

Universidad de las Ciencias Informáticas

Facultad 3



Título: Análisis y diseño de un sistema de evaluación del desempeño de la Facultad 3.

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

Autor(es): Osmar Leosvanys Febles Urquiza.

Noel Martinez Ramirez.

Tutor(es): Ing. Disnier Alberto Camejo Domínguez

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

<Nombre autor> <nombre tutor>

Firma del Autor Firma del Tutor

Agradecimientos

A toda nuestra familia, por todo el Cariño y el Amor que nos han dado, y contribuir a nuestra formación como persona.

A nuestro Tutor, por todo el apoyo que nos brindó durante el desarrollo de la tesis, y su abnegada preocupación por el trabajo y por nuestras necesidades.

A nuestros Amigos por estar siempre pendientes a nosotros, y que ayudaron de una manera u otra a la formación de este trabajo.

A la Revolución, por formar esta escuela, y darnos la oportunidad de hacer otra familia y de graduarnos como Ingeniero Informático.

En General a todas las personas que hicieron posible este logro, y que han formado parte de nuestras vidas, que sepan que aquí tienen dos Amigos para Toda la Vida que pueden contar con nosotros, de veras mil Gracias.

Dedicatoria Noel

A mis Padres y mi Abuela, por ser los responsables de todos mis logros, por educarme de la mejor manera posible, de veras Gracias, ustedes son los mejores. A mi hermano que siempre está apoyándome en todo, a Mis Tíos, Mis Primos y a toda la familia de mi Mujer por hacerme saber en todo momento que puedo contar con ellos. A mi Mujer Orquídea por estar junto a mí a lo largo de estos cinco años, en los momentos malos y en los buenos, te dedico este trabajo a Ti, mi Amor y Gracias por darme ese Hijo que es lo que se recogió de todo el Amor y la Ternura que hemos cosechado en todos estos años.

Dedicárselo principalmente a mi Hermano de años, Néstor Alain, por ser la persona que ha estado conmigo desde hace 8 años en todo momento, apoyándonos mutuamente y saliendo de los problemas juntos. A los negros de la Calle, Gracias hermanos por enseñarme el verdadero concepto de amistad y juntos conocer la calle.

A mis tres hermanitos que vienen conmigo desde primer año, Wilkie, Pedro y Lennon, Gracias de verdad de a corazón por su amistad incalculable, por todos los momentos que hemos pasado y por estar siempre cuando los necesité, que sepan que en Playa tienen un hermano que nunca los olvidara. A mis hermanos de la Aldea, Dayniel, Delio, Carlos, Roberto y el Cory, por ser aldeanos igual que yo y por permitir que entre todos formáramos una familia, a los de mi grupo y a todos mis amigos, Iliana, J Carlos, Chung, Husein, Hiran, Yuli, Julio, Anier, Eliover, Niudis, Nodalis, Dunia, Yoalis, Isel y Darian, a todos los que me ayudaron cuando estaba en cama, de verdad mil Gracias sin ustedes no hubiera podido aprobar el semestre. A mi compañero de Tesis, ese guajiro cerrado que se convirtió en la persona con la que mejor he trabajado y en un hermano mas.

A mi Hijo "Kelly Brayan", por ser desde ya, mi principal razón de ser y por ser lo mejor que me ha pasado en la vida, esto es para ti Hijo mío, Te Quiero.

De forma general, a todas las personas que de una manera u otra han formado parte de mi vida, a los que no han valorado mi amistad, gracias, de ustedes también aprendí y a los que sí la valoraron, Gracias, Gracias, Gracias, Gracias por existir.

Big Noe.

Dedicatoria Omar

Le dedico este trabajo principalmente a la persona más importante para mí y a la que ha sabido comportarse correctamente en todos los momentos decididos de mi vida, que no se te olvide nunca mami esto es para ti (DALIA ROSA) mi madre, tú fuiste la encargada de darme todo el ánimo para realizar este trabajo y poder pasar estos 5 años de mi vida en la universidad que sinceramente han sido los mejores de mi vida gracias MAMI.

También es tuyo lber por tener siempre fe en mí y mantener siempre la responsabilidad de hermano mayor y aprovecho para decirte que siempre vas hacer mi ejemplo a seguir, TE QUIERO.

A ti Lizbeth gracias por darme la fuerza que necesitaba en los momentos críticos de mi 5 año y bueno esto también fue parte tuya gracias mi vida.

Es también para (los míos) ese grupo gigante que nunca me voy a olvidar de ellos por hacer de mi lo que soy en este momento y estar conmigo en todos los momentos tanto malos como buenos, ustedes saben con quienes estoy hablando ok: Piry, Delio, Robert, Noel, Lenon, Anier, Pedro, Joel, Disnier. No tengo más nada que decirle en esta hoja ustedes saben que no hablo mucho así que ya saben no duden nada. LOS QUIERO.

Te lo dedico a ti Javier Ernesto por ser la primera persona que sale de mi sangre y quiero que sepas que eres lo máximo para mí. Te amo mi chama.

Gracias mis hermanos de la vida por permitir que esto sucediera: Raúl, Alejandro, Miguelito, Williams, Yoe. LOS QUIERO MIS HERMANOS.

Este último párrafo es para ti por ser la mujer que me ha dado fuerza en la vida y demostrarme que el amor si existe y que si no fuera por ti créeme que esto no hubiera podido suceder ok, a pesar de las indiferencias que hemos tenido quiero que este acto que hoy te estoy brindando quede como un pacto de amor entre nosotros. Mi Yane.

RESUMEN

En el mundo existen muchos sistemas informáticos que se dedican a la evaluación del desempeño de sus empleados, muchos de estos sistemas son propietarios lo cual le impide cumplir con las expectativas de la propuesta ya que esta va tener su código abierto y libre acceso debido a la política de la universidad con respecto a la migración de software libre.

En la propuesta se presenta el análisis y diseño de un sistema que ayude a la toma de decisiones para la evaluación del desempeño en la facultad #3, para el desarrollo del diseño de la herramienta se utilizó la ayuda de los patrones de diseño dándole claridad, flexibilidad, rapidez a dicho proceso.

Para el desarrollo de la investigación se empleó como metodología el Proceso Unificado de Desarrollo (RUP), los artefactos que se generaron fueron usando como lenguaje de modelado el Lenguaje Unificado de Modelado (UML) y auxiliados por el Visual Paradigm como herramienta de Ingeniería de Software Asistida por Computadora (CASE). Para garantizar la calidad de los artefactos generados se aplicaron métricas, que arrojaron resultados positivos.

PALABRAS CLAVE

Reportes, Requisitos, Casos de uso, Clases, Análisis, Diseño.

TABLA DE CONTENIDOS

INTRODUCCIÓN	10
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	13
1.1 Introducción	13
1.2 Conceptos relacionados al problema	13
1.2.1 Sistema.....	13
1.2.2 Analista del Sistema.....	13
1.2.3 Diseño.....	13
1.2.4 Análisis y Diseño del Sistema.....	14
1.3 Sistema de gestión de evaluación de desempeño	14
1.3.1 Sistemas de gestión de evaluación de desempeño en el mundo.....	14
1.3.2 Sistemas de gestión de evaluación de desempeño en cuba.	16
1.4 Sistemas Gestores de Base de Datos	17
1.4.1 MySQL.....	17
1.4.2 SGBD: PostgreSQL.....	17
1.5 Metodologías existentes	18
1.6 Lenguajes	24
1.6.1 Hypertext Markup Language (HTML).	24
1.6.2 Cascading Style Sheets (CSS).....	24
1.6.3 PHP (Hypertext Preprocessor).	25
1.6.4 JavaScript	26
1.6.5 Lenguaje Unificado de Modelado (UML).	26
1.7 Herramientas	27
1.7.1 Macromedia Dreamweaver, herramienta de diseño web.	27
1.7.2 Eclipse	28
1.7.3 Visual Paradigm	28
1.8 Framework	29
1.8.1 Symfony.....	29
1.8.2 Zend Studio.....	30

1.9 Patrones	30
1.9.1 <i>Patrones de diseño</i>	30
1.9.2 <i>Patrones de arquitectura</i>	33
 CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA	35
 2.1 Introducción	35
2.2 Modelo de dominio	35
2.2.1 <i>¿Por qué modelo de dominio?</i>	35
2.2.2 <i>Representación del modelo del dominio</i>	37
2.3 Especificación de los requisitos del Software	37
2.3.1 <i>Requerimientos Funcionales</i>	37
2.4 Descripción del Sistema propuesto	42
2.4.1 <i>Descripción de los actores del sistema</i>	43
2.4.2 <i>Casos de Uso del Sistema</i>	44
2.4.3 <i>Modelado del diagrama de CU del sistema</i>	44
2.4.4 <i>Descripción expandida de los Casos de Uso del sistema</i>	46
2.4.5 <i>Conclusiones Parciales</i>	72
 CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA	73
 3.1 Introducción	73
3.2 Modelo de análisis	73
3.3 Clases de análisis	73
3.3.1 <i>Diagrama de clases de análisis</i>	74
3.4 Diseño	80
3.4.1 <i>Modelo de diseño</i>	81
3.4.2 <i>Diagrama de clases de diseño</i>	81
3.5 Diagramas de Secuencia	90
3.6 Diagrama de Clases Persistentes	91
3.8 Principios de diseño	92
3.8.1 <i>Estándares de la interfaz de aplicación</i>	92

3.9 Concepción general de la ayuda.....	92
3.10 Tratamiento de Excepciones.....	92
3.11 Conclusiones Parciales.....	93
CAPÍTULO 4: VALIDACIÓN DE LA PROPUESTA.....	94
4.1-Introducción.....	94
4.2 -Métricas.....	94
4.2.1. Métricas de la calidad de la especificación.....	95
4.2.2. Métricas de Casos de uso.....	97
4.2.3. Métricas para el Modelo de Diseño.....	98
4.5 Conclusiones Parciales.....	101
CONCLUSIONES GENERALES.....	102
RECOMENDACIONES.....	103
REFERENCIAS BIBLIOGRAFICAS.....	104
ANEXOS.....	106
GLOSARIO.....	134

INTRODUCCIÓN

Las tecnologías en los últimos 50 años se han caracterizados por un avance vertiginoso y para nadie es ajeno que esto ha cambiado la forma de vida en la sociedad, donde son necesarias desde un centro de trabajo, escuelas, hasta en el propio hogar, llegando al punto que somos dependientes de estas.

La Universidad de las Ciencias Informáticas (UCI) desde su surgimiento en el año 2002, tiene como principal objetivo el desarrollo de la industria cubana del software y lograr que esta sea una ciudad totalmente automatizada. Para su desarrollo y evolución fomentó la creación de diferentes estrategias y servicios adicionales, necesarios e indispensables para la satisfacción y buen funcionamiento de la misma. En particular nuestra Universidad sin tecnología es como una sociedad prehistórica, en todas partes son necesarias, en el docente, en los comedores, en los aptos, entre otros. Por lo que cada día se hace más necesario que el trabajo de todas las personas sea de forma automatizada.

Unas de las actividades que no están automatizadas es la toma de decisiones para evaluar a un profesor, y el trabajo es un poco engorroso el tener que estar esperando por las actividades del que sería evaluado y lo mismo, el tener que llevar las actividades a cada persona que se evalúa. Es por eso que surge la necesidad de desarrollar una aplicación para la toma de decisiones de la evaluación del desempeño que se le da a cada profesor, y así hacer más fácil el trabajo, estableciendo como **Problema Científico** lo siguiente :

¿Cómo gestionar la evaluación de desempeño en la facultad 3?

El Objeto de Estudio:

Estará enmarcado en los procesos de desarrollo del software.

Enfocando el Campo de Acción:

En el desarrollo del análisis y diseño del proceso de gestión de evaluación de desempeño en la facultad 3.

La **idea a defender** es el análisis y diseño para un nuevo Sistema de Gestión en software libre para la evaluación de desempeño en la facultad 3, que permita darle flexibilidad, rapidez y eficiencia a dicho proceso.

Teniendo como **Objetivo General**:

Desarrollar el análisis y diseño para una herramienta que automatice el proceso de gestión de evaluación de desempeño en la facultad 3 del cual se derivan los siguientes **Objetivos Específicos**:

- Investigar sobre los sistemas implementados de Evaluación de desempeño
- Estudiar la bibliografía contemporánea asociada a los conceptos y lenguajes a utilizar.
- Identificar y describir los requerimientos del sistema a proponer.
- Realizar el análisis y diseño de la aplicación informática a desarrollar.
- Diseñar todas las clases que se implementarán.

Para que este trabajo cumpla con su propósito se plantearon un grupo de **Tareas de Investigación**:

- Realizar un análisis de los sistemas similares existentes en el mundo y en Cuba.
- Fundamentar las metodologías y herramientas a utilizar para el desarrollo del trabajo.
- Realizar del levantamiento de requisitos para la aplicación a desarrollar.
- Realizar el Análisis y Diseño de la aplicación Web a desarrollar.
- Realizar el estudio de la factibilidad para el sistema final.

Se emplearán métodos empíricos y teóricos de la investigación científica con el fin de dar cumplimiento a las tareas expuestas anteriormente. Dentro de los métodos empíricos se utilizará la entrevista, la cual posibilitará obtener información referente a cómo se espera que funcione la aplicación web, esta será realizada a profesores involucrados con la evaluación de desempeño.

En los métodos teóricos se utilizará el análisis y la síntesis porque permiten llegar a ideas generales a partir de condiciones específicas, y la inducción y deducción porque permiten concluir con ideas particulares partiendo de las generales, además de que se pueden utilizar en las diferentes etapas de la investigación del trabajo.

La propuesta de este trabajo contribuye a mejorar el proceso de gestión de evaluación del desempeño en la Facultad 3 para el logro de un eficiente servicio, garantizando con el empleo de esta nueva estrategia a

desarrollar, una mejor alternativa para el gestión de información entre un sistema altamente escalable y los usuarios del mismo.

El contenido del presente trabajo se estructura en 4 capítulos:

Capítulo 1. Fundamentación Teórica: Se exponen los fundamentos generales que sirven de soporte teórico en la solución del problema. Se analizan las herramientas y lenguajes de programación idóneas para el desarrollo de Gestión de evaluación de desempeño. Se plantea la metodología a emplear en el desarrollo del mismo. Se realiza un estudio de la arquitectura a emplear en la propuesta de sistema y un análisis de algunos sistemas similares a este a nivel internacional como nacional.

Capítulo 2. Características del sistema: Se abordará sobre todo lo referente al sistema realizando una propuesta que permitirá explicar y justificar la definición de los actores y casos de uso del mismo, así como una exposición de los requerimientos funcionales y no funcionales respectivamente.

Capítulo 3. Análisis y diseño del sistema: En este capítulo se modelan los principales artefactos que se obtienen como resultado del flujo de trabajo de análisis y diseño. En la fase de análisis se muestran los diagramas de clases de análisis, y en la fase de diseño se obtienen los diagramas de secuencia y de clases de diseño.

Capítulo 4. Validación de la propuesta: En este capítulo se realiza un estudio de la factibilidad del proceso de evaluación del desempeño en la facultad #3, utilizando como principal método las métricas para determinar la calidad de la especificación de los requisitos, del diseño de casos de uso y del diseño.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

En este capítulo se realiza una profunda investigación de diferentes sistemas similares tanto en el ámbito internacional como nacional. Se describen además las tecnologías actuales, herramientas y lenguajes de desarrollo utilizados para el análisis y diseño de un sistema sobre las cuales se apoya la propuesta.

1.2 Conceptos relacionados al problema

1.2.1 Sistema

Sistema es un conjunto de elementos que forman una actividad, un procedimiento o un plan de procedimientos que buscan una meta o metas comunes mediante la manipulación de datos para obtener resultados modulares o complementarios.

1.2.2 Analista del Sistema

El analista del sistema surge por la necesidad de recopilar, desglosar, catalogar y analizar información necesaria de una empresa para poder proponer nuevos y mejores métodos o modificar los actuales para que así aumente el desempeño de los departamentos dentro de la organización.

Las funciones del analista del sistemas para la década de los ochenta se definieron como aquel que debería conocer los procedimientos para indagar y ser capaz de proponer un verdadero sistema racionalizado, además de esto debería tener conocimientos sobre los modernos sistemas de información, bases del diseño, en fin, sobre todo en computación ya que estos eran los factores considerados como que justificaban esta especialidad.

1.2.3 Diseño

El diseño se basa en la estructura estática y dinámica, donde su propósito principal es crear el esqueleto concreto en el cual la implementación completa se basa para la ejecución del sistema. Este al igual que el análisis, no termina realmente hasta que el sistema final es entregado, durante esta fase se alcanza un

grado de terminación al poner en su lugar la mayoría de las decisiones estratégicas de diseño y al establecer políticas para diferentes problemas tácticos.

1.2.4 Análisis y Diseño del Sistema

El análisis y diseño de sistemas se refiere al proceso de examinar la situación de una empresa con el propósito de manejarla con métodos y procedimientos más adecuados.

Este se puede dividir en dos partes: el análisis de sistemas que comprende la planificación, el levantamiento inicial de información y el estudio en detalle del sistema actual para luego recomendar o estructurar las especificaciones necesarias para el nuevo sistema; y el diseño que consiste en llevar a cabo el sistema por medio de la clasificación y empleo de la información de manera que se pueda ofrecer una alternativa mucho más viable.

Se determina, a modo de resumen, que el análisis especifica qué es lo que el sistema debe hacer, mientras que el diseño establece cómo alcanzar el objetivo.

1.3 Sistema de gestión de evaluación de desempeño.

1.3.1 Sistemas de gestión de evaluación de desempeño en el mundo.

La necesidad de mejorar el proceso de evaluación de desempeño en el mundo actual hace imprescindible la implementación de distintas aplicaciones Web que le permitan a los profesionales estar alerta respecto a su rol, tareas y funciones, saber cómo ejecutarlas y mejorarlas, su atención se contara con más precisión en las tareas.

A continuación se caracterizan algunos sistemas de gestión de evaluación de desempeño en diferentes lugares en el mundo:

Herramienta para la Planificación y Evaluación del Desempeño del Empleado.

Este sistema ha sido desarrollado en EEUU por Management Sciences for Health (MSH) en 1988. También ha sido adaptado y utilizado por los empleados de oficinas de MSH en el exterior y por afiliados

alrededor del mundo, incluyendo Filipinas, Madagascar, y Bangladesh. Provee a la organización un medio para manejar el rendimiento de los empleados a través de la identificación de objetivos críticos para el desempeño de todos los empleados; estos objetivos están ligados a las metas prioritarias de la organización y son revisados dentro de un tiempo específico.

Esta Herramienta contiene tres partes:

1. Planificación del Rendimiento y Sistema de Revisión.
2. Una guía para desarrollar los Objetivos del Desempeño.
3. Una guía para desarrollar las Descripciones de Trabajos.

Ventajas:

Un Sistema de Planificación del Rendimiento y Sistema de Revisión provee ventajas tanto al empleado como a la organización. Para el empleado, le da respuestas a tres preguntas críticas:

¿Qué tengo que hacer?

¿Qué tal lo hago?

¿Qué significado tiene mi trabajo para la organización?

Provee a la organización un medio de dirigirla planificación y revisión del trabajo en una manera sistemática. También provee información objetiva que puede guiar decisiones de la gerencia en promociones, transferencias, asignaciones de tareas y necesidades acerca del desarrollo de los empleados. A pesar de que este sistema no debe ser utilizado solamente como una guía para definir salarios, puede proveer tanto al empleado como a la organización una base objetiva para determinar un aumento de salario o premios meritorios.

Limitaciones:

La aplicación requiere un compromiso completo de la organización, apoyo de los administradores y una filosofía de participación gerencial. Una inversión para entrenar a todo el personal en el sistema del PP&R es requerida cuando se introduce, con capacitación posterior para los empleados nuevos en la organización. Se recomienda capacitación continua y desarrollo de habilidades para los supervisores.

Sistema de gestión del desempeño.

Sistema de gestión utilizado en el Banco de la República de Colombia creado en el 2003, un medio para afianzar el modelo de competencia. Con este nuevo sistema se ha logrado la implementación de un sistema más estructurado, que posibilita la divulgación y formación para el afianzamiento y una contribución al cambio cultural. Además es una aplicación que le permite al empleado obtener la información clara y disponible para su desarrollo y evaluación en la organización.

Sistema de Gestión del Desempeño en Liberty Seguros.

El Sistema de Gestión del Desempeño, desarrollado e implantado por **Entel** (empresa innovadora de Consultoría, Tecnología y Outsourcing) en el año 2005, se mantiene en la actualidad con un servicio de actualizaciones. Su principal objetivo es: aumentar el rendimiento de los empleados, estableciendo acciones de mejora, formación y desarrollo. Es una herramienta que permite la gestión y aplicación del programa de evaluación del desempeño por los managers y la posterior visualización por parte de los colaboradores. Se utiliza para fijar el rumbo a los profesionales, hacer un seguimiento y corrección a lo largo del año y, al final del periodo, evaluar los resultados obtenidos por los mismos.

1.3.2 Sistemas de gestión de evaluación de desempeño en cuba.

Procedimiento para la Evaluación del Desempeño en el Sector Hotelero Cubano.

Sistema desarrollado por Dr. Noel Varela Izquierdo, MSc. Guillermo A. López Calvajar y MSc. Eduardo Concepción Morales en la Universidad de Cienfuegos. Este trabajo consiste en el diseño, elaboración y aplicación para la evaluación del desempeño en las instalaciones turísticas y entidades afines, conformado a partir de las concepciones actuales de la evaluación del desempeño en el mundo. Como resultado principal, el trabajo aporta un nuevo procedimiento para la evaluación del desempeño, con la aplicación de la teoría de los conjuntos difusos para eliminar la subjetividad y lograr la diferenciación cualitativa y cuantitativa en la evaluación del personal y automatizado para facilitar su aplicación (SISED).

1.4 Sistemas Gestores de Base de Datos

1.4.1 MySQL

Es uno de los gestores de bases de datos más populares de internet para aplicaciones pequeñas, que no realicen muchas transacciones al mismo tiempo, con PHP hace la combinación perfecta. Está desarrollado bajo la filosofía de código abierto y es multiplataforma.

Su velocidad, estabilidad y seguridad es alta. Presenta como gran desventaja que no implementa integridad de los datos, ni soporte de vistas, ni sub-consultas, esto lo hace un poco deficiente a la hora de desarrollar un sistema grande, con muchos clientes y muchas transacciones a la vez. (18)

1.4.2 SGBD: PostgreSQL.

PostgreSQL es un sistema gestor de base de datos libre. Tiene soporte total para transacciones, vistas, procedimientos almacenados, almacenamiento de objetos de gran tamaño. Se destaca en ejecutar consultas complejas, consultas sobre vistas, subconsultas y joins de gran tamaño. Permite la definición de tipos de datos personalizados e incluye un modelo de seguridad completo. Como toda herramienta de software libre PostgreSQL tiene entre otras ventajas, su código fuente está disponible sin costo alguno y es multiplataforma. Fue diseñado para ambientes de alto volumen. Escala muy bien al aumentar el número de CPUs y la cantidad de RAM. Tiene mejor soporte para vistas y procedimientos almacenados en el servidor, además tiene ciertas características orientadas a objetos. (1)

Características

Objeto-Relacional PostgreSQL aproxima los datos a un modelo objeto-relacional, y es capaz de manejar complejas rutinas y reglas. Ejemplos de su avanzada funcionalidad son consultas SQL declarativas, control de concurrencia multi-versión, soporte multi-usuario, transacciones, optimización de consultas, herencia, y arreglos.

Altamente Extensible Soporta operadores, funciones, métodos de acceso y tipos de datos definidos por el usuario.

Integridad Referencial PostgreSQL soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos.

MVCC, o Control de Concurrencia Multi-Versión (Multi-Version Concurrency Control), es la tecnología que PostgreSQL usa para evitar bloqueos innecesarios. MVCC está considerado mejor que el bloqueo a nivel

de fila porque un lector nunca es bloqueado por un escritor. En su lugar, PostgreSQL mantiene una ruta a todas las transacciones realizadas por los usuarios de la base de datos. PostgreSQL es capaz entonces de manejar los registros sin necesidad de que los usuarios tengan que esperar a que los registros estén disponibles. (1)

Cliente/Servidor PostgreSQL usa una arquitectura proceso-por-usuario cliente/servidor. Hay un proceso maestro que se ramifica para proporcionar conexiones adicionales para cada cliente que intente conectar a PostgreSQL.

Ventajas

1. Instalación ilimitada.
2. Mejor soporte que los proveedores comerciales.
3. Ahorros considerables en costos de operación.
4. Estabilidad y confiabilidad legendarias.
5. Extensible.
6. Multiplataforma.
7. Diseñado para ambientes de alto volumen.
8. Herramientas gráficas de diseño y administración de bases de datos.

1.5 Metodologías existentes

Una metodología es un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar nuevo software. La metodología indica cómo hay que obtener los distintos productos parciales y finales. (3)

Todo desarrollo de software es riesgoso y difícil de controlar, pero si no se lleva una metodología de por medio, lo que se obtiene son clientes insatisfechos con el resultado y desarrolladores aún más insatisfechos. Muchas veces no se toma en cuenta el utilizar una metodología adecuada lo que trae consigo que los proyectos, en problemas son los que salen del presupuesto, tienen importantes retrasos, o simplemente no cumplen con las expectativas del cliente.

Para dar una idea de que metodología se puede utilizar y cual se adapta a este medio se describen tres de ellas de las que se considera las más importantes tal como: XP, OMT y FDD y por último la metodología a seguir: RUP.

Extreme Programming (XP)

Es una de las metodologías de desarrollo de software más exitosas en la actualidad, utilizada para proyectos de corto plazo, pequeño equipo y cuyo plazo de entrega era ayer. 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. (3)

Características de XP, la metodología se basa en:

- . Pruebas Unitarias: se basa en las pruebas realizadas a los principales procesos, de tal manera que adelantándose en algo hacia el futuro, se pueden hacer pruebas de las fallas que pudieran ocurrir.

- . 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.

¿Qué es lo que propone XP?

- . El manejo del cambio se convierte en parte sustantiva del proceso

- . El cliente o el usuario se convierten en miembro del equipo

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.

Se recomienda para proyectos de corto plazo con equipos pequeños y rotables en cuanto a roles. Se Basa en los Use Stories (historias de uso), que al igual que el anterior definen los detalles técnicos sin meterse con los detalles de implementación. En cuanto a carga de trabajo, XP es proceso ligero pero no se les asignan roles organizativo al equipo, roles como el modelado o generación de la documentación, esto es reemplazado por la presencia de un representante especializado del cliente.

Este proceso está muy orientado a la implementación, esto lo hace bueno para el equipo de desarrollo ya que no debe preocuparse de la documentación y ya que los equipos rotan todos aprenden de todos, por otro lado el cliente también se siente satisfecho pues recibe un software que se adapta exactamente a sus

deseas, pero esto implica que para esto debió designar a una persona totalmente involucrada en el negocio, lo que podría implicar que esta persona deje de hacer sus funciones para estar totalmente disponible al equipo de desarrollo, razón por la cual se considera mejor la utilización de este proceso para desarrollos internos, pues debe haber una gran confianza entre el cliente y el equipo de desarrollo, como se hace mención anteriormente era poco probable que el cliente pueda prescindir de sus empleados esto incurriría en un coste adicional para el cliente. Por último como se podría representar todo lo que se debe sin documentación alguna (dependencia entre componentes por ejemplo) si no se anota ni se archiva nada y como alguien más puede tomar el lugar de uno de los miembros del equipo, o hacer mejoras en el sistema, esto crearía una dependencia con el equipo de desarrollo. (4)

OMT (Object Modeling Technique)

Es una de las metodologías de análisis y diseño orientados a objetos, más maduros y eficientes que existen en la actualidad. La gran virtud que aporta esta metodología es su carácter de abierta (no propietaria), que le permite ser de dominio público y, en consecuencia, sobrevivir con enorme vitalidad. Esto facilita su evolución para acoplarse a todas las necesidades actuales y futuras de la ingeniería de software. (5)

Ventajas

1. Proporciona una serie de pasos perfectamente definidos al desarrollador.
2. Tratamiento especial de la herencia.
3. Facilita el mantenimiento dada la gran cantidad de información que se genera en el análisis.
4. Es fuerte en el análisis.

Desventajas

1. Hay pocos métodos para encontrar inconsistencias en los modelos.
2. Interacción de objetos no soportada explícitamente en ninguna herramienta gráfica.
3. Al ser un análisis iterativo es difícil de saber cuándo comenzar con el diseño
4. Es débil en el diseño.

FDD (Feature Driven Development)

Este proceso se considera como punto medio entre los procesos pesados y ágiles, aunque en la práctica es más similar a XP. Pensado para proyectos relativamente cortos, también está basado en iteraciones que producen un software funcional que puede ser visto, probado y monitorizado por el cliente.

Estas iteraciones son decididas en base a las funcionalidades que el software debe tener, funcionalidades definidas por el cliente, este proceso está dividido en cinco fases:

1. Desarrollo de un Modelo General.
2. Construcción de la Lista de Funcionalidades.
3. Diseñar en base a las funcionalidades definidas.
4. Implementar en base a las mismas funcionalidades.

Aquí en el equipo de trabajo si existen jerarquías, siempre debe haber un jefe de proyecto, y aunque es un proceso considerado ligero también incluye documentación (la mínima necesaria para que algún nuevo integrante pueda entender el desarrollo de inmediato).

Desventaja

Aunque también genera documentación es la mínima necesaria para comprender el código, si bien es cierto el quipo cuenta con cierta libertad también es cierto que estos dependen directamente del jefe de proyecto quien es el responsable directo de proyecto.

El principal problema de este proceso está representado por la necesidad de contar con miembros experimentados en el equipo, miembros que puedan definir los roles y funciones de cada uno, que marquen el camino a seguir desde el principio con la elaboración del modelo global, el hecho de ser una combinación de XP y RUP es lo que lo hace más atractivo a la implementación, claro siempre y cuando la experiencia de esa gente de mayor jerarquía no cree dependencias.

RUP (Rational Unified Process)

RUP es un proceso de ingeniería de software que constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

Provee lineamientos, templates para herramientas que guían una implementación efectiva para las mejores prácticas de un software.

Es un proceso basado en componentes, que utiliza el Lenguaje Unificado de Modelado (UML) para preparar todos los esquemas de un sistema software.

Entre sus principales características se encuentran :(6)

1. Forma disciplinada de asignar tareas y responsabilidades (quién hace qué, cuándo y cómo)
2. Pretende implementar las mejores prácticas en Ingeniería de Software
3. Desarrollo interactivo
4. Administración de requisitos
5. Uso de arquitectura basada en componentes
6. Control de cambios
7. Modelado visual del software
8. Verificación de la calidad del software

Es uno de los procesos más generales que existe, se basa en la documentación generada en cada uno de sus cuatro fases:

1. Intercepción (puesta en marchar).
2. Elaboración (definición, análisis y diseño).
3. Construcción (implementación).
4. Transición (fin del proyecto y puesta en producción) en las cuales se ejecutarán varias iteraciones (según el tamaño del proyecto).

RUP se basa en Use Case (casos de uso) para describir lo que se tiene y lo que se espera del software, está muy orientado a la arquitectura del sistema a implementarse, documentándose de la mejor manera, gracias a su plan de desarrollo con el que se controla el desarrollo se pueden reconocer los problemas y fallos de forma temprana y corregirlos.

RUP define las actividades en grupos lógicos definiéndose 9 flujos de trabajo principales. Los 6 primeros son conocidos como flujos de ingeniería y los tres últimos como de apoyo. El proceso de desarrollo se divide en cuatro fases.

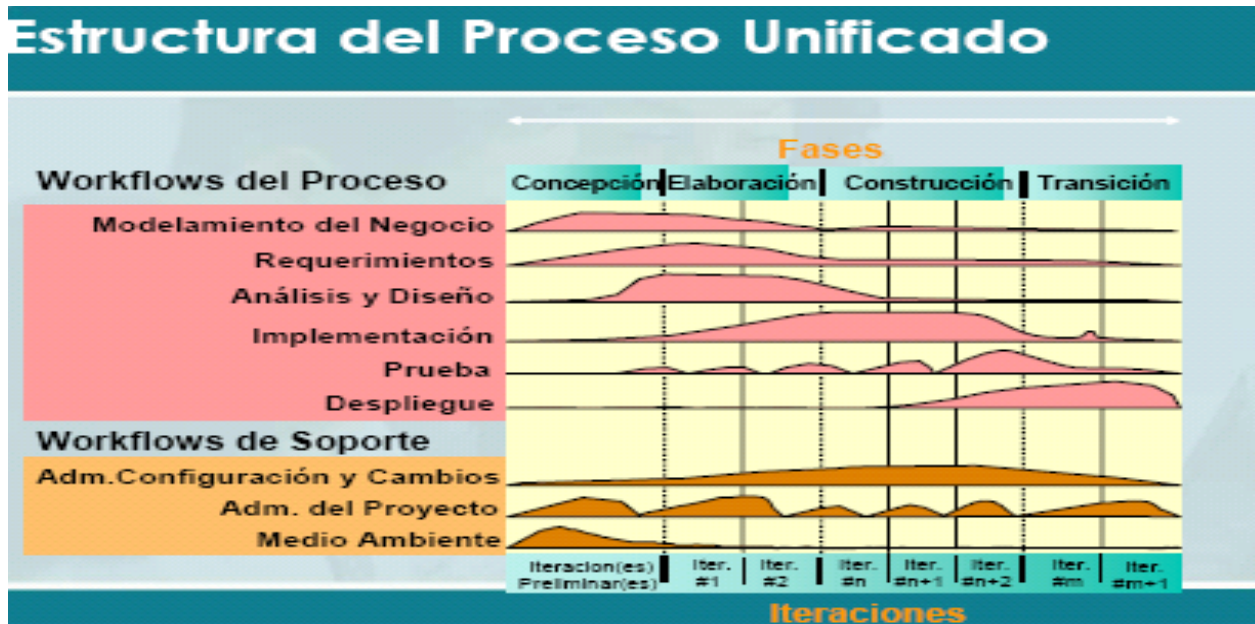


Figura # 1 RUP en dos dimensiones.

Flujos de trabajo:

Modelamiento del negocio: Describe los procesos de negocio, mediante la identificación de participantes y actividades que requieren automatización.

Requerimientos: Define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.

Análisis y diseño: Describe cómo el sistema será realizado a partir de la funcionalidad prevista y las restricciones impuestas (requerimientos).

Implementación: Define cómo se organizan las clases y objetos en componentes, cuáles nodos se utilizarán y la ubicación en ellos de los componentes y la estructura de capas de la aplicación.

Prueba (Testeo): Busca los defectos a lo largo del ciclo de vida.

Instalación: Realiza actividades (empaquete, instalación, asistencia a usuarios, etc.).

Administración del proyecto: Involucra actividades con las que se busca producir un producto que satisfaga las necesidades de los clientes.

Administración de configuración y cambios: Describe cómo controlar los elementos producidos por todos los integrantes del equipo de proyecto en cuanto a: utilización/actualización concurrente de elementos, control de versiones, etc.

Ambiente: Contiene actividades que describen los procesos y herramientas que soportarán el equipo de trabajo del proyecto; así como el procedimiento para implementar el proceso en una organización.

El ciclo de vida de RUP se caracteriza por:

Dirigido por casos de uso: Los casos de uso reflejan lo que los usuarios futuros necesitan, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. A partir de aquí los casos de uso guían el proceso de desarrollo.

Centrado en la arquitectura: La arquitectura muestra la visión común del sistema completo, por lo que describe los elementos del modelo más importantes para su construcción. El modelo de arquitectura se representa a través de vistas en las que se incluyen los diagramas de UML.

Iterativo e Incremental: RUP propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, unos más que otros. Las iteraciones hacen referencia a pasos en los flujos de trabajo, y los incrementos, al crecimiento del producto.

1.6 Lenguajes

1.6.1 *Hypertext Markup Language (HTML).*

HTML es un lenguaje de hipertexto cuyas siglas significan Lenguaje de Marcas Hipertexto. HTML consiste en códigos estándar o "marcas" que son usadas para definir la estructura de la información en una página Web. HTML es usado para crear documentos en la World Wide Web. Una página Web es la unidad básica de información accesible desde la World Wide Web⁷. HTML define varios aspectos de una página web, incluyendo títulos, negritas, itálicas, imágenes, párrafos y conexiones de hipertexto a otros recursos, es por ello que puede ser comparado con el procesamiento de palabras. (14)

1.6.2 *Cascading Style Sheets (CSS).*

CSS, Cascading Style Sheets, hojas de estilo en cascada, fue introducido en 1996 como el estándar para añadir información de estilo a los documentos HTML. El lenguaje y reglas de CSS hacen este trabajo de forma muy simple y clara, facilitando la tarea de diseñar páginas Web atractivas y vistosas. Hojas de Estilo

en Cascada es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir, o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura. Esta forma de descripción de estilos ofrece a los desarrolladores el control total sobre estilo y formato de sus documentos. (7)

1.6.3 PHP (Hypertext Preprocessor).

Es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor. PHP es un lenguaje de programación de estilo clásico, con variables, sentencias condicionales, bucles, funciones.

Diseñado para, entre otras cosas, aumentar, incrementar el dinamismo de las páginas Web. Originalmente se trataba de un conjunto de macros concebidas para ayudar en el mantenimiento de páginas Web. Desde entonces, sus características han ido creciendo hasta convertirse en un lenguaje de programación completo, capaz de manejar entornos que integran grandes bases de datos. Su popularidad se basa, en gran parte, a su sintaxis similar a la del lenguaje de programación C, su rapidez y simplicidad.

PHP también tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos tales como Linux y Windows, y puede interactuar con los servidores de Web más populares ya que existe en versión CGI, módulo para Apache, e ISAPI8.

Es multiplataforma, funciona tanto para Unix (con Apache) como para Windows (con Microsoft Internet Information Server) de forma que el código que se haya creado para una de ellas no tiene por qué modificarse al pasar a la otra. (8)

Porqué utilizar PHP y no otro lenguaje:

1. PHP no soporta directamente punteros, como el C, de forma que no existen los problemas de depuración provocados por estos.
2. Se pueden hacer grandes cosas con pocas líneas de código. Lo que hace que merezca la pena aprenderlo.
3. El código PHP es mucho más legible que el de PERL, todo el que haya programado PERL podrá corroborar esta afirmación.

4. Viene acompañado por una excelente biblioteca de funciones que permite realizar cualquier labor (acceso a base de datos, encriptación, envío de correo, XML, creación de PDF ...)
5. Al poderse encapsular dentro de código HTML se puede recoger el trabajo del diseñador gráfico e incrustar el código PHP posteriormente.
6. Esta siendo utilizado con éxito en varios millones de sitios web.
7. Hay multitud de aplicaciones PHP para resolver problemas concretos (tiendas virtuales, periódico) listas para usar.
8. Es multiplataforma, funciona en todas las plataformas que soporten apache.
9. Es software libre. Se puede obtener en la web y su código está disponible bajo la licencia GPL.

1.6.4 JavaScript

Se trata de un lenguaje de tipo script compacto y guiado por eventos, diseñado específicamente para el desarrollo de aplicaciones cliente-servidor dentro del ámbito de Internet. Los programas Java Script van incrustados en los documentos HTML, y se encargan de realizar acciones en el cliente, como pueden ser pedir datos, confirmaciones, mostrar mensajes, crear animaciones, comprobar campos. (13)

Es el lenguaje que nos permite interactuar con el navegador de manera dinámica y eficaz, proporcionando a las páginas web dinamismo y vida. (10)

1.6.5 Lenguaje Unificado de Modelado (UML).

El Lenguaje Unificado de Modelado (UML) es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software. Está compuesto por diversos elementos gráficos que se combinan para conformar diagramas. Debido a que el UML es un lenguaje, cuenta con reglas para combinar tales elementos. UML es un lenguaje que permite la modelación de sistemas con tecnología orientada a objetos.

Los diagramas son entes importantes de UML, cuya finalidad es presentar diversas perspectivas de un sistema, a las cuales se les conoce como modelo. Un modelo UML describe lo que supuestamente hará un sistema, pero no dice cómo implementarlo. El modelo gráfico de UML tiene un vocabulario en el que se identifican: elementos, relaciones y diagramas. (6)

Características:

1. Permite modelar sistemas utilizando técnicas orientadas a objetos (OO).
2. Permite especificar todas las decisiones de análisis, diseño e implementación, construyéndose así modelos precisos, no ambiguos y completos.
3. Puede conectarse con los lenguajes de programación (Ingeniería directa e inversa).
4. Permite documentar todos los artefactos de un proceso de desarrollo (requisitos, arquitectura, pruebas, versiones, etc.).
5. Cubre las cuestiones relacionadas con el tamaño propio de los sistemas complejos y críticos.
6. Es un lenguaje muy expresivo que cubre todas las vistas necesarias para desarrollar y luego desplegar los sistemas.
7. Existe un equilibrio entre expresividad y simplicidad, pues no es difícil de aprender ni de utilizar.
8. UML es independiente del proceso, aunque para utilizarlo óptimamente se debería usar en un proceso que fuese dirigido por los casos de uso, centrado en la arquitectura, iterativo e incremental.

¿Por qué utilizar PHP y PostgreSQL?

El departamento de Informatización sigue una política de utilización de software libre, es por esto que se plantea como propuesta de solución en este sentido cumpliendo con estos principios. Además como se expuso anteriormente PHP y PostgreSQL dadas sus características constituyen una buena alternativa en este sentido.

1.7 Herramientas

1.7.1 Macromedia Dreamweaver, herramienta de diseño web.

Macromedia Dreamweaver es uno de de los programas más utilizados en todo el mundo para la creación de páginas web. Es empleado tanto por profesionales como por personas que se inician en la creación de su primera página web.

Es una de las herramientas más completas y estandarizadas a nivel mundial, con varios años de experiencia. Incluye una gran cantidad de funcionalidades que le permiten al diseñador explotar al máximo sus potencialidades, se integra con múltiples lenguajes y plataformas incluyendo PHP, incluye herramientas para trabajar aplicaciones que manejan XML, así como mejoras a su manejo de hojas de estilo (CSS) y guía para revisar los diseños y una barra de código para tener acceso a las funciones

frecuentes. Además de sus ventajas existía dominio de la herramienta por parte del equipo de trabajo por lo que se decide su utilización. (11)

1.7.2 Eclipse

Eclipse es un IDE (Entorno de Desarrollo Integrado) tan potente como popular que incorpora un sin fin de utilidades para simplificar la labor de los programadores. Aparte de ser un entorno de desarrollo completo, una de las particularidades más interesantes para la comunidad es que es de código libre y gratuito.

Para Eclipse existen diversos plug-ins o añadidos para proveer de nuevas utilidades al programa, enfocadas a diversos usos que los distintos tipos de programadores pueden necesitar. Unos de los añadidos de Eclipse que más pueden interesar a los desarrolladores de páginas web sería el módulo para programación en PHP. (15)

Entre estos esta PDT (PHP Development Tools, Eclipse) trabaja para proveer un IDE completamente funcional para PHP para la plataforma Eclipse. PDT es independiente de plataforma, estando disponible en Windows o Linux. Entre las características en la versión actual (1.0) se encuentran:

1. Editor sensible al contexto, el cual provee de resaltamiento de código, asistente de código y autocompletado de código.
2. Ayuda para eliminar errores de código de PHP adicionales.
3. Integración con el modelo del proyecto Eclipse, que permite para inspeccionar el uso de las vistas del contorno del fichero y del proyecto.
4. Soporte para el debug incremental del código de PHP
5. Extensos frameworks que permiten a los desarrolladores fácilmente extender PDT para crear nuevas e interesantes herramientas orientadas al desarrollo de PHP.

1.7.3 Visual Paradigm

Visual Paradigm es una herramienta CASE (Computer-Aided Software Engineering) que permite realizar ingeniería tanto directa como inversa. Es una herramienta 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 el control de versiones. Cabe destacar igualmente su robustez, usabilidad y portabilidad. Visual Paradigm para UML es una herramienta profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas

y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. (16)

1.8 Framework

1.8.1 *Symfony*

Symfony es un completo framework diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. El resultado de todas estas ventajas es que no se debe reinventar la rueda cada vez que se crea una nueva aplicación web. Symfony está desarrollado completamente con PHP 5. Ha sido probado en numerosos proyectos reales y se utiliza en sitios web de comercio electrónico de primer nivel. Symfony es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. Se puede ejecutar tanto en plataformas Linux como en plataformas Windows.

1. Fácil de instalar y configurar en la mayoría de plataformas
2. Independiente del sistema gestor de bases de datos
3. Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.

Symfony automatiza la mayoría de elementos comunes de los proyectos web, como por ejemplo [16]:

1. La capa de internacionalización que incluye Symfony permite la traducción de los datos y de la interfaz, así como la adaptación local de los contenidos.
2. La capa de presentación utiliza plantillas y layout que pueden ser creados por diseñadores HTML sin ningún tipo de conocimiento del framework. Los helpers incluidos permiten minimizar el código utilizado en la presentación, ya que encapsulan grandes bloques de código en llamadas simples a funciones.
3. Los formularios incluyen validación automatizada y relleno automático de datos, lo que asegura la obtención de datos correctos y mejora la experiencia de usuario.

Los datos incluyen mecanismos de escape que permiten una mejor protección contra los ataques producidos por datos corruptos. (13)

5. Las interacciones con Ajax son muy fáciles de implementar mediante los helpers que permiten encapsular los efectos Java Script compatibles con todos los navegadores en una única línea de código. Es un framework maduro, bien documentado, recomendable para aplicaciones web complejas con mucha lógica de negocio pues permite el mantenimiento y las ampliaciones futuras de la aplicación, con un código ligero, legible y efectivo.

1.8.2 Zend Studio.

Se trata de un programa de la casa Zend, impulsores de la tecnología de servidor PHP, orientada a desarrollar aplicaciones web, como no, en lenguaje PHP. El programa, además de servir de editor de texto para páginas PHP, proporciona una serie de ayudas que pasan desde la creación y gestión de proyectos hasta la depuración de código. El programa entero está escrito en Java, lo que a veces supone que no funcione tan rápido como otras aplicaciones de uso diario. Sin embargo, esto ha permitido a Zend lanzar con relativa facilidad y rapidez versiones del producto para Windows, Linux y MacOS, aunque el desarrollo de las versiones de este último sistema se retrase un poco más. Zend Studio consta de dos partes en las que se dividen las funcionalidades de parte del cliente y las del servidor.

Las dos partes se instalan por separado, la del cliente contiene el interfaz de edición y la ayuda. Permite además hacer depuraciones simples de scripts, aunque para disfrutar de toda la potencia de la herramienta de depuración habrá que disponer de la parte del servidor, que instala Apache y el módulo PHP o, en caso de que estén instalados, los configura para trabajar juntos en depuración. (17)

1.9 Patrones

1.9.1 Patrones de diseño.

Un patrón es una solución recurrente a un problema dentro de un contexto dado. Los patrones surgen de la experiencia de seres humanos al tratar de lograr ciertos objetivos, además capturan la experiencia existente y probada para promover buenas prácticas.

Según Larman, un patrón es una descripción de un problema y su solución, que recibe un nombre y que puede emplearse en otros contextos; además indica la manera de utilizarlo en circunstancias diversas (Larman, 1999).

Los Patrones de Diseño (Design Patterns) son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces.

Estos se dividen en tres grandes categorías:

Patrones Creacionales: solucionan problemas de creación de instancias. Nos ayudan a encapsular y abstraer dicha creación.

Patrones Estructurales: solucionan problemas de composición (agregación) de clases y objetos.

Patrones de Comportamiento: soluciones respecto a la interacción y responsabilidades entre clases y objetos, así como los algoritmos que encapsulan.

Ventajas de los patrones de diseño:

- Los patrones de diseño proponen una forma de reutilizar la experiencia de los desarrolladores, para ello clasifica y describe formas de solucionar problemas que ocurren de forma frecuente en el desarrollo.
- Están basados en la recopilación del conocimiento de los expertos en desarrollo de software.
- Es una experiencia real, probada y que funciona. Es historia y nos ayuda a no cometer los mismos errores.

Patrones GRASP

Para el resultado de la investigación se emplea un grupo de patrones relacionados con el diseño de software, llamados patrones GRASP (General Responsibility Assignment Software Patterns) los que describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones. Este grupo de patrones está muy relacionado con los problemas básicos del diseño. La asignación correcta de las responsabilidades en el diseño orientado a objetos garantiza la alta

cohesión de las clases y el bajo acoplamiento de los mismos, lo que posibilita más extensibilidad, adaptabilidad y menos tiempo para el mantenimiento del diseño.

Experto

Asigna una responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplir la responsabilidad, es un principio básico que suele utilizarse en el diseño orientado a objetos, permitiendo que se conserve el encapsulamiento, soportando un bajo acoplamiento y una alta cohesión.

Creador

Guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos. El propósito fundamental de este patrón es encontrar un creador que se debe conectar con el objeto producido en cualquier evento. Se brinda soporte a un bajo acoplamiento, lo cual supone menos dependencias respecto al mantenimiento y mejores oportunidades de reutilización.

Bajo Acoplamiento

Este patrón se basa principalmente en la concepción de tener las clases lo menos ligadas entre sí que se pueda. De tal forma que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de las clases, potenciando la reutilización, disminuyendo la dependencia y además son fáciles de entender por separadas.

Alta Cohesión

Este patrón propone asignar la responsabilidad de manera que la complejidad se mantenga dentro de límites manejables asumiendo solamente las responsabilidades que deben manejar, evadiendo un trabajo excesivo. Su utilización mejora la claridad y facilidad con que se entiende el diseño, simplifica el mantenimiento y las mejoras de funcionalidad, generan un bajo acoplamiento, soporta mayor capacidad de reutilización.

Patrón Controlador

Es un patrón que sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa, de tal forma que es la que recibe los datos del usuario y la que los envía a las distintas clases según el método llamado. Asigna las responsabilidades de capturar los eventos del sistema a las clases.

Patrón Indirección:

Se asigna la responsabilidad a una clase intermedia para que medie entre otros componentes o servicios, y éstos no terminen directamente acoplados. El intermediario crea una indirección entre el resto de los componentes o servicios.

1.9.2 Patrones de arquitectura

Modelo Vista Controlador (MVC)

Modelo Vista Controlador (MVC) es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. Frecuentemente se ve en aplicaciones web, donde la vista es la página HTML y el código es el que provee de datos dinámicos a la página, el modelo es el Sistema de Gestión de Base de Datos y el controlador representa la lógica de negocio.

Modelo: Esta es la representación específica de la información con la cual el sistema opera. La lógica de datos asegura la integridad de estos y permite derivar nuevos datos.

Vista: Esta presenta el modelo en un formato adecuado para interactuar, usualmente es la interfaz de usuario.

Controlador: Responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista.

Tanto la vista como el controlador dependen del modelo, el cual no depende de las otras clases. Esta separación permite construir y probar el modelo, independientemente de la representación visual. El patrón de arquitectura MVC tiene flexibilidad para cambiar las vistas y los controladores. Además tiene vista sincronizada y se adapta al cambio.

Conclusiones Parciales.

En este capítulo se exponen las condiciones y problemas que rodean el objeto de estudio a través de los conceptos y definiciones planteadas. Se evidencia la necesidad de implementar un software que permita la interoperabilidad, organización y consumo de los servicios Web. Se realizó la justificación de cada herramienta, metodología a utilizar y los lenguajes que se emplearon para el desarrollo del sistema.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

2.1 Introducción

Este capítulo constituye la propuesta de solución para automatizar el proceso de Evaluación del Desempeño en la facultad #3. Para describir la solución se realiza el modelo de dominio, así como un glosario de términos para definir los diversos conceptos que serán utilizados, así como las relaciones que entre estos conceptos se establecen, en aras de lograr la utilización de un vocabulario común.

2.2 Modelo de dominio

2.2.1 ¿Por qué modelo de dominio?

Teniendo en cuenta que la definición del proceso del negocio que tienen que ver con el objeto de estudio se hace difícil encontrarlo, por lo que se ve a simple vista la necesidad de describir el funcionamiento de la herramienta mediante una serie de conceptos, entidades y sus relaciones, agrupándolos en un modelo del dominio con el fin de tener un fácil entendimiento del sistema.

Se ha decidido realizar un modelo de dominio ya que es una alternativa apropiada dado el escenario del problema, además de que no se considera necesario un modelamiento completo del negocio, las fronteras del negocio no se pueden determinar con claridad y los procesos del negocio no están claramente definidos.

Un Modelo del dominio captura los tipos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan los principales conceptos. Muchos de los objetos o clases pueden obtenerse de una especificación de requisitos. La modelación del dominio tiene como objetivo fundamental la comprensión y descripción de las clases más importantes en el sistema.

Con la aplicación del Modelo del dominio se detectaron las siguientes entidades y conceptos (objetos):

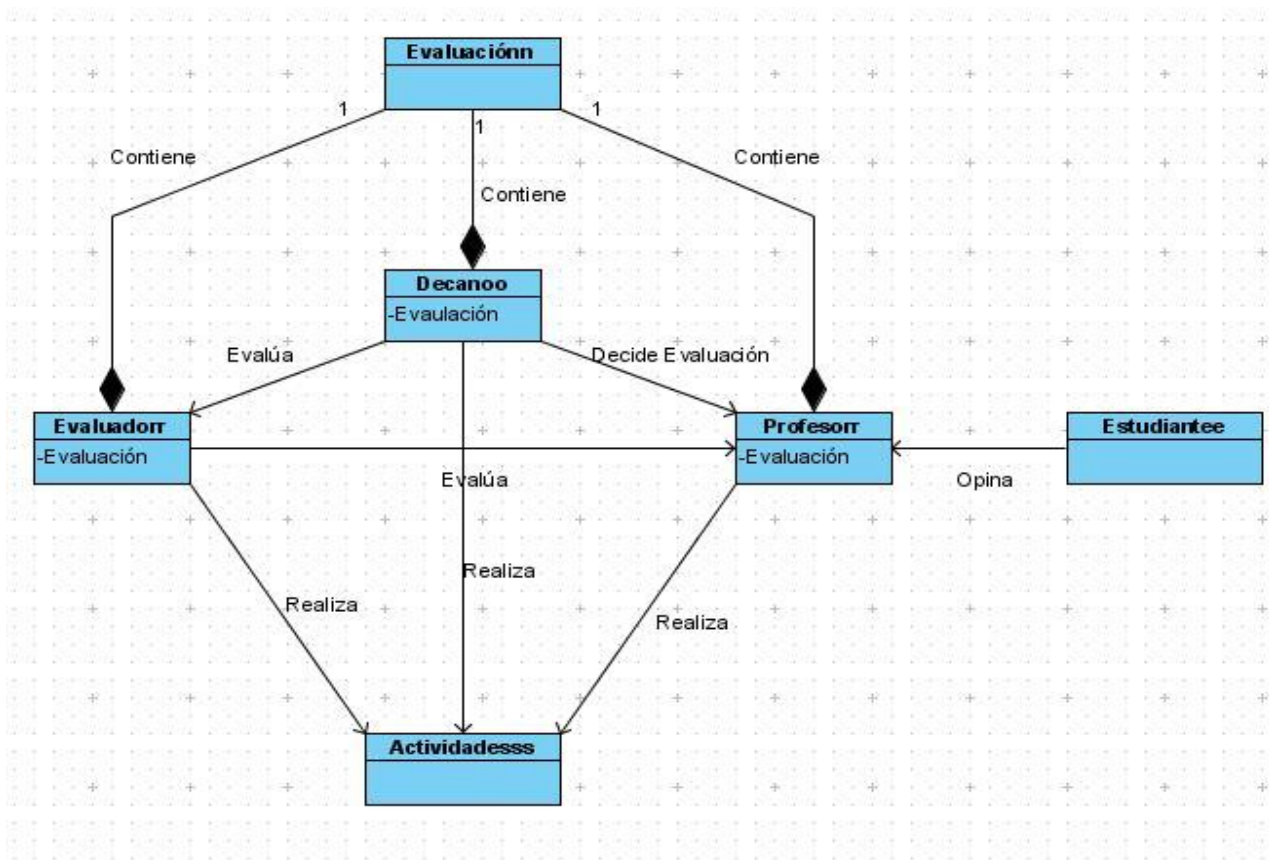
- **Evaluación:** Es el resultado de todo el proceso, donde cada profesor va a contener una evaluación al final del trimestre.
- **Decano:** Es el responsable de darle la evaluación a todos los evaluadores así como también la decisión final sobre la evaluación de los profesores de su facultad y de cumplir con las actividades que se le asignen.

- **Evaluador:** Es el responsable de darle la evaluación a los profesores que le pertenecen y de cumplir con las actividades que se le asignen.
- **Profesor:** Es el responsable de cumplir con las actividades que se le asignen.
- **Estudiantes:** Va a tener la oportunidad de opinar sobre los profesores que le imparten clase.
- **Actividades:** Son todas aquellas tareas que se le asignan a los profesores y estos deben de darle cumplimiento.

Principales Eventos:

- **Contiene:** Expresa que A está contenido lógicamente en B, Evaluador, Profesor y Decano contienen una evaluación.
- **Evalúa:** Expresa que A evalúa a B, el Decano evalúa al Evaluador y el Evaluador al Profesor.
- **Decide Evaluación:** Expresa que A decide si es correcta o no la evaluación que tienen B, el Decano decide si la evaluación de los profesores es la correcta y es el que decide la evaluación final.
- **Realiza:** Expresa que A realiza lo que contiene B, el Decano, el Evaluador y el Profesor realizan Actividades.
- **Opina:** Expresa que A opina sobre B, los Estudiantes Opinan sobre sus Profesores.

2.2.2 Representación del modelo del dominio



2.3 Especificación de los requisitos del Software

Para modelar el sistema que se pretende construir se identifican sus requisitos, tanto funcionales como no funcionales, y se modelan los funcionales en términos de casos de uso del sistema.

2.3.1 Requerimientos Funcionales

Los Requerimientos Funcionales son capacidades o condiciones que el sistema debe cumplir, los mismos se mantienen invariables sin importar con que propiedades o cualidades se relacionen.

RF1) Autenticar usuario

- 1.1) Entrar usuario y contraseña del dominio UCI.
- 1.2) Autorizar acceso de acuerdo al tipo de usuario.
- 1.3) Validar usuario y contraseña (IP de la máquina que hace la entrada, Hora de la entrada, Nombre, Usuario, Grupo, Solapín, Foto).

RF2) Gestionar Profesor

- 2.1) Adicionar profesor
- 2.2) Eliminar profesor
- 2.3) Modificar profesor

RF3) Gestionar Actividad

- 3.1) Agregar Actividad.
- 3.2) Modificar Actividad (Cambiándola de Fecha, de lugar o de hora).
- 3.2) Eliminar Actividad.

RF4) Generar Reportes

- 4.1) Reportar los profesores con sus evaluaciones.
- 4.2) Reportar los profesores que cumplieron en tiempo y forma todas las actividades.
- 4.3) Reportar los profesores con sanción.
- 4.4) Reportar los profesores incumplidores del trimestre.
- 4.5) Reportar profesores con pérdida de requisitos.
- 4.6) Reportar los profesores que realizaron tareas no reflejadas por plan de trabajo.

RF5) Consultar Actividades cumplidas (Dado un profesor determinado)

- 5.1) Actividades en un día
- 5.2) Actividades en la semana
- 5.3) Actividades en el mes
- 5.4) Actividades en el trimestre

RF6) Registrar indisciplinas

6.1) Comisión disciplinaria.

6.2) Sanciones.

RF7) Gestionar Página Principal

7.1) Agregar Tareas más Importantes.

7.2) Quitar Tareas.

7.3) Agregar Profesores más destacados.

7.4) Quitar Profesor.

7.5) Modificar ED.

7.6) Actualizar Horario.

RF8) Justificar

8.1) Permitir al profesor señalar las actividades que no puede cumplir en el trimestre teniendo conocimiento previo de futuras afectaciones, ya sea pase de profesores u otras.

8.2) Justificar el fallo de una actividad inmediata.

RF9) Actividades Asignadas

9.1) Mostrar Listado de actividades asignadas al usuario.

9.3) Ver Tareas cumplidas.

9.4) Ver Tareas por cumplir.

RF10) Proponer Evaluación

10.1) Generar Evaluación.

10.2) Emitir un reporte de los Profesores con sus Evaluaciones.

10.3) Manda un correo al Evaluador superior con todas las evaluaciones.

RF11) Estado de la Actividad

11.1) Cumplida

11.2) Incumplida

RF12) Verificar Actividad

RF13) Imprimir Reportes

RF14) Autoevaluación

RF15) Recordar Actividad

RF16) Gestionar Foro

- 16.1) Crear Foro.
- 16.2) Eliminar Foro.
- 16.3) Editar Foro

RF17) Opinar Sobre Profesor

RF18) Temas en Foro.

- 18.2) Insertar Tema.
- 18.3) Opinar sobre tema.

2.3.2 Requerimientos No Funcionales

Usabilidad:

1. El sistema debe permitir el acceso concurrente de un gran número de usuarios.
2. El sistema podrá ser usado por cualquier persona que posea conocimientos básicos en el manejo de la computadora y de un ambiente Web en sentido general.

Rendimiento:

1. El sistema debe ser lo más eficiente posible para poder lograr un tiempo de respuesta adecuado.
2. La velocidad de procesamiento de la información debe ser rápida.

3. Aplicación de las diferentes técnicas de elaboración en la Web para facilitar el rápido acceso a sus páginas.

Soporte:

1. El sitio debe permitir posteriores modificaciones y actualizaciones a fin de alcanzar mayor funcionalidad o dado que cambien algunos elementos del negocio.

Portabilidad:

1. Necesidad de que el sistema sea multiplataforma.

Seguridad:

1. La información manejada por el sistema debe estar protegida de acceso no autorizado y divulgación.
2. La información manejada por el sistema será objeto de cuidadosa protección contra la corrupción y estados inconsistentes por las personas autorizadas.
3. Los usuarios autorizados se les garantizará el acceso al sitio y que los dispositivos o mecanismos utilizados para lograr la seguridad no ocultarán o retrasarán a los mismos para realizar las transferencias necesarias en un momento dado.
4. Se deben crear grupos de usuarios que tendrán asignados permisos de acción sobre cada información manejada por el sistema para lo cual se requiere la autenticación del usuario.

Funcionalidad:

1. El sistema debe estar accesible desde la intranet de la Universidad.
2. Capacidad de búsqueda rápida.
3. Navegación fácil.

Requerimientos de Hardware:

1. En el Cliente:
 - Se requiere que las computadoras de los usuarios estén en la red.
2. En el Servidor.
 - Se requiere que el servidor tenga, al menos 1Gb de RAM y al menos de CPU 2.40 GHz.

Requerimientos de Software:

1. Se utilizará como lenguaje de programación PHP5 y como gestor de Base de Datos PostgreSQL.
2. En el cliente debe existir un navegador Web (Internet Explorer o Mozilla).
3. Sistema Operativo Windows o Linux.

2.4 Descripción del Sistema propuesto

El sistema de evaluación de desempeño debe de cumplir con distintas funcionalidades para un mejor funcionamiento del mismo. Como Actores del sistema se tiene al **Evaluador** (Jefe de Año, Jefe de Polo, Jefe de Departamento y el Decano), el Administrador, Profesores y Estudiantes.

Primeramente el sistema comenzara dejando que el usuarios se autentique, autorizando el acceso según su rol, si es **Evaluador** lo enviara a su página, donde podrán realizar distintas operaciones como por ejemplo: Adicionar, eliminar, modificar o buscar actividades, cambiar una evaluación siempre que el profesor no tenga una indisciplina cometida en dicho trimestre, registrar indisciplinas, ver todos los profesores que le pertenece, si es el Administrador entrara a la página de administración donde podrá actualizar la página principal con las actividades más importantes de la facultad y la universidad, poner como se desempeña la evaluación de desempeño en la universidad, también un listado de los profesores más destacados del pasado trimestre, actualizar las tareas de la semanas como son: la gestión del

horario docente, matutino, reuniones. Si es un **Profesor** podrá si es tiempo de entregar el plan de trabajo registrarlo en el sistema, registrar al final del trimestre su Autoevaluación durante ese periodo, justificar actividades que no pueda cumplir o que incumplió y consultar actividades asignadas. Si es **Estudiante** se llevara directamente a una página, donde van a estar solamente los profesores que les imparten clase y podrán opinar sobre ellos. Si es el **Decano** es el que proporciona la decisión final sobre las evaluaciones de todos los profesores, la enviara a su página donde podrá realizar diversas funcionalidades que le permita un mejor funcionamiento de su trabajo, ejemplos de estas funciones son: tendrá la posibilidad de Adicionar, Modificar o Eliminar a un profesor, poder cambiar un profesor internamente en la facultad, ver todas las evaluaciones, el plan de trabajo de cada profesor y podrá ver una serie de reportes.

El Sistema brindara muchas funcionalidades como son: Proponer evaluación, según el cumplimiento de las actividades de dicho profesor, consultar actividades de cada profesor, toda una series de reportes ya definidos, tendrá un foro para que los estudiantes opinen de sus profesores y los profesores podrán poner temas de conversación con el fin de mejorar la calidad de la facultad por la parte de docencia, tendrá un chat por donde los profesores podrán conversar entre ellos, justificar algunas actividades y a si dicho proceso se hace más rápido y va a dar la posibilidad de mandar correos automáticamente a la persona que lo necesite recordándole de una actividad.

2.4.1 Descripción de los actores del sistema

Los actores del sistema representan cada trabajador del negocio (inclusive si fuera un sistema ya existente) que tiene actividades a automatizar, además de los actores del negocio que interactúan de alguna forma con el sistema. Los actores del sistema:

- No son parte de él.
- Pueden intercambiar información con él.
- Pueden ser un recipiente pasivo de información.
- Pueden representar el rol que juega una o varias personas, un equipo o un sistema automatizado.

Actor del Sistema	Justificación
Jefe de años	Encargado de darles la evaluación a los profesores Guías.
Jefe de departamentos	Encargado de darles la evaluación a los profesores

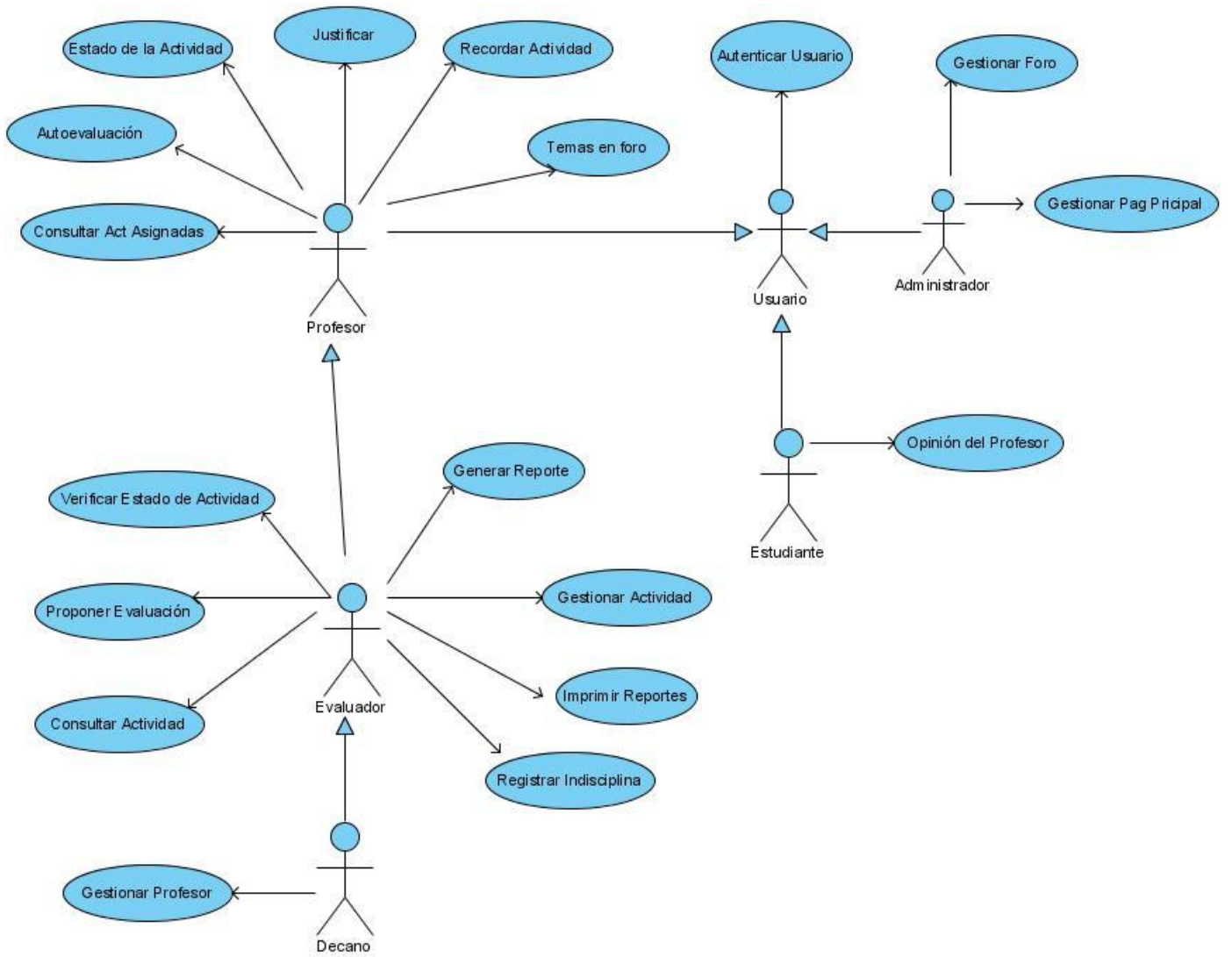
	que pertenecen a su departamento.
Jefe de polo productivo	Encargado de darles la evaluación a los profesores que pertenecen a su proyecto.
Decano	Encargado de darles la evaluación a los evaluadores y es la decisión final de la evaluación de todos los profesores.

2.4.2 Casos de Uso del Sistema

Los Casos de uso del sistema son procesos que responden a las funcionalidades definidas en los requerimientos funcionales. Representan las funcionalidades que debe hacer el sistema

2.4.3 Modelado del diagrama de CU del sistema

Un Diagrama de Casos de Uso muestra la relación entre los actores y los casos de uso del sistema. Representa la funcionalidad que ofrece el sistema en lo que se refiere a su interacción externa.



2.4.4 Descripción expandida de los Casos de Uso del sistema

Nombre del Caso de Uso	Autenticar Usuario
Actor	Usuario
Propósito	Permitir a los usuarios acceder al sistema
Precondiciones	El usuario debe pertenecer al dominio UCI
Poscondiciones	Se identificó y autenticó el usuario. Se mostró la sesión de trabajo del usuario autenticado. Se guardaron los datos y se cerró de manera segura la sesión de trabajo.
Resumen	El caso de uso se inicia cuando el actor interactúa con la interfaz de Bienvenida y accede a la opción de autenticarse en el sistema, introduciendo usuario y contraseña del dominio UCI. El sistema consulta internamente que haya coincidencias entre los datos introducidos y los datos almacenados, permitiendo o no el acceso del usuario al sistema.
Referencias	
Curso Normal de Eventos	
Acciones del actor	Respuesta del sistema
1. El caso de uso se inicia cuando el actor accede a la página de Bienvenida y a la opción de inicio de sesión.	

	<p>Brinda la posibilidad de introducir los datos de autenticación:</p> <p>Usuario.</p> <p>Contraseña</p> <p style="padding-left: 40px;">Ambos referentes al dominio UCI.</p> <p>Y permite:</p> <p>Iniciar sesión.</p>
2. Introduce usuario y la contraseña correspondiente selecciona la opción de iniciar sesión.	
	Valida los datos introducidos.
	Inicia la sesión correspondiente de acuerdo al usuario autenticado.
	El caso de uso termina.
Curso Alternativo de los eventos	
4. a El usuario introducido es incorrecto.	
	4. a.1 Muestra el siguiente mensaje de error: "El usuario introducido es incorrecto, por favor rectifique este dato".
	4. a.b Regresa al paso 2 del Curso Normal de Eventos.
4. b La contraseña introducida es incorrecta.	
	4. b.1 Muestra el siguiente mensaje de error: "La contraseña introducida es incorrecta, por favor rectifique este dato".
	4. b.2 Regresa al paso 2 del Curso Normal de Eventos.
4. c Se introducen datos erróneos por	

más de tres intentos consecutivos.	
	4. c.1 Muestra el siguiente mensaje de información: "Ha sobrepasado el número de intentos permitidos, su usuario será bloqueado por cuestiones de seguridad".
	4. c.2 Inhabilita el usuario.
	4. c.3 El caso de uso termina.

Nombre del Caso de Uso	Gestionar Profesor
Actor	Administrador
Propósito	Garantizar que se adicione, que se elimine o se modifique a un profesor.
Precondiciones	El usuario debe pertenecer al dominio UCI El usuario debe tener los permisos necesarios para poder ejecutar esta acción. Para eliminar a un profesor debe de estar registrado.
Poscondiciones	Se adiciona, se elimina o se modifica a un profesor.
Resumen	El caso de uso se inicia cuando el actor accede a la opción de Gestionar Profesor. El sistema muestra una interfaz con las opciones de Modificar, Eliminar y Adicionar. El usuario selecciona la opción que desea.
Referencias	RF2
Curso Normal de Eventos	
Acciones del actor	Respuesta del sistema
1. El caso de uso se inicia cuando el	

actor selecciona la opción de Gestionar Profesor.	
	2. Muestra las opciones de Modificar Profesor, Eliminar Profesor, Adicionar Profesor.
3. Selecciona la opción deseada.	
	4. Gestiona la selección <ul style="list-style-type: none"> ➤ Adicionar Profesor ➤ Eliminar Profesor ➤ Modificar Profesor
	5. Finalizar el caso de uso.
Curso Alternativo de los eventos	
4.a Accede a la opción que le permite Adicionar Profesor	
	4. a.1 Muestra los Campos a llenar. <ul style="list-style-type: none"> ➤ Nombre del Profesor.
4.a.2 Llena todos los Campos	
4. a.3 Adiciona al profesor.	
	4. a.4 Verifica que el profesor no existe, en caso contrario ir a CA1.
	4. a.5 El Sistema actualiza la Base de Dato.
	4.a.6 El Caso de Uso se Termina
CA1: Muestra un mensaje de error informando que el usuario ya existe.	
4. b Accede a la opción que le permite Eliminar Profesor.	
	4. b.1 Muestra un listado con todos los profesores que le pertenecen, con su foto.
4. b.2 Selecciona al profesor que desea	

eliminar.	
4. b.3 Le da al botón Eliminar.	
	4.b.4 Muestra el mensaje “Esta seguro de que desea continuar”
4.b.5 Le da al Botón Aceptar	
	4.b.6 Se elimina y se actualiza la Base de Dato
	4.b.7 El caso de Uso se termina
4.c Accede a la opción que le permite Modificar Profesor	
	4. c.2 Muestra todos los profesores con sus fotos que le pertenecen a ese evaluador.
4. c.3 Selecciona al profesor que desea modificar.	
	4. c.4 Muestra todos los datos del profesor.
4. c.5 Cambia lo que desea Modificar.	
	4.c.6 Se guardan los cambios en la Base de Dato
	4. c.7 El Caso de Uso se termina.

Nombre del Caso de Uso	Gestionar Actividad
Actor	Evaluador
Propósito	Garantizar que se adicione, que se elimine o se modifique alguna actividad.
Precondiciones	El usuario debe pertenecer al dominio UCI El usuario debe tener los permisos necesarios para poder ejecutar esta acción.

	Para eliminar a una actividad debe de estar registrada.
Poscondiciones	Se adiciona, se elimina o se modifica una actividad.
Resumen	El caso de uso se inicia cuando el actor accede a la opción de Gestionar Actividad. El sistema muestra una interfaz con las opciones de Modificar, Eliminar y Adicionar. El usuario selecciona la opción que desea.
Referencias	RF3
Curso Normal de Eventos	
Acciones del actor	Respuesta del sistema
El caso de uso se inicia cuando el actor selecciona la opción de Gestionar Actividad.	
	1. Muestra las opciones de Modificar Actividad, Eliminar Actividad, Adicionar Actividad.
2. Selecciona la opción deseada.	
	3. Gestiona la selección <ul style="list-style-type: none"> ➤ Adicionar Actividad ➤ Eliminar Actividad ➤ Modificar Actividad
	4. Finalizar el caso de uso.
Curso Alternativo de los eventos	
4.a Accede a la opción que le permite Adicionar Actividad	
	4. a.1 Se muestran todos los profesores que le pertenecen.

4. a.2 Selecciona el o los profesores que se le va asignar dicha actividad.	
4. a.3 Se marca el botón de adicionar.	
	4. a.4 Se muestra un texto en blanco para escribir la actividad.
4. a.5 Escribe la actividad y la adiciona.	
	4. a.6 El Sistema actualiza las actividades de los profesores.
	4.a.7 El Caso de Uso se Termina
4. b Accede a la opción que le permite Eliminar Actividad.	
	4. b.1 Muestra un listado con todas las actividades.
4. b.2 Selecciona la o las actividades que desea eliminar.	
4. b.3 La Elimina.	
	4.b.4 Muestra el mensaje “Esta seguro de que desea Eliminar esta actividad”
4.b.5 Le da al Botón Aceptar	
	4. b.6 Se elimina y se actualiza las actividades de los profesores.
	4.b.7 El caso de Uso se termina
4. c Accede a la opción que le permite Modificar Actividad.	
	4. c.1 Muestra todas las actividades.
4. c.2 Selecciona la actividad que desea modificar.	
	4. c.3 Muestra los datos de la actividad. <ul style="list-style-type: none"> ➤ Fecha ➤ Lugar

	➤ Hora
4. c.4 Selecciona lo que desea modificar.	
4. c.5 Lo modifica.	
	4. c.6 Automáticamente se le manda un mensaje al profesor con la actividad actualizada.
	4. c.7 El Caso de Uso se termina.

Nombre del Caso de Uso	Generar Reportes
Actor	Evaluador
Propósito	Permitir al usuario generar reportes de acuerdo a los permisos que este posea.
Precondiciones	El usuario debe pertenecer al dominio UCI El usuario debe tener permisos para generar reportes.
Poscondiciones	Se identificó y autenticó el usuario. Se generaron los reportes correspondientes.
Resumen	El caso de uso se inicia cuando el actor accede a la opción de Generar Reportes. El sistema muestra una interfaz con los tipos de reportes que se pueden generar. El usuario selecciona los reportes que desea y el sistema genera los datos solicitados.
Referencias	RF4
Curso Normal de Eventos	
Acciones del actor	Respuesta del sistema

<p>5. El caso de uso se inicia cuando el actor accede a la opción que permite generar reportes.</p>	
	<p>6. El sistema le da la posibilidad de escoger un trimestre y le muestra todos los reportes que se pueden generar:</p> <ul style="list-style-type: none"> ➤ Reportes de los profesores con sus evaluaciones. ➤ Reportes de las Actividades, ➤ Profesores con Sanción ➤ Reportar los profesores incumplidores. ➤ Reportar la perdida de requisitos. ➤ Reportar los profesores que realizaron tareas no reflejadas por plan de trabajo y dichas tareas. ➤ Reporte de los profesores que cumplieron en tiempo y forma todas las actividades.
<p>7. Escoge el reporte.</p>	
	<p>8. El sistema se lo muestra y le da la opción de imprimirlo.</p>
<p>9. El actor ve el reporte y si quiere imprimirlo ver CU Imprimir Reporte.</p>	
	<p>10. El Caso de Uso se Termina.</p>

Nombre del Caso de Uso	Consultar Actividades
Actor	Evaluador
Propósito	Garantizar que le evaluador vea las actividades de un profesor.
Precondiciones	El usuario debe pertenecer al dominio UCI El usuario debe tener los permisos necesarios para poder ejecutar esta acción.
Poscondiciones	Se consulta las actividades de un profesor.
Resumen	El caso de uso se inicia cuando el actor accede a la opción de Ver Mis Profesores. El Usuario debe escoger el nombre del profesor que desea consultar y el sistema muestra la interfaz del profesor. El actor accede a la opción de Ver Actividades.
Referencias	RF5
Curso Normal de Eventos	
Acciones del actor	Respuesta del sistema
1. El caso de uso se inicia cuando el actor selecciona la opción de Ver Mis Profesores.	
	2. Le muestra una lista con los Nombres y Fotos de todos los profesores que pertenecen al actor.
3. Selecciona al profesor deseado.	
	4. Muestra la interfaz del profesor.

5. Selecciona la opción de Ver Actividades.	
	6. Muestra una interfaz con las opciones: <ul style="list-style-type: none"> ➤ Actividades en un Día ➤ Actividades en una Semana ➤ Actividades en un Mes ➤ Actividades en un Trimestre.
7. Selecciona el Periodo que desea consultar.	
	8. Si tiene al menos una actividad se muestra, en caso contrario ir a CA1.
	9. El Caso de Uso se Termina
CA1: Muestra un mensaje informando que el Profesor no tiene actividad en dicho Periodo.	

Nombre del Caso de Uso	Registrar Indisciplinas.
Actor	Evaluador
Propósito	Garantiza que el evaluador registre todas las disciplinas que se le reporten de sus profesores.
Precondiciones	El usuario debe pertenecer al dominio UCI El usuario debe tener los permisos necesarios para poder ejecutar esta acción.
Poscondiciones	Se registra una disciplina.

Resumen	El caso de uso se inicia cuando el actor accede a la opción de Registrar Indisciplinas. El Usuario escribe el nombre del profesor que desea adicionarle una indisciplina y la registrar.
Referencias	RF6
Curso Normal de Eventos	
Acciones del actor	Respuesta del sistema
1. El caso de uso se inicia cuando el actor selecciona la opción de Registrar Indisciplinas.	
	2. Le muestra una interfaz con campos para llenar: <ul style="list-style-type: none"> ➤ Nombre del profesor ➤ Tipo de indisciplina ➤ Sanción ➤ Problemas que trajo dicha indisciplina.
3. Llena los Campos mostrados.	
4. Registra la indisciplina.	
	5. Se actualiza la Base de Datos.
	6. El Caso de Uso se Termina.

Nombre del Caso de Uso	Gestionar Página Principal
Actor	Evaluador
Propósito	Tener actualizada la página principal del sistema.
Precondiciones	El usuario debe pertenecer al dominio UCI, y tiene que ser el administrador del

	sistema.
Poscondiciones	Se muestran las actividades más importantes a nivel de Facultad como a nivel de Universidad, se muestran los profesores más destacados del trimestre y, se mantiene informado como se desarrolla la ED en la Universidad o se Actualiza el Horario.
Resumen	El caso de uso se inicia cuando el actor accede a la opción de Gestionar Página Principal, y actualiza la página principal, puede ser reflejando la tareas más importantes, los profesores más destacado, modificando la ED o actualizando el Horario Docente.
Referencias	RF7
Curso Normal de Eventos	
Acciones del actor	Respuesta del sistema
1. El caso de uso se inicia cuando el actor accede a la opción de Actualizar Página Principal.	
	2. Muestra las opciones de Agregar Tareas más Importantes, Agregar Profesores más Destacados, Quitar Tareas, Quitar Profesor, Modificar como se Desarrolla la ED en la Universidad, Actualizar Horario.
3. Selecciona la opción deseada.	
	4. Gestiona la selección ➤ Agregar Tareas más

	<p>Importantes.</p> <ul style="list-style-type: none"> ➤ Agregar Profesores más Destacados. ➤ Quitar Tareas. ➤ Quitar Profesor. ➤ Modificar ED. ➤ Actualizar Horario
	5. Finalizar el caso de uso.
Curso Alternativo de los eventos	
4. a Accede a la opción que le permite Agregar Tareas más Importantes.	
	<p>4.a.1 Le muestra unos campos a llenar:</p> <ul style="list-style-type: none"> ➤ Nombre. ➤ Fecha. ➤ Hora.
4. a.2 Llena los campos.	
4. a.3 Se marca el botón de adicionar.	
	4. a.4 Se actualiza la página.
	4.a.5 El Caso de Uso se Termina
4. b Accede a la opción que le permite Agregar Profesor más Destacado.	
	<p>4.b.1 Le muestra unos campos a llenar:</p> <ul style="list-style-type: none"> ➤ Nombre. ➤ Apellidos. ➤ Principal Actividad.
4. b.2 Llena los campos.	
4. b.3 Se marca el botón de adicionar.	
	4. b.4 Se actualiza la página.

	4.b.5 El caso de Uso se termina
4. c Accede a la opción que le permite Quitar Tareas.	
	4. c.1 Muestra las tareas que están en la página principal.
4. c.2 Escoge las que desea eliminar.	
4. c.3 Se marca el botón de eliminar.	
	4. c.4 El Caso de Uso se termina.
4. d Accede a la opción que le permite Quitar Profesor.	
	4. d.1 Muestra los profesores que están en la página principal.
4. d.2 Escoge los que quiere eliminar.	
4. c.3 Se marca el botón de eliminar.	
	4. c.4 El Caso de Uso se termina.
4.e Accede a la opción que le permite Modificar ED.	
	4. e.1 Le muestra la descripción de cómo se desempeña la ED anteriormente.
4. e.2 Hace los cambios que se dieron a conocer por la Universidad.	
	4. e.3 Se actualiza dicho proceso de Evaluación.
	4. e.4 El Caso de Uso se Termina.
4. f Accede a la opción que le permite Actualizar Horario.	
	4. f.1 Le Muestra el Horario Viejo.
4. f.2 Realiza los cambios correspondientes a la semana y lo actualiza.	
	4. f.3 Guarda los cambios correspondientes.

	4.f.4 El Caso de Uso se Termina
--	---------------------------------

Nombre del Caso de Uso	Justificar
Actor	Profesor
Propósito	Justificar una actividad.
Precondiciones	El usuario debe pertenecer al dominio UCI Debe tener actividades asignadas.
Poscondiciones	Se justificó una actividad.
Resumen	El caso de uso se inicia cuando el actor accede a la opción de Justificar. El sistema muestra una interfaz con las opciones de Actividad Inmediata o Actividades Futuras.
Referencias	RF8
Curso Normal de Eventos	
Acciones del actor	Respuesta del sistema
1. El caso de uso se inicia cuando el actor selecciona la opción de Justificar.	
	2. Muestra las opciones de Actividad Inmediata y Actividades Futuras.
3. Selecciona la opción deseada.	
	4. Gestiona la selección <ul style="list-style-type: none"> ➤ Actividad Inmediata. ➤ Actividades Futuras.
	5. Finalizar el caso de uso.
Curso Alternativo de los eventos	
4. a Accede a la opción de Actividad Inmediata.	

	4. a.1 Le muestra todas sus actividades.
4. a.2 Escoge la que va a justificar.	
	4. a.3 Le Muestra un formulario para que Justifique el porqué.
4. a.4 La Justifica.	
	4. a.5 El Sistema manda la justificación con el nombre de la actividad al su Evaluador.
	4.a.6 El Caso de Uso se Termina
4. b Actividades Futuras.	
	4. b.1 Muestra todas sus actividades.
4. b.2 Marca Todas las actividades que no puede cumplir y las adiciona a la lista de Futuras Actividades que no se Cumplirán.	
	4.b.3 Le Muestra varias opciones: <ul style="list-style-type: none"> ➤ Justificación. ➤ Fecha de Salida. ➤ Posible Fecha de regreso.
4. b.4 Llena todos los campos.	
	4. b.5 El sistema manda la lista de Futuras Actividades que no se Cumplirán y los datos solicitados a su Evaluador y a el Decano.
	4.b.7 El caso de Uso se termina

Nombre del Caso de Uso	Consultar Actividades Asignadas.
Actor	Profesor

Propósito	Tener conocimientos de sus tareas y el estado de cada una.
Precondiciones	El usuario debe pertenecer al dominio UCI. No puede ser un estudiante.
Poscondiciones	Se muestran las tareas que le pertenecen.
Resumen	El caso de uso se inicia cuando el actor desde su página correspondiente accede a la opción de Actividades Asignadas y el sistema le muestra varias opciones.
Referencias	RF9
Curso Normal de Eventos	
Acciones del actor	Respuesta del sistema
1. El caso de uso se inicia cuando el actor accede a la opción de Actividades Asignadas.	
	2. El sistema le muestra estas opciones: <ul style="list-style-type: none"> ➤ Todas las Actividades. ➤ Actividades Cumplidas. ➤ Actividades por Cumplir
3. Selecciona una opción.	
	4. Le muestra los resultados correspondientes a la opción escogida.
	5. El Caso de Uso se Termina.

Nombre del Caso de Uso	Proponer Evaluación
Actor	Evaluador
Propósito	Propone una Evaluación
Precondiciones	El usuario debe pertenecer al dominio UCI. Tiene que ser un Evaluador. No puede pasar del día 24 del último mes del trimestre.
Poscondiciones	Se generan las evaluaciones de los profesores.
Resumen	El caso de uso se inicia cuando el actor desde su página correspondiente accede a la opción de Generar Evaluación y el sistema comienza a generar las evaluaciones, emite un reporte sobre esas evaluaciones, y le manda un correo al evaluador superior con las mismas.
Referencias	RF10
Curso Normal de Eventos	
Acciones del actor	Respuesta del sistema
6. El caso de uso se inicia cuando el actor accede a la opción de Generar Evaluación.	
	7. El sistema empieza a generar las evaluaciones.
	8. Emite un reporte de todos los profesores con sus evaluaciones.
	9. Envía un correo al evaluador superior con dichas evaluaciones.
	10. El Caso de Uso se Termina.

Nombre del Caso de Uso	Estado de la Actividad
Actor	Profesor
Propósito	Cambiar el estado de la actividad.
Precondiciones	El usuario debe pertenecer al dominio UCI. El estado de la actividad tiene que ser Nula.
Poscondiciones	Se le puso a la actividad Cumplida o Incumplida
Resumen	El caso de uso se inicia cuando el actor desde su página correspondiente accede a todas sus actividades, escoge la actividad que desea y le cambia el estado.
Referencias	RF11
Curso Normal de Eventos	
Acciones del actor	Respuesta del sistema
El caso de uso se inicia cuando el actor accede a la opción de Mis Actividades	
	1. El sistema le muestra las actividades.
Va a la opción Estado.	
	2. El sistema le muestra dos opciones: ➤ Cumplida ➤ Incumplida
Escoge una opción.	

	3. Se actualiza el estado de la Actividad.
	4. El Caso de Uso se Termina.

Nombre del Caso de Uso	Verificar Estado
Actor	Evaluador
Propósito	Ver si el estado de la actividad es el correcto.
Precondiciones	El usuario debe pertenecer al dominio UCI. Tiene que ser un Evaluador.
Poscondiciones	Se Acepta o no la actividad.
Resumen	El caso de uso se inicia cuando el actor desde su página correspondiente accede a la opción Verificar Actividad y después escoge al profesor que desea chequear.
Referencias	RF12
Curso Normal de Eventos	
Acciones del actor	Respuesta del sistema
1. El caso de uso se inicia cuando el actor accede a la opción de Verificar Actividad.	
	2. El sistema le muestra todos los profesores que le pertenece.
3. Escoge al profesor que desea chequear.	
	4. Le muestra todas sus actividades con su estado.
5. Marca la actividad que desea	

cambiarle su estado.	
	6. Le muestra la opción de Cambiar Estado: <ul style="list-style-type: none"> ➤ Cumplida ➤ Incumplida
7. Le cambia el Estado.	
8. Pone la justificación del Cambio	
	9. Se actualiza el estado de la actividad.
	10. El Caso de Uso se Termina.

Nombre del Caso de Uso	Imprimir Reportes
Actor	Evaluador
Propósito	Imprimir reportes
Precondiciones	El usuario debe pertenecer al dominio UCI. Tiene que ser un Evaluador. Debe seleccionar al menos un reporte.
Poscondiciones	Se imprime al menos un reporte.
Resumen	El caso de uso se inicia cuando el actor desde su página correspondiente accede a la opción Imprimir Reporte y el sistema le muestra todos los posibles reportes que puede imprimir.
Referencias	RF13
Curso Normal de Eventos	
Acciones del actor	Respuesta del sistema
1. El caso de uso se inicia cuando el actor accede a la opción de Imprimir Reportes.	

	2. El sistema le muestra todos los posibles reportes.
3. El actor escoge los reportes que desea imprimir.	
	4. Imprime los reportes correspondientes.
5. Recoge lo imprimido	
	6. El Caso de Uso se Termina.

Nombre del Caso de Uso	Autoevaluación
Actor	Profesor
Propósito	Hacer la autoevaluación.
Precondiciones	El usuario debe pertenecer al dominio UCI. No puede pasar del día 20 del último mes del trimestre.
Poscondiciones	Se guarda la autoevaluación en el sitio.
Resumen	El caso de uso se inicia cuando el actor desde su página correspondiente accede a la opción Mi Autoevaluación y el sistema le da la posibilidad de redactarla.
Referencias	RF14
Curso Normal de Eventos	
Acciones del actor	Respuesta del sistema
1. El caso de uso se inicia cuando el actor accede a la opción de Mi Autoevaluación.	
	2. El sistema le da la posibilidad de redactar su autoevaluación.
3. La redacta y la salva en el	

sistema.	
	4. Guarda todos los cambios.
	5. El Caso de Uso se Termina.

Nombre del Caso de Uso	Recordar Actividad
Actor	Usuario
Propósito	Permitir recordar una actividad
Precondiciones	El usuario debe pertenecer al dominio UCI. No puede ser un estudiante.
Poscondiciones	Se inserta el día que necesitas que te recuerden una actividad.
Resumen	El caso de uso se inicia cuando el actor desde su página correspondiente accede a la opción de Recordar e inserta el día que quiere que se le recuerde.
Referencias	RF15
Curso Normal de Eventos	
Acciones del actor	Respuesta del sistema
1. El caso de uso se inicia cuando el actor accede a la opción de Recordar.	
	2. El Sistema le Muestra todas sus Actividades.
3. Escoge cual desea que se le recuerde.	
	4. El sistema muestra un interfaz con los siguientes campos a llenar: <ul style="list-style-type: none"> ➤ Día. ➤ Hora

5. El usuario llena los campos correspondientes.	
	6. El sistema guarda esos datos.
	7. El Caso de Uso se termina

Nombre del Caso de Uso	Opinar sobre Profesor
Actor	Estudiante.
Propósito	Opinar sobre los profesores que le den clase.
Precondiciones	El usuario debe pertenecer al dominio UCI El usuario debe ser un estudiante.
Poscondiciones	Se justificó una actividad.
Resumen	El caso de uso se inicia cuando el actor accede a la opción de Opinar. El sistema muestra una interfaz con todos sus profesores, el actor escoge sobre quien quiere opinar y opina.
Referencias	RF17
Curso Normal de Eventos	
Acciones del actor	Respuesta del sistema
1. El caso de uso se inicia cuando el actor selecciona la opción de Opinar.	
	2. Muestra todos los profesores que tiene que ver con el actor.
3. Selecciona al profesor.	
	4. Le muestra una interfaz para que ponga su opinión.
5. Opinan sobre dicho profesor.	
	6. Le manda opinión al Profesor y se guarda.

	7. Finalizar el caso de uso.
--	------------------------------

Nombre del Caso de Uso	Temas en Foro
Actor	Profesor
Propósito	Opinan sobre un tema que se inserto con anterioridad.
Precondiciones	El usuario debe pertenecer al dominio UCI Tiene que ser un profesor.
Poscondiciones	Se inserto un tema o se opinó sobre un tema.
Resumen	El caso de uso se inicia cuando el actor accede a la opción de Foro. El sistema muestra una interfaz con las opciones de Insertar Tema o de Opinar.
Referencias	RF18
Curso Normal de Eventos	
Acciones del actor	Respuesta del sistema
1. El caso de uso se inicia cuando el actor selecciona la opción de Foro.	
	2. Muestra las opciones de Insertar Tema y de Opinar.
3. Selecciona la opción deseada.	
	4. Gestiona la selección <ul style="list-style-type: none"> ➤ Insertar Tema ➤ Opinar.
	5. Finalizar el caso de uso.
Curso Alternativo de los eventos	
4. a Accede a la opción de Insertar Tema.	

	4.a.1 Muestra las opciones de: <ul style="list-style-type: none"> ➤ Nombre del Tema. ➤ Explicación.
4. a.2 Llena los campos.	
	4. a.3 Se actualiza el foro.
	4.a.6 El Caso de Uso se Termina
4. b Accede a la opción de Opinar.	
	4. b.1 Muestra todos los temas.
4. b.2 Escoge el Tema.	
4. b.3 Opina sobre dicho tema.	
	4. b.4 Actualiza foro.
	4. c.7 El Caso de Uso se termina.

2.4.5 Conclusiones Parciales

Se realizó una amplia descripción referente al negocio y sistema del trabajo, se abordaron detalladamente las distintas etapas que contienen los mismos para su mayor entendimiento, y se mostró una representación y descripción de todos sus procesos, para ello se expusieron tablas y modelados que contienen la información necesaria para llevar a cabo un mejor desarrollo futuro.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

3.1 Introducción

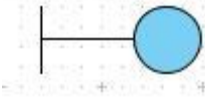
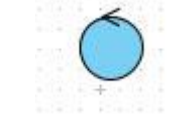
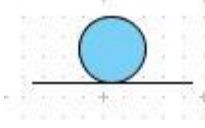
El análisis consiste en obtener una visión del sistema que se preocupa de ver qué hace, de modo que sólo se interesa por los requisitos funcionales. Por otro lado, el diseño es un refinamiento del análisis que tiene en cuenta los requisitos no funcionales, en definitiva cómo cumple el sistema sus objetivos. El diseño debe ser suficiente para que el sistema pueda ser implementado sin ambigüedades.

3.2 Modelo de análisis

En la construcción del modelo de análisis se tienen que identificar las clases que describen la realización de los casos de uso, los atributos y las relaciones entre ellas. Con esta información se construye el Diagrama de clases del análisis, que por lo general se descompone para agrupar las clases en paquetes. Esta descomposición tiene impacto por lo general en el diseño e implementación de la solución.

3.3 Clases de análisis

Se centran en los requisitos funcionales y son evidentes en el dominio del problema porque representan conceptos y relaciones del dominio. Tienen atributos y entre ellas se establecen relaciones de asociación, agregación / composición, generalización / especialización y tipos asociativos. RUP propone clasificar a las clases en:

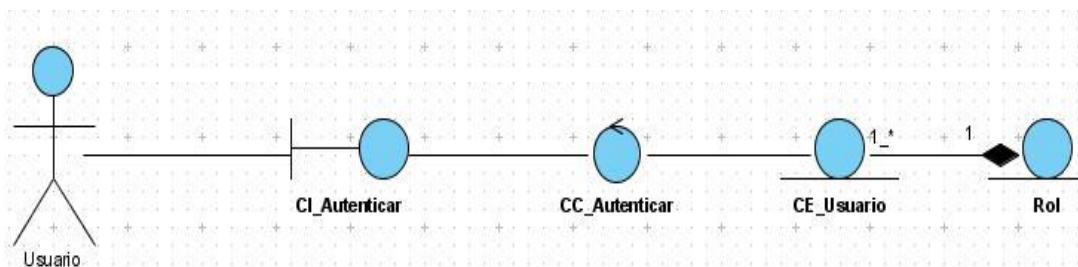
	<p>Clases Interfaz: Modelan la interacción entre el sistema y sus actores.</p>
	<p>Clases Entidad: Modelan información que posee larga vida y que es a menudo persistente.</p>
	<p>Clases Controladoras: Coordinan la realización de uno o unos pocos casos de uso coordinando la actividades de los objetos que implementan la funcionalidad del caso de uso.</p>

3.3.1 Diagrama de clases de análisis

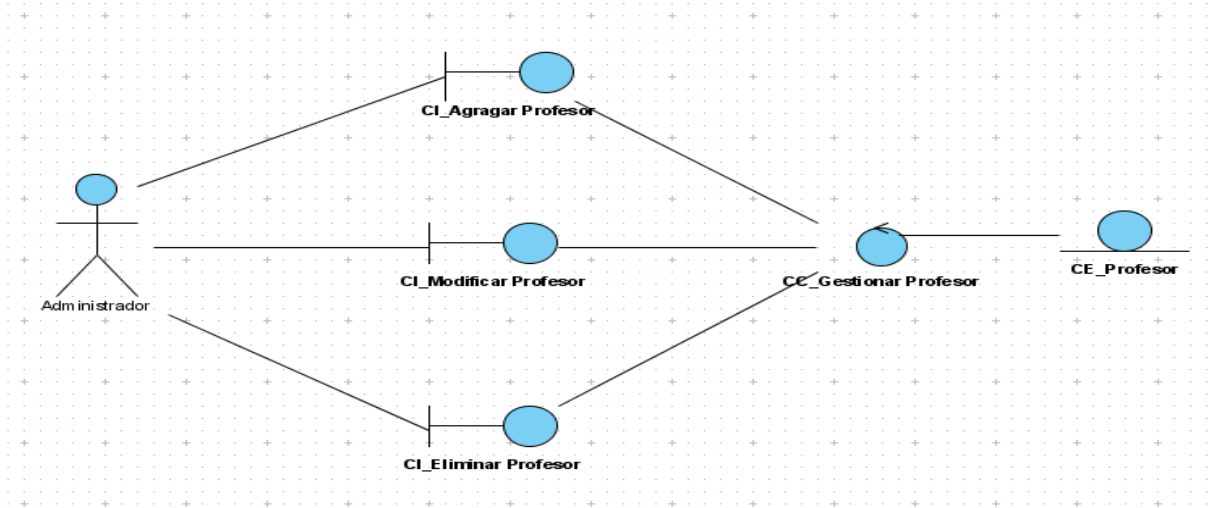
Un diagrama de clases de análisis es un artefacto en el que se representan los conceptos en un dominio del problema, representa las cosas del mundo real. El modelo de análisis puede considerarse como una primera aproximación al modelo de diseño. Son específicamente unos diagramas estáticos que muestran qué es lo que interactúa, pero no cómo interactúa.

A continuación se muestran los diagramas de clases del análisis de todos los casos de uso del sistema:

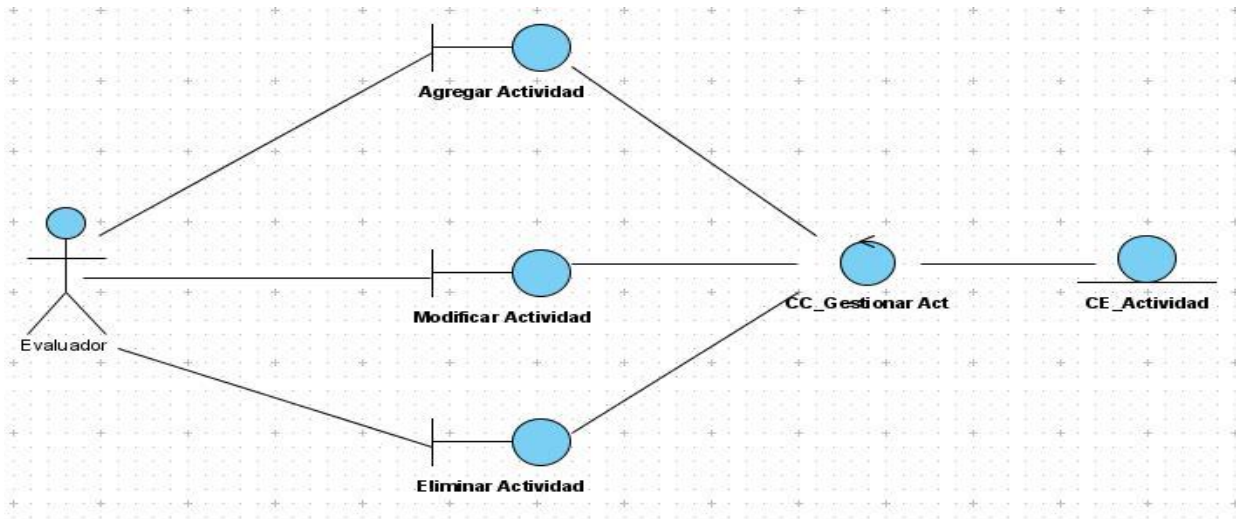
Autenticar Usuario



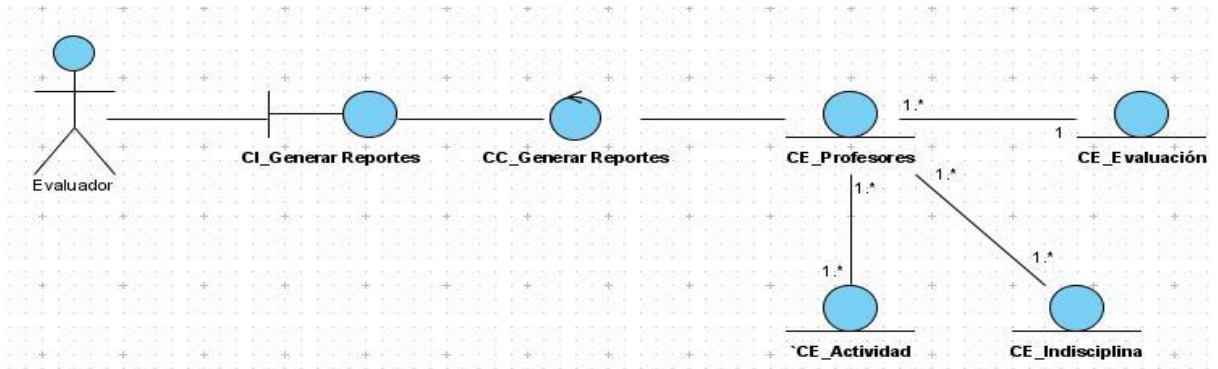
Gestionar profesor



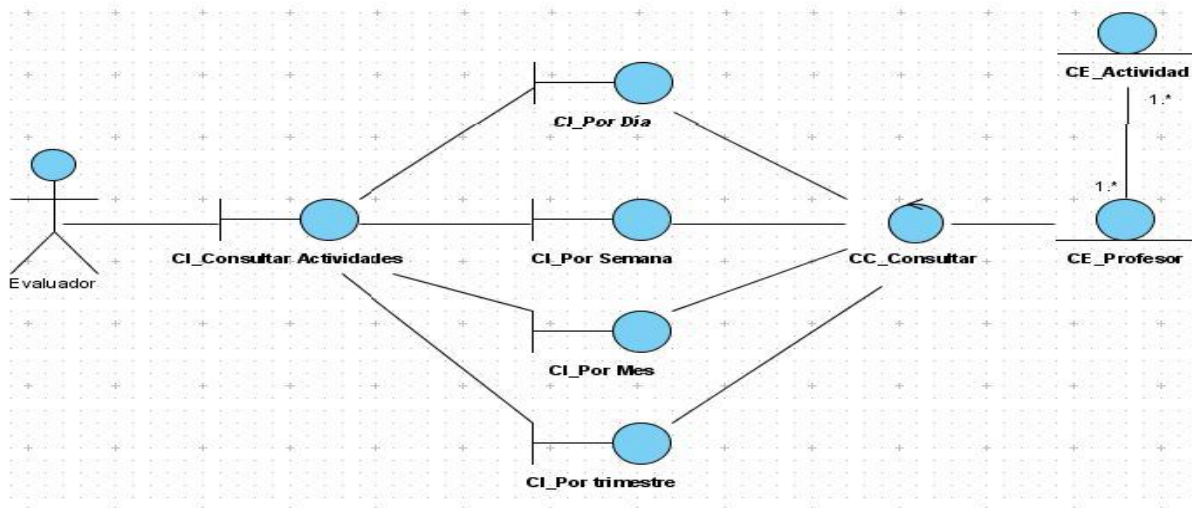
Gestionar Actividad



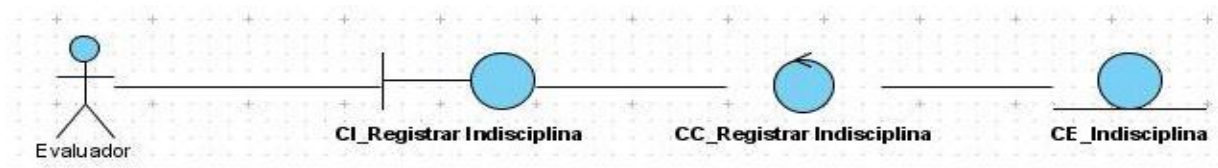
Generar Reportes



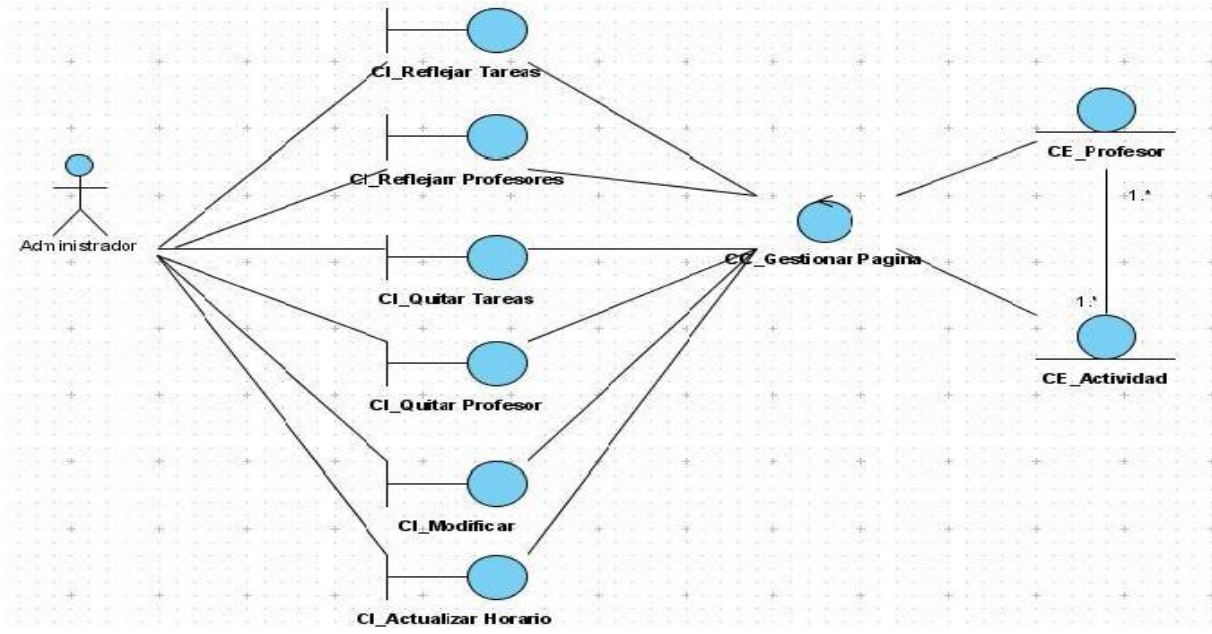
Consultar Actividad



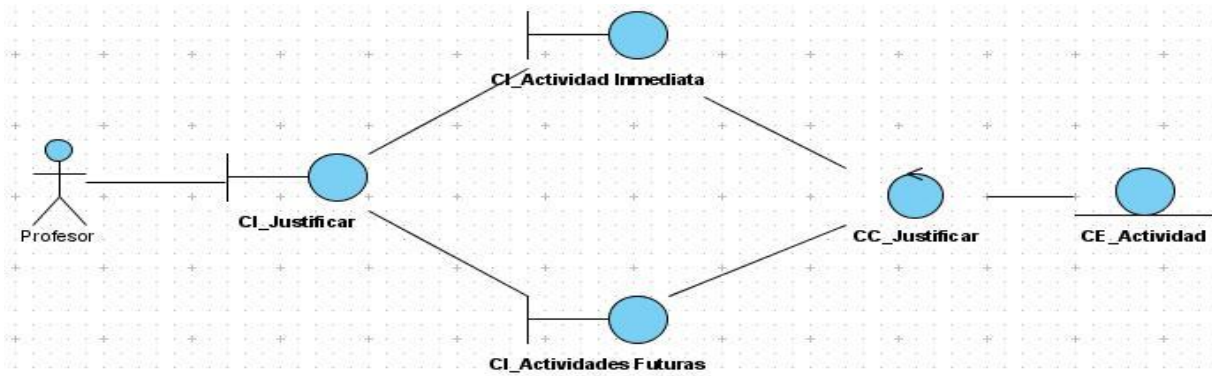
Registrar Indisciplina



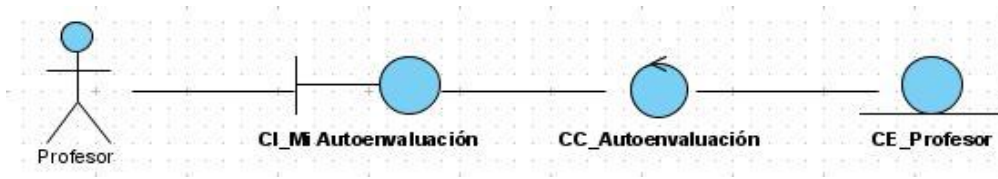
Gestionar Página Principal



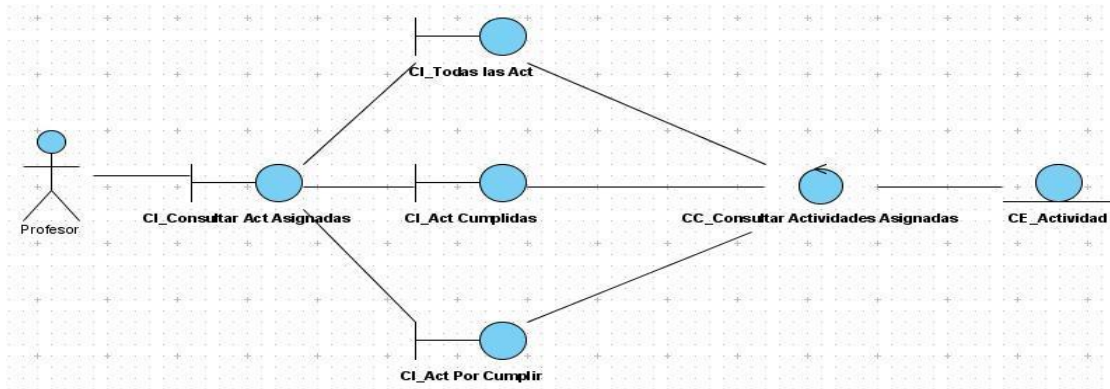
Justificar



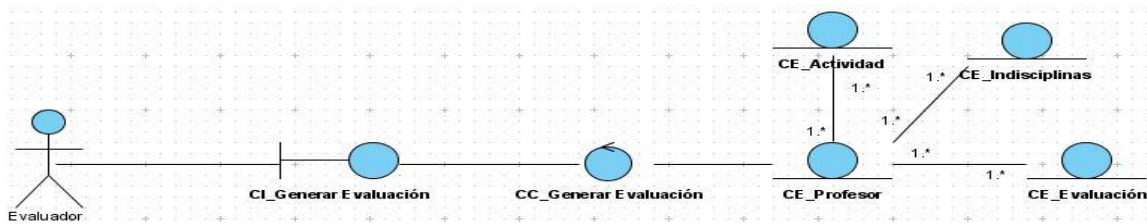
Autoevaluación



Actividades Asignadas



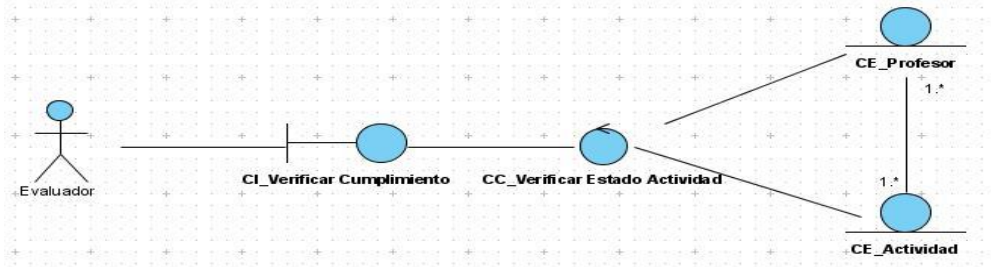
Proponer Evaluación



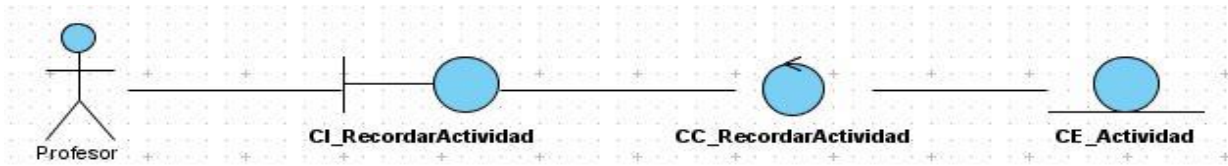
Estado de la Actividad



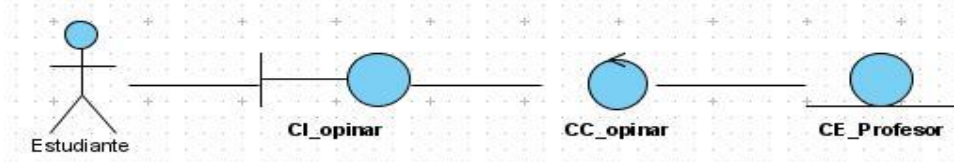
Verificar Actividad



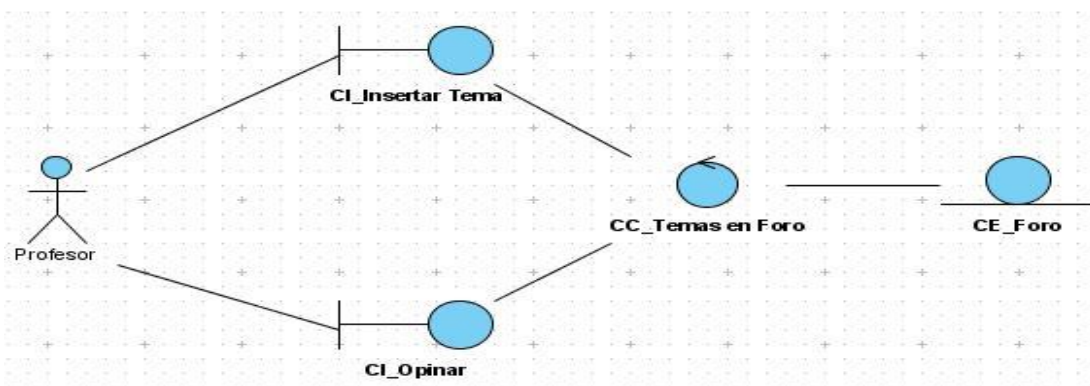
Recordar Actividad



Opinar Sobre Profesor



Temas en Foro



Diagramas de Interacción.

Los diagramas de interacción muestran una interacción concreta: un conjunto de objetos y sus relaciones, junto con los mensajes que se envían entre ellos. Modelan el comportamiento dinámico del sistema; el flujo de control en una operación. Describe la interacción entre objetos, los objetos interactúan a través de mensajes para cumplir ciertas tareas. Las interacciones proveen un comportamiento y típicamente implementan un Caso de Uso.

En el análisis se usan los diagramas de colaboración, ya que el objetivo principal es identificar las funcionalidades de cada objeto y las responsabilidades sobre ellos.

En el Anexo 1 se pueden consultar los diagramas de colaboración para cada uno de los escenarios de los casos de usos.

3.4 Diseño

En el diseño se modela el sistema y se encuentra su forma para que soporte todos los requisitos- incluyendo los requisitos no funcionales y otras restricciones que le suponen. Una entrada esencial en el diseño es el resultado del análisis, esto es el modelo de análisis. Los propósitos del diseño son:



- Adquirir una comprensión en profundidad de los aspectos relacionados con los requisitos no funcionales y restricciones relacionadas con los lenguajes de programación, componentes reutilizables, sistemas operativos, tecnologías de distribución y concurrencia, tecnologías de interfaz de usuario, tecnologías de gestión de transacciones, etc.
- Crear una entrada apropiada y un punto de partida para actividades de implementación subsiguientes capturando los requisitos o subsistemas individuales, interfaces y clases.
- Ser capaces de descomponer los trabajos de implementación en partes más manejables que puedan ser llevadas a cabo por diferentes equipos de desarrollo, teniendo en cuenta la posible concurrencia.


3.4.1 Modelo de diseño

El modelo de diseño es un modelo de objetos que adquiere una comprensión en profundidad de los aspectos relacionados con los requisitos no funcionales y restricciones relacionadas con los lenguajes de programación, componentes reutilizables, sistemas operativos, tecnologías de distribución concurrencias y, tecnologías de interfaz de usuario. Además ayuda a descomponer los trabajos de implementación en partes más manejables que puedan ser llevadas a cabo por diferentes equipos de desarrollo, teniendo en cuenta la posible concurrencia.

3.4.2 Diagrama de clases de diseño

Un diagrama de clases de diseño es un diagrama que muestra un conjunto de interfaces, colaboraciones y sus relaciones. Los diagramas de clases de diseño se utilizan para modelar principalmente la vista de diseño estática de un sistema. Esto incluye modelar el vocabulario del sistema, las colaboraciones o esquemas. Los diagramas de clases, son importantes, no sólo para visualizar, especificar y documentar modelos estructurales, sino también para construir sistemas ejecutables, aplicando ingeniería directa e inversa.

	Server Page: Representa la página Web que tiene código que se ejecuta en el servidor. Este código interactúa con recursos en el servidor. Las operaciones representan las funciones del código y los atributos las variables visibles dentro del alcance de la página. Esta clase sólo puede tener relaciones con objetos en el servidor.
	Client Page: Una instancia de Página Cliente es una página Web, con formato HTML. Mezcla de datos, presentación y lógica. Son interpretadas por el navegador. Sus atributos son las variables declaradas dentro del script que son accesibles para páginas cualquier función dentro de la página. Cada página cliente es construida por una sola página de servidor.

	<p>Form: Colección de elementos de entrada que son parte de una página cliente. Se relaciona directamente con la etiqueta de igual nombre del HTML. Sus atributos son los elementos de entrada del formulario (Text Field, Text Área, Button, Label, Radio Button, Radio Group, Select, Check Box y Hidden Fields).</p>
<p>→ <<Build>></p>	<p>Build: Representa una asociación especial que relaciona las páginas cliente con las páginas servidor, de forma general se expresa como que las páginas que se encuentran en el servidor construyen las páginas en el cliente. Debe ser una relación direccional, donde una página servidor puede construir una o más páginas cliente.</p>
<p>→ <<Submit>></p>	<p>Submit: Es la relación que se crea siempre entre una página servidor y un formulario, a través de esta relación el formulario manda los valores de sus campos al servidor, para ser procesados por la página servidor.</p>

A continuación se muestra una representación de los diagramas de clases del diseño:

Diagrama de Clase de Diseño: Autenticar Usuario

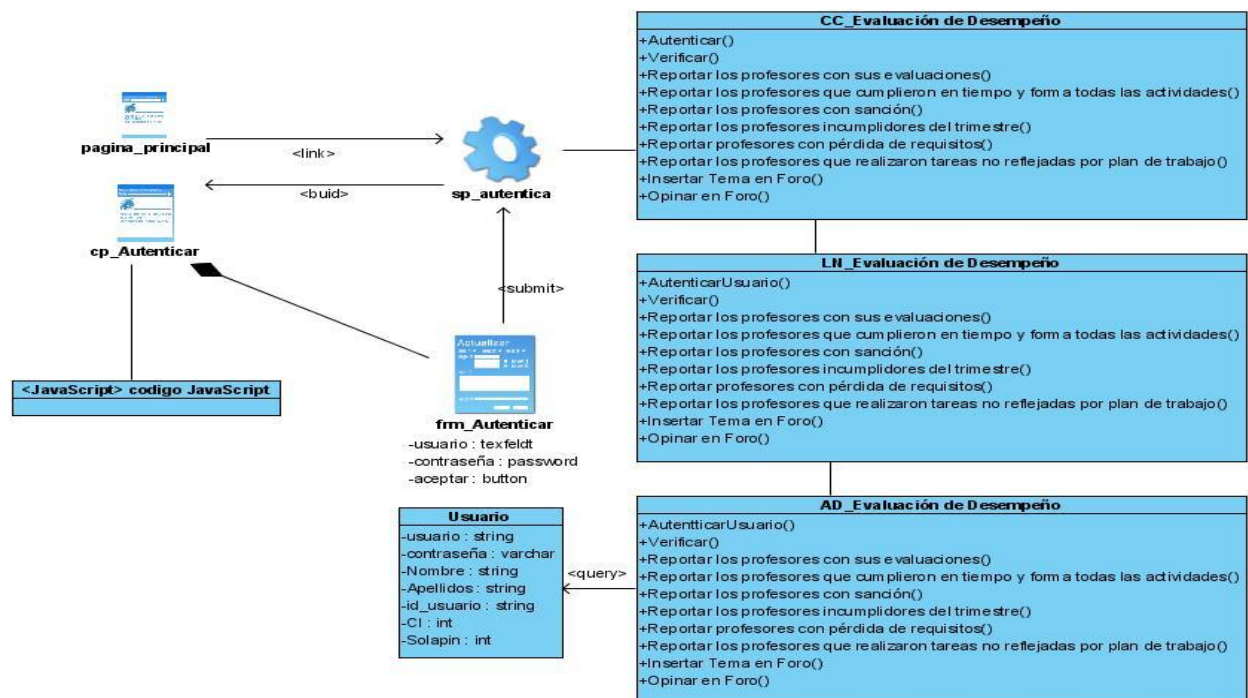


Diagrama de Clase de Diseño: Gestionar Profesor

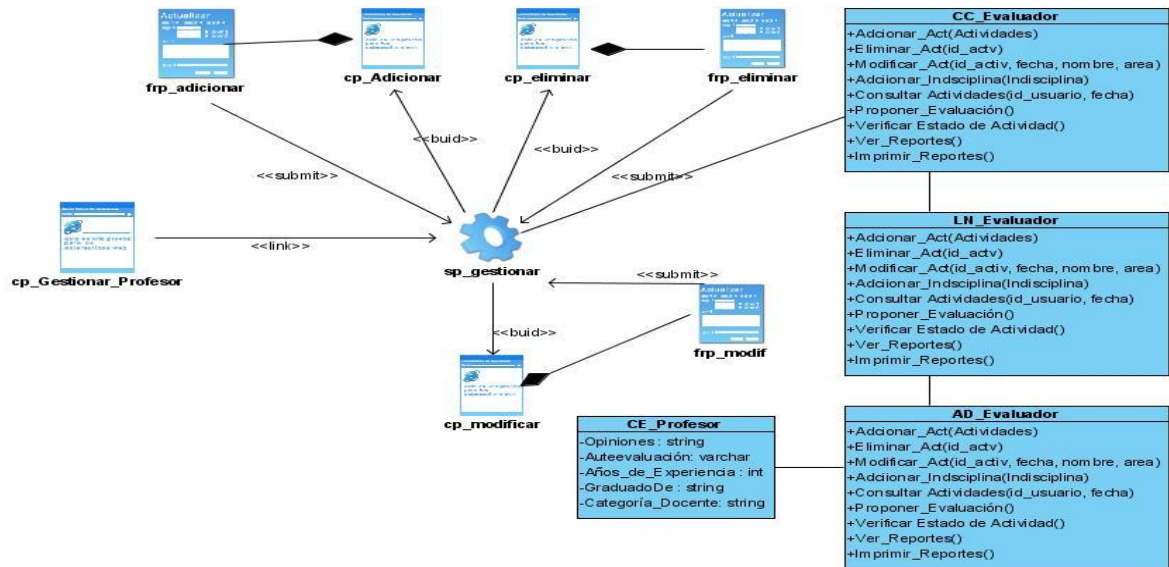


Diagrama de Clase de Diseño: Gestionar Actividad

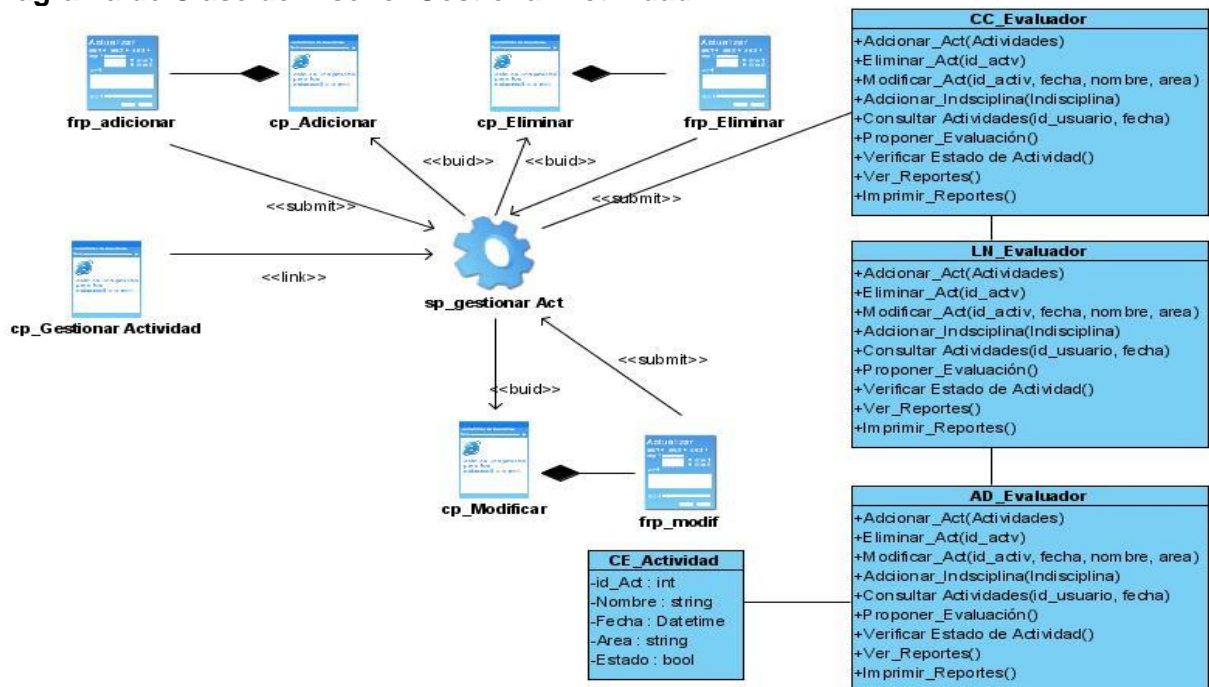


Diagrama de Clase de Diseño: Generar Reportes

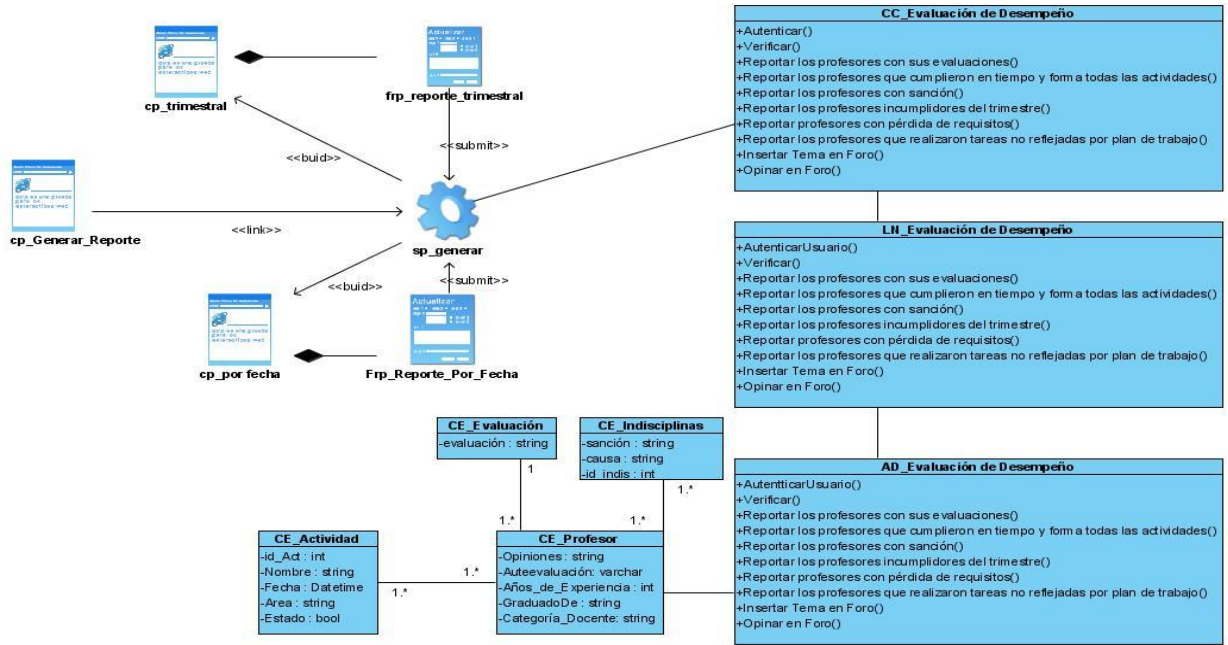


Diagrama de Clase de Diseño: Consultar Actividad

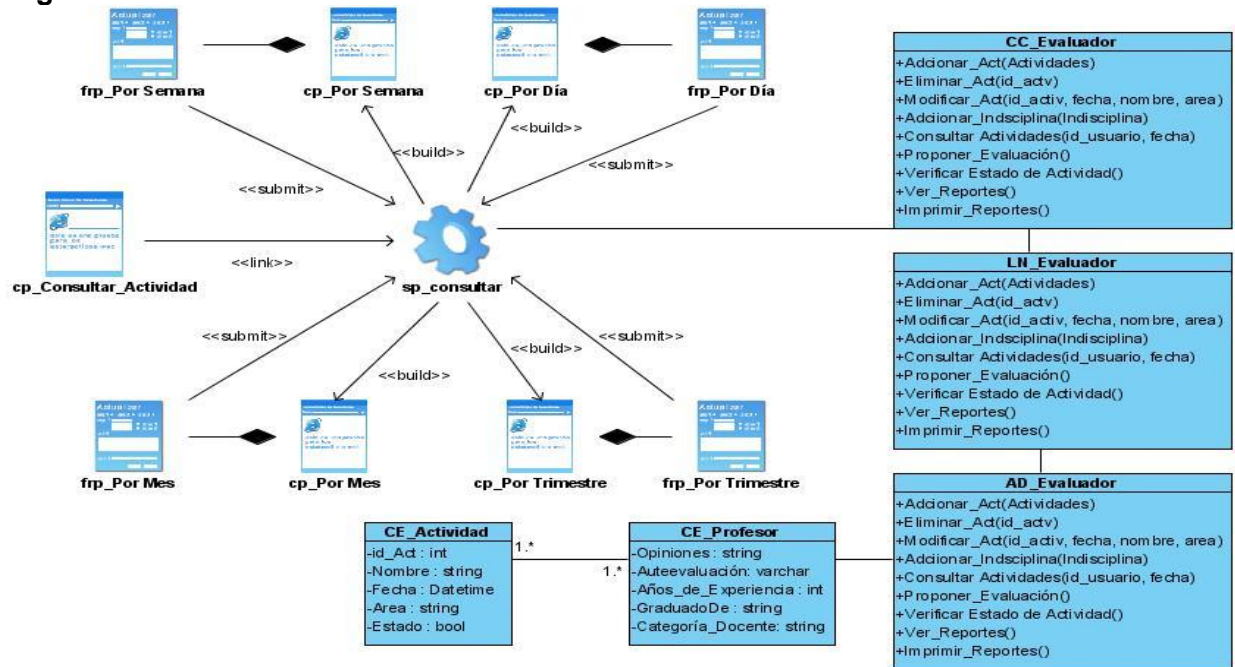


Diagrama de Clase de Diseño: Registrar Indisciplina

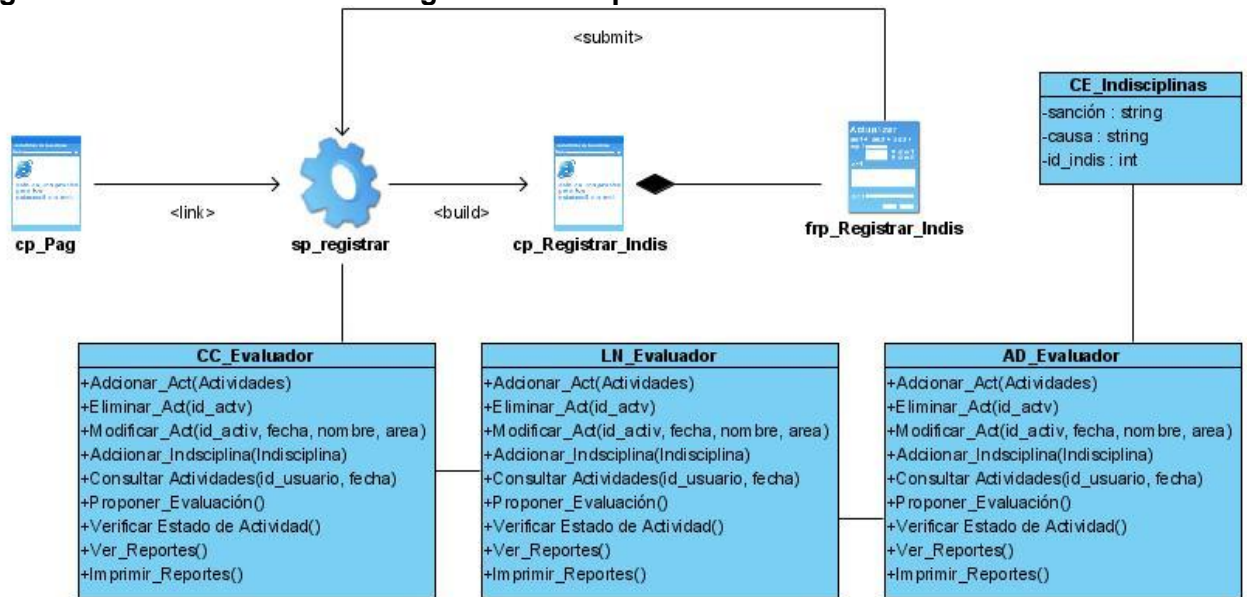


Diagrama de Clase de Diseño: Gestionar Página Principal

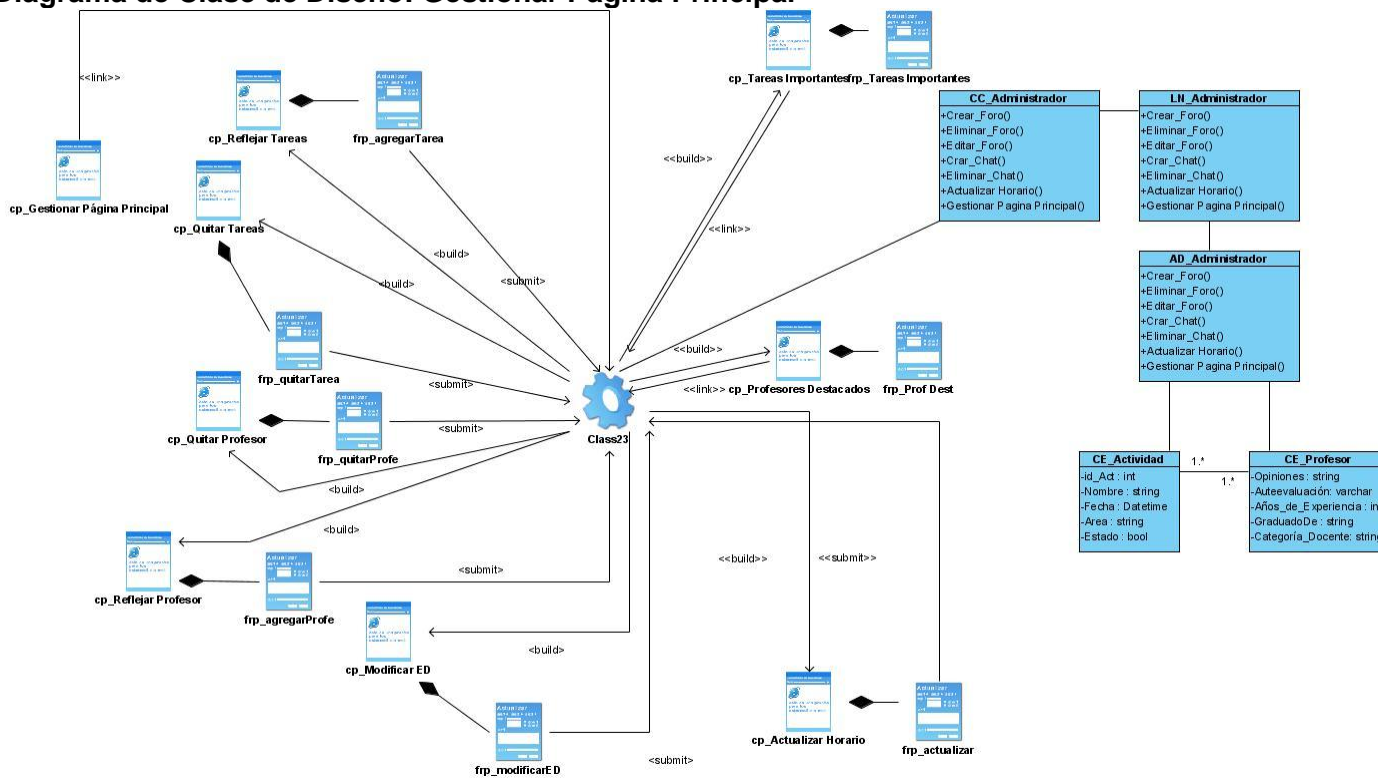


Diagrama de Clase de Diseño: Justificar

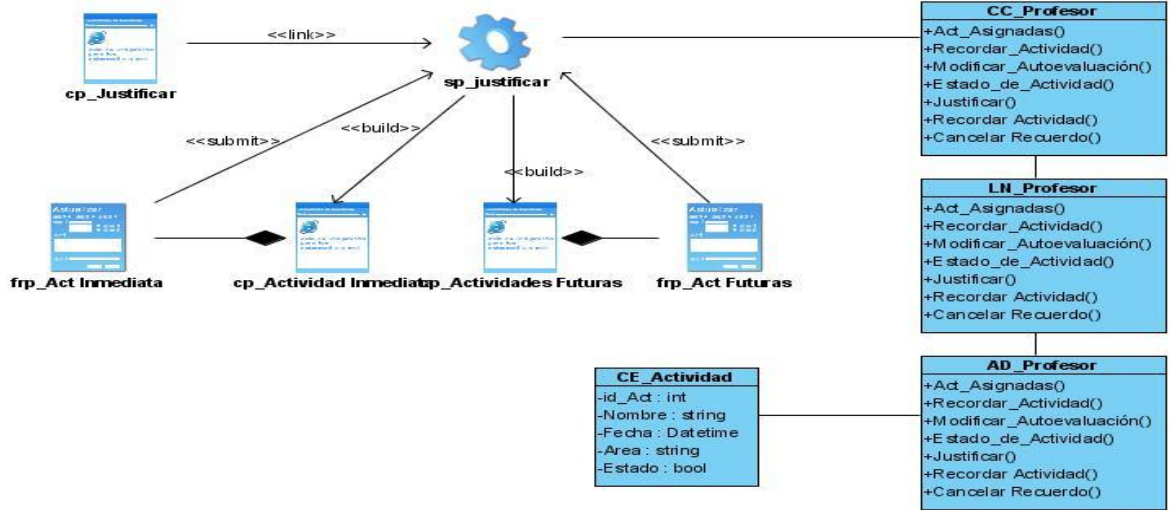


Diagrama de Clase de Diseño: Actividades Asignadas

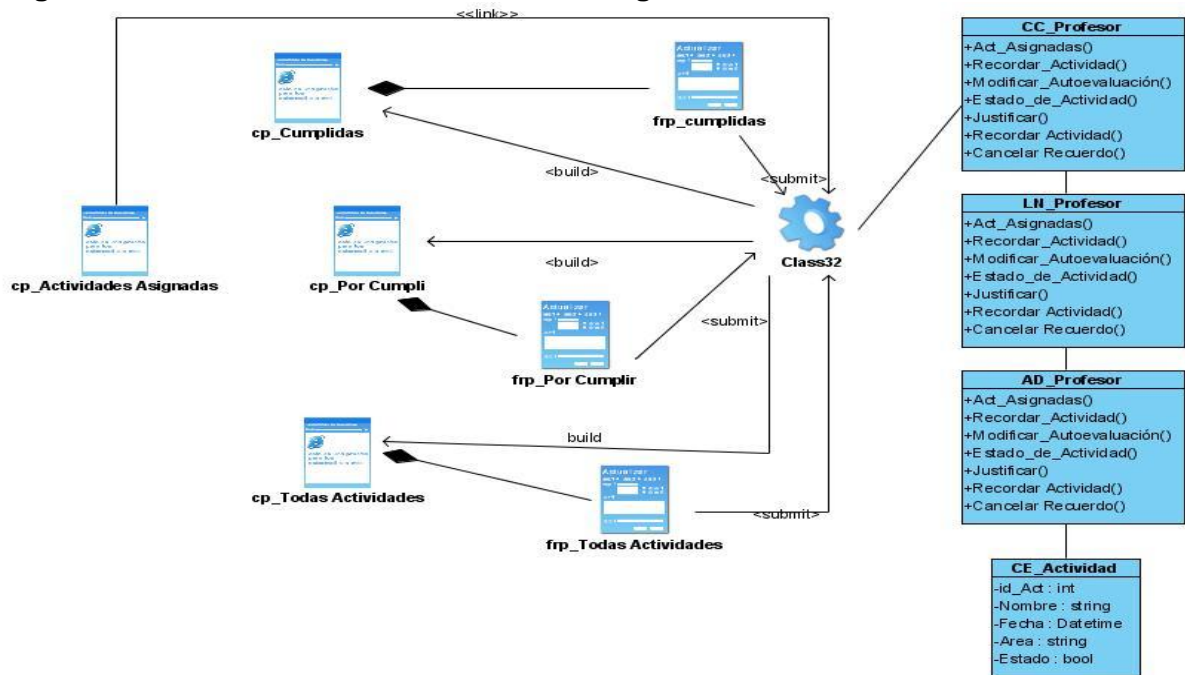


Diagrama de Clase de Diseño: Proponer Evaluación

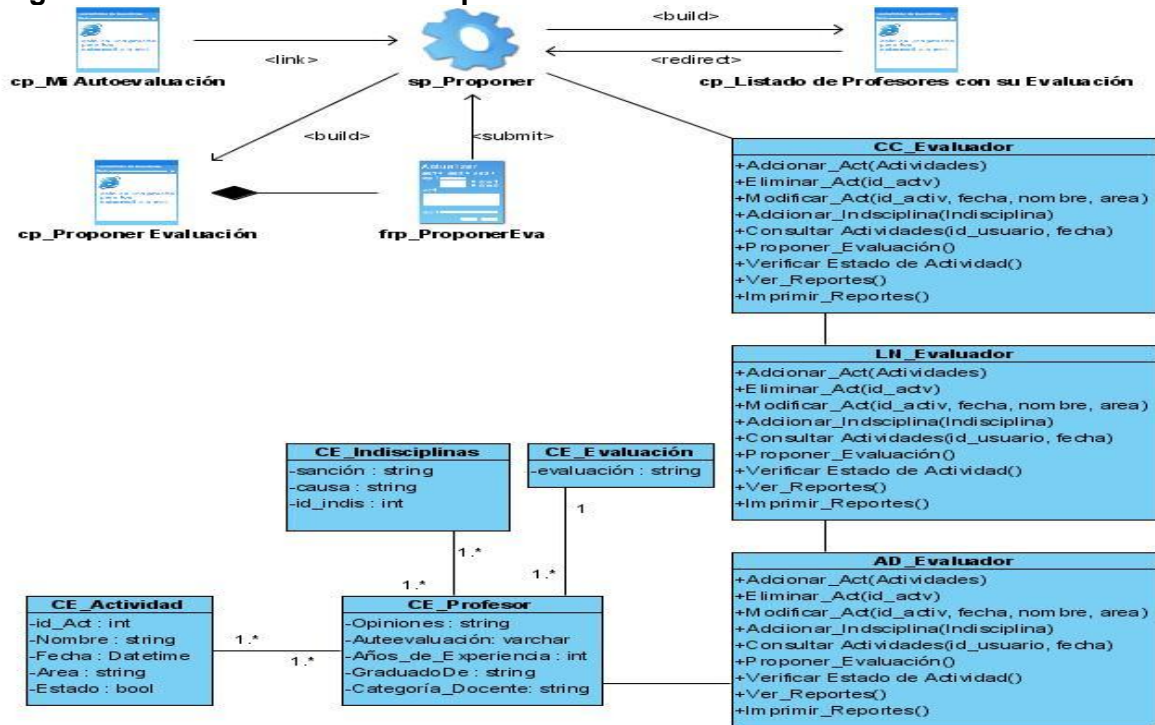


Diagrama de Clase de Diseño: Estado de la Actividad

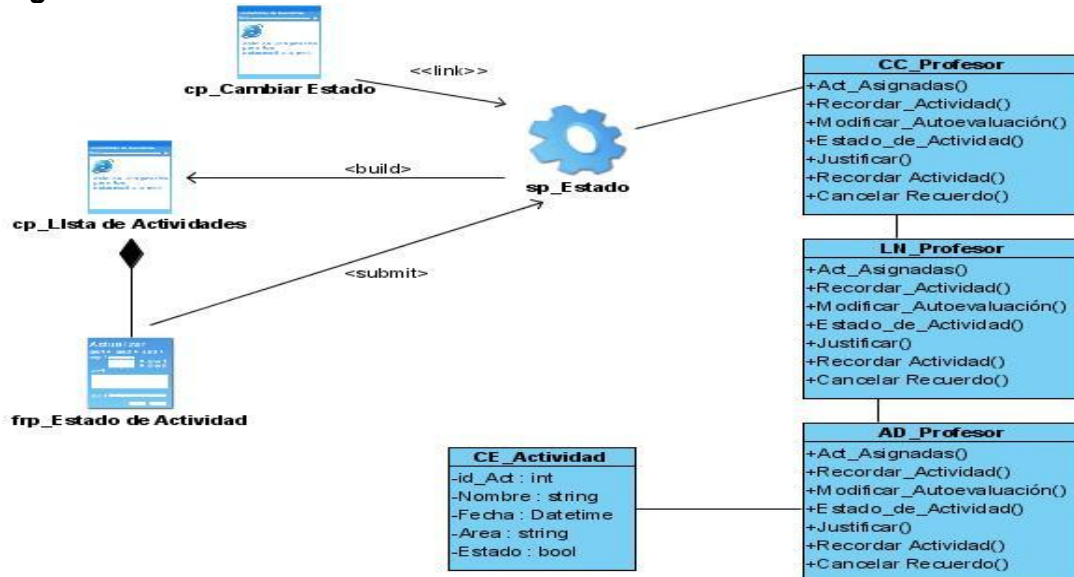


Diagrama de Clase de Diseño: Recordar Actividad

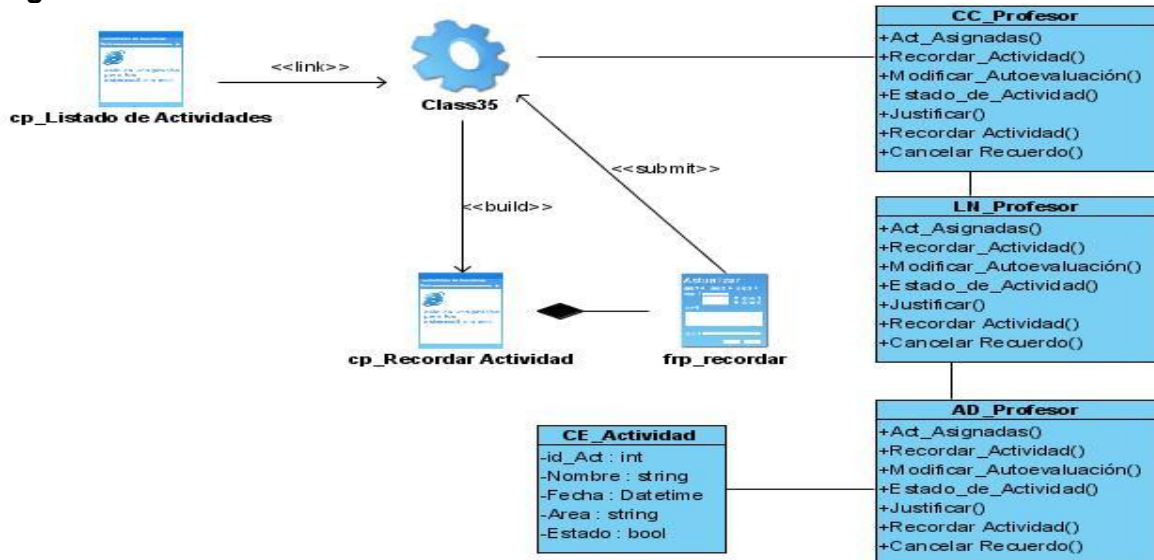


Diagrama de Clase de Diseño: Opinar Sobre Profesor

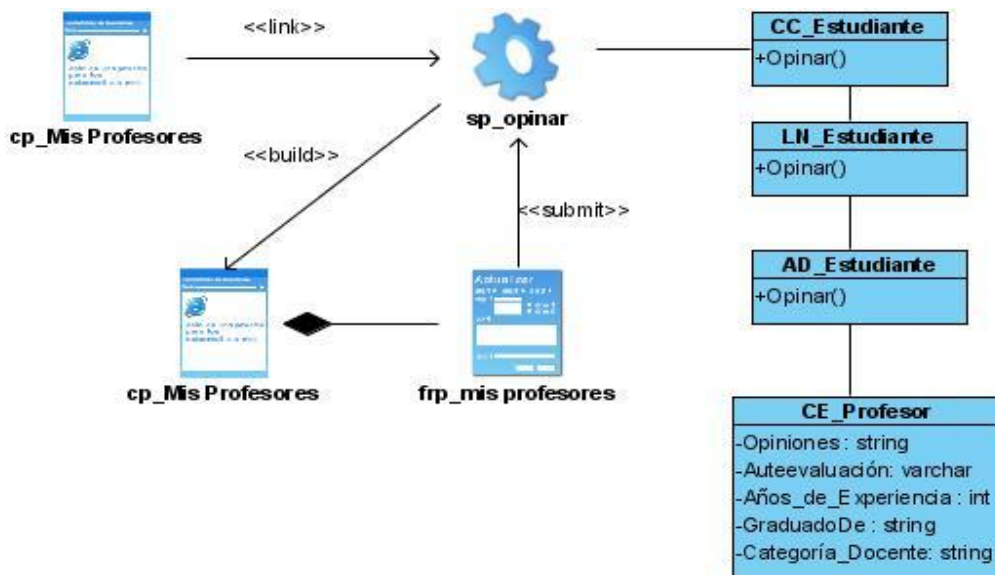
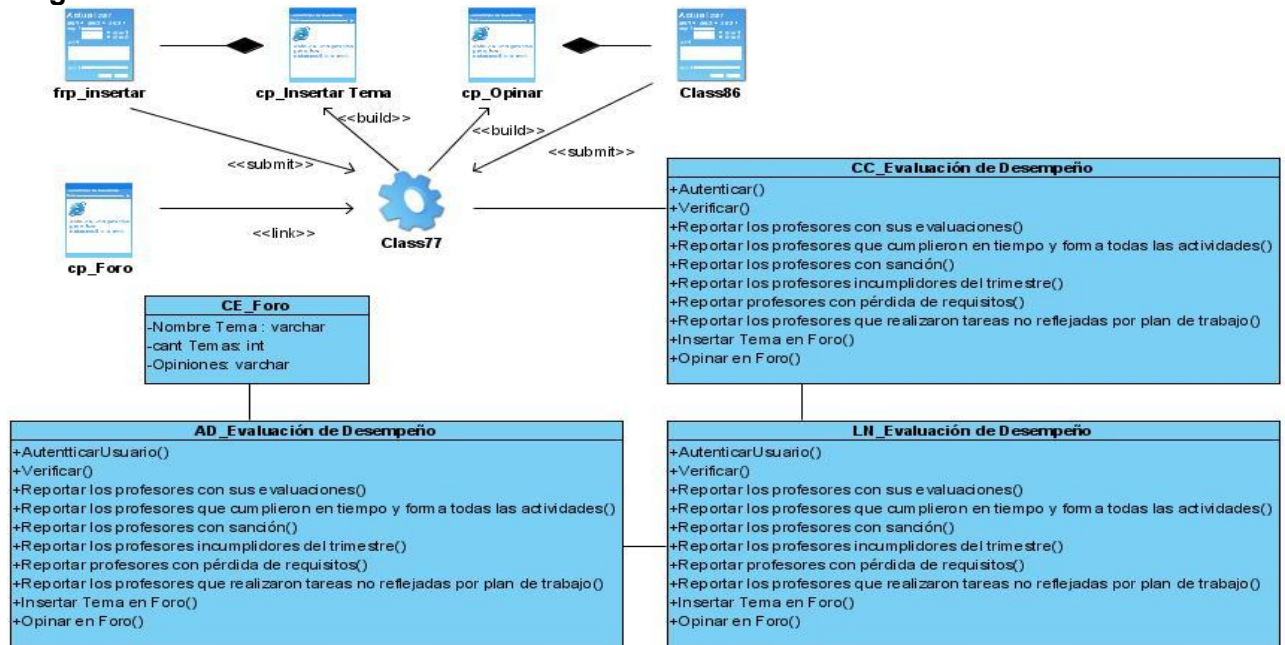


Diagrama de Clase de Diseño: Temas en Foro



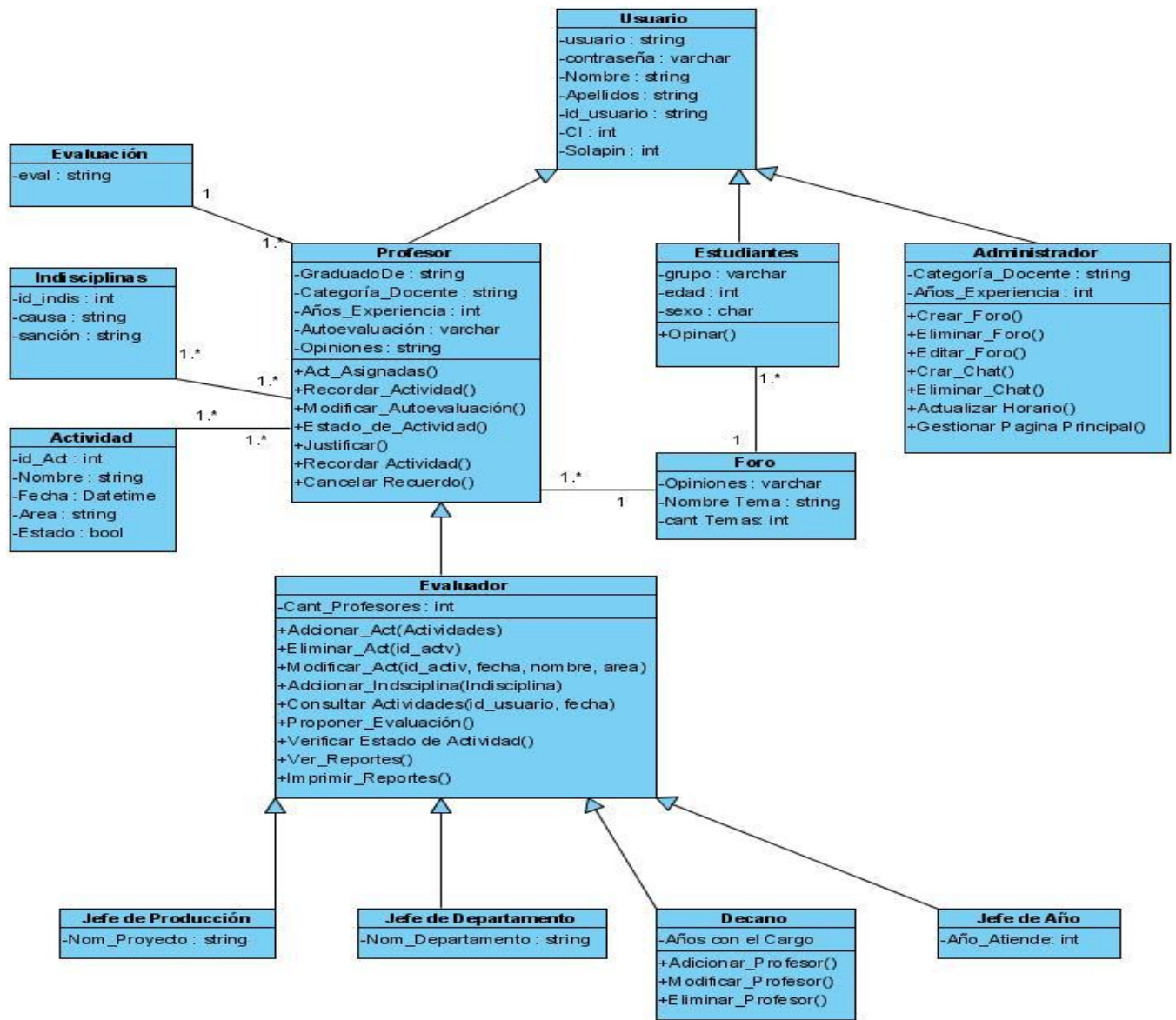
3.5 Diagramas de Secuencia

Los diagramas de secuencia forman parte del modelado dinámico del sistema y proporcionan una vista detallada de los casos de uso. Son diagramas que muestran la interacción organizada de objetos, mediante mensajes que se envían entre sí, en una secuencia de tiempo. Son útiles para observar la vida de los objetos en un sistema, identificar llamadas a realizar o posibles errores del modelado estático que imposibiliten el flujo de información.

En el Anexo 2 se pueden consultar los diagramas de secuencia para cada uno de los escenarios de los casos de usos.

3.6 Diagrama de Clases Persistentes

Las clases persistentes necesitan guardar su estado en un medio permanente ya que estas proporcionan un almacenamiento físico permanente de la información de la clase para la copia de seguridad en caso de caída del sistema, o para el intercambio de información.



3.8 Principios de diseño

El diseño ha sido elaborado pensando en los usuarios finales, siendo este consistente en todas las páginas con el fin de lograr que el usuario se sienta cómodo y alcance rápidamente su objetivo. Para ello se ha seguido una serie de principios que son de importancia para el buen funcionamiento de la aplicación ya que contribuyen con el desarrollo del sistema, estos se expondrán a continuación.

3.8.1 Estándares de la interfaz de aplicación

Para la aplicación se definió un estándar para todas las páginas. Se usarán principalmente colores azules en diferentes tonalidades, blanco, gris claro y negro con el fin de dar claridad al diseño y hacer más agradable la vista. Se utilizará el color rojo para resaltar los errores de campos requeridos, y para los mensajes de operaciones no válidas mostrará una ventana explicando brevemente cual es el error. De forma general se realizarán varias operaciones por página, de forma que el usuario no tenga que moverse tanto dentro de la aplicación y así evitar la sobrecarga en las páginas.

3.9 Concepción general de la ayuda

En la página principal de la aplicación se mostrará una serie de informaciones como son:

- Como se realiza la evaluación de Desempeño en la UCI.
- Las tareas más importantes tanto a nivel de Facultad como a nivel de Universidad.
- Horarios (Docente y de Reuniones).
- Profesores más Destacados del Trimestre.

3.10 Tratamiento de Excepciones

Con el fin de lograr un buen funcionamiento en la aplicación se le dará tratamiento a las excepciones que sean indispensables, por ello se prevendrán los errores por parte del usuario pidiéndole solo los datos necesarios para así verificar la integridad de los mismos al introducirlos y antes de permitir el acceso a alguna interfaz o información, o antes de ser almacenados evitando así la inconsistencia.

Los campos obligatorios serán verificados a través de validaciones que utilizará el sistema, tanto en la parte del cliente, como en la parte del servidor. Cuando se introduzca información en un formulario y falten datos se mostrará un mensaje de alerta mostrando el campo o dato que falta por introducir.

Las excepciones que ocurran internamente en la aplicación serán capturadas por el sistema y mostradas como mensaje de error al usuario. Estos mensajes aparecerán en otra ventana, serán específicos y entendibles para los usuarios.

3.11 Conclusiones Parciales

En el desarrollo de este capítulo se muestra una representación gráfica de los diferentes diagramas tanto del análisis como del diseño correspondiente al sistema. Estos son una representación más amplia ya que dan una mejor idea al programador a la hora de implementar las diferentes funcionalidades porque se escenifica los diferentes escenarios importantes del proceso.

CAPÍTULO 4: VALIDACIÓN DE LA PROPUESTA

4.1-Introducción

La planificación de proyectos consiste en estimar los resultados del proyecto desde la primera fase del trabajo hasta la entrega del producto final al cliente y los valores de costo, se hace referencia a los beneficios que traerá consigo esta propuesta demostrando así la factibilidad del mismo.

4.2 -Métricas

El IEEE Standard Glossary of Software Engineering Terms [IEEE93] define métrica como una medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado. Una medida proporciona una indicación cuantitativa de la extensión, cantidad, dimensiones, capacidad o tamaño de algunos atributos de un proceso o producto.

Hay cuatro razones para medir los procesos del software, los productos y los recursos:

- Caracterizar
- Evaluar
- Predecir
- Mejorar

Caracterizamos para entender mucho mejor todos los procesos, productos, recursos, entornos y para establecer las líneas base para diferentes comparaciones con posteriores evaluaciones. Evaluamos con el objetivo de determinar el estado del diseño además de valorar la consecución de los objetivos de calidad y para evaluar el impacto de la tecnología y las mejoras del proceso.

Predecimos para poder planificar con el objetivo establecer metas alcanzables para el coste, planificación, y calidad de manera que se puedan aplicar los recursos apropiados. Medimos para mejorar, la medición nos facilita recoger la información cuantitativa que nos ayuda a identificar obstáculos, problemas de raíz, ineficiencias y otras oportunidades para mejorar la calidad del producto y el rendimiento del proceso.

4.2.1. Métricas de la calidad de la especificación

Davis y sus grupo de trabajo (Pressman, 2005) proponen una lista de características que pueden emplearse para valorar la calidad del modelo de análisis y la correspondiente especificación de requisitos: especificidad (ausencia de ambigüedad), compleción, corrección, comprensión, capacidad de verificación, consistencia interna y externa, capacidad de logro, concisión, trazabilidad, capacidad de modificación, exactitud y capacidad de reutilización. Aunque muchas de estas características parecen de naturaleza cualitativa, Davis sugiere que todas pueden representarse usando una o más métricas.

Supongamos que hay n_r requisitos en una especificación:

$$n_r = n_f + n_{nf}$$

Donde: n_f es el número de requisitos funcionales.

n_{nf} es el número de requisitos no funcionales.

Especificidad

Para determinar la especificidad (ausencia de ambigüedad) de los requisitos, realizamos una métrica basada en la consistencia de la interpretación de los revisores para cada requisito:

$$Q_i = n_{ui} / n_r$$

Donde: n_{ui} es el número de requisitos para los que todos los revisores tuvieron interpretaciones idénticas.

Cuanto más cerca este de 1 el valor de Q, menor será la ambigüedad de la especificación.

Para evaluar la métrica de la especificidad de los requisitos se realizaron dos revisiones por varios revisores. El objetivo principal de estas revisiones era obtener los requisitos con el mayor grado de claridad posible y sin ambigüedades.

Resultados:

$$n_r = n_f + n_{nf}$$

$$n_r = 19 + 18$$

$$n_r = 37$$

Primera Revisión

$$Q_i = n_{ui} / n_r$$

$$Q_i = 30 / 37$$

$$Q_i = 0.8$$

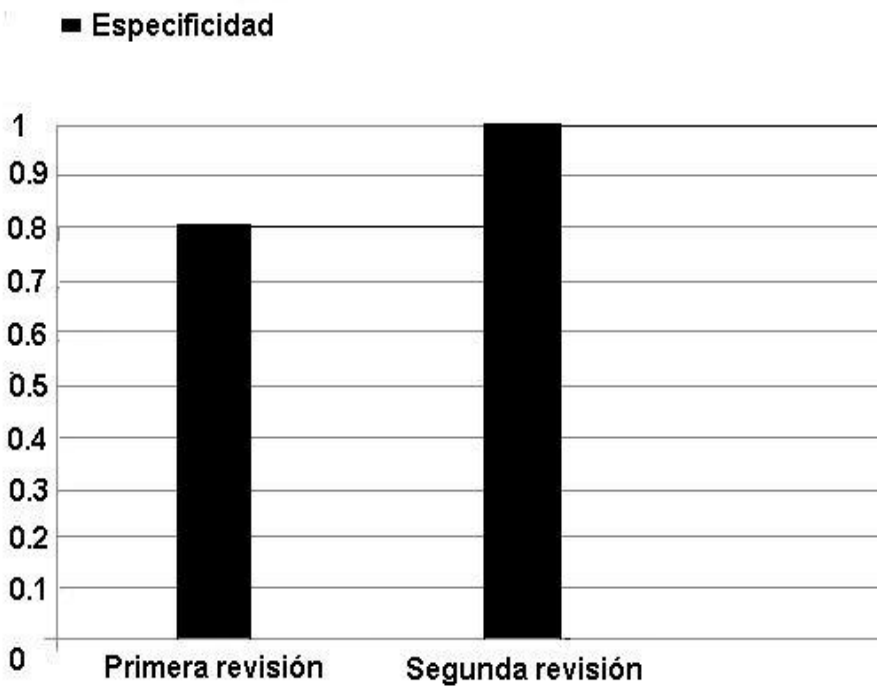
Segunda Revisión

$$Q_i = n_{ui} / n_r$$

$$Q_i = 37 / 37$$

$$Q_i = 1.0$$

Resultados de la métrica



4.2.2. Métricas de Casos de uso

Métricas NOAS /NOS

NOAS: Numero de pasos del actor.

NOS: Numero de pasos.

Esta heurística se basa en la idea de que un caso de uso sirve para expresar una interacción actor-sistema. Por ello, el número de pasos de actor y el de pasos del sistema deben estar en torno al 50%,

considerando también la posibilidad de que existan pasos de inclusión o extensión en los que se realice otro caso de uso.

Nombre del CU	NOAS	NOS	Valor de la métrica
Gestionar Profesor	9	20	47%
Proponer Evaluación	4	7	53%
Consultar Actividades	4	8	50%
Autenticarse	7	13	54%
Generar Reporte	3	7	48%

Tabla: Aplicación de la métrica NOAS/NOS al DCUS

Esta métrica fue aplicada a cada uno de los casos de uso del sistema, como se muestra en la tabla anterior, para cada uno de los casos de uso el valor de esta métrica está en el rango habitual, con valores bastante cercanos al 50%, lo que implica que los mismos están bastante completos pues no se obvian participaciones del actor o del sistema, o sea, que se incluye todo lo que debe hacer tanto el actor como el sistema, para lograr el objetivo del caso de uso; y los pasos del actor y del sistema tienen un desglose adecuado. En el caso de las acciones del actor, si realiza varias, estas están separadas por comas, lo que constituye un solo paso.

4.2.3. Métricas para el Modelo de Diseño

La serie de métricas CK

Uno de los conjuntos de métricas más ampliamente referenciados han sido los propuestos por Chidamber y Kemerer. Normalmente conocidas como la serie de métricas CK, los autores han propuesto seis métricas basadas en clases para sistemas orientados a objetos.

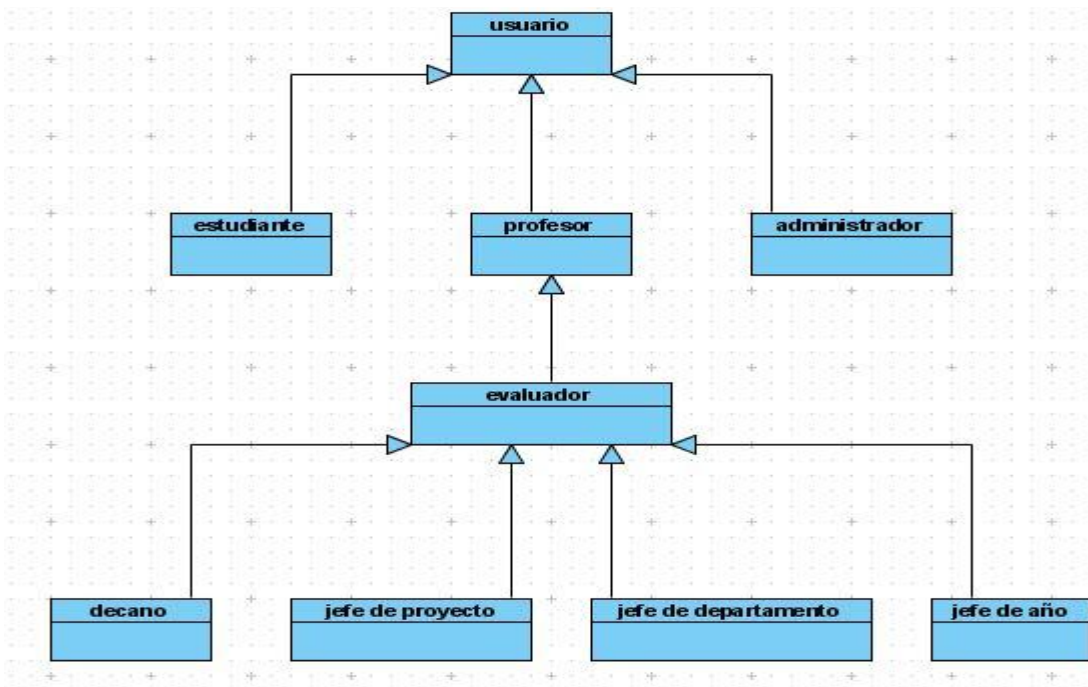
De esas seis métricas se aplicarán las siguientes:

✓ **Árbol de profundidad de herencia (APH)**

Esta métrica se define como la máxima longitud del nodo a la raíz del árbol. A medida que el APH crece es posible que clases de más bajos niveles hereden muchos métodos. Esto conlleva dificultades potenciales cuando se intenta predecir el comportamiento de una clase. Una jerarquía de clases profunda (el APH largo) también conduce a una complejidad de diseño mayor. Por el lado positivo los valores de APH grandes implican un gran número de métodos que se reutilizarán.

Por su parte, algunos autores sugieren un umbral de 6 niveles como indicador de un abuso en la herencia en distintos lenguajes de programación.

Resultado: A partir de los datos obtenidos después de aplicar la métrica APH se obtuvo que los niveles más altos de herencia son de 4, lo cual se encuentra dentro del umbral definido para determinar que el diseño no es complejo, no existe un alto acoplamiento y no es de difícil mantenimiento.



Métricas Propuestas por Lorenz y Kidd

✓ Tamaño de clase (TC)

El tamaño general de una clase puede medirse determinando las siguientes medidas:

- Total de operaciones (tanto heredadas como privadas de la instancia), que se encapsulan dentro de la clase.
- El número de atributos (atributos tanto heredados como privados de la instancia), encapsulados por la clase.

Un TC grande afecta los indicadores de calidad definidos para esta métrica por los especialistas:

Reutilización: reduce la reutilización de la clase.

Implementación: complica la implementación.

Responsabilidad: la clase debe tener bastante responsabilidad.

Las medidas o umbrales para los parámetros de calidad han sido una polémica a nivel mundial en el diseño de sistemas. Algunos especialistas plantean umbrales para estas métricas según se muestra en la tabla 4.2.3, estos fueron los aplicados en el diseño de este sistema.

No de Operaciones y/o Atributos	
TC	Umbral
Pequeño	≤ 20
Medio	> 20 y ≤ 30
Grande	> 30

Tabla 4.2.3 Umbrales para TC

Resultado: Esta métrica fue aplicada al diagrama de clases persistente donde hay un total de 12 clases, todas las clases se pueden considerar de tamaño pequeño. Estos valores demuestran que los indicadores de calidad reutilización, implementación y responsabilidad no se ven afectados.

4.5 Conclusiones Parciales.

En este Capítulo se realizó la validación que requiere la realización del sistema, se demostró la eficiencia de uno de los métodos para capturar la funcionalidad de un sistema, las métricas para determinar la calidad de la especificación de los requisitos, del diseño de casos de uso y del diseño.

CONCLUSIONES GENERALES

Al concluir el presente trabajo se arriba a las siguientes conclusiones:

- Se realizó un buen estudio de arte que permitió conocer a sistemas similares y así poder sacar funcionalidades y características nuevas al sistema.
- El análisis de los procesos de negocio ya elaborados, junto a la interacción con los clientes del sistema y aplicando técnicas para la **elicitación** de requerimientos, propició que se obtuvieran resultados satisfactorios en la identificación de los requerimientos que debe cumplir el sistema.
- La especificación de los artefactos del modelo del sistema facilitó un mayor entendimiento y un acuerdo común entre los clientes y los desarrolladores, con respecto a las funcionalidades que el sistema debe brindar.
- Se ha realizado el análisis y diseño de una aplicación Web que permitirá darle flexibilidad, rapidez y eficiencia a dicho proceso.
- El análisis de los artefactos obtenidos en el modelo del sistema y el de diseño permitió medir el grado de factibilidad de estos en función de la calidad, funcionalidad, ambigüedad, reutilización y consistencia.

RECOMENDACIONES

Con el objetivo de disponer de una herramienta que abarque de manera automatizada todos los procesos de la evaluación de desempeño en la facultad #3 se recomienda:

- Trabajar en la continuación de las siguientes fases, la implementación del sistema y las pruebas para la obtención del producto final.
- Continuar con la investigación para aumentar las funcionalidades del sistema.
- Realizar el análisis y diseño de los casos de uso del sistema que no se tuvieron en cuenta para el desarrollo de esta iteración, considerando la importancia que tienen para la utilización del sistema en la facultad.

REFERENCIAS BIBLIOGRAFICAS.

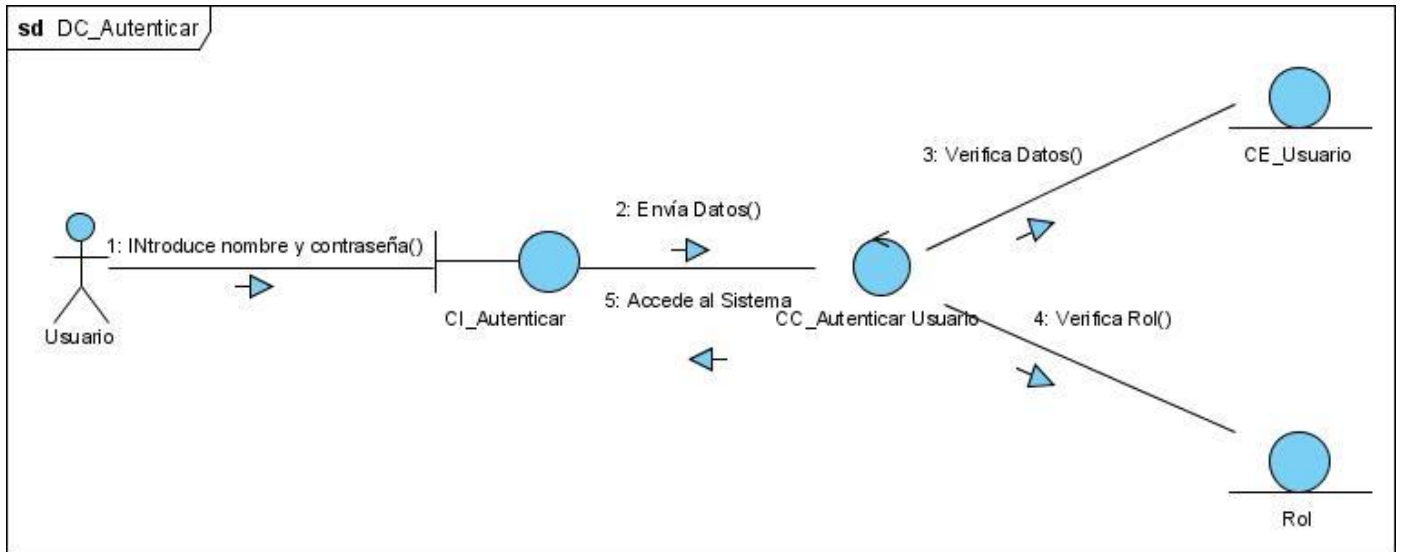
1. PostgreSQL Práctico. [En línea] [Citado el: 17 de Enero de 2009.] Disponible en: <http://www.sobl.org/traduccion/practical-postgres/node19.html>.
2. Metodologías de Desarrollo de Software. [En línea] [Citado el: 10 de Febrero de 2009.] Disponible en <http://www.scribd.com/doc/2050925/metodologias-de-desarrollo-software>.
3. Metodologías De Desarrollo De Software. [En línea] [Citado el: 15 de Enero de 2009.] Disponible en: http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html.
4. Metodologías RUP y XP - [PROCESOS DE DESARROLLO]. [En línea] [Citado el: 13 de Enero de 2009.] Disponible en: <http://jackopc.blogspot.com/2007/05/metodologias-rup-y-xp-procesos-de.html>.
5. Metodología OMT. [En línea] [Citado el: 15 de Febrero de 2009.] Disponible en <http://www.monografias.com/trabajos13/metomt/metomt.shtml>.
6. Wikipedia Proceso Unificado de Rational. [En línea] [Citado el: 6 de Febrero de 2009.] Disponible en http://es.wikipedia.org/wiki/Proceso_Unificado_de_Rational#Demostrar_valor_.
7. Guía Breve de CSS. [En línea] [Citado el: 3 de Febrero de 2009.] Disponible en: <http://www.w3c.es/divulgacion/guiasbreves/HojasEstilo>.
8. Lenguaje PHP. [En línea] [Citado el: 16 de Enero de 2009.] Disponible en: <http://www.virtual.unal.edu.co/cursos/sedes/manizales/4060029/lecciones/cap11-2.html>.
9. Programación en PHP. [En línea] [Citado el: 10 de Febrero de 2009.] Disponible en: http://es.wikibooks.org/wiki/Programaci%C3%B3n_en_PHP.
10. Java Script. [En línea] [Citado el: 26 de Enero de 2009.] Disponible en: <http://www.webestilo.com/javascript/>.
11. Curso de Macromedia Dreamweaver. [En línea] [Citado el: 26 de Enero de 2009.] Disponible en: <http://www.aulafacil.com/AulaDream/Dream/temario.htm>.
12. PDT: Eclipse + PHP. [En línea] [Citado el: 13 de Enero de 2009.] Disponible en: <http://www.desarrolloweb.com/articulos/pdt-eclipse-php.html>.
13. Symfony, la guía definitiva. [En línea] [Citado el: 6 de Febrero de 2009.] Disponible en: <http://www.librosweb.es/symfony/index.html>.

14. HTML: Conceptos básicos. [En línea] [Citado el 8 de febrero de 2009.] Disponible en :
<http://www.webestilo.com>.
15. Eclipse: [En línea] [Citado el 15 de febrero de 2009.] Disponible en:
<http://www.eclipse.org/>
16. Visual Parading: [En línea] [Citado el 20 de febrero de 2009.] Disponible en:
[http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_\(M%C3%8D\)_14720_p/](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(M%C3%8D)_14720_p/)
17. Zend Studio: [En línea] [Citado el 25 de febrero de 2009.] Disponible en:
<http://www.desarrolloweb.com/articulos/1178.php>
18. MySQL: [En línea] [Citado el 17 de febrero de 2009.] Disponible en:
<http://dev.mysql.com/doc/refman/5.0/es/index.html>

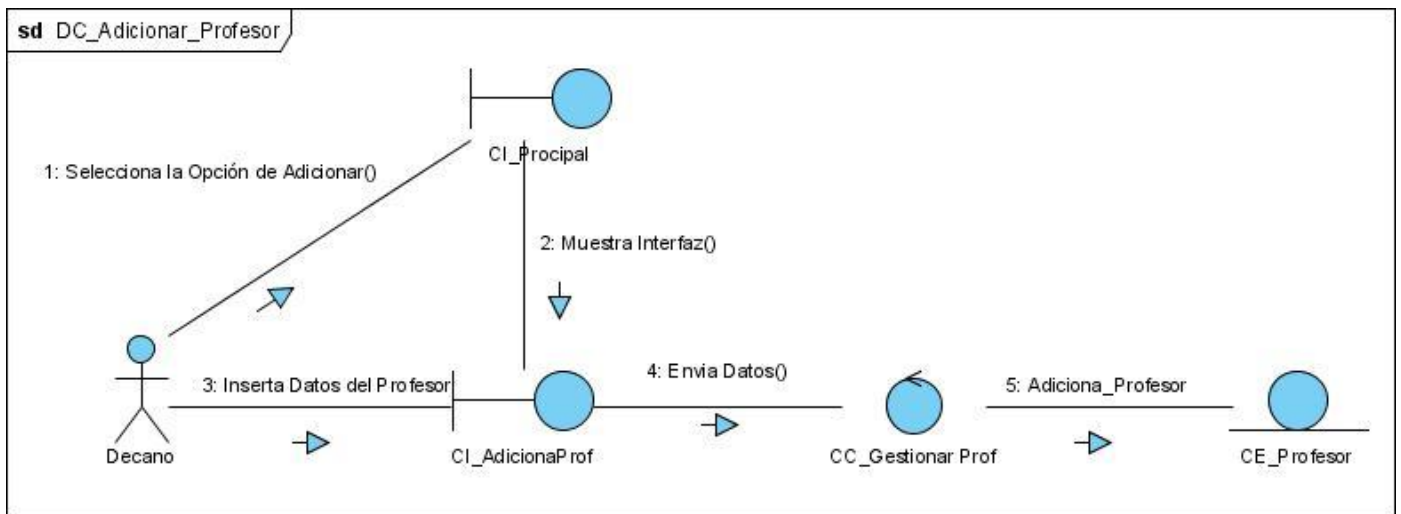
ANEXOS

Anexo 1: Diagramas de Colaboración

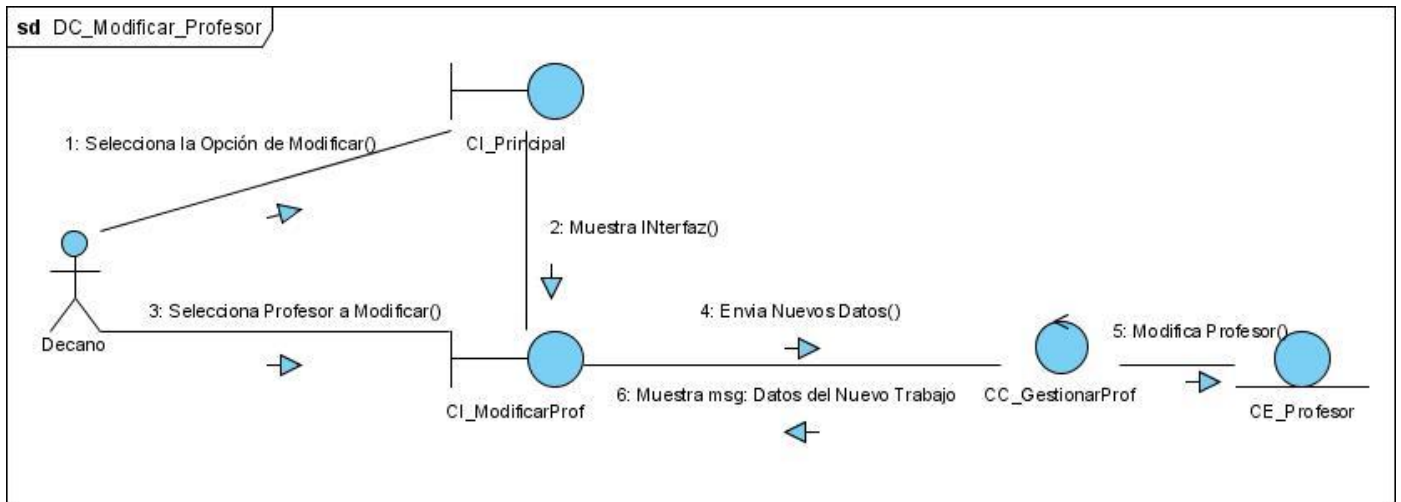
Diagramas de Colaboración: Autenticar.



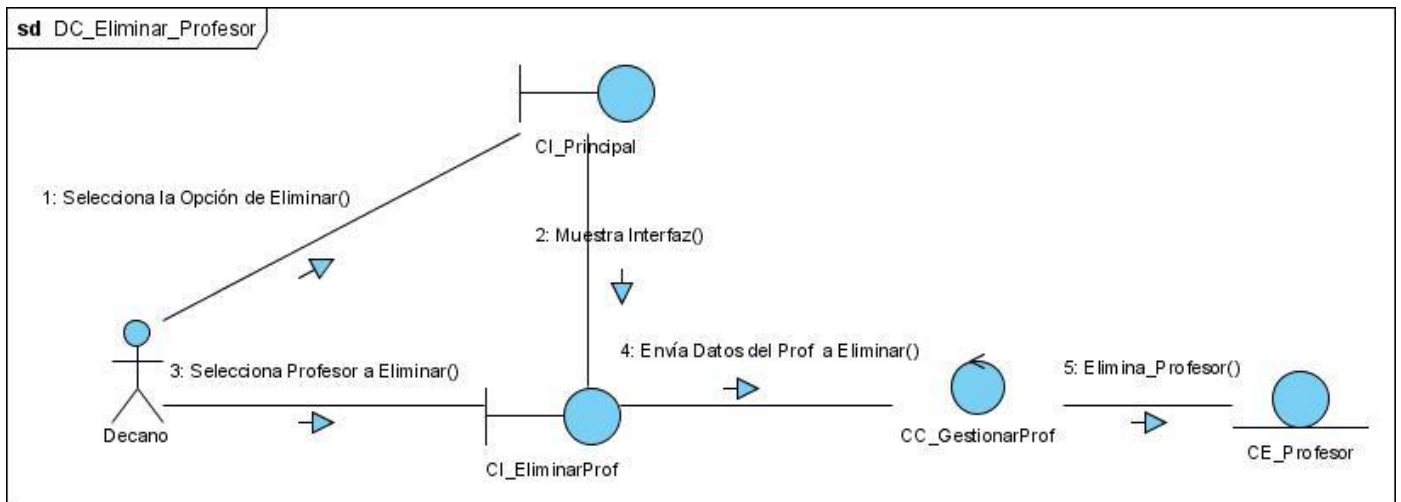
Diagramas de Colaboración: Gestionar Profesor (Adicionar).



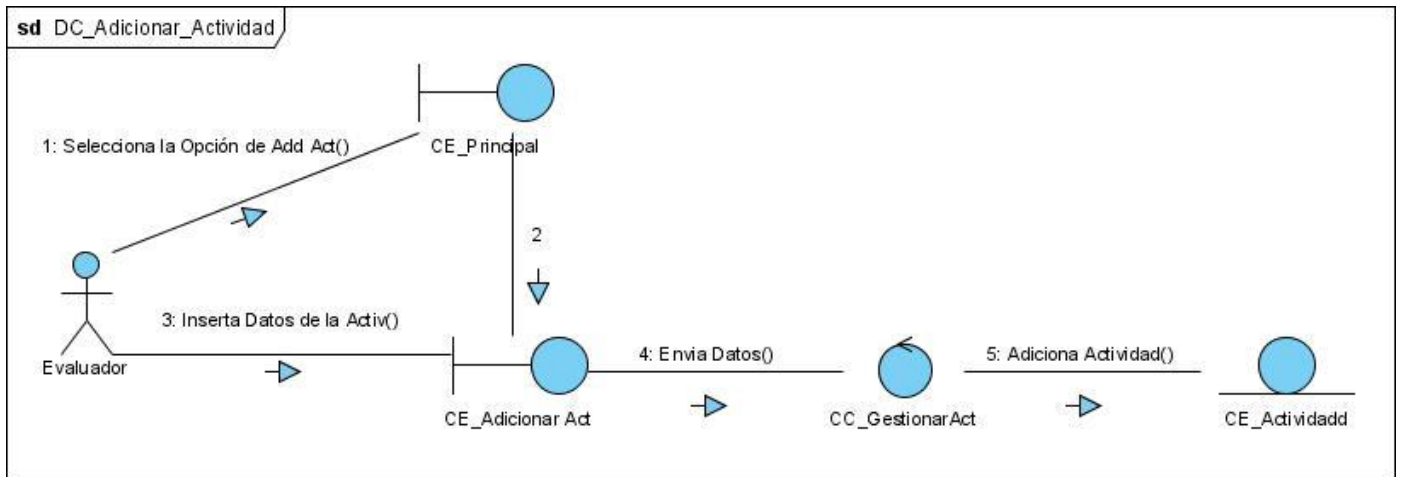
Diagramas de Colaboración: Gestionar Profesor (Modificar).



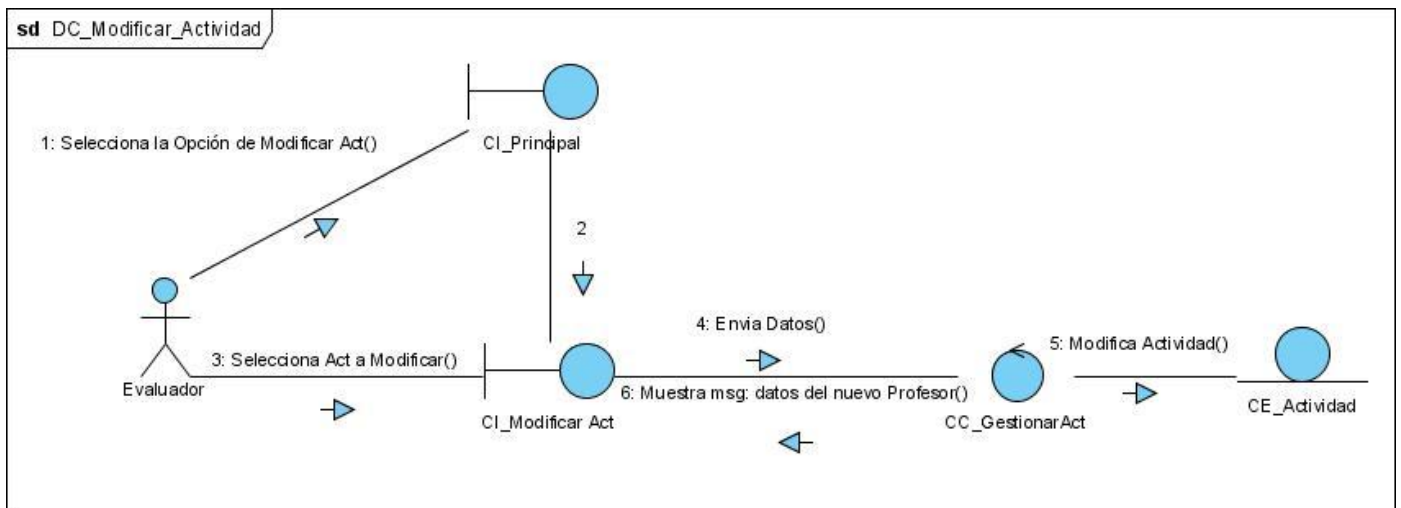
Diagramas de Colaboración: Gestionar Profesor (Eliminar).



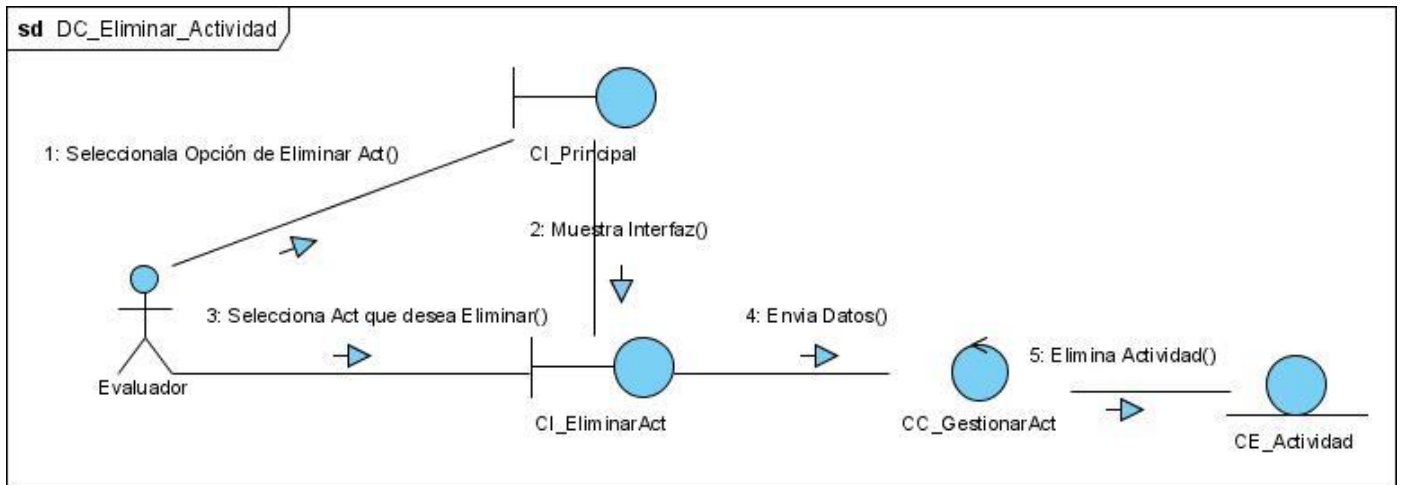
Diagramas de Colaboración: Gestionar Actividad (Adicionar).



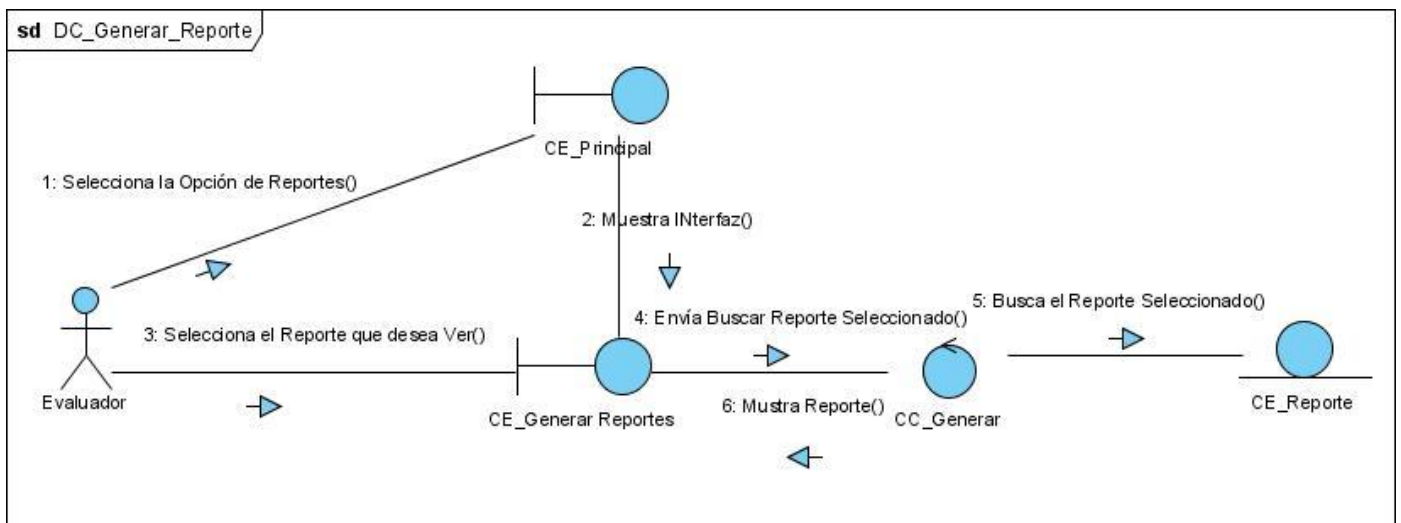
Diagramas de Colaboración: Gestionar Actividad (Modificar).



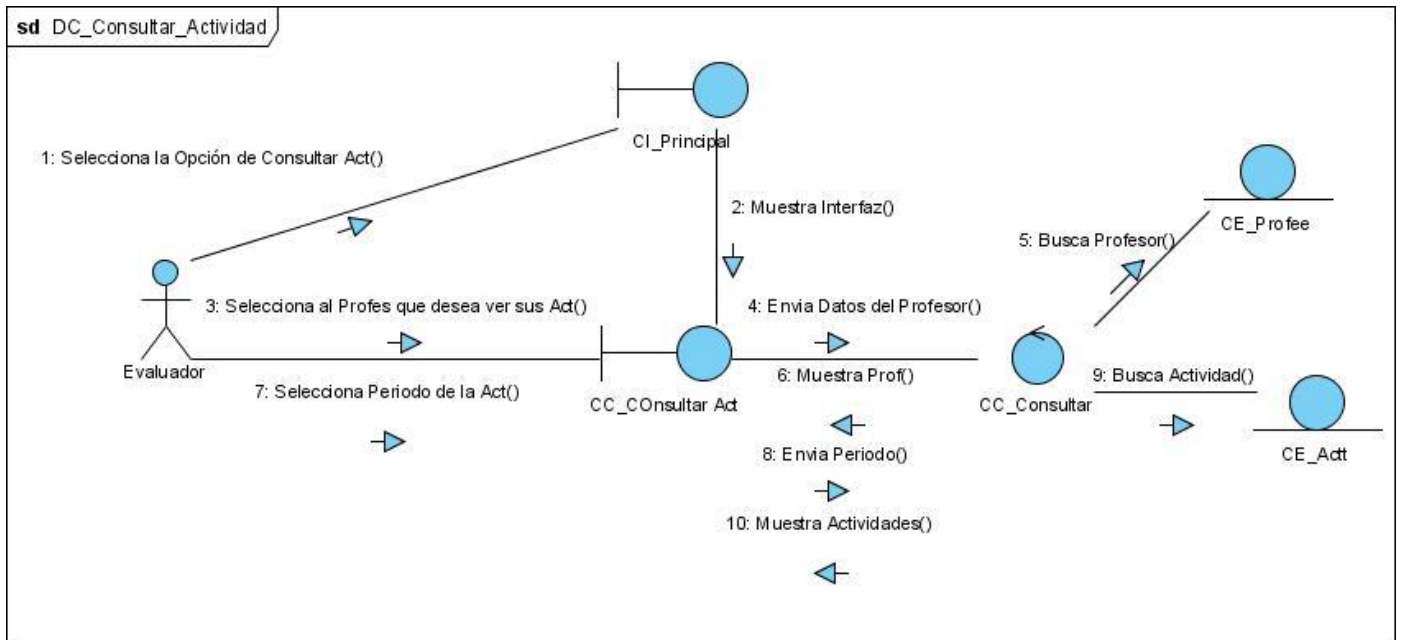
Diagramas de Colaboración: Gestionar Actividad (Eliminar).



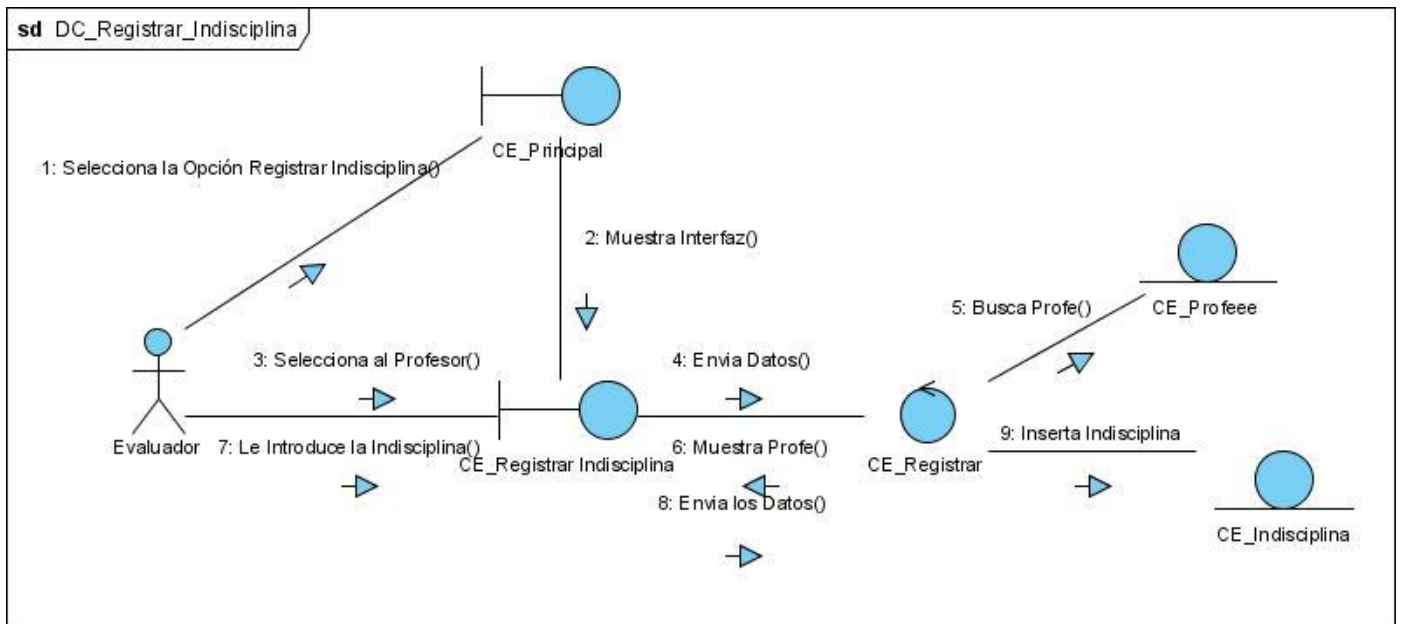
Diagramas de Colaboración: Generar Reporte.



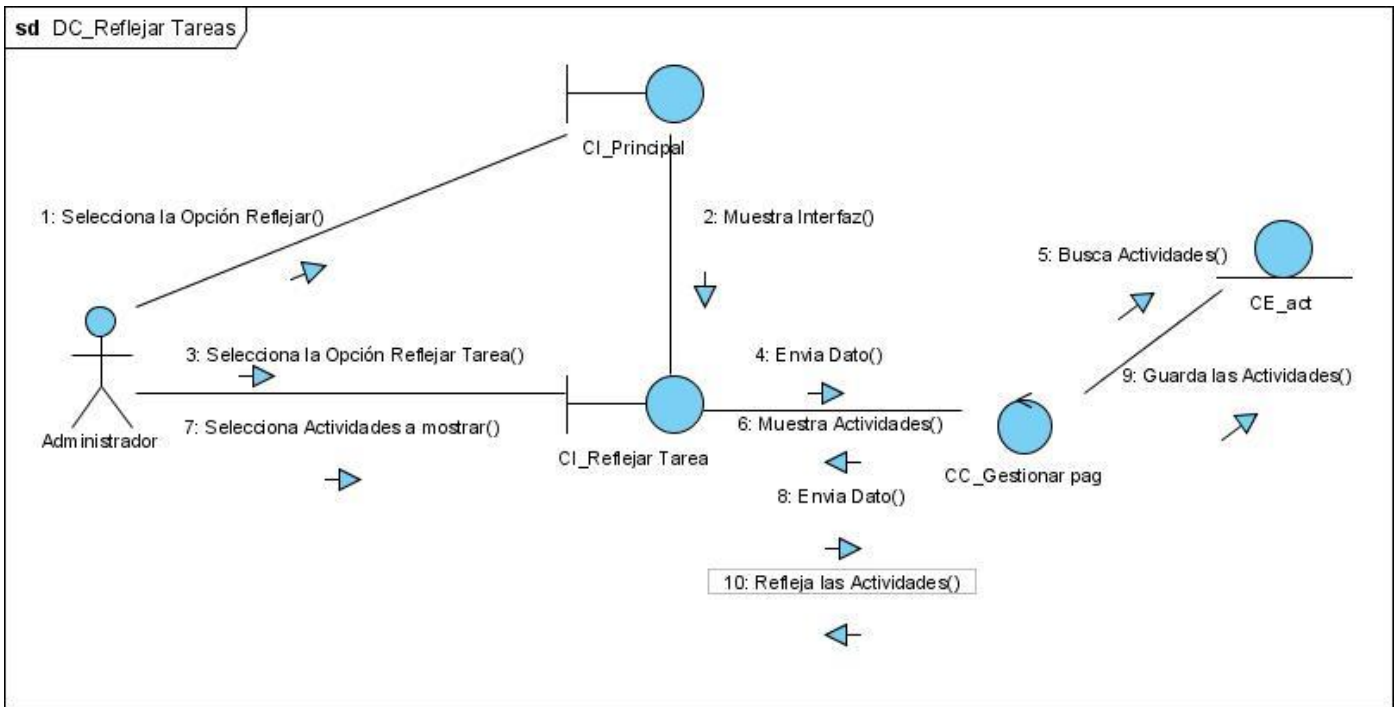
Diagramas de Colaboración: Consultar Actividad.



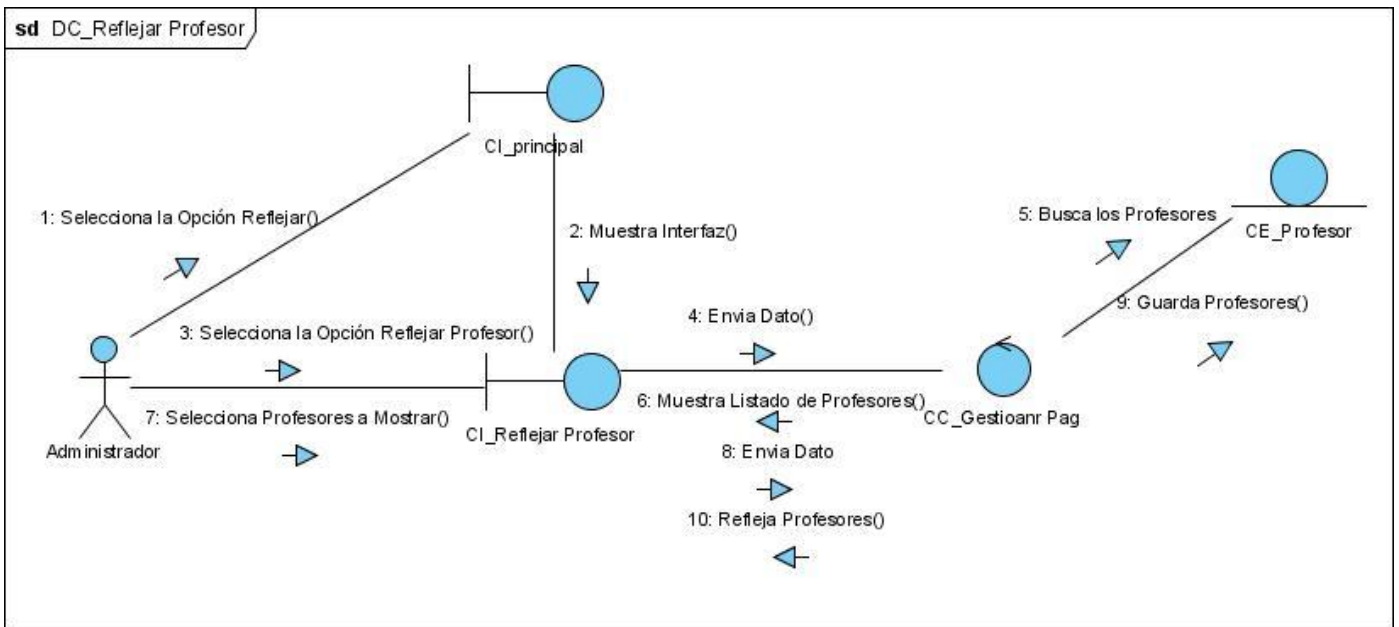
Diagramas de Colaboración: Registrar Indisciplina.



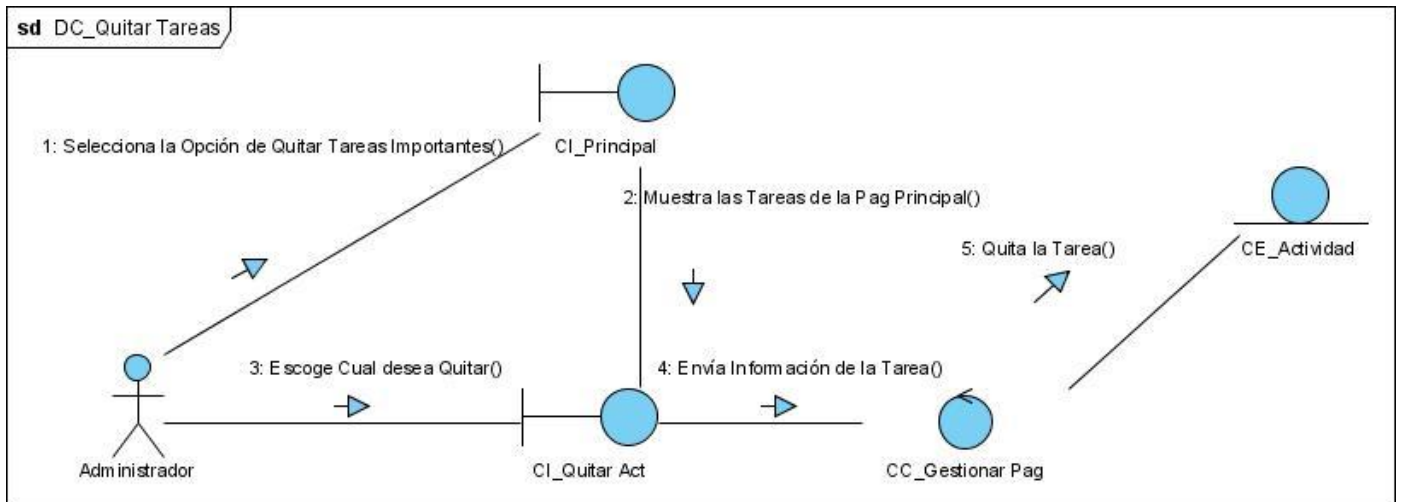
Diagramas de Colaboración: Gestionar Página Principal (Reflejar Tareas)



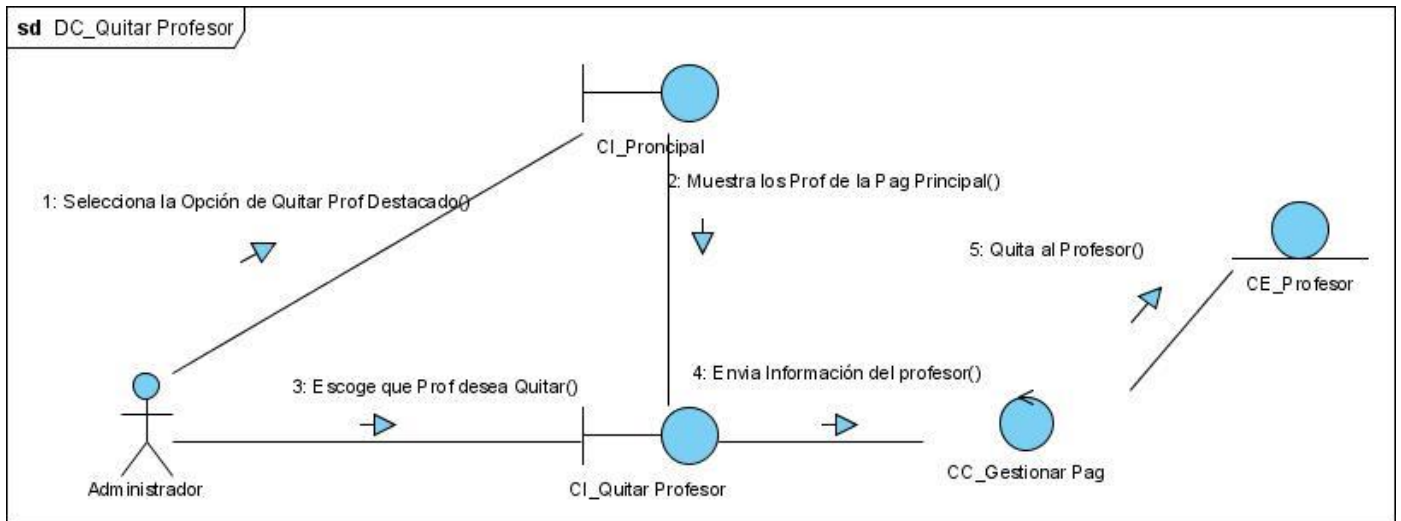
Diagramas de Colaboración: Gestionar Página Principal (Reflejar Profesor)



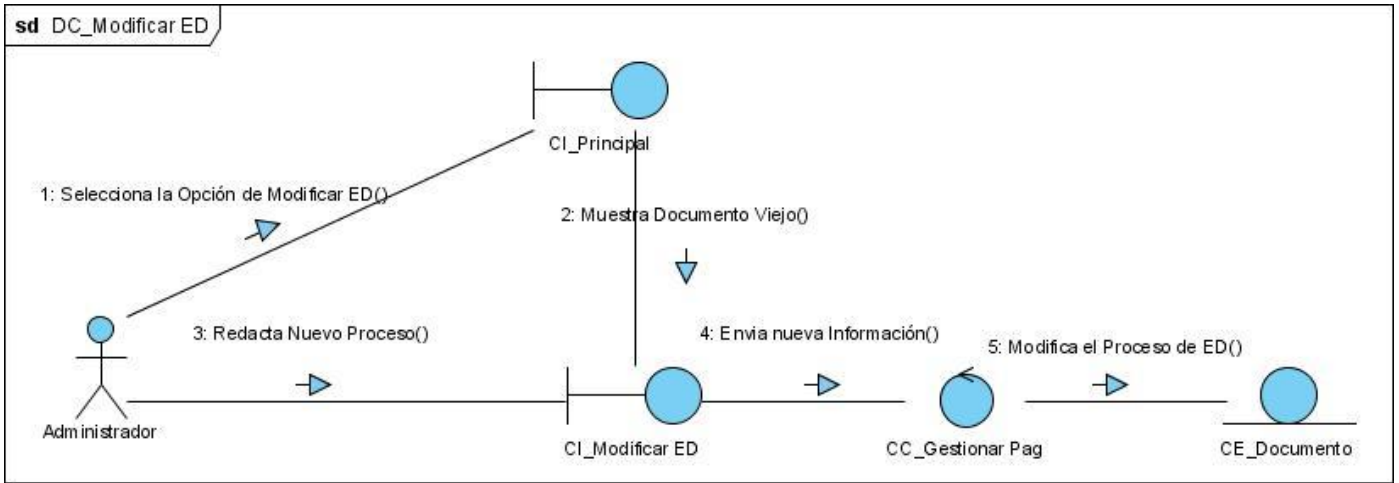
Diagramas de Colaboración: Gestionar Página Principal (Quitar Tareas)



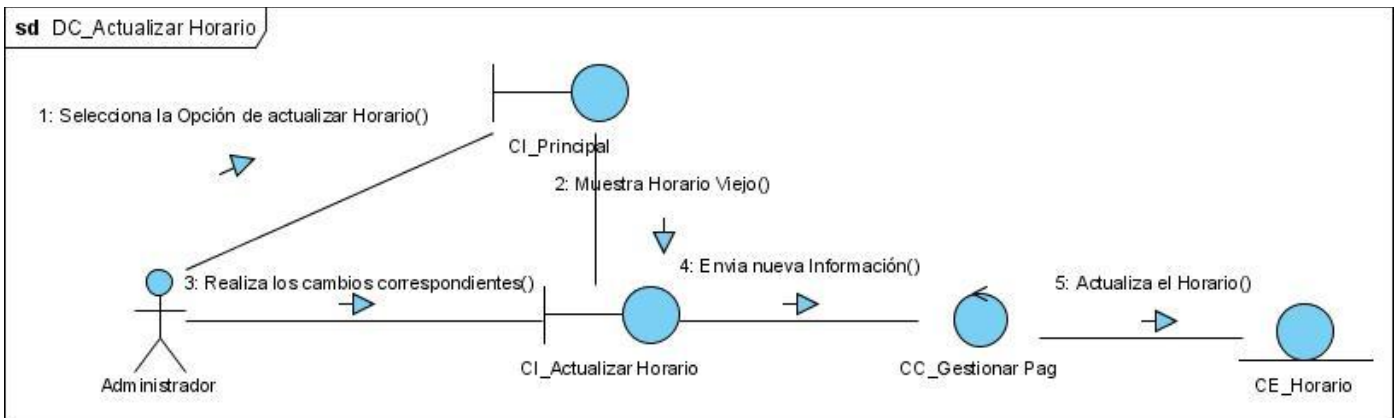
Diagramas de Colaboración: Gestionar Página Principal (Quitar Profesor).



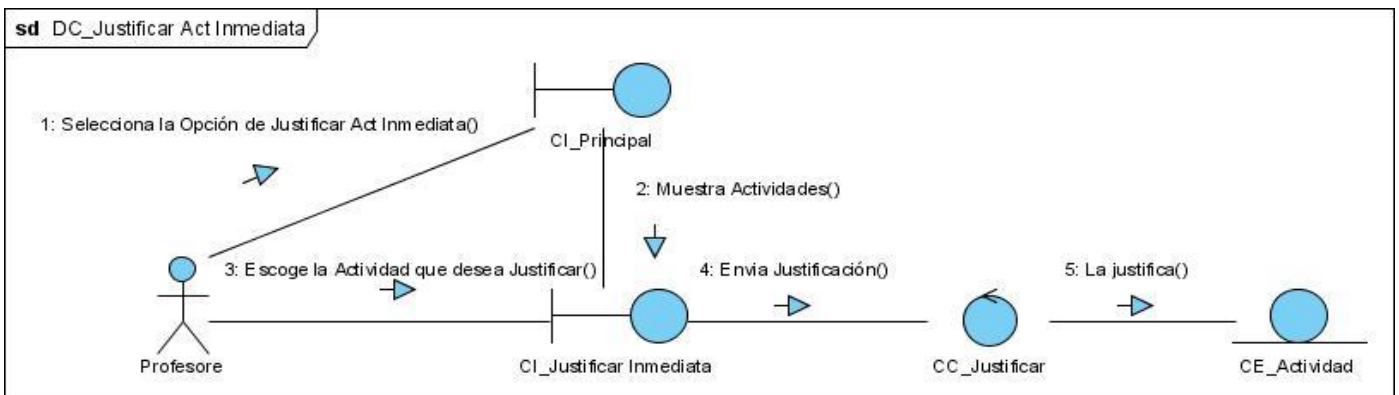
Diagramas de Colaboración: Gestionar Página Principal (Modificar ED)



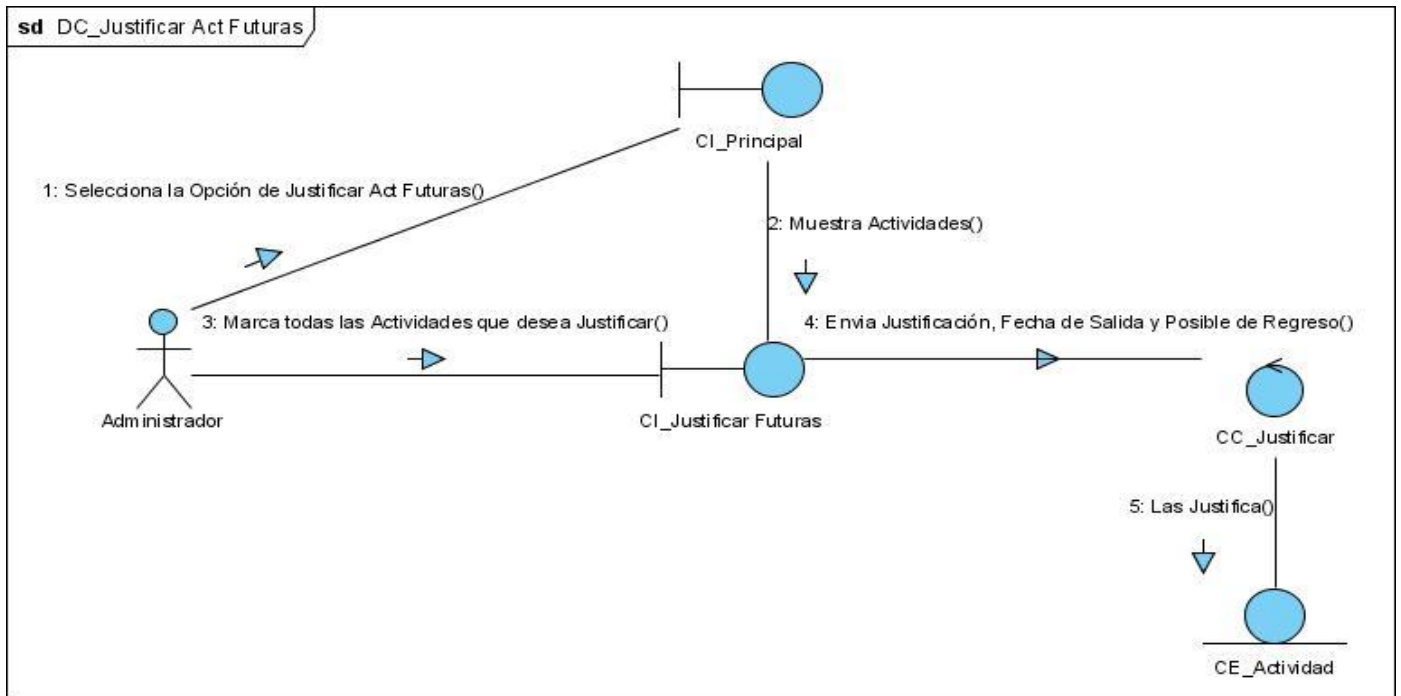
Diagramas de Colaboración: Gestionar Página Principal (Actualizar Horario)



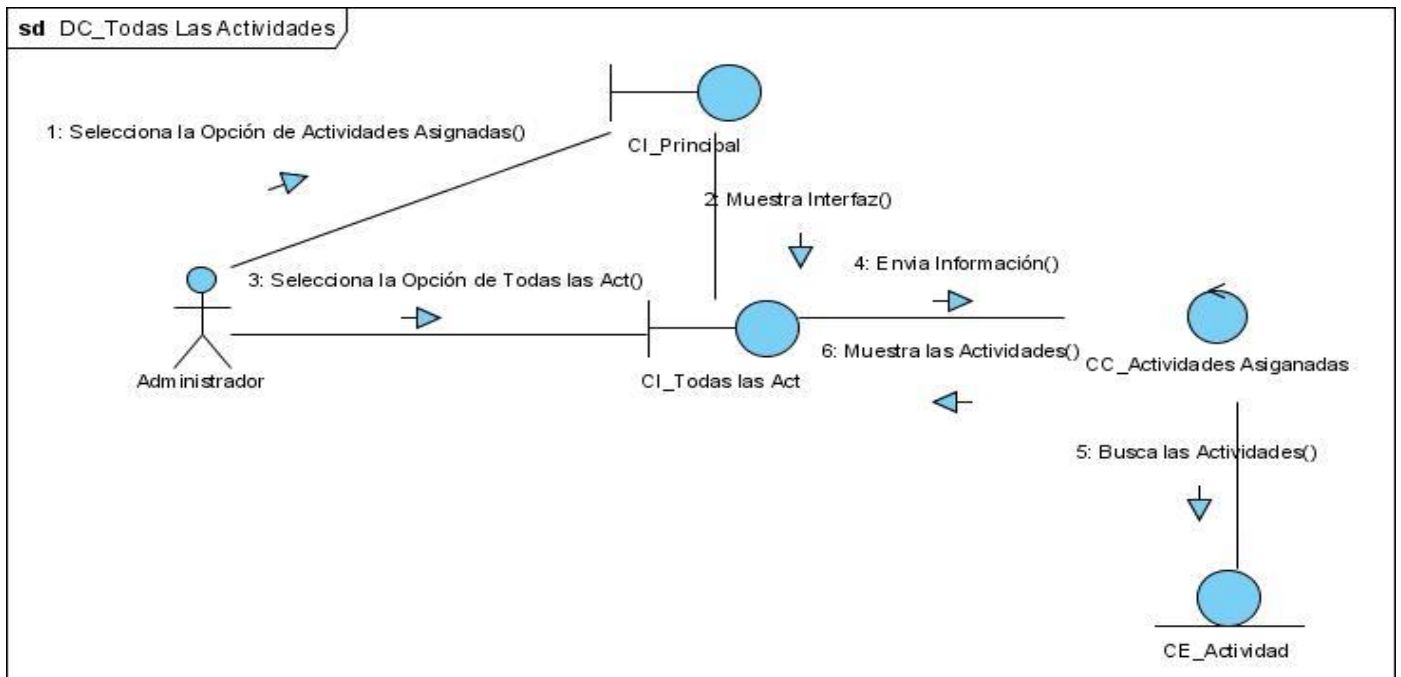
Diagramas de Colaboración: Justificar (Actividad Inmediata)



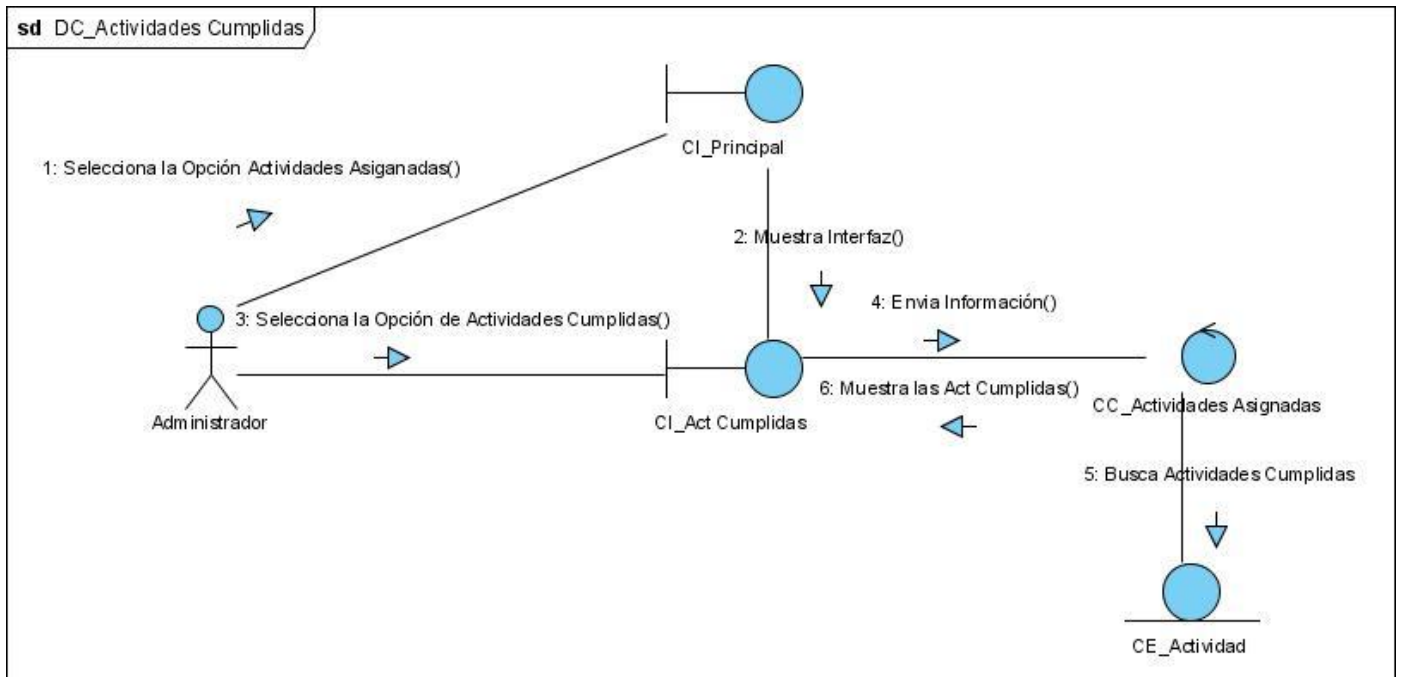
Diagramas de Colaboración: Justificar (Actividades Futuras).



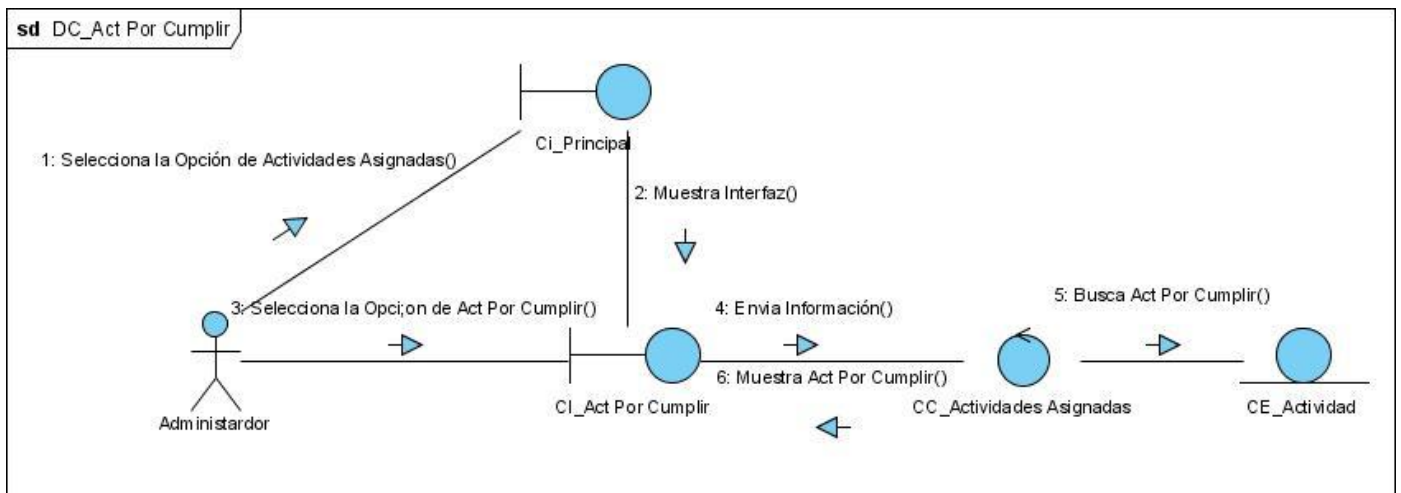
Diagramas de Colaboración: Actividades Asignadas (Todas las Actividades).



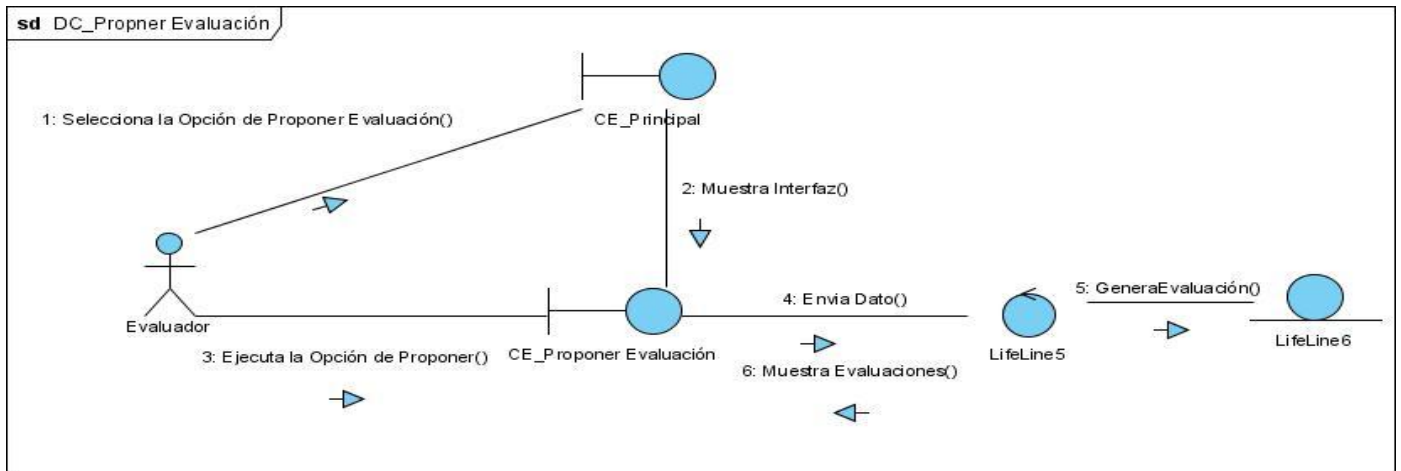
Diagramas de Colaboración: Actividades Asignadas (Actividades Cumplidas).



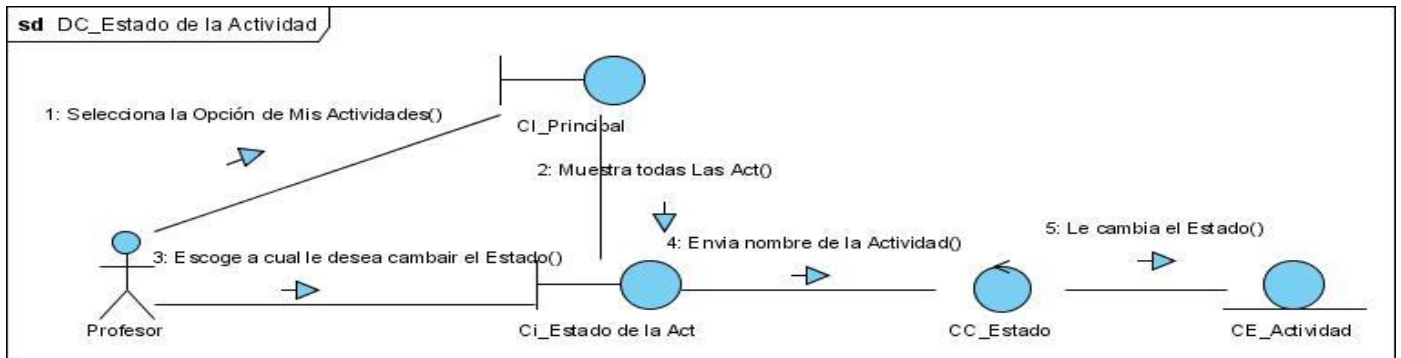
Diagramas de Colaboración: Actividades Asignadas (Actividades Por Cumplir).



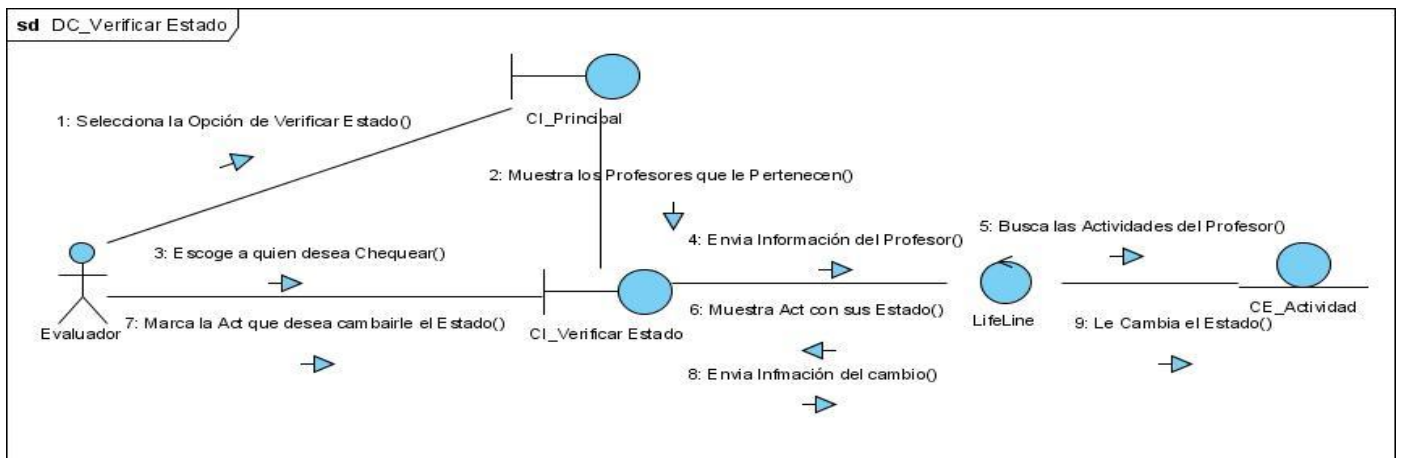
Diagramas de Colaboración: Proponer Evaluación.



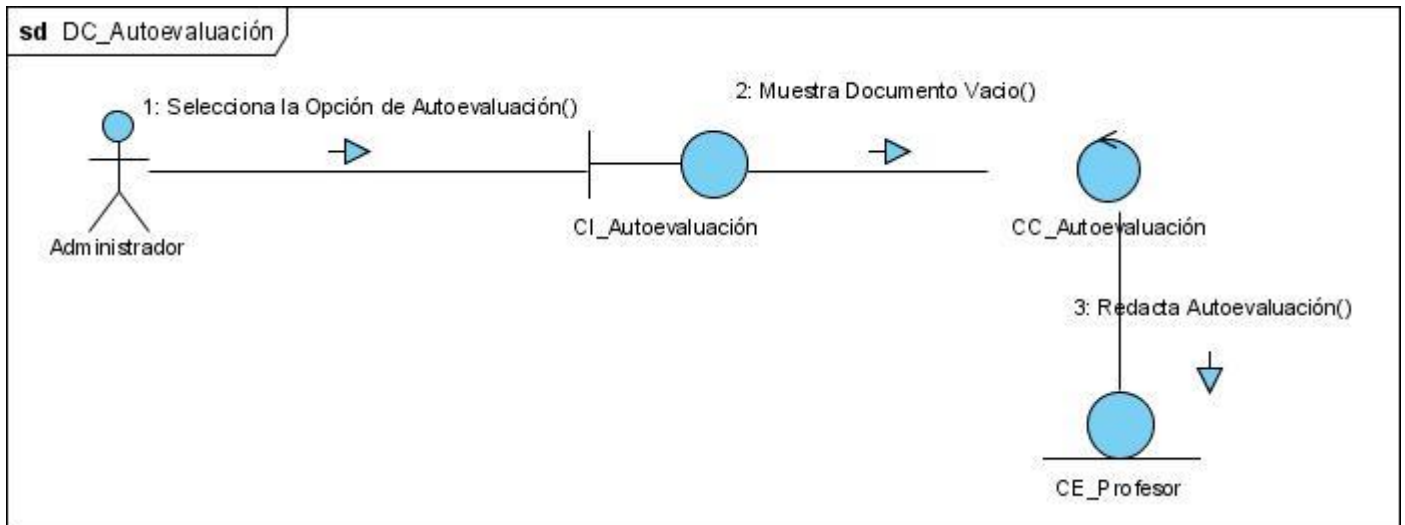
Diagramas de Colaboración: Estado de la Actividad.



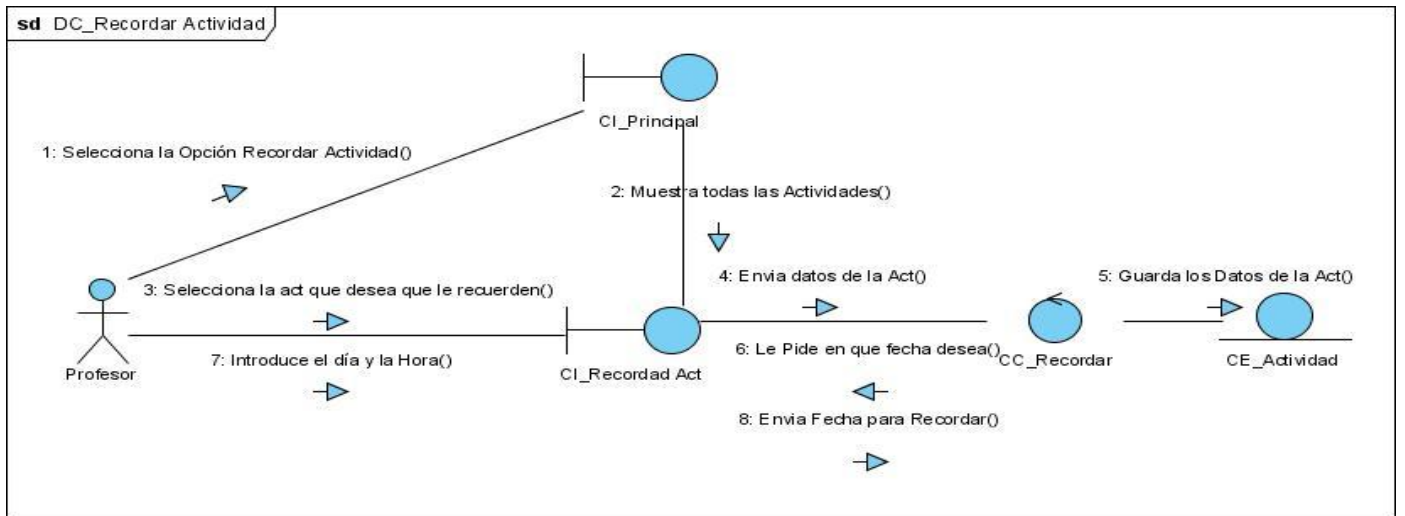
Diagramas de Colaboración: Verificar Actividad.



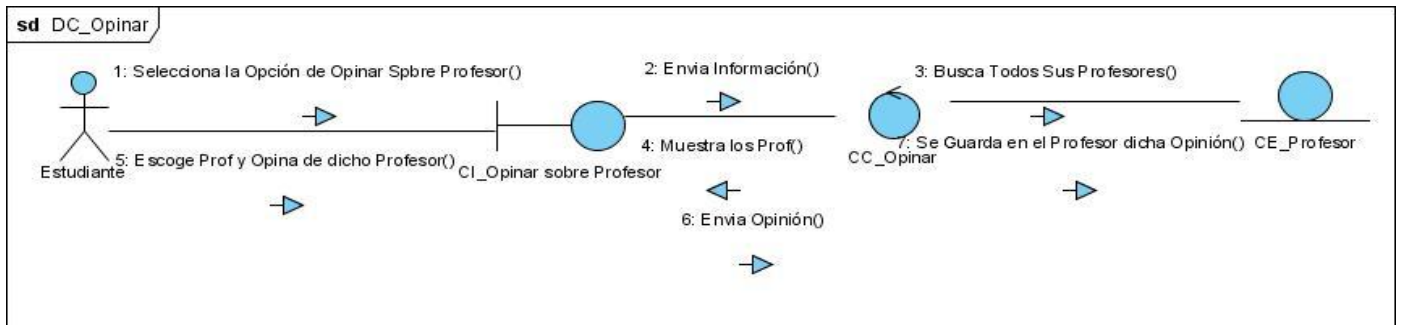
Diagramas de Colaboración: Autoevaluación.



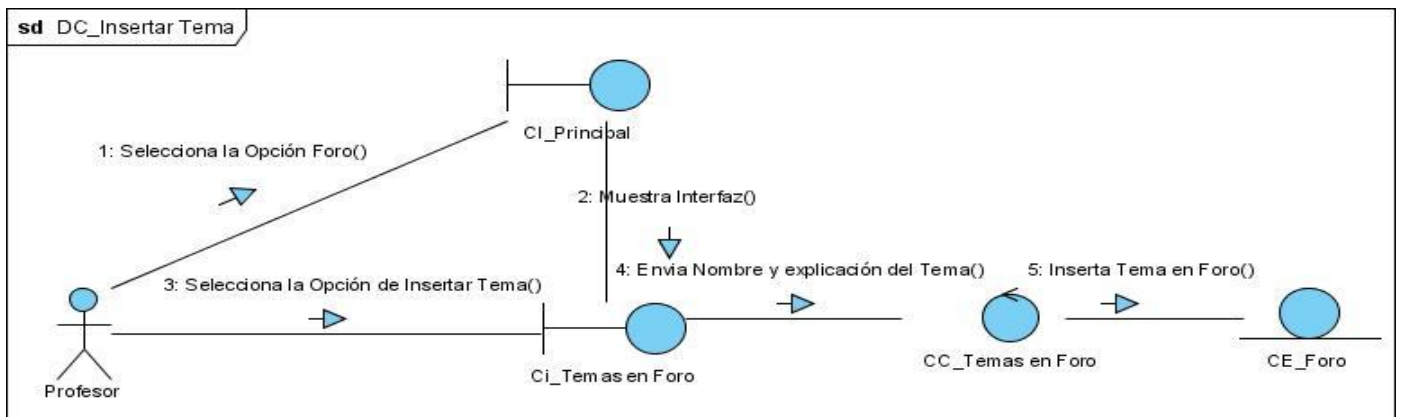
Diagramas de Colaboración: Recordar Actividad.



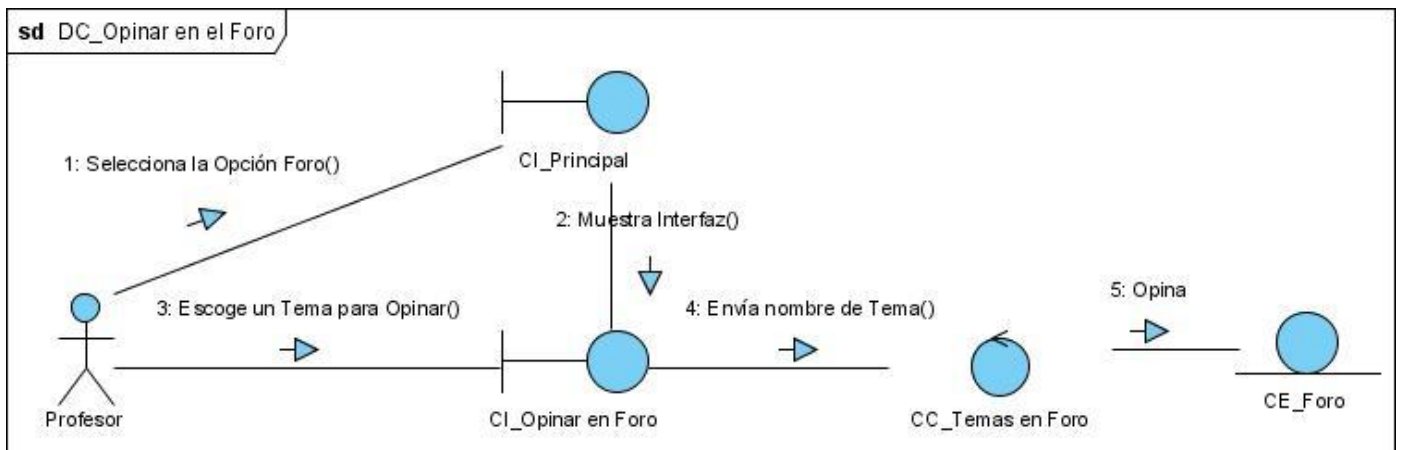
Diagramas de Colaboración: Opinar.



Diagramas de Colaboración: Temas en Foro (Insertar Tema).

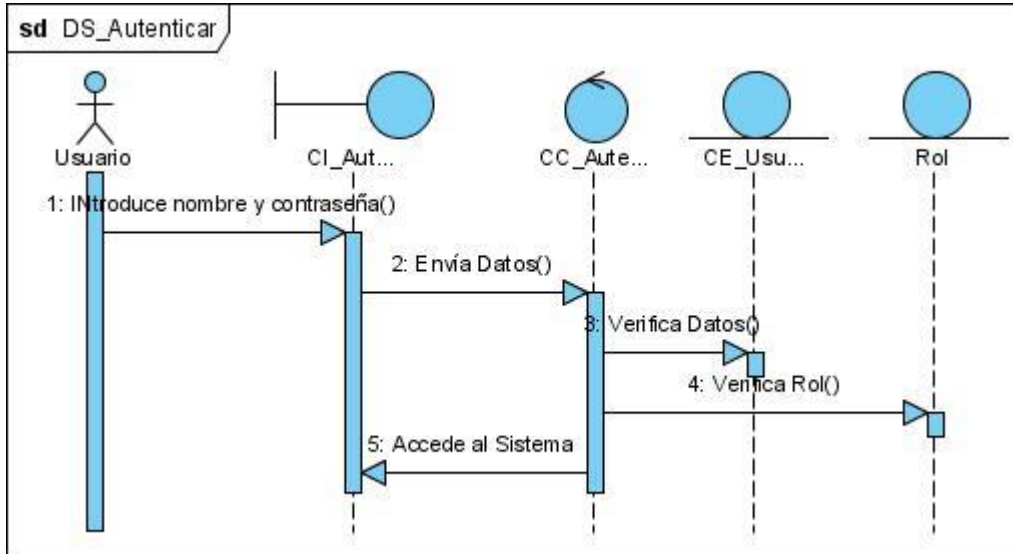


Diagramas de Colaboración: Temas en Foro (Opinar en el Foro).

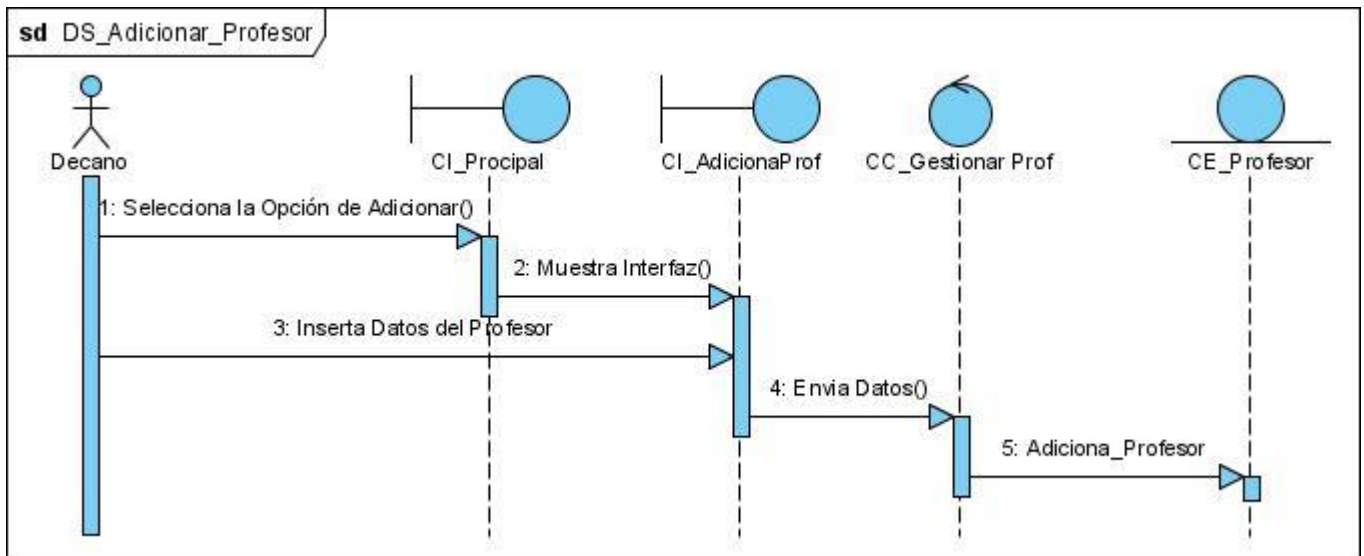


Anexo 2: Diagramas de Secuencia.

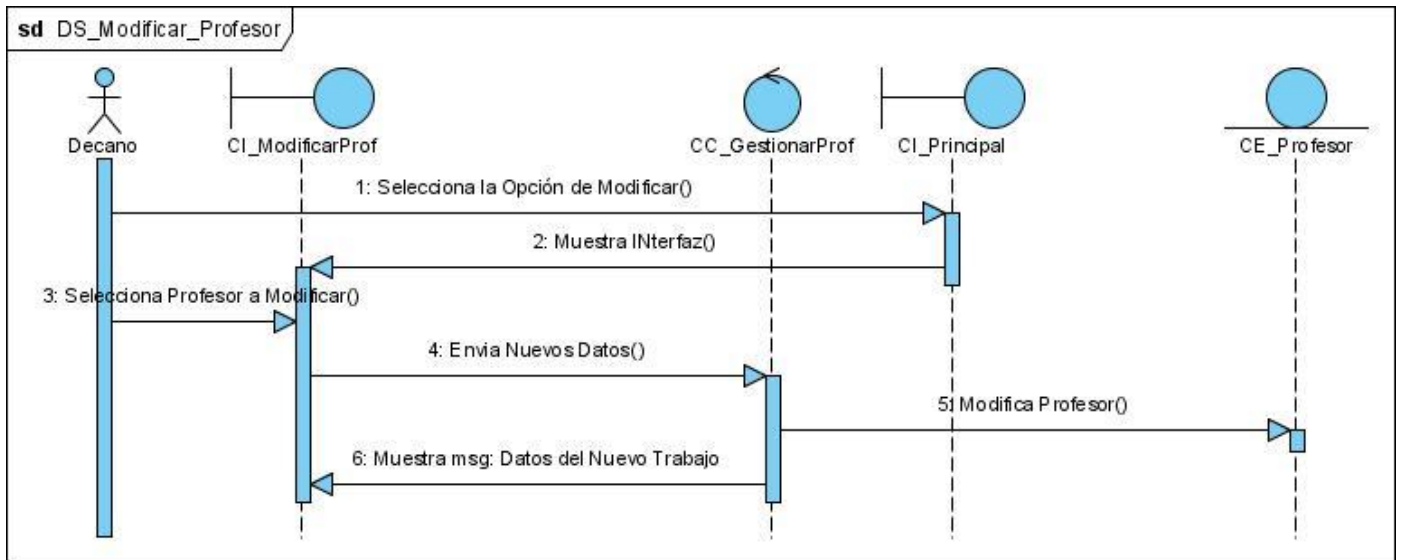
Diagramas de Secuencia: Autenticar.



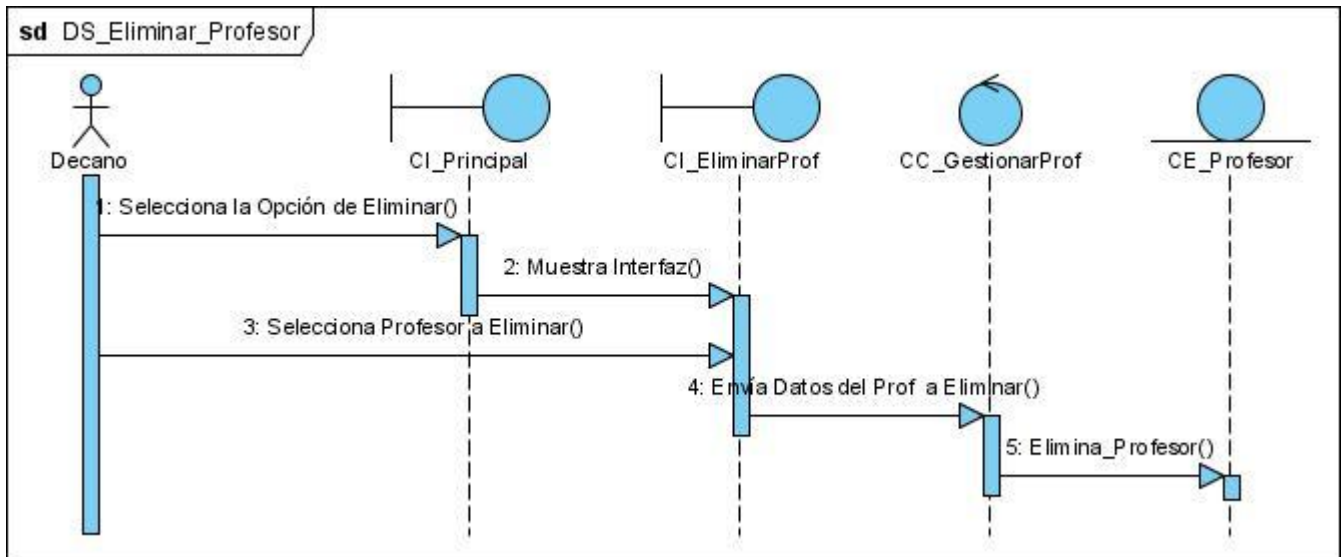
Diagramas de Secuencia: Gestionar Profesor (Adicionar).



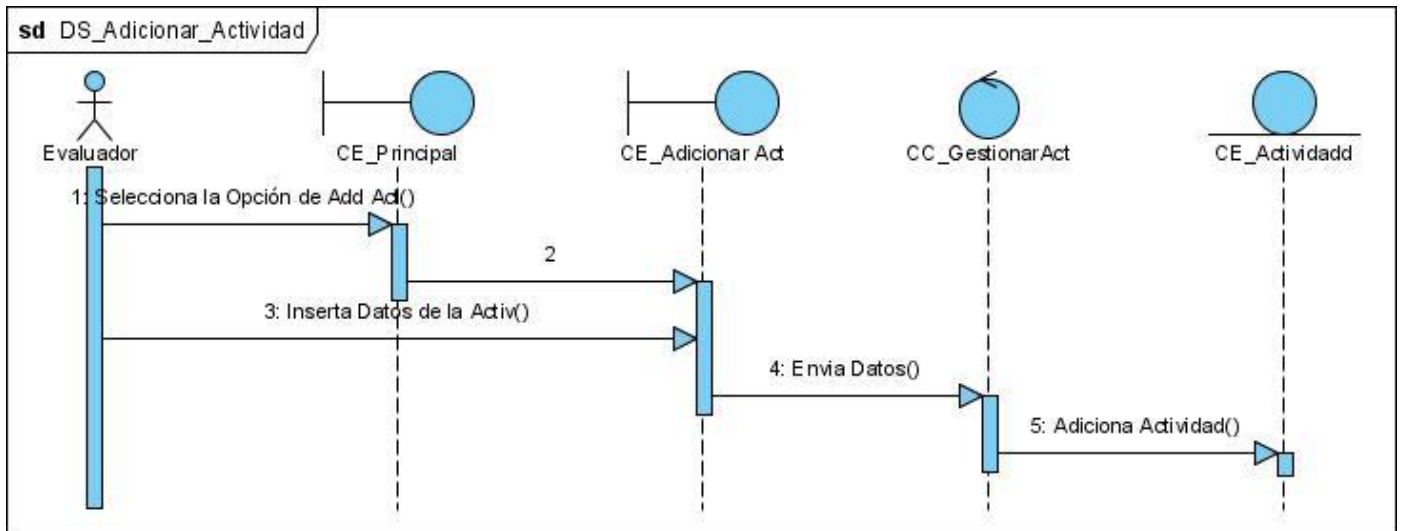
Diagramas de Secuencia: Gestionar Profesor (Modificar).



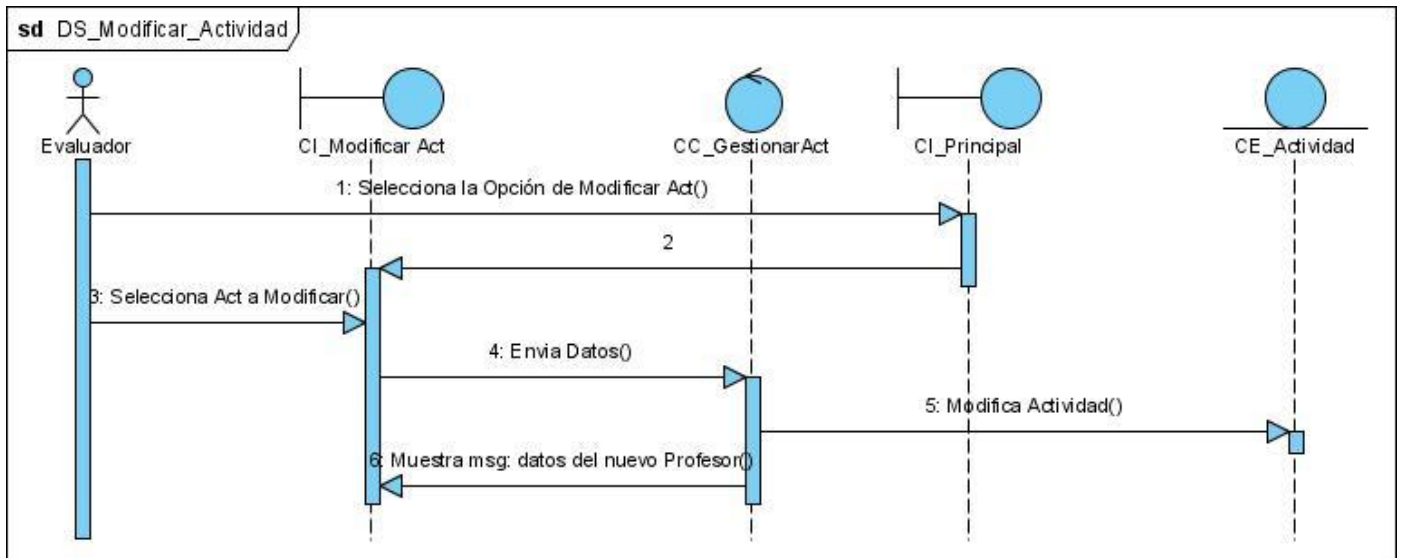
Diagramas de Secuencia: Gestionar Profesor (Eliminar).



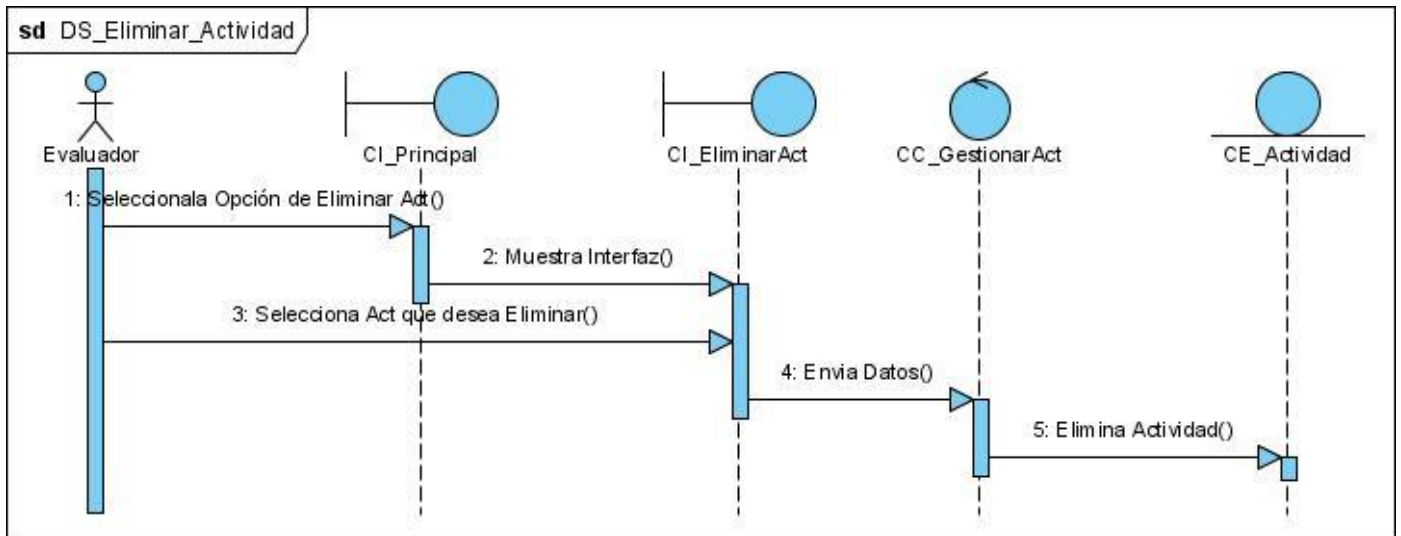
Diagramas de Secuencia: Gestionar Actividad (Adicionar).



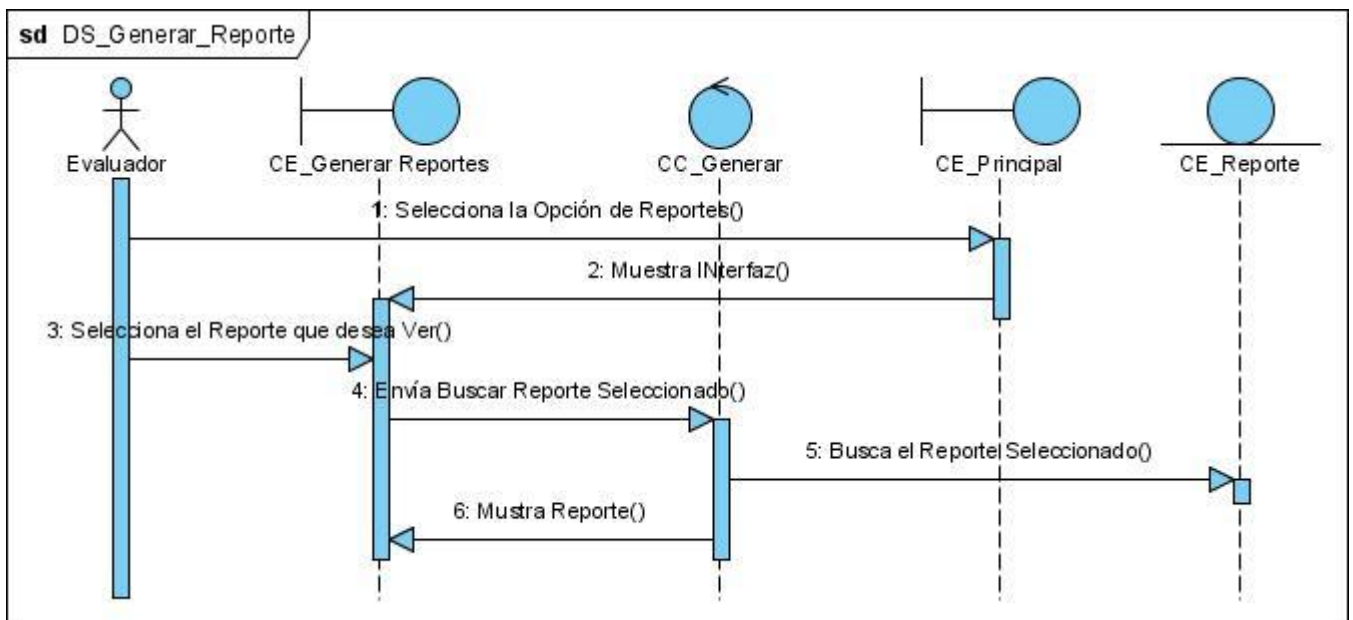
Diagramas de Secuencia: Gestionar Actividad (Modificar).



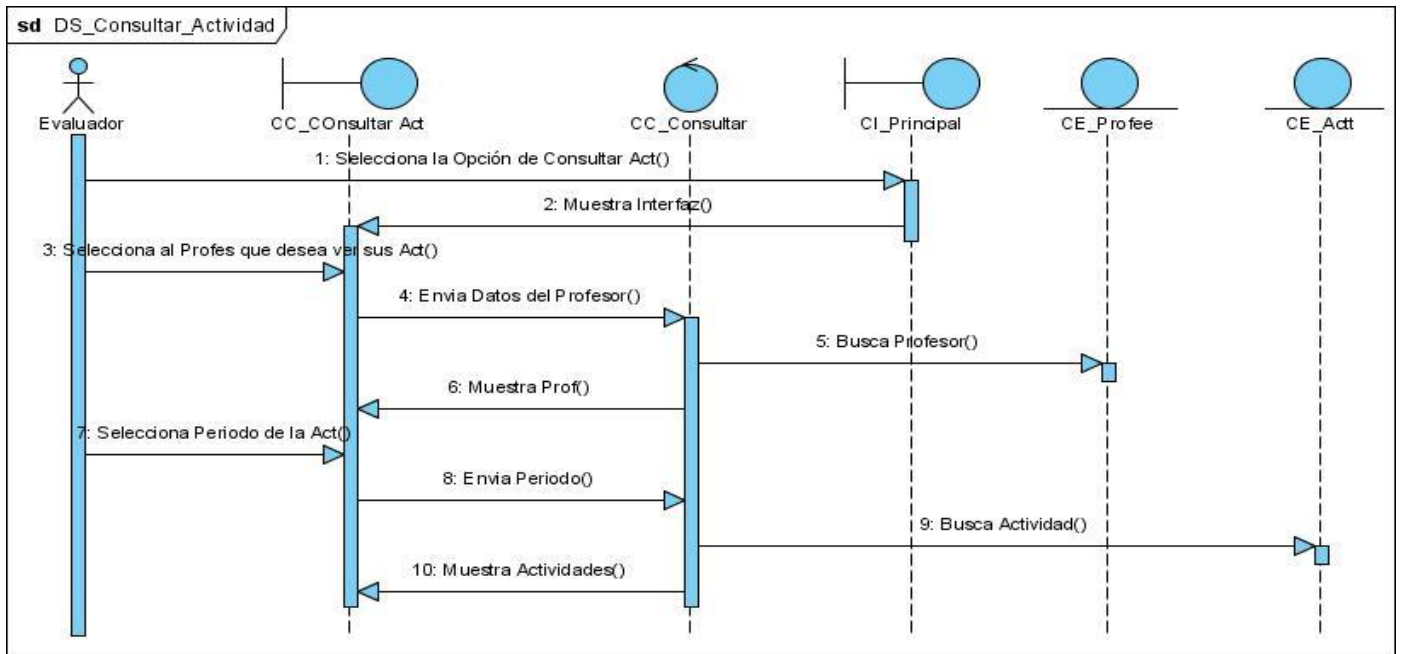
Diagramas de Secuencia: Gestionar Actividad (Eliminar).



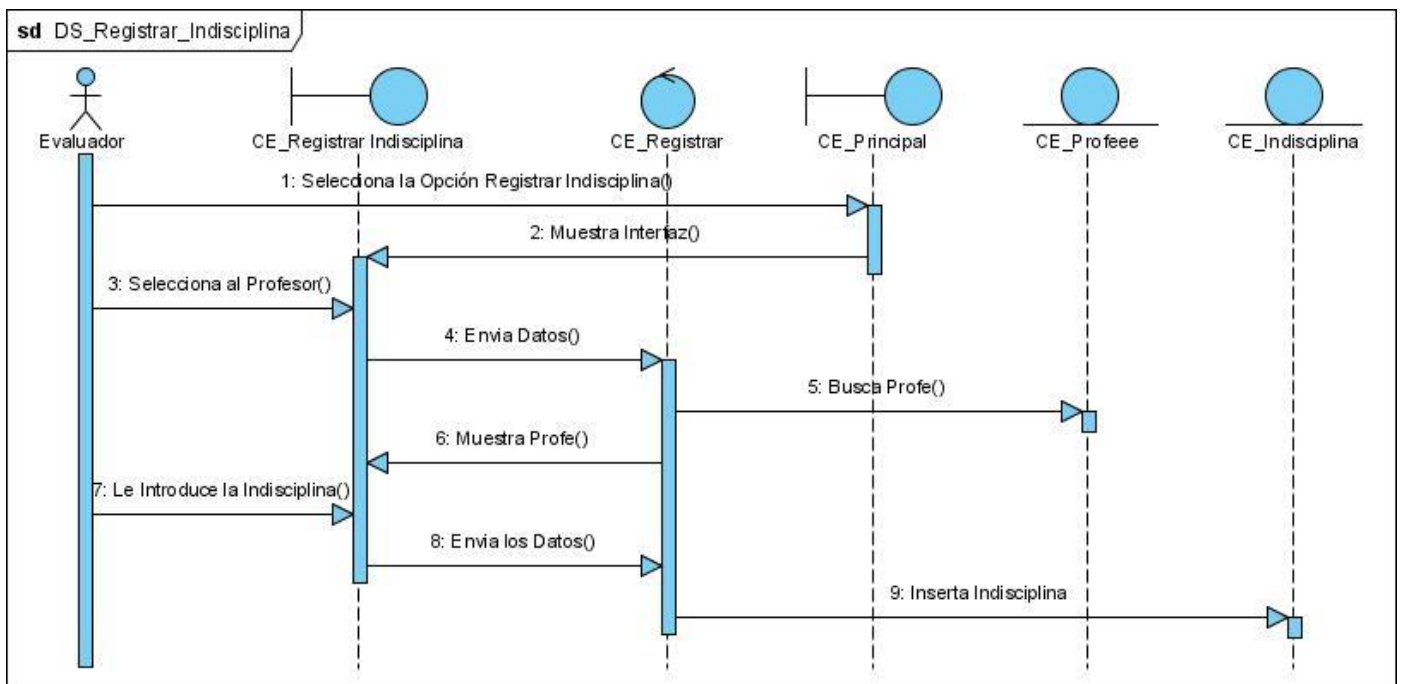
Diagramas de Secuencia: Generar Reportes.



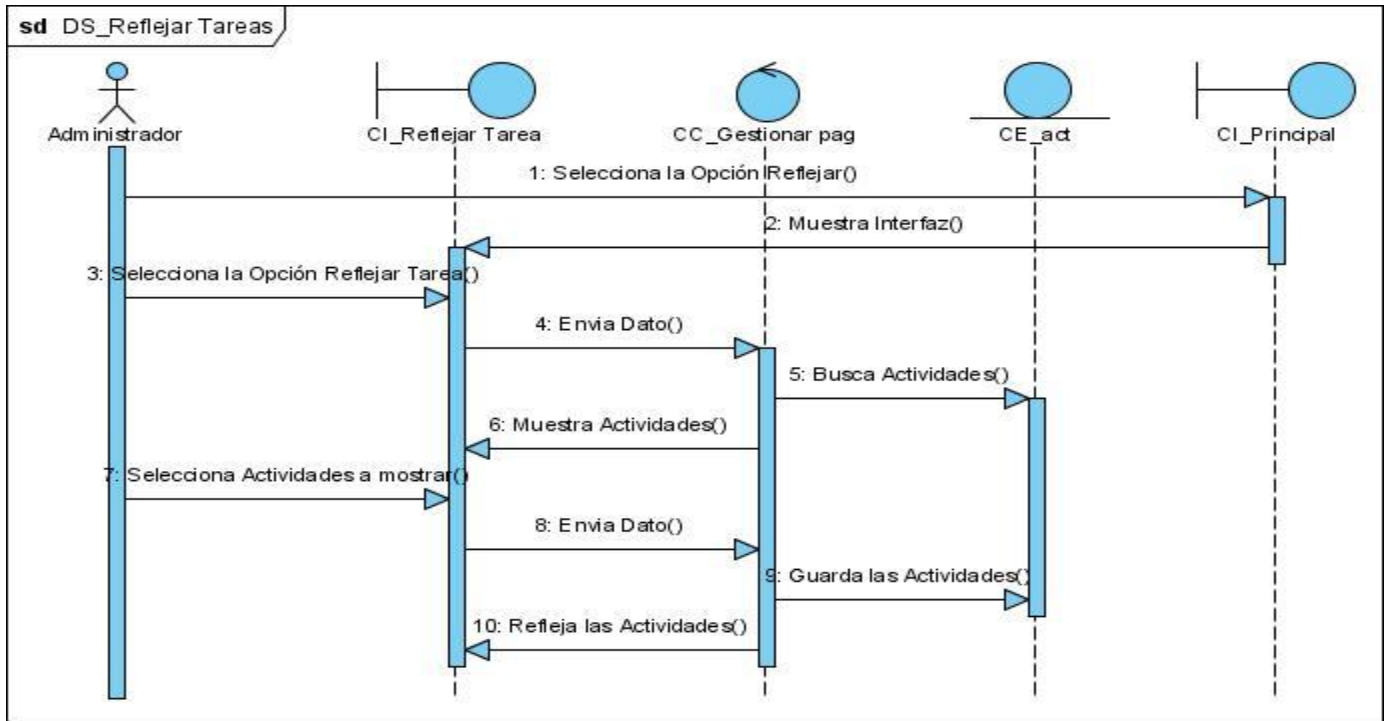
Diagramas de Secuencia: Consultar Actividades.



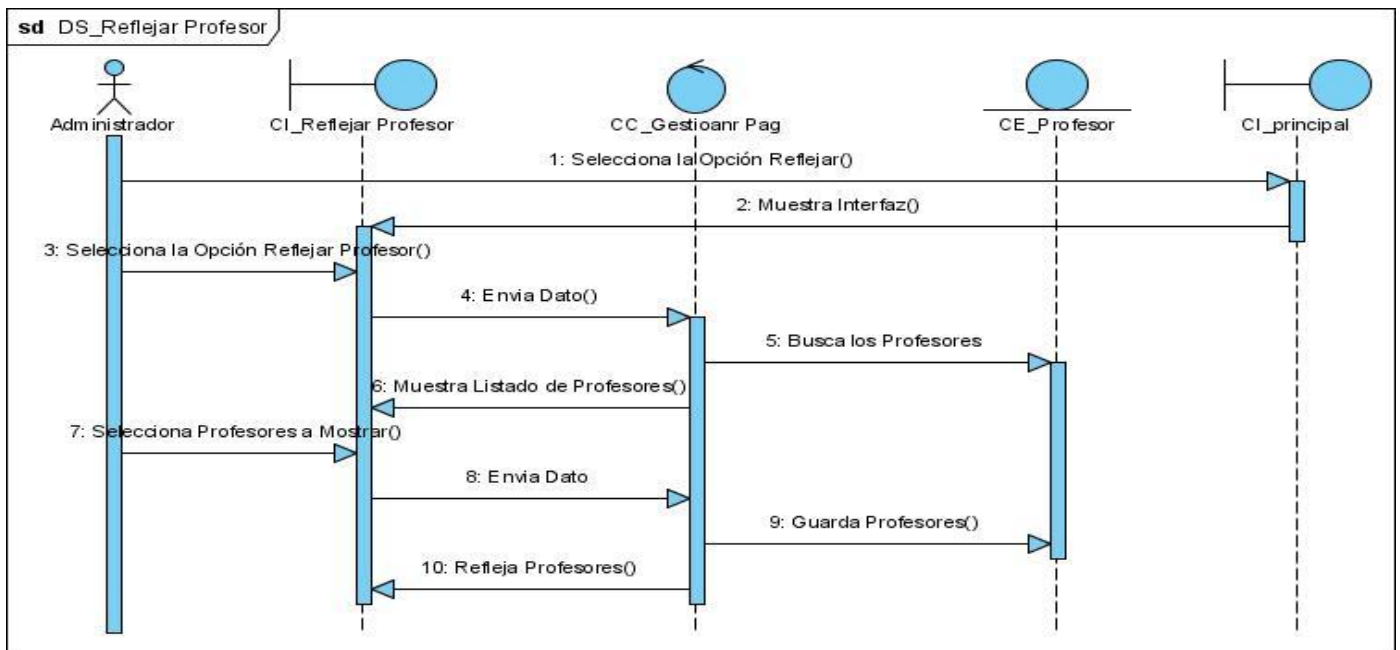
Diagramas de Secuencia: Registrar Indisciplina.



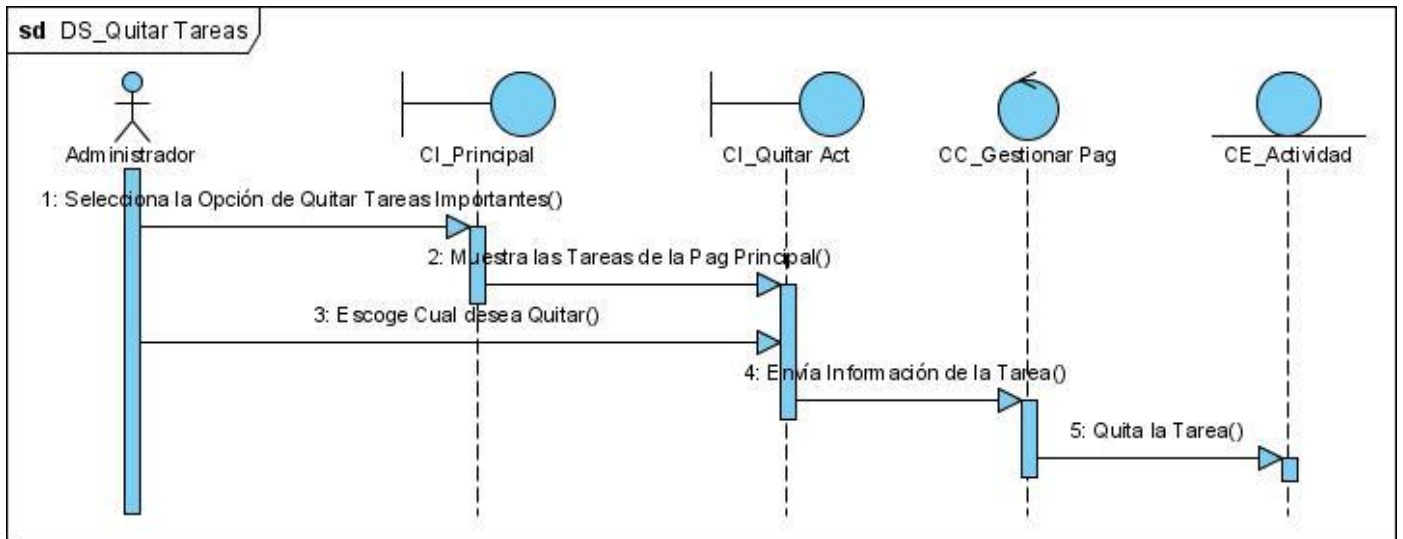
Diagramas de Secuencia: Gestionar Página Principal (Reflejar Tareas).



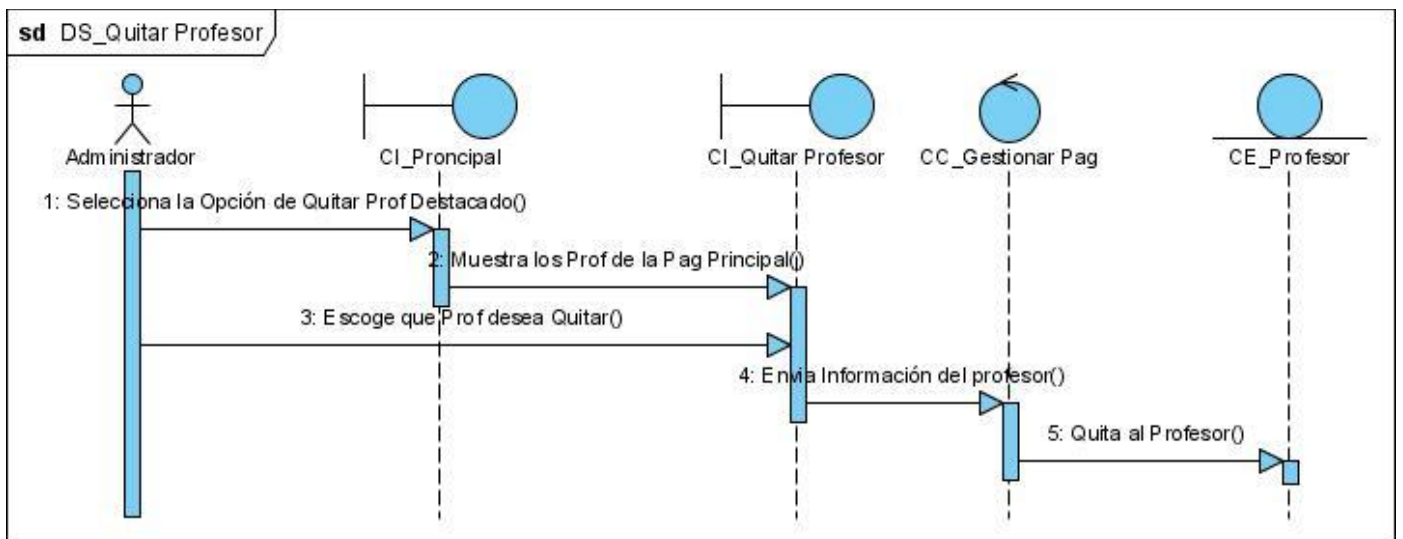
Diagramas de Secuencia: Gestionar Página Principal (Reflejar Profesor).



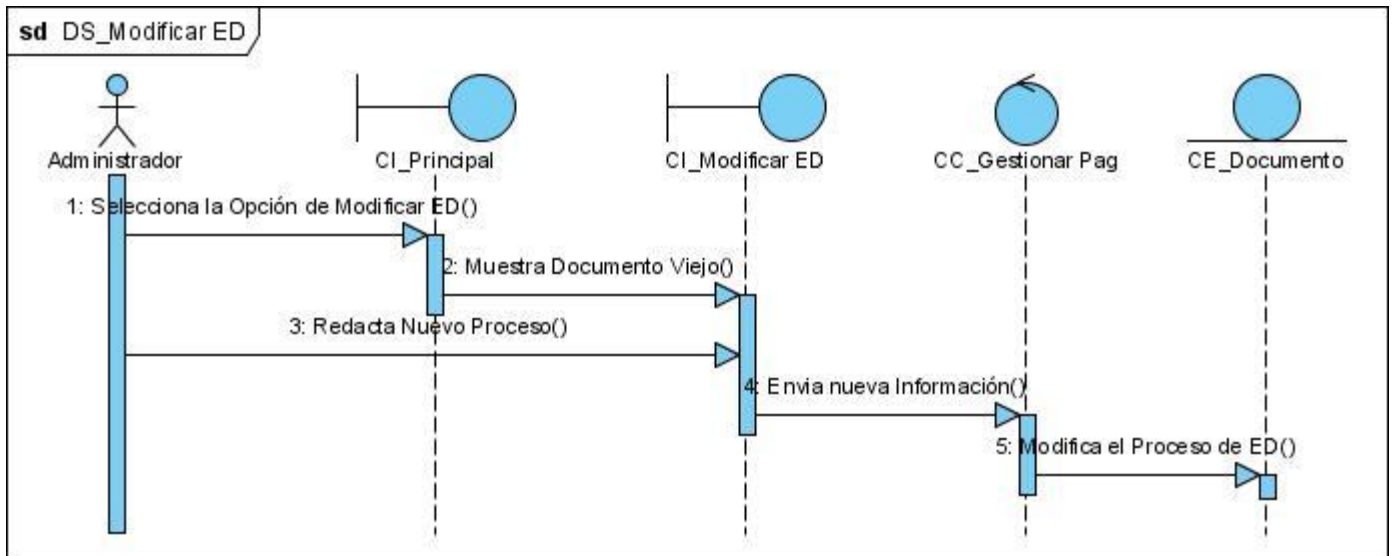
Diagramas de Secuencia: Gestionar Página Principal (Quitar Tareas).



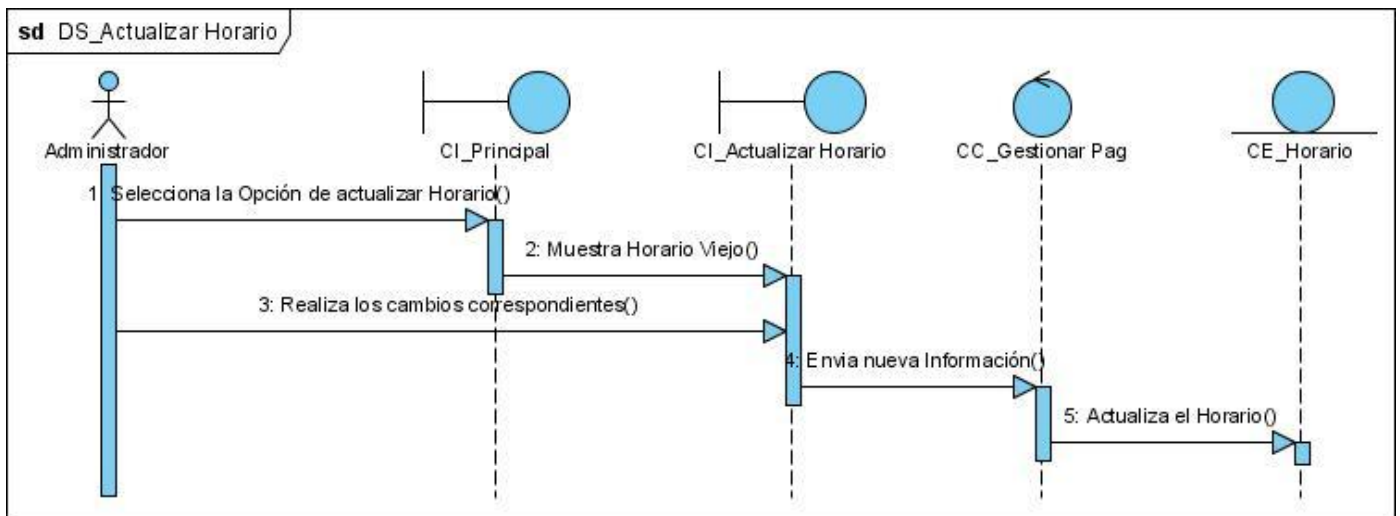
Diagramas de Secuencia: Gestionar Página Principal (Quitar Profesor).



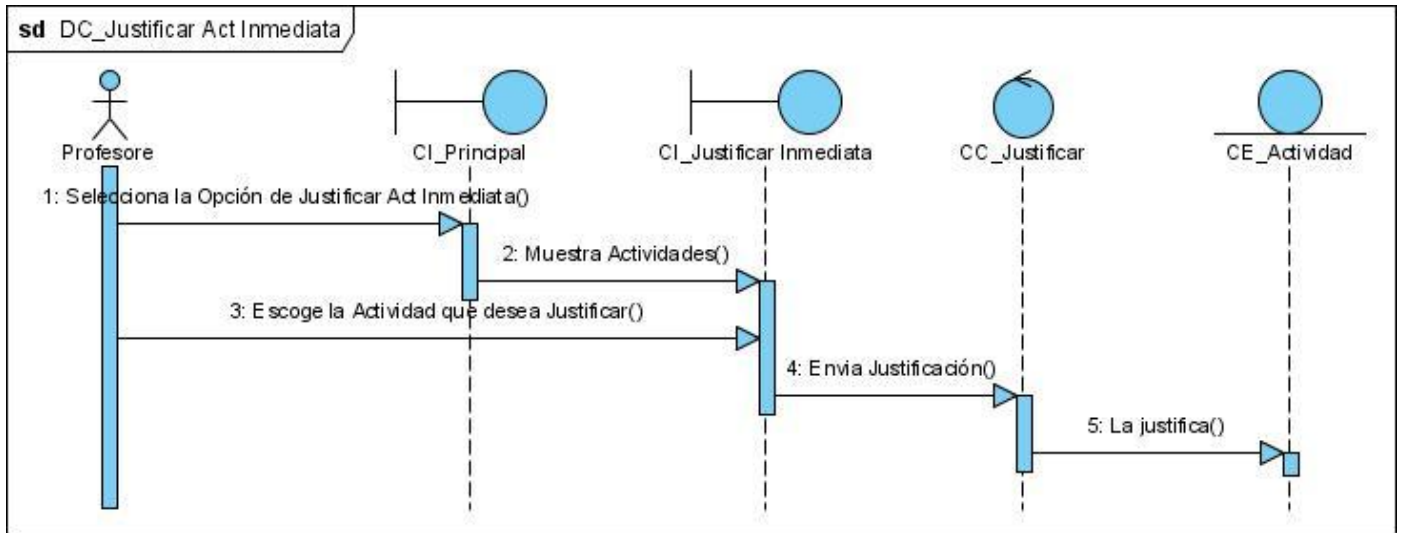
Diagramas de Secuencia: Gestionar Página Principal (Modificar ED).



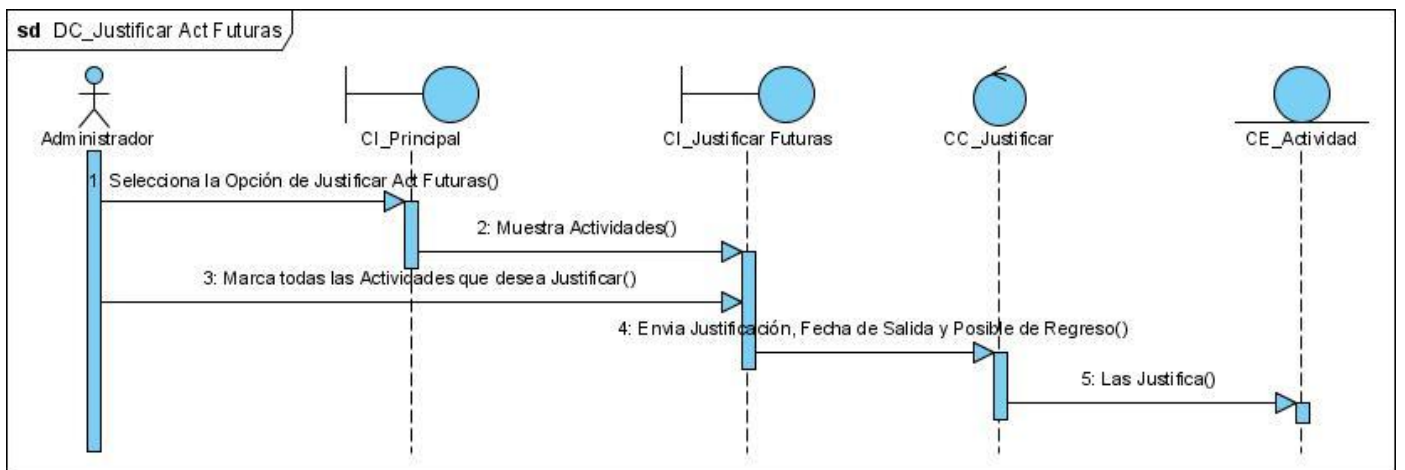
Diagramas de Secuencia: Gestionar Página Principal (Actualizar Horario).



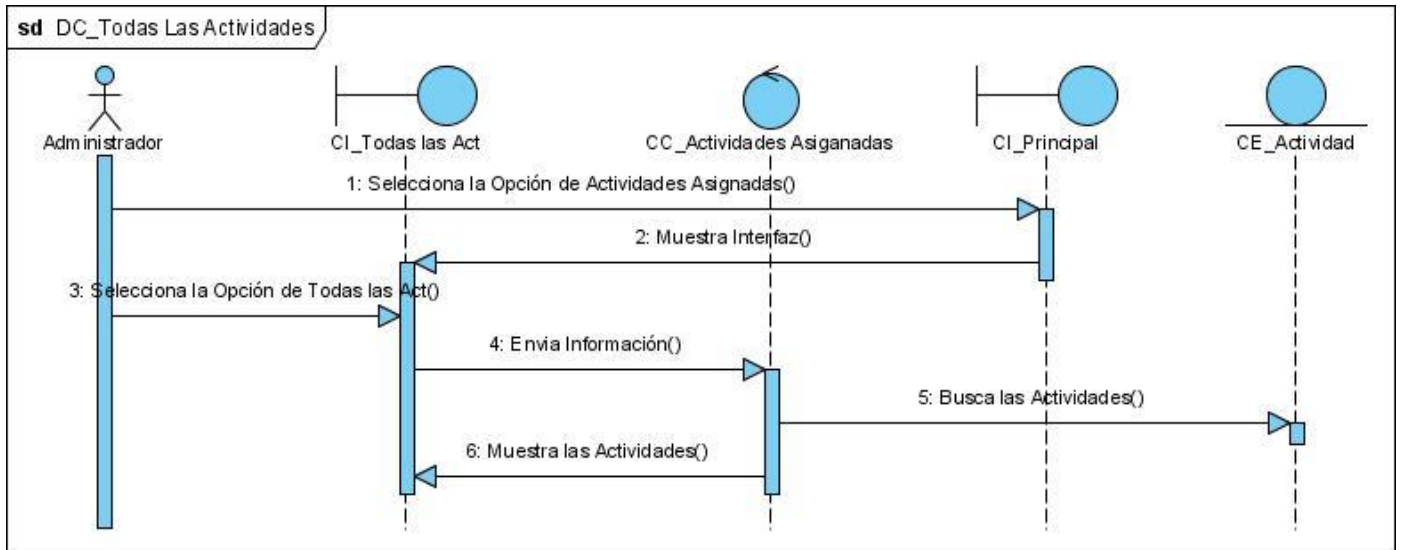
Diagramas de Secuencia: Justificar (Actividades Inmediatas).



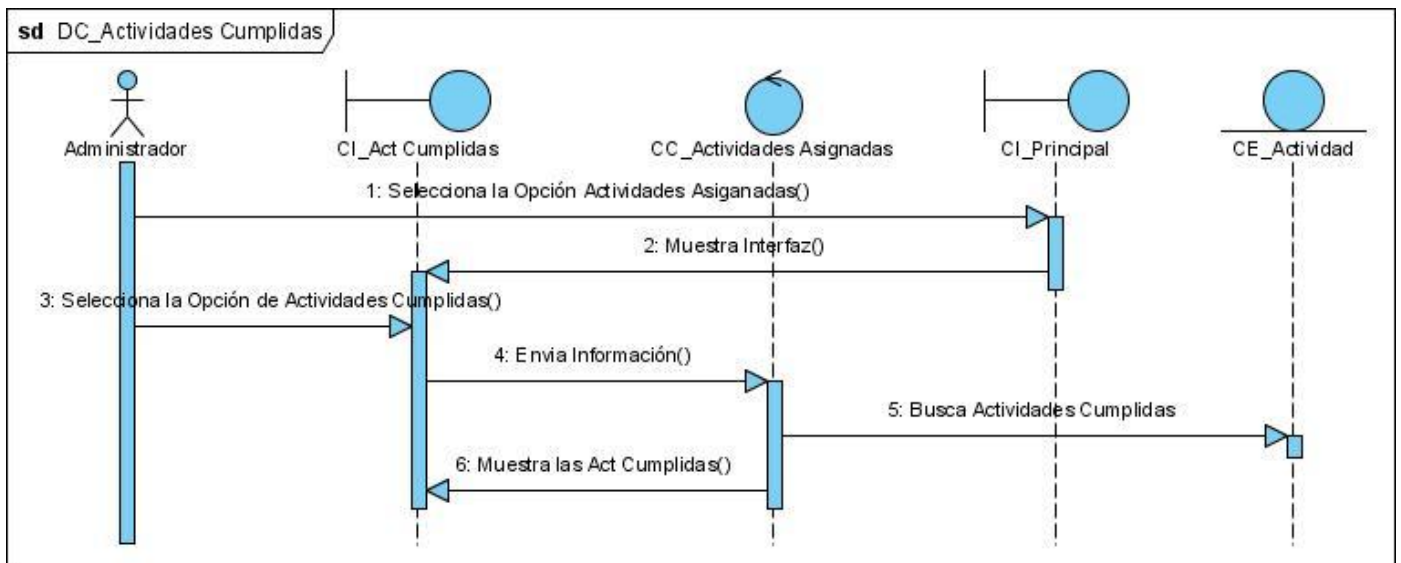
Diagramas de Secuencia: Justificar (Actividades Futuras).



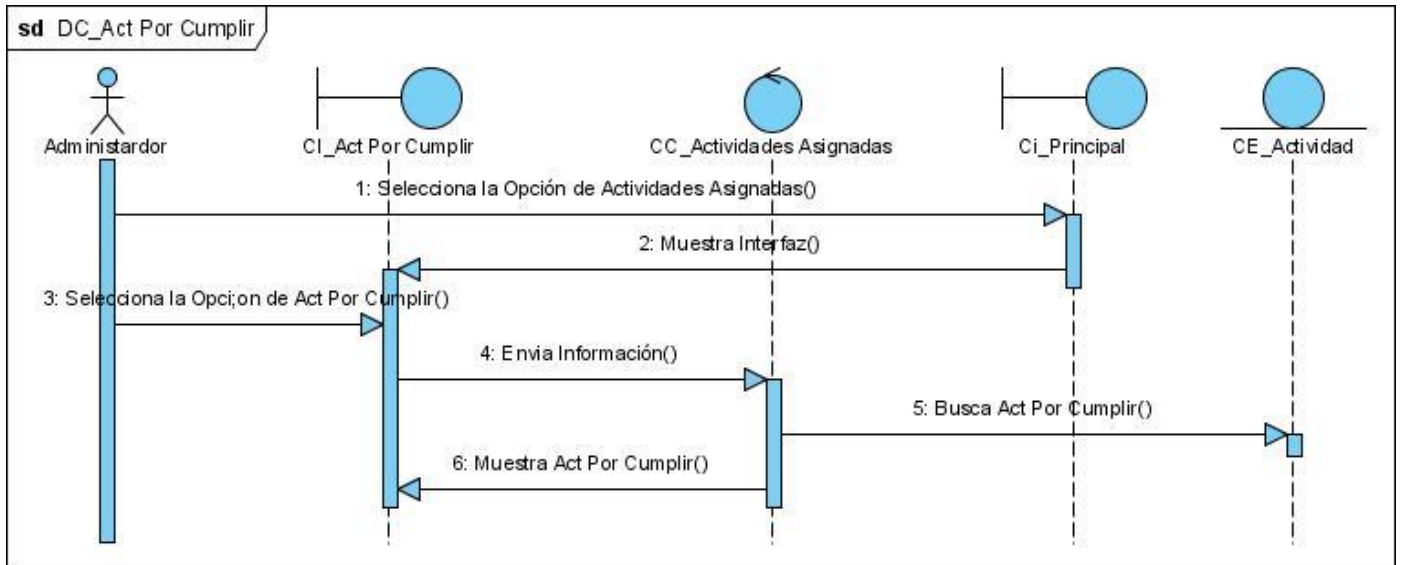
Diagramas de Secuencia: Actividades Asignadas (Todas las Actividades).



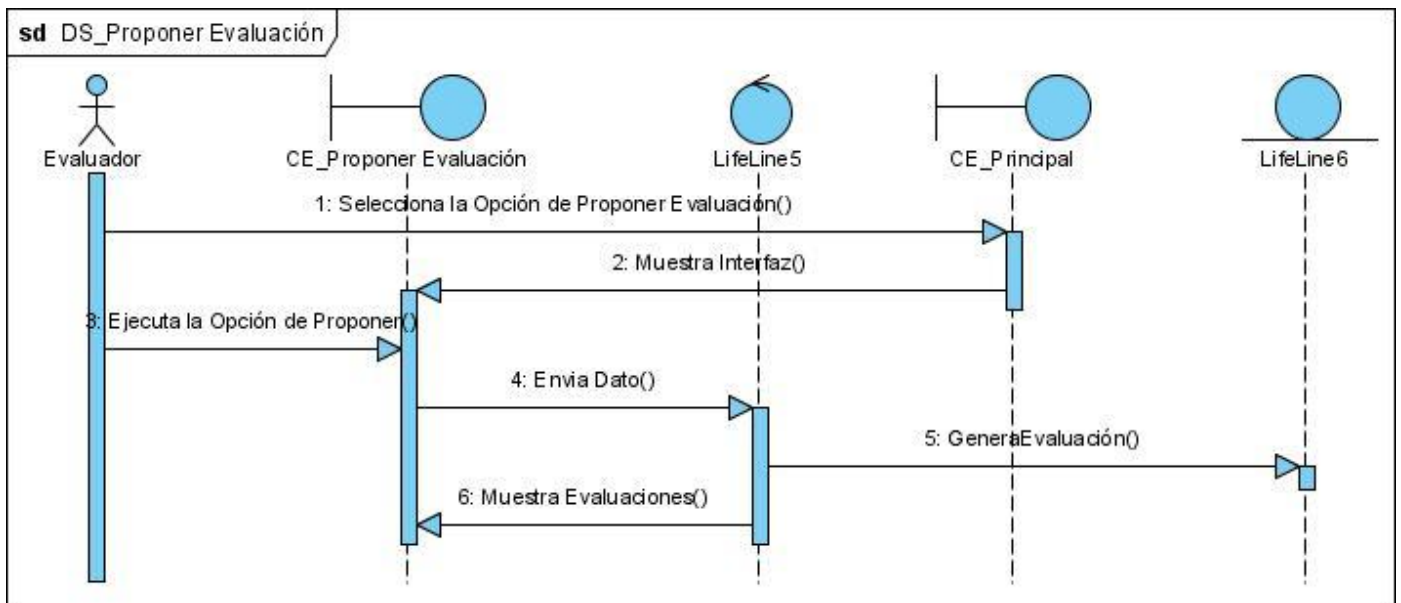
Diagramas de Secuencia: Actividades Asignadas (Actividades Cumplidas).



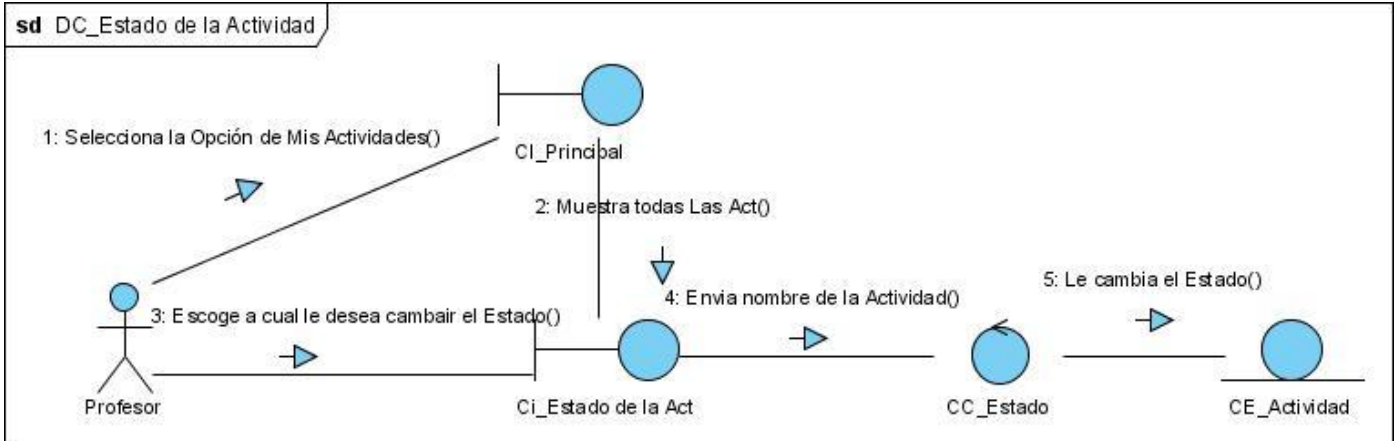
Diagramas de Secuencia: Actividades Asignadas (Actividades Por Cumplir).



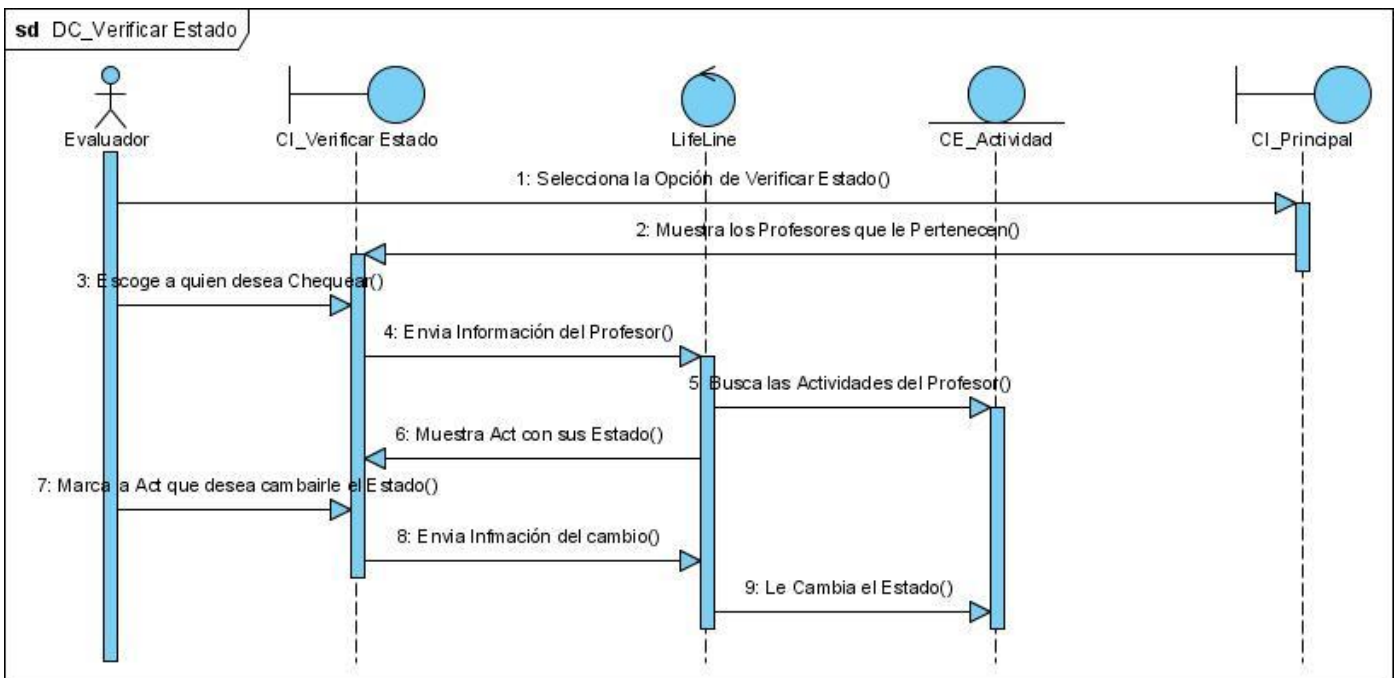
Diagramas de Secuencia: Proponer Evaluación.



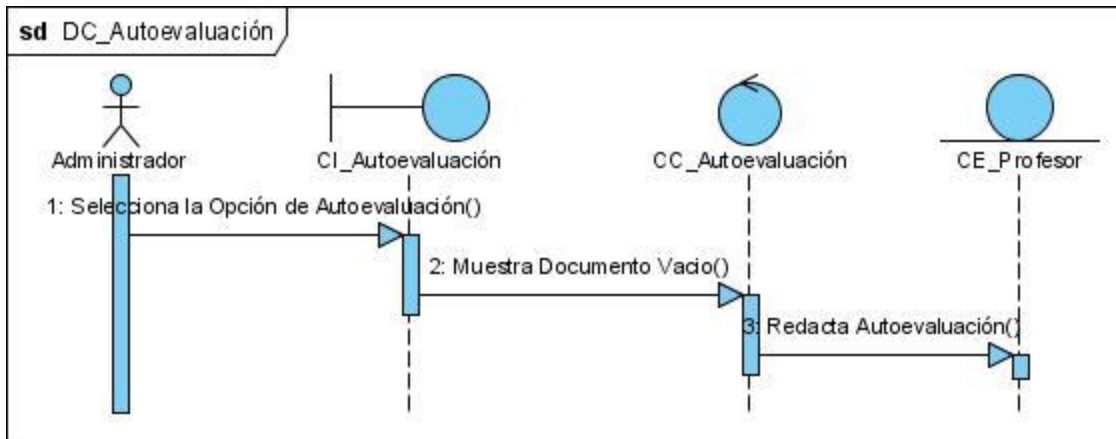
Diagramas de Secuencia: Estado de la Actividad.



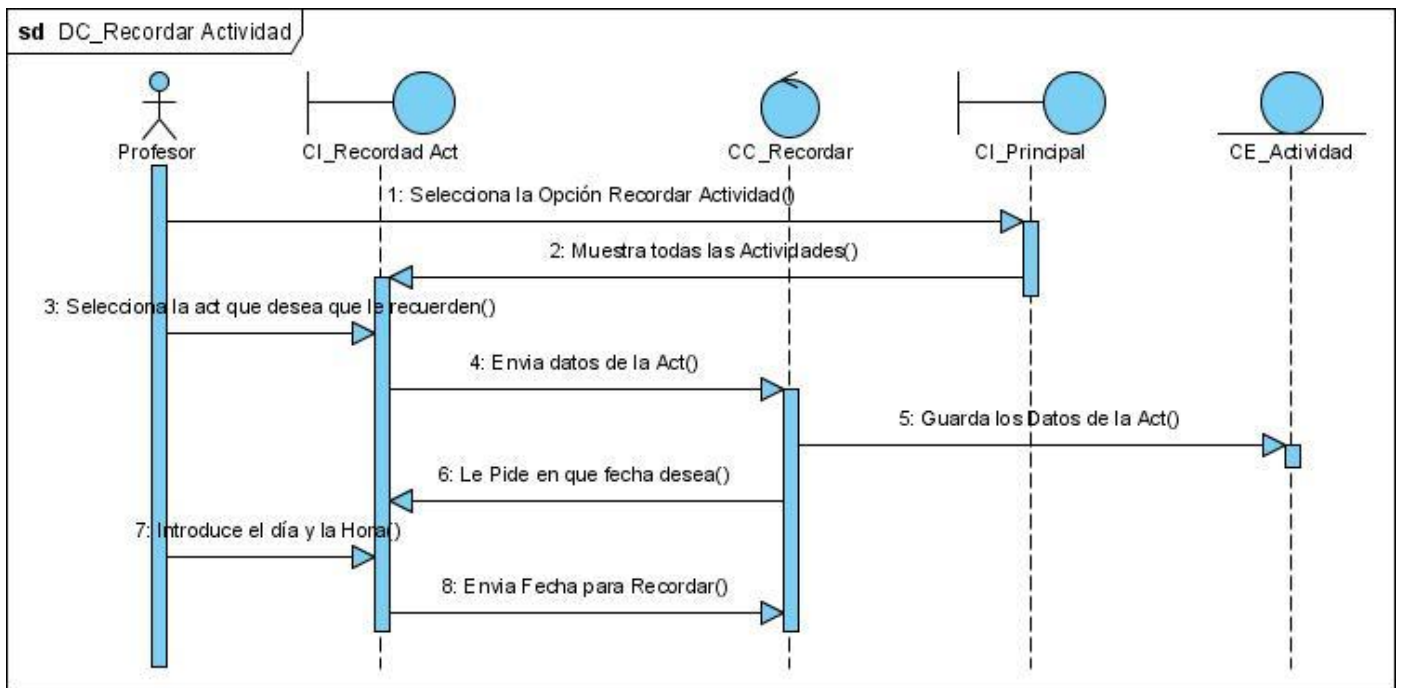
Diagramas de Secuencia: Verificar Actividad.



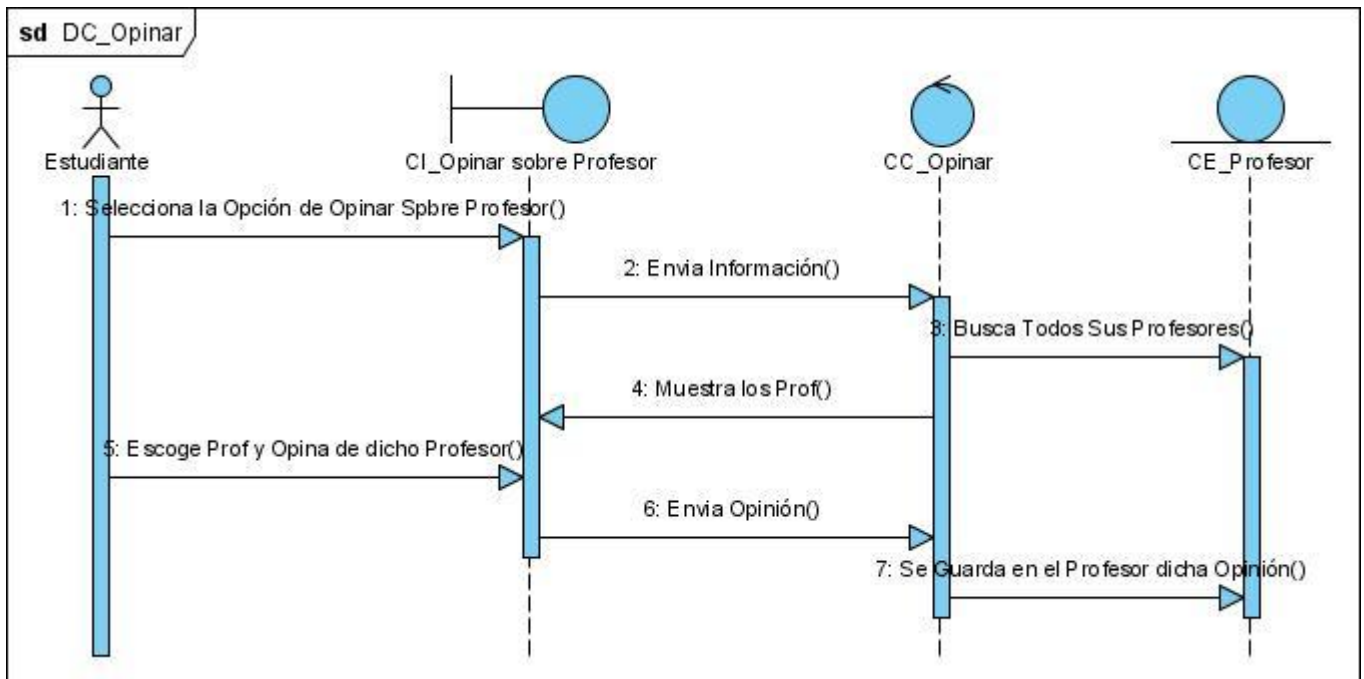
Diagramas de Secuencia: Autoevaluación.



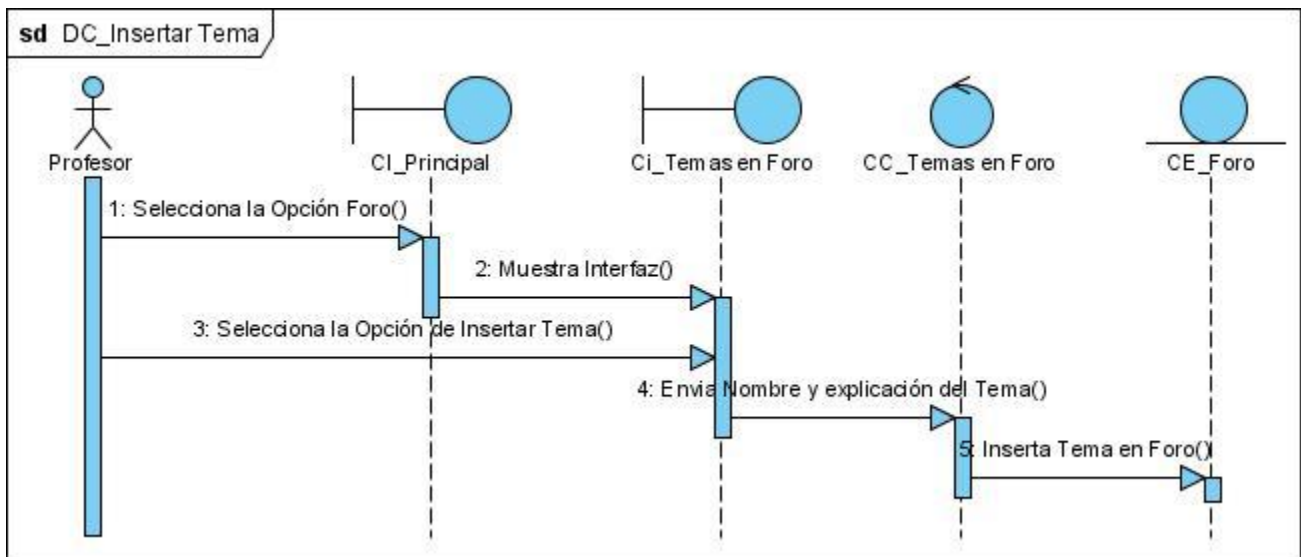
Diagramas de Secuencia: Recordar Actividad.



Diagramas de Secuencia: Opinar.

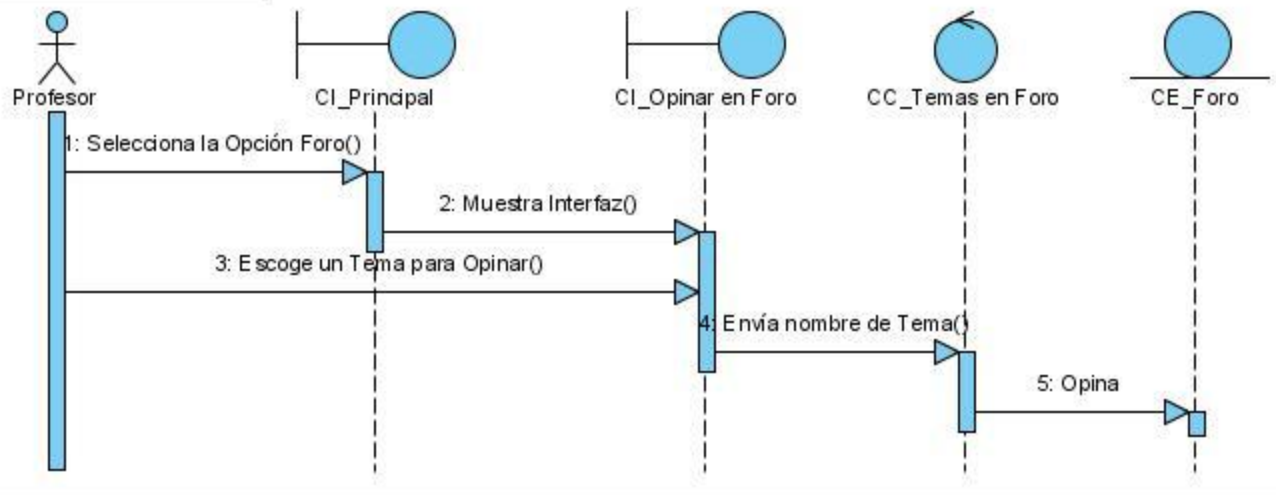


Diagramas de Secuencia: Tema en Foro (Insertar Tema).



Diagramas de Secuencia: Tema en Foro (Opinar en el Foro).

sd DC_Opinar en el Foro



GLOSARIO

Actores: Roles pertenecientes a los usuarios, agrupados según sus iteraciones con las funcionalidades del sistema.

SGBD: Sistema Gestor de Base de Dato.

Requerimientos: Condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo.

Persistente: Conjunto de datos y elementos que deber ser almacenados por el espacio de tiempo que se requiera, para dar soporte de información a un sistema u organización.

GRASP (General Responsibility Assignment Software Patterns): Patrones generales de software para asignación de responsabilidades.

DCUS: Diagrama de casos de uso del sistema. Representación de la relación de los CU con los actores del sistema.

XP (Extreme Programming).

Multiplataforma: es un término usado para referirse a los programas, sistemas operativos, lenguajes de programación, u otra clase de software, que puedan funcionar en diversas plataformas.

MySQL (Lenguaje Estructurado de Consulta): es un sistema de gestión de base de datos relacional, multihilo y multiusuario.

CGI (Common Gateway Interface): Es un mecanismo de comunicación entre el servidor web y una aplicación externa cuyo resultado final de la ejecución son objetos.

ISAPI 8: (Internet Server Application Programming Interface).

CSS (Hojas de estilo en cascada): Es usado para definir la presentación de un documento estructurado escrito en HTML o XML.

IDE (Integrated Development Environment): Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica.

