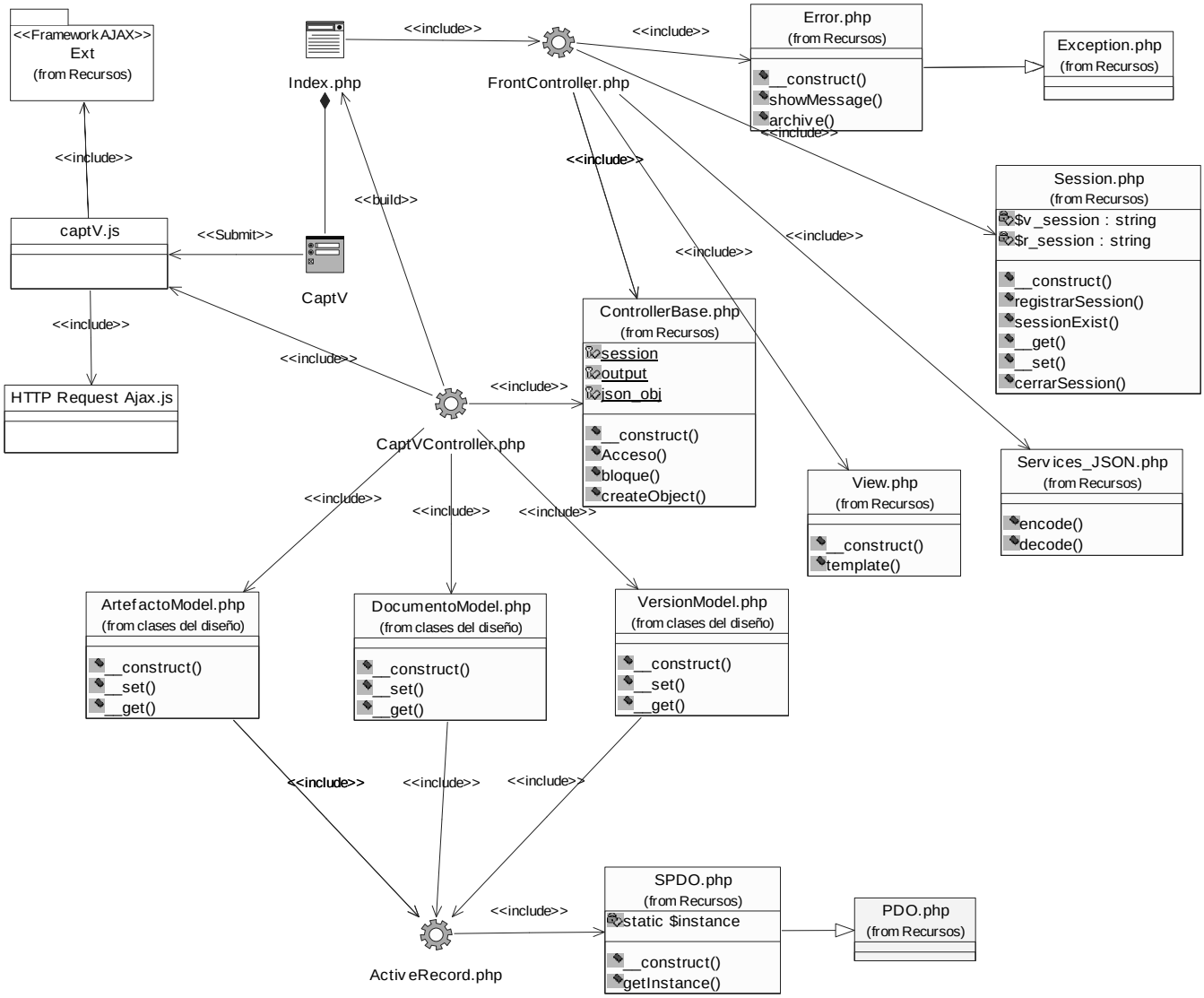
3.3.4 Paquete FT Mod Negocio

CU Capturar Vocabulario Negocio



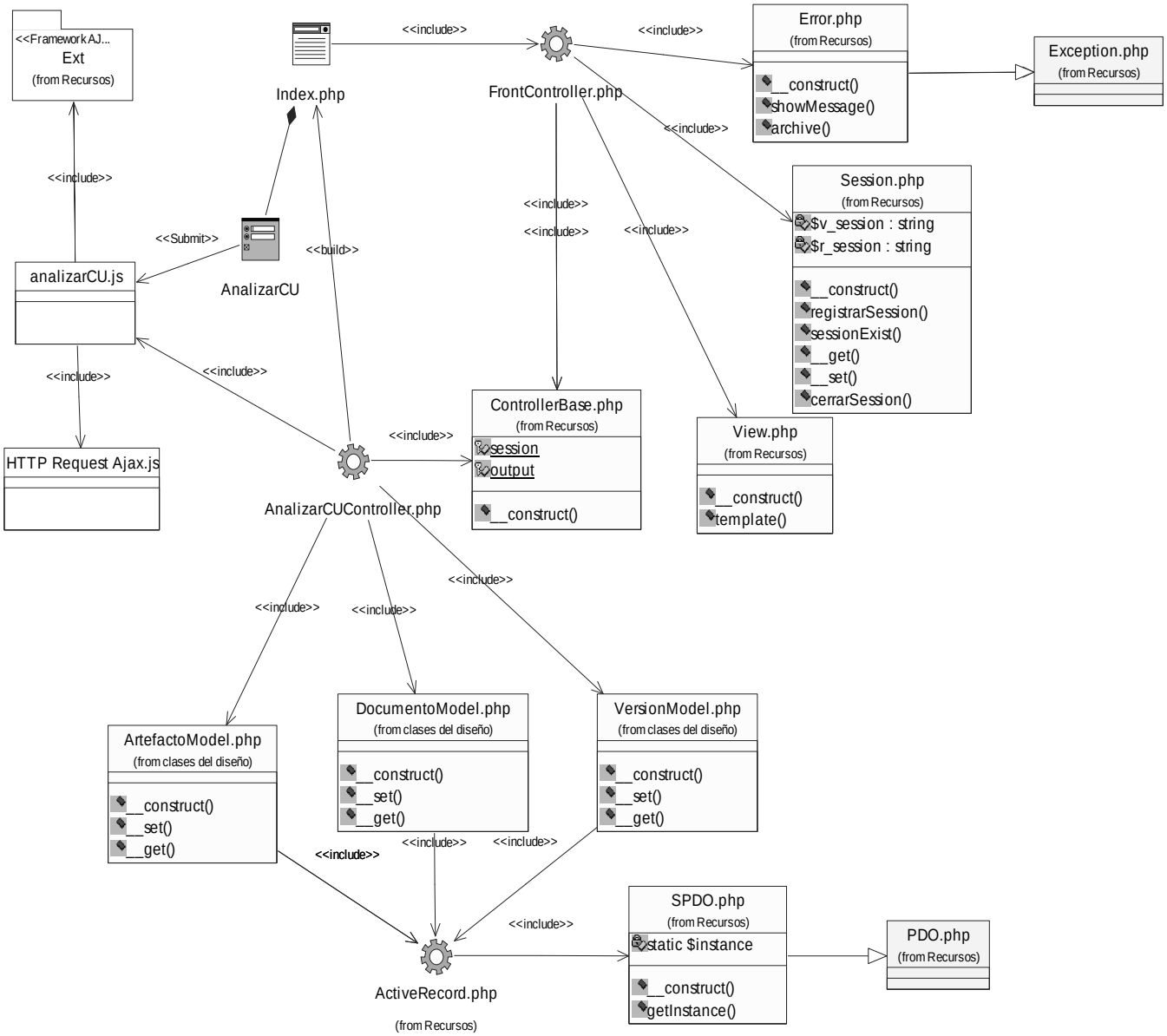
3.3.5 Paquete FT Requerimiento

CU Capturar Vocabulario



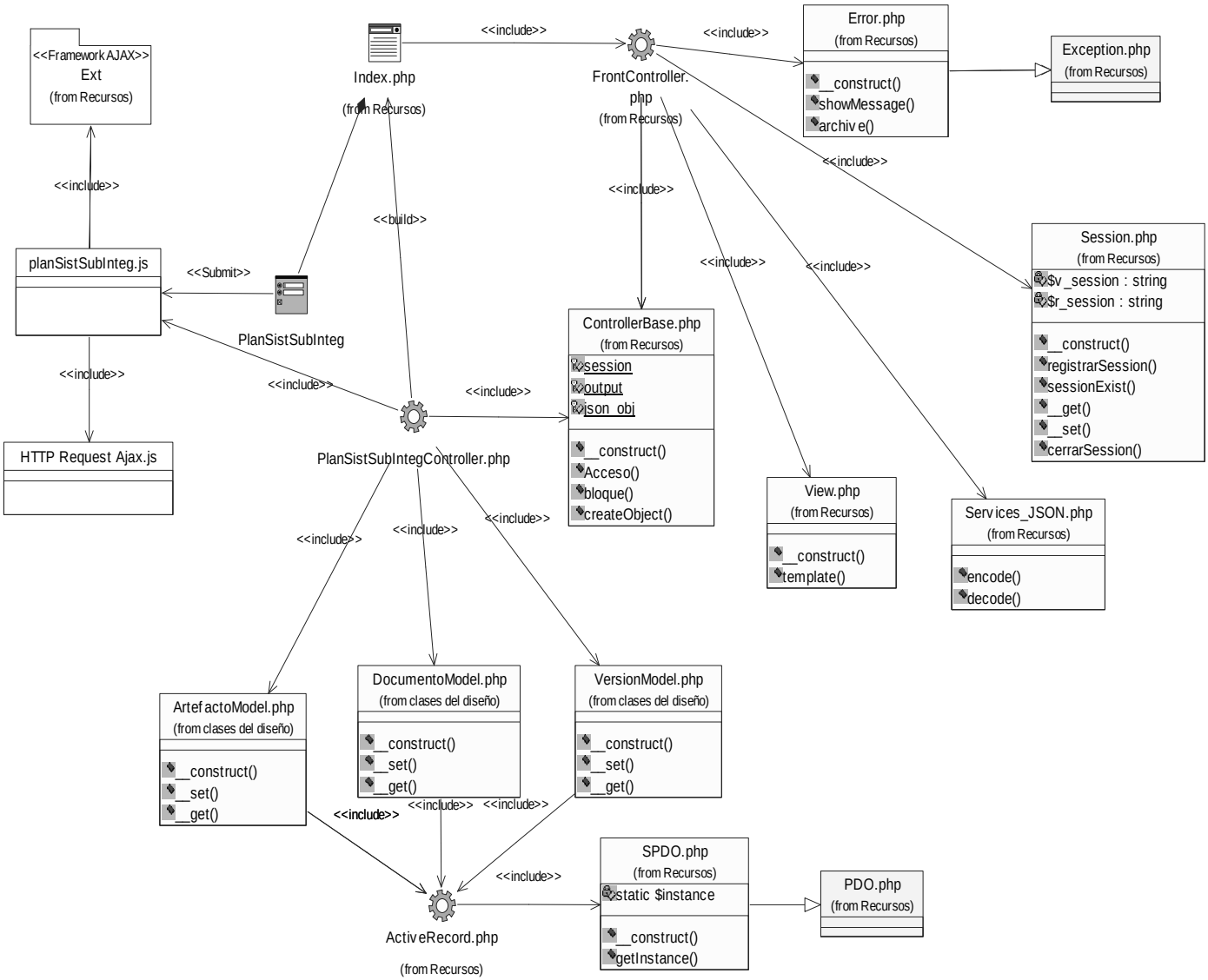
3.3.6 Paquete FT Análisis-Diseño

CU Analizar Caso de Uso



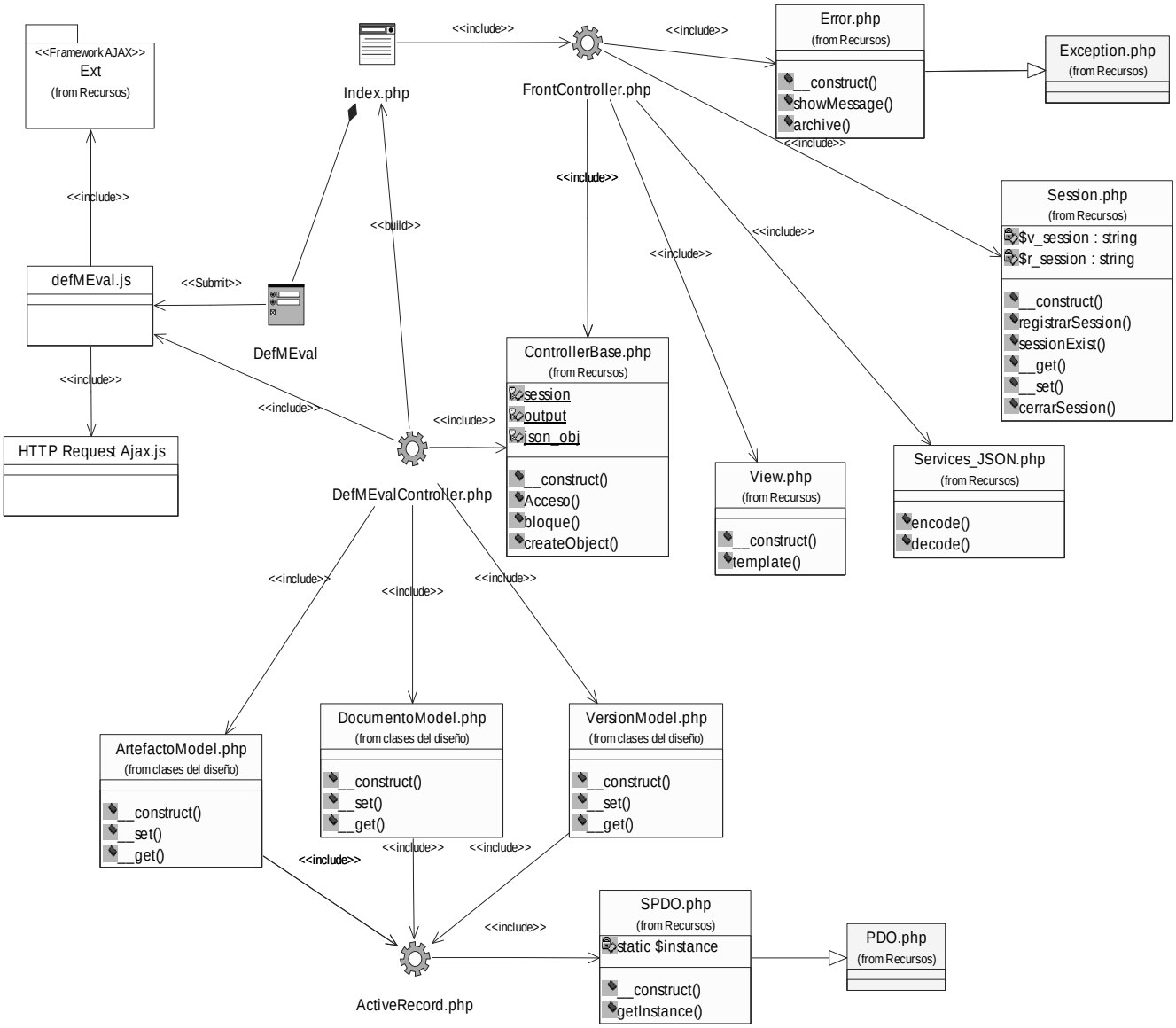
3.3.7 Paquete FT Implementación

CU Elaborar Plan de Sistema y Subsistema de Integración



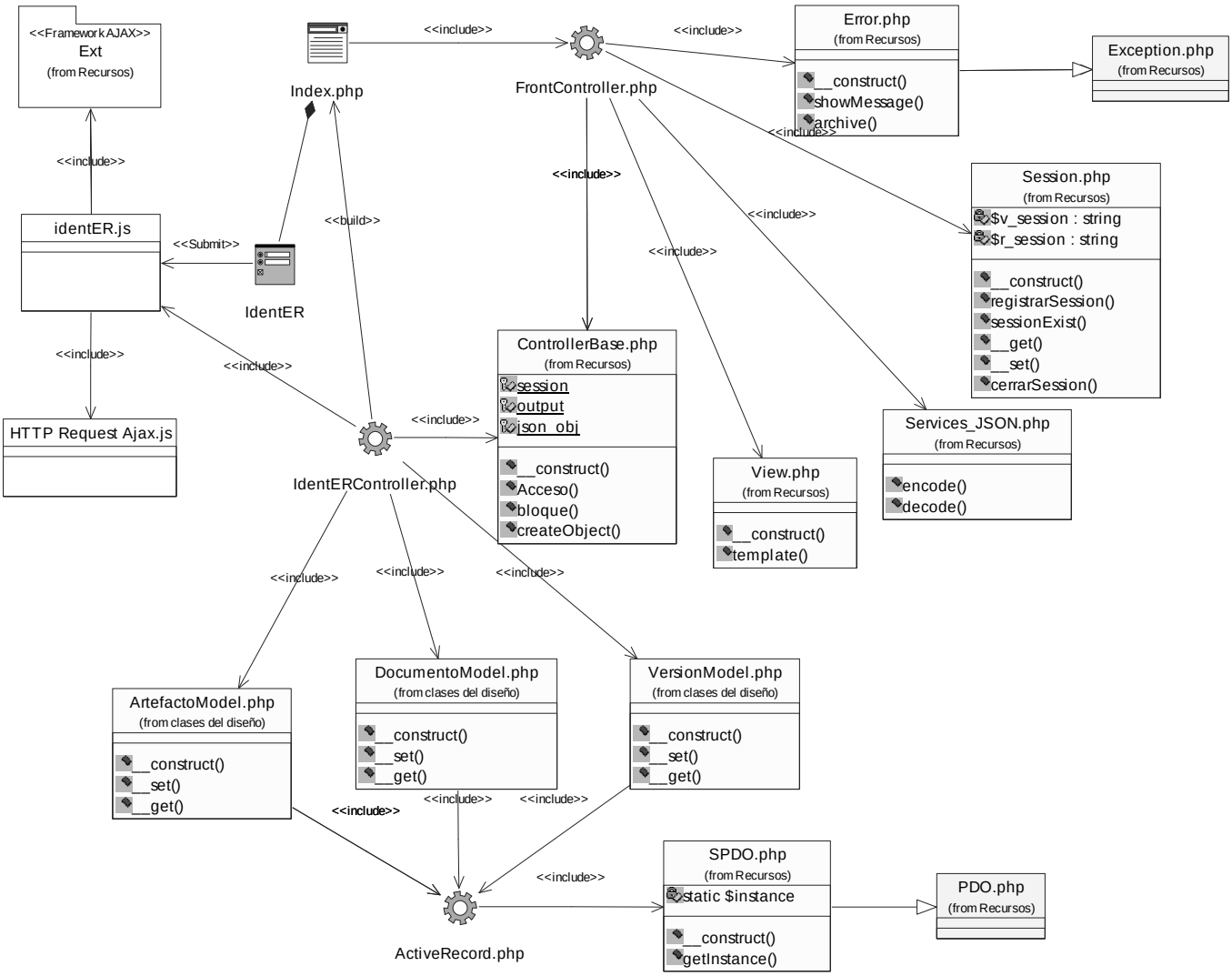
3.3.8 Paquete FT Prueba

CU Definir Misión de la Evaluación



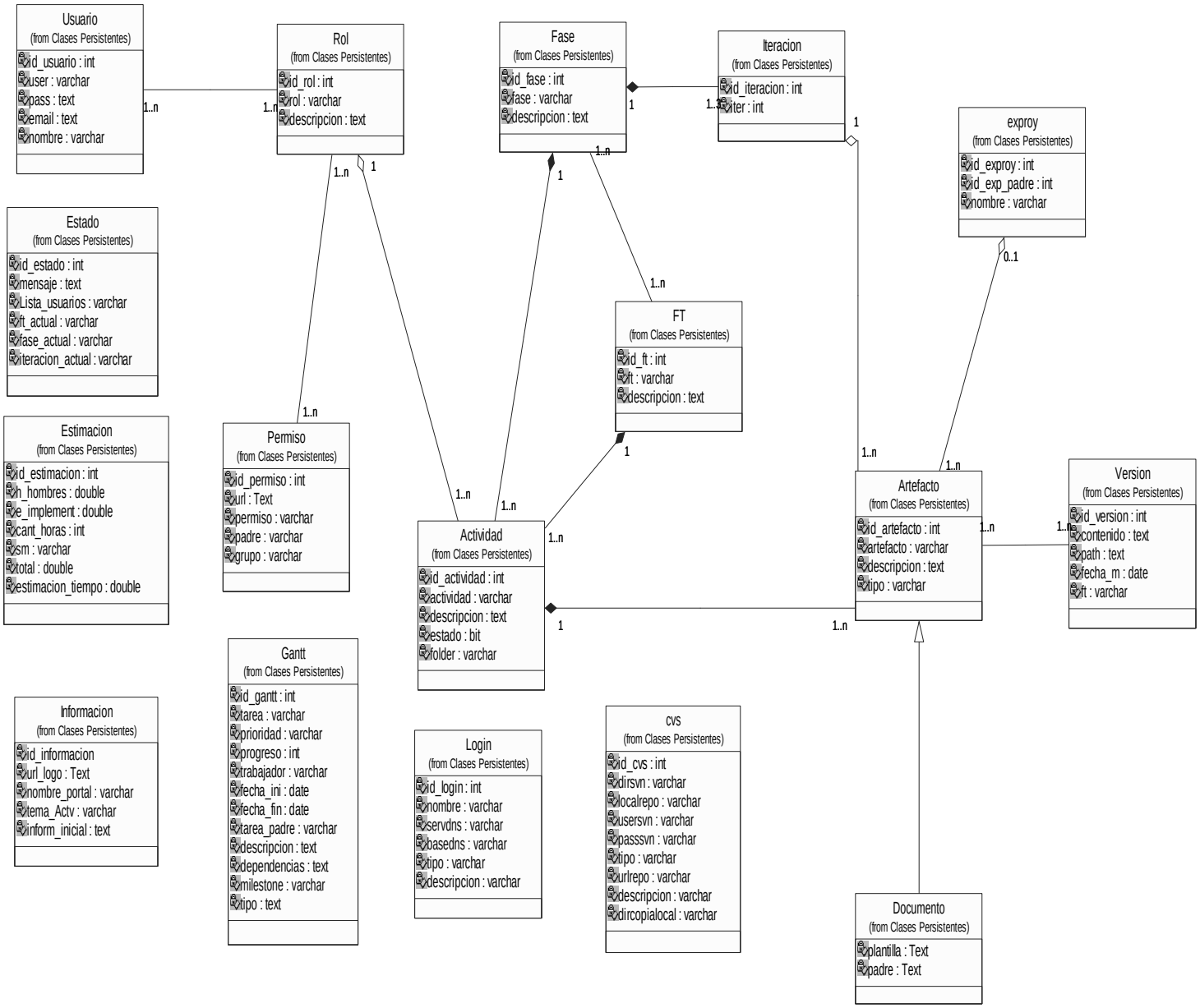
3.3.9 Paquete FT Gestión de Proyecto

CU Identificar y Evaluar Riesgos

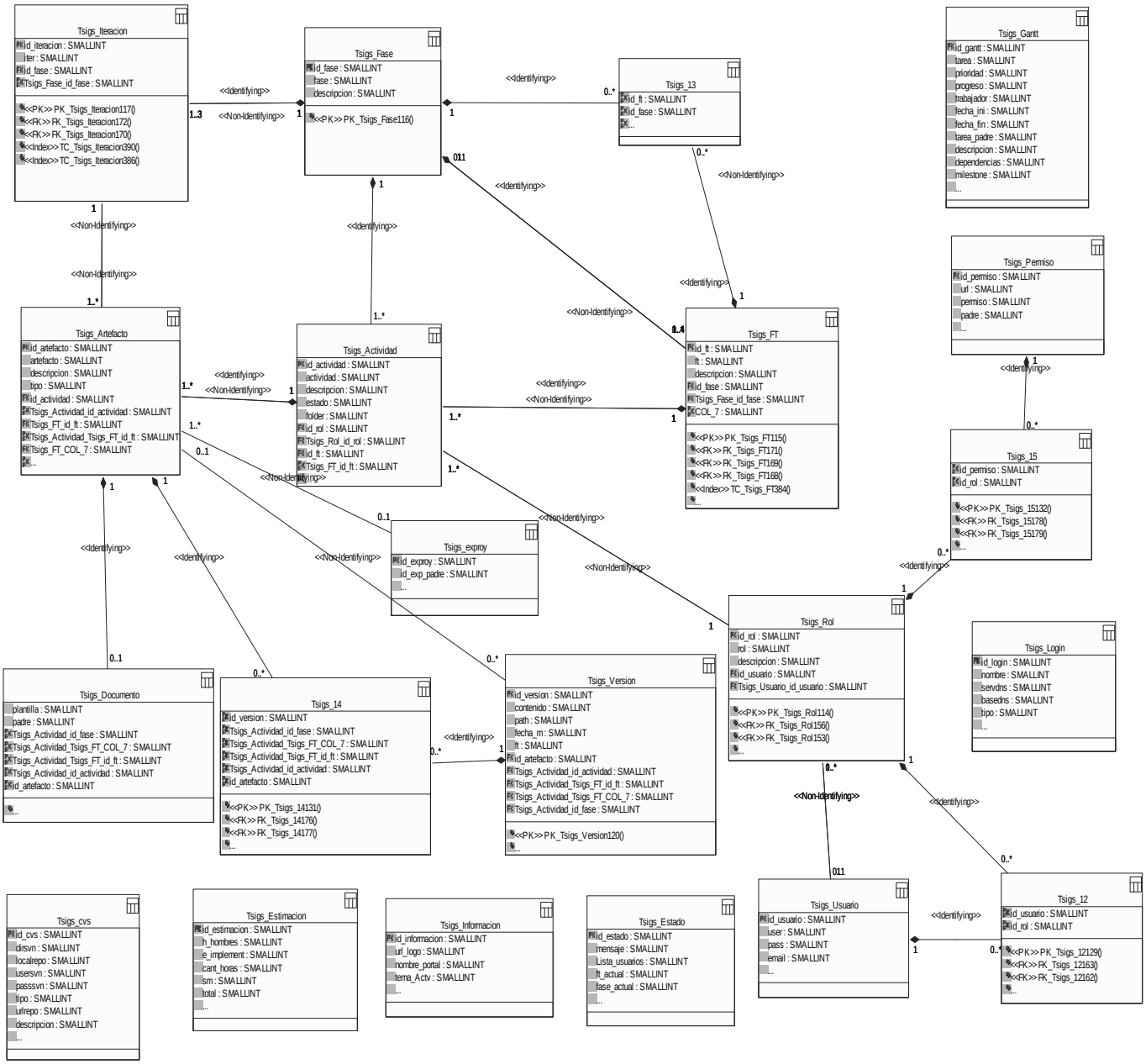


Para no extender en demasía el documento las restantes diagramas se encuentran en el **Anexo 3**.

3.4 Diagrama de clases persistentes



3.5 Modelo de Datos.



3.6 Descripción de las clases.

Nombre: ActiveRecord	
Tipo de clase :Entidad	
Atributo	**Tipo**
$_pdo	
$_classname	
$_classModel	
$table	
$keyField	
Para cada responsabilidad:	
Nombre:	public function __construct() public function setup() public function nomClass() public function getPrimaryKey() public function findByPk($id) public function add($fields, $function_fields = array()) public function update($fields, $strict = true, $function_fields = array()) public function delete($records, $and = null) public function find_all($where = NULL, $order_by = 'ASC', $condition = null , $herencia = null, $limit_offset = null)

	public function FindBySql($query, $condition = null) public function create($array_val = null) public function cantTuplas() public function query($query, $id = null) public function queryUser($query) public function join($array_select, $array_tables, $where, $where_datos, $condition = null, $order_by = NULL) public function ultimate($string) public function buscar($value, $query) public function lastIdInsert($fields)
Descripción:	Es la clase encargada de manejar los datos utilizando patrón de arquitectura ActiveRecord

Nombre: SPDO	
Tipo de clase :Entidad	
Atributo	**Tipo**
$instance	
Para cada responsabilidad:	
Nombre:	public function __construct() public function getInstance()

Descripción:	Es la clase encargada de manejar los datos utilizando patrón de arquitectura singleton herdando de la clase PDO de PHP

Nombre: ControllerBase	
Tipo de clase :Entidad	
Atributo	**Tipo**
$view	
$session	
$json_obj	
$output	
Para cada responsabilidad:	
Nombre:	public function __construct() public function bloque() public function createObject($dir_file, $classname) public function Acceso($permiso) public function AccesoMod($actividad) public function eliminarDir($carpeta)
Descripción:	Esta clase permite que todos los controladores hereden de ella

Nombre: Error	
Tipo de clase :Entidad	
Para cada responsabilidad:	
Nombre:	public function __construct($message, $code = null) public function showMessage() public function archive()
Descripción:	Esta clase permite el tratamiento de errores

Nombre: FrontController	
Tipo de clase :Controladora	
Para cada responsabilidad:	
Nombre:	public function main() public function folders($name)
Descripción:	Esta clase permite como controlador general delegar todas acciones hacia los controladores utilizando para ello una arquitectura MVC

Nombre: Services_JSON
Tipo de clase :Entidad
Para cada responsabilidad:

Nombre:	public function encode () public function decode () public function utf162utf8($utf16) public function utf82utf16($utf8) public function name_value($name, $value) public function reduce_string($str) public function isError($data, $code = null)
Descripción:	Esta clase se encarga de enviar los datos desde el servidor(PHP) hacia el cliente(EXTjs)

Nombre: Session	
Tipo de clase :Entidad	
Atributo	**Tipo**
$v_session	
$r_session	
$instance	
Para cada responsabilidad:	
Nombre:	public function __construct() public function singleton ()

	public function cerrarSession() public function registrarSession($session ,$valor) public function sessionExist($session) public function __get($atribute) public function __set($atribute, $value)
Descripción:	Esta clase permite manejar una parte de la seguridad del sitio

Nombre: View	
Tipo de clase :Entidad	
Para cada responsabilidad:	
Nombre:	public function template($plantilla, $obj, $index = null)
Descripción:	Motor de plantillas

Para no extender en demasía el documento las descripciones de las clases se encuentran en el **Anexo 4**.

3.7 Conclusiones

Se finaliza la etapa de análisis del sistema, realizando el modelo de clases de Análisis. También se concluye así con la etapa de diseño del sistema. Se graficaron los diagramas de diseño web del sistema y se graficaron además los diagramas de clases persistentes y modelo de datos que posibilitan una mayor claridad a la hora de implementar. Se describieron las clases involucradas en el desarrollo del sistema. Con todos estos elementos, se tiene una idea más precisa sobre los elementos constitutivos del sistema.

Capítulo 4

Implementación y Prueba

4.1 Introducción

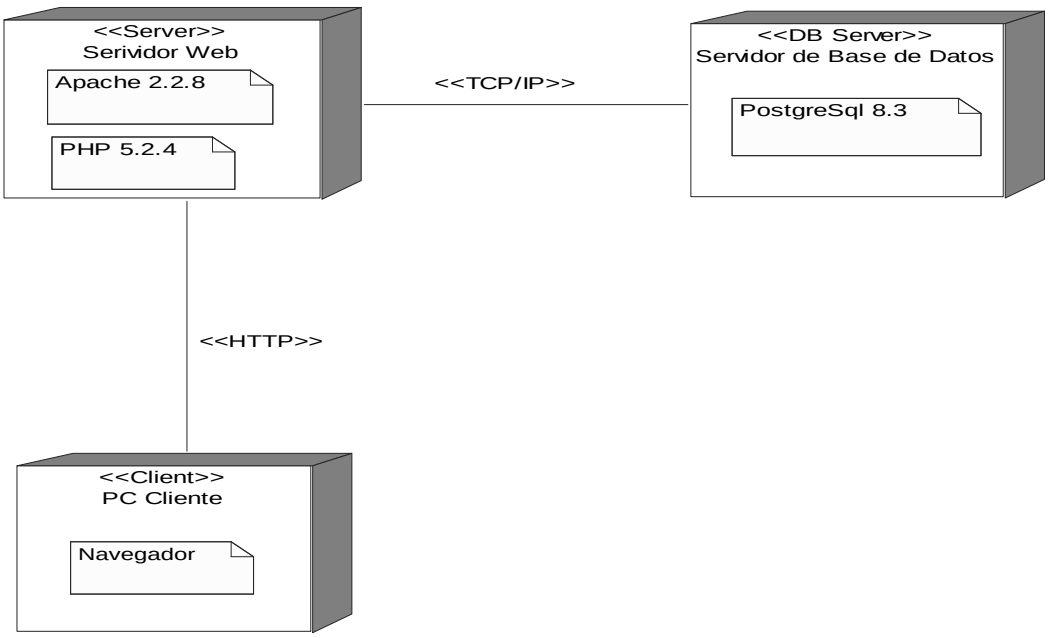
En el capítulo anterior se modelaron los diagramas de clases de análisis y los diagramas de clases web. En este capítulo se pasa a la implementación del software en el lenguaje de programación escogido anteriormente, se desarrolla el código de una manera certificada, definiendo estándares de programación, codificando y se realizan pruebas unitarias del producto y se modelaran los diagramas de despliegues y componentes para una mayor vista para la implementación del producto.

4.2 Modelo de Despliegue

El modelo de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. Se utiliza como entrada fundamental en las actividades de diseño e implementación debido a que la distribución del sistema tiene una influencia principal en su diseño.

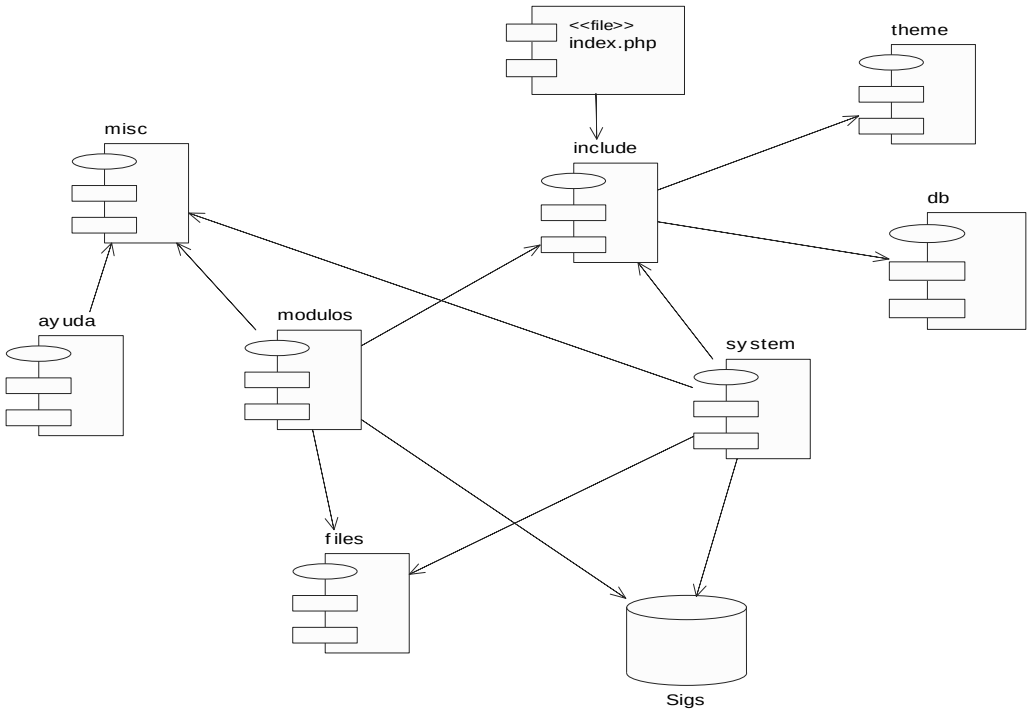
Elementos:

- Procesadores: Nodos que tienen capacidad de procesamiento, computadoras por lo general.
- Dispositivos: Nodos que no tienen capacidad de procesamiento.
- Protocolos: Estándares que deben existir implementados en la red entre máquinas, para efectuar cierta comunicación.

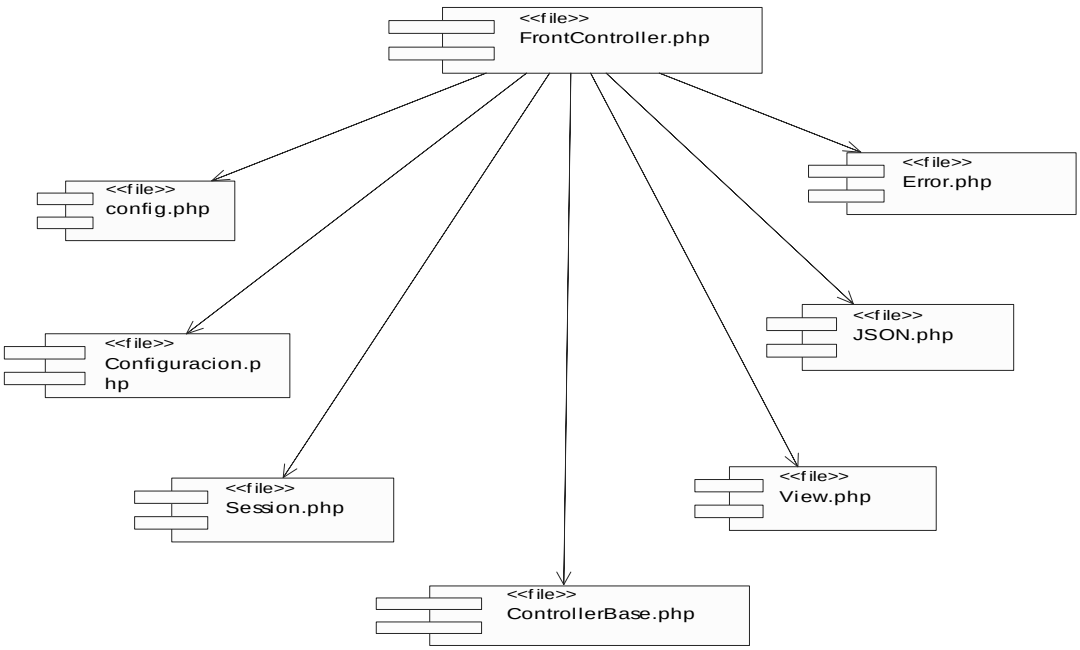


4.3 Diagrama de componentes

4.3.1 Diagrama de Componentes



4.3.2 Diagrama de paquete include



Para no extender en demasía el documento las restantes descripciones se encuentran en el **Anexo 5**.

4.4 Modelo de prueba

4.4.1 Pruebas de Caja Negra

¿Qué es una prueba?

La prueba del software es un conjunto de actividades que se lleva a cabo sistemáticamente, que puede planificarse por adelantado y ejecutar una vez construido el código para la revisión final de las especificaciones, del diseño y de la codificación del software.

Cualquier producto de ingeniería puede ser probado mediante una de estas técnicas:

1. Conociendo la funcionalidad específica para la cual fue diseñado el producto, se pueden llevar a

cabo pruebas que demuestren que cada función es completamente operativa. Pruebas de caja negra.

2. Conociendo el funcionamiento del producto se pueden desarrollar pruebas que aseguren que “todas las piezas encajen”, o sea, que la operación interna se ajusta a las especificaciones y que todos los componentes internos se han comprobado de forma adecuada. Pruebas de caja blanca.

1. El método que se usó para el diseño de los casos de pruebas fue una variante del de particiones equivalentes, que es un tipo de prueba de caja negra, propuesto por RUP, este método consta de 3 pasos fundamentales y se basa totalmente y usa como artefacto de entrada los casos de usos y opcionalmente el documento de reglas del negocio. Los pasos son:

    - Para cada caso de uso, generar un sistema completo de escenarios.
    - Identificar los casos de prueba.
    - Identificar valores de datos para las pruebas.

4.4.2 Caso de Uso: Gestionar Flujo de Trabajo

1. Sistema completo de escenarios

Nombre del escenario	Flujo donde empieza	Alternativo
Escenario1 “Registro exitoso”	Flujo Básico	
Escenario 2 “Faltan datos obligatorios”	Flujo Básico	A1
Escenario 3 “Ya existe el flujo de trabajo”	Flujo Básico	A2
Escenario 4 “Modificación exitosa”	Flujo Básico	

Escenario 5 “Faltan datos en la modificación”	Flujo Básico	A3
Escenario 6: “No selección del elemento a modificar”	Flujo Básico	A4
Escenario 7 “Eliminación exitosa”	Flujo Básico	
Escenario 8: “No selección del elemento a eliminar”	Flujo Básico	A5

Tabla # 1 Matriz Parcial de escenarios

2. Casos de Pruebas

Sección Adicionar Flujo de Trabajo

Id del escenario	**Escenario**	**Nombre del FT**	**Descripción del FT**	**Respuesta del Sistema**
EC 1	Escenario 1: “Registro exitoso”	V	V	El sistema muestra un mensaje informándole al administrador que ya ha sido efectuado el registro del flujo de trabajo.
EC 2	Escenario 2: “Faltan datos obligatorios”	Alguno de los datos toma valor I.		El sistema emite un mensaje para que llene los campos obligatorios.
EC 3	Escenario 3: “Ya existe el	Nombre existente	V	El sistema muestra un mensaje informativo.

	flujo de trabajo”			

Tabla # 2 Matriz de Casos de Prueba de la sección “Adicionar Flujo de Trabajo”

Sección Editar Flujo de Trabajo

Id del escenario	Escenario	Nombre del FT	Descripción del FT	Respuesta del Sistema
EC 4	Escenario 4: “Modificación exitosa”	V	V	El sistema muestra un mensaje informándole al administrador que ya se han actualizado los datos.
EC 5	Escenario 5: “Faltan datos en la modificación”	Alguno de los datos toma valor I.		El sistema emite un mensaje para que llene los campos obligatorios.
EC 6	Escenario 6: “No selección del elemento a modificar”	N/A.	N/A.	El sistema muestra un mensaje informando que seleccione el elemento a modificar.

Tabla # 3 Matriz de Casos de Prueba de la sección “Editar Flujo de Trabajo”

Sección Eliminar Flujo de Trabajo

Id del escenario	Escenario	Respuesta del Sistema
EC 7	Escenario: 7 “Eliminación exitosa”	El sistema emite un mensaje confirmando que se eliminó exitosamente.

EC 8	Escenario 8: “No selección del elemento a eliminar”	El sistema muestra un mensaje informando que seleccione el elemento a eliminar.

Tabla # 4 Matriz de Casos de Prueba de la sección “Eliminar Flujo de Trabajo”

Las celdas de la tabla contienen V, I, o N/A. **V** indica válido, **I** indica inválido, y **N/A** que no es necesario proporcionar un valor del dato en este caso, ya que es irrelevante.

3. Valores de Datos para las Pruebas

Sección Adicionar Flujo de Trabajo

Id del escenario	**Escenario**	**Nombre del FT**	**Descripción del FT**	**Respuesta del Sistema**	**Resultado de la prueba**
EC 1	Escenario 1: “Registro exitoso”	Implementación	Flujo de Trabajo Implementación.	El sistema muestra un mensaje informándole al administrador que ya ha sido efectuado el registro del flujo de trabajo.	Respuesta esperada.
EC 2	Escenario 2: “Faltan datos obligatorios”	“ ”	“ ”	El sistema emite un mensaje para que llene los campos obligatorios	Respuesta esperada.
EC 3	Escenario	Implementació		El sistema muestra un	Respuesta

	3: “Ya existe el flujo de trabajo”	n		mensaje informando que el flujo de trabajo ya existe.	esperada.

Tabla # 5 Matriz de Casos de Prueba con los valores de los datos

Sección Editar Flujo de Trabajo

Id del escenario	**Escenario**	**Selección de un FT a Eliminar**	**Nombre del FT**	**Descripción del FT**	**Respuesta del Sistema**	**Resultado de la prueba**
EC 4	Escenario: 4 “Modificación exitosa”	Implementación	Implementación	Flujo de trabajo que tiene mayor peso el la fase de construcción	El sistema muestra un mensaje informándole al administrador que ya se han actualizado los datos.	Respuesta esperada.
EC 6	Escenario 6: “No selección del elemento a modificar”	“”	N/A	N/A	El sistema muestra un mensaje informando que seleccione el elemento a modificar.	Respuesta esperada.

Tabla # 6 Matriz de Casos de Prueba con los valores de los datos

Sección Eliminar Flujo de Trabajo

Id del escenario	Escenario	Selección de un FT a Eliminar	Respuesta del Sistema	Resultado de la prueba
EC 7	Escenario: 7 "Eliminación exitosa"	Implementación	El sistema emite un mensaje confirmando que se eliminó exitosamente.	Respuesta esperada.
EC 8	Escenario 8: "No selección del elemento a eliminar"	""	El sistema muestra un mensaje informando que seleccione el elemento a eliminar.	Respuesta esperada.

Tabla # 7 Matriz de Casos de Prueba con los valores de los datos

4.5 Conclusiones

Se finaliza la etapa de Implementación del sistema, habiéndose el modelo de despliegues y los diagramas de componentes. También se concluye así con la etapa de prueba del sistema. Se describieron los casos de pruebas del sistema a partir de las descripciones textuales de los casos de usos, definiendo entradas válidas y no válidas, obteniendo como resultado que el sistema cumpla con las funcionalidades descritas por las descripciones de los casos de usos. Con todos estos elementos, se tiene una idea más precisa sobre los elementos constitutivos del sistema que propone.

CONCLUSIONES

En el período de prueba al que fue sometido el Sistema de Ingeniería y Gestión de Software, arrojaron como resultado el cumplimiento del objetivo propuesto en este trabajo. Haciendo posible que los distintos proyectos de desarrollo de software presenten una herramienta más que integre los distintos procesos que se llevan a cabo en el ciclo de vida de un producto. Se considera, que el presente trabajo arrojó el diagnóstico general sobre los procesos de ingeniería y gestión de software en la UCI, así como el análisis, diseño, implementación y prueba de dicho sistema.

El estudio realizado sobre las distintas metodologías de desarrollo (tradicionales y agiles), así como su utilización en el desarrollo de la aplicación, se obtuvo la conclusión de que la metodología RUP se adapta al problema a tratar, debido a que el sistema está orientado a esa metodología, así como la misma es adaptable a cualquier proceso de desarrollo. Para la implementación del sistema y con el objetivo de lograr una mayor extensibilidad y adaptabilidad al entorno de trabajo, se utilizaron herramientas libres como el PostgreSQL y PHP en su versión 5 como lenguaje de programación permitiendo así su implantación en la mayoría de las plataformas, la técnica de AJAX con el framework Ext JS con el fin de brindar un producto mas amigable al usuario final y rapidez en ejecución.

Teniendo en cuenta lo anterior, el Sistema para la Ingeniería y Gestión de Software, se convierte en una herramienta útil para los distintos grupos de desarrollo de software que utilicen los procesos de ingeniería, gestión de la configuración y gestión de proyecto como guías de desarrollo.

RECOMENDACIONES

El resultado arrojado luego de la realización del presente trabajo aporta un conjunto de mejoras al proceso de desarrollo de software en los proyectos productivos existentes en la universidad. En pos de lograr la mejora de los elementos que componen la propuesta, se recomienda el desarrollo continuado de la misma, incrementando su eficiencia a través de la incorporación de nuevas funcionalidades referentes a los procesos de Ingeniería y Gestión de Software. Además se aconseja llevar el desarrollo de la aplicación a las metodologías ágiles como alternativa latente en el desarrollo de software en la actualidad, lo cual permite que la totalidad de los proyectos productivos del centro sean vinculados a la propuesta.

La aplicación debe ser empleada cumpliendo a cabalidad con los requisitos establecidos por la metodología de desarrollo representada, ya que es la única vía existente para mejorar la calidad del producto final, así como la conformidad de los usuarios.

El proceso de implantación de la aplicación debe ser realizado primeramente en proyectos pilotos en pos de verificar el nivel de adaptación de los desarrolladores a la nueva herramienta, posteriormente y teniendo en cuenta los resultados arrojados por este proceso de prueba, la propuesta debe ser extendida a diversos grupos de desarrollo.

BIBLIOGRAFÍA

Referencias Bibliográficas

1. Chile, D.d.C.d.l.C.d.l.U.d. *Ingeniería de Software*. [cited 2007 6 de diciembre 2007]; Available from: http://www.dcc.uchile.cl/web/printer-27562.html.
2. Perfetti, A.L., *Tesis de Magister en Ingeniería de Software*. 2005: Buenos Aires.
3. Peña, R. *Gestión de Proyecto*. 2007 [cited; Available from: http://www.gestiopolis.com/recursos/documentos/fulldocs/ger/gestioproyecto.htm.
4. Sacristán, L. *Aplicaciones de Gestión de Proyectos*. 2007 [cited 2007; Available from: http://sentidoweb.com/2007/05/17/aplicaciones-de-gestion-de-proyectos.php.
5. Morfeo, P., *Estudio de Mejoras y Nuevas Funcionalidades para G-Forge*. 2005, Madrid.
6. *Open Workbench*. 2007 [cited; Available from: http://www.puntoorg.org/open_workbench_gesti_n_de_proyectos_para_windows_.
7. Sanchez, M.A.M., et al., *Metodologías De Desarrollo De Software*. 2005: Perú S.A.C.
8. *Getting Started with UML*. January 02, 2007 [cited; Available from: http://www.omg.org/UML/.
9. *HTML y XHTML*. [cited; Available from: http://www.librosweb.es/xhtml/capitulo1/html_y_xhtml.html.
10. *CSS*. [cited; Available from: http://www.librosweb.es/css/capitulo1/que_es_css.html.
11. *Javascript*. [cited; Available from: http://www.maestrosdelweb.com/editorial/%C2%BFque-es-javascript/.
12. *ASP*. [cited; Available from: http://www.maestrosdelweb.com/editorial/aspintro/.
13. *PERL*. [cited; Available from: http://www.desarrolloweb.com/articulos/541.php.
14. *JSP*. [cited; Available from: http://www.desarrolloweb.com/articulos/831.php.
15. *Python*. [cited; Available from: http://www.desarrolloweb.com/articulos/1325.php.
16. *SGBD*. [cited; Available from: http://www.eubd.ucm.es/html/personales/enred/mantonia/docauto/tema5/tema5.htm.
17. *Microsoft SQL Server*. [cited; Available from: http://es.wikipedia.org/wiki/Microsoft_SQL_Server.
18. *Oracle* [cited; Available from: http://es.wikipedia.org/wiki/Oracle.
19. *MySQL* [cited; Available from: http://es.wikipedia.org/wiki/MySQL.
20. *Internet Information Services* [cited; Available from: http://es.wikipedia.org/wiki/IIS.
21. *Zope*. [cited; Available from: http://es.wikipedia.org/wiki/Zope.
22. *FPDF*. 2006 [cited; Available from: http://www.fpdf.org/.
23. *jpgraph*. 2005.
24. Ben Collins-Sussman, B.W.F., C. Michael Pilato, *Control de versiones con Subversion*. 2004. 301.
25. *Frameworks ExtJS*. 2007 [cited; Available from: http://extjs.com.
26. Larman, C., *UML y Patrones*. 1999. 357-359.
27. [cited; Available from: http://www.phpizza.com/es/.
28. Company, T.P. *Zend Studio 5 LAS SOLUCIONES MÁS COMPLETAS PARA EL DESARROLLO DE PHP*. 2007 [cited; Available from: http://www.zend.com/en/products/studio/.
29. Company, T.P. *Performance, Management, Integration, and Enterprise Scalability*. 2007 [cited; Available from: http://www.zend.com/en/products/platform/.

30. *EMS SQL Management Studio for PostgreSQL*. [cited; Available from: http://www.sqlmanager.net/en/products/studio/postgresql.
31. *Acerca de Dreamweaver*. [cited; Available from: http://www.tci.cl/cursos/acerca_de.php.
32. Incorporated, A.S. *Información sobre la integración de las funciones de Studio 8*. 2007 [cited; Available from: http://www.adobe.com/es/products/studio/integration/.
33. *Rational Rose Enterprise*. [cited; Available from: http://www-306.ibm.com/software/awdtools/developer/rose/enterprise/index.html.

GLOSARIO DE TÉRMINOS

.Net: Es un proyecto de Microsoft para crear una nueva plataforma de desarrollo de software con énfasis en transparencia de redes, con independencia de plataforma de hardware y que permita un rápido desarrollo de aplicaciones.

Arquitectura Cliente-Servidor: Consiste básicamente en que un programa -el cliente- realiza peticiones a otro programa -el servidor- que le da respuesta. Aunque esta idea se puede aplicar a programas que se ejecutan sobre una sola computadora, es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de computadoras.

Bug: Un defecto de software (computer bug en inglés), es el resultado de un fallo o deficiencia durante el proceso de creación de programas de ordenador o computadora (software).

BSD: Son las iniciales de Berkeley Software Distribution (en español, Distribución de Software Berkeley) y se utiliza para identificar un sistema operativo derivado del sistema Unix nacido a partir de las aportaciones realizadas a ese sistema por la Universidad de California en Berkeley.

Control de Versiones: Una versión, revisión o edición de un producto, es el estado en el que se encuentra en un momento dado su desarrollo o modificación. Se llama control de versiones a la gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración del mismo.

Copyright: El derecho de autor es un conjunto de normas y principios que regulan los derechos morales y patrimoniales que la ley concede a los autores (los derechos de autor), por el solo hecho de la creación de una obra literaria, artística o científica, tanto publicada o que todavía no se haya publicado.

DOM: Document Object Model (una traducción al español para nada literal, pero apropiada, podría ser Modelo en Objetos para la representación de Documentos), abreviado DOM, es esencialmente un modelo computacional a través de la cual los programas y scripts pueden acceder y modificar dinámicamente el contenido, estructura y estilo de los documentos HTML y XML.

FTP: (File Transfer Protocol) es un protocolo de transferencia de archivos entre sistemas conectados a

una red TCP basado en la arquitectura cliente-servidor, de manera que desde un equipo cliente nos podemos conectar a un servidor para descargar archivos desde él o para enviarle nuestros propios archivos independientemente del sistema operativo utilizado en cada equipo.

GNU: El proyecto GNU fue iniciado por Richard Stallman con el objetivo de crear un sistema operativo completamente libre.

GNU GPL: GNU General Public License o simplemente su acrónimo del inglés GNU GPL, es una licencia creada por la Free Software Foundation a mediados de los 80, y está orientada principalmente a proteger la libre distribución, modificación y uso de software.

HTTP/HTTPS: El protocolo de transferencia de hipertexto (HTTP, HyperText Transfer Protocol) es el protocolo usado en cada transacción de la Web (WWW). Hypertext Transfer Protocol Secure (en español: Protocolo seguro de transferencia de hipertexto), mas conocido por su acrónimo HTTPS, es un protocolo de red basado en el protocolo HTTP, destinado a la transferencia segura de datos de hipertexto, es decir, es la versión segura de HTTP.

Hibernate: Es una herramienta de Mapeo objeto-relacional para la plataforma Java (y disponible también para .Net con el nombre de NHibernate) que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) que permiten establecer estas relaciones.

IDE: Un entorno de desarrollo integrado o en inglés Integrated Development Environment ('IDE') es un programa compuesto por un conjunto de herramientas para un programador.

LAMP: El acrónimo LAMP se refiere a un conjunto de subsistemas software necesarios para alcanzar una solución global, en este caso configurar sitios web o Servidores dinámicos con un esfuerzo reducido. Esto se consigue mediante la unión de las siguientes tecnologías Linux, Apache, MySQL y (Perl, PHP o Python).

Linux: Es un sistema operativo tipo Unix (también conocido como GNU/Linux) que se distribuye bajo la Licencia Pública General de GNU (GNU GPL), es decir que es software libre.

MacOS X: Es el actual sistema operativo de la familia de ordenadores Macintosh.

SQL: Lenguaje de consulta estructurado (SQL Structured Query Language) es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones sobre las mismas.

W3C: World Wide Web Consortium, abreviado W3C, es un consorcio internacional que produce estándares para la World Wide Web.

Wiki: Es un sitio web cuyas páginas web pueden ser editadas por múltiples voluntarios a través del navegador web. Los usuarios pueden crear, modificar o borrar un mismo texto que comparten.