

Universidad de las Ciencias Informáticas



Título: “Análisis y Diseño de los Procesos de Gestión de Personal para Akademos v2.0.”

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

Autor(es):

Mireydis Orellana León

Yurisleis Alonso Beatón

Tutor(es):

Ing. Alexey Peña Paz

Ing. Catherine Muñoz Velázquez

Ciudad de La Habana, Junio 2009.

“Año 50 de la Revolución”

“Si buscas resultados distintos, no hagas siempre lo mismo”

Albert Einstein



DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores del trabajo titulado:

Análisis y Diseño de los Procesos de Gestión de Personal para Akademos v2.0 y autorizamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Yurisleis Alonso Beatón

Mireydis Orellana León

Ing. Alexey Peña Paz

Ing. Catherine Muñoz Velázquez



OPINIÓN DE LOS TUTORES DEL TRABAJO DE DIPLOMA.

Título: Análisis y Diseño de los Procesos de Gestión de Personal para Akademos v2.0.

Autor(es): Yurisleis Alonso Beatón y Mireydis Orellana León

Los tutores del presente Trabajo de Diploma consideran que durante su ejecución los estudiantes mostraron las cualidades que a continuación se detallan.

Los diplomantes mostraron alta consagración, independencia, creatividad y responsabilidad los cuales se manifiestan en este resultado, trabajo realizado con calidad y acorde a las exigencias requeridas.

Además demostraron en todo momento estar capacitados para asumir y acometer correctamente el trabajo propuesto. Su labor fue desempeñada de forma ardua y constante lo que le permitió desenvolverse fácilmente y avanzar sin grandes contratiempos.

Por todo lo anteriormente expresado consideramos que Yurisleis Alonso Beatón y Mireydis Orellana León están aptos para ejercer como Ingenieros Informáticos; y proponemos que se le otorgue al Trabajo de Diploma la calificación de 5 puntos.

Además se considera que los resultados de este trabajo ameritan ser presentados también en eventos científicos técnicos como baluartes de nuestra sociedad en la gestión de proyectos.

Ing. Catherine Muñoz Velázquez

Ing. Alexey Peña Paz

Fecha



Agradezco:

A mis padres por ser la luz que guía mi vida desde el momento en que me trajeron al mundo y por darme su infinito amor, comprensión y apoyo.

A mi hermanito que es lo que más quiero en este mundo.

A mi Abu que es como si fuera mi otra mamá.

A mi novio Lisandro, mi tati, que sin su amor y comprensión no hubiera soportado estos cinco años tan lejos de mis seres más queridos.

A mi familia por estar pendiente de mí cada día que me pase aquí en la Universidad y por darme su apoyo incondicional.

A mis suegros Isela y Roberto que me han tratado en estos años como una hija.

A la revolución por darme la oportunidad de cumplir el más grande de mis sueños, graduarme.

A mis grandes amigos de todos los tiempos Yuleiky y Alberto por vivir pendientes de mí.

A mis compañeros de grupo Acralys, Dariel y Enmanuel que me apoyaron y ayudaron para que el presente trabajo saliera adelante, y en especial a Yisel, que fue la que más me tuvo que soportar en esos días.

A mis tutores Alexey y Catherine por toda la ayuda brindada.

A mis compañeras de cuarto Ailyn, Noralis y Dania Rosa.

A mi compañero de tesis Yuri.

A Marianny, gracias por prestarnos algo de su poco tiempo.

A todos los que de una forma u otra nos ayudaron para que el presente trabajo saliera adelante y nos pudiéramos graduar, a todos muchas gracias.

Mireydís



Agradezco:

A mis padres que siempre confiaron en mí.

A mis abuelos por ser como mis padres desde siempre.

A mi tía Rosa por toda su ayuda, apoyo y preocupación desde que vine para la Universidad y tratarme como su propio hijo.

A mi compañera de tesis Mireydis.

A mis tutores Alexey y Catherine por guiarnos todo el tiempo en la tesis.

A mis compañeros de grupo Enmanuel, Dariel, Yisel y Acralys por formar parte de este trabajo.

A mi gran amigo Henry por tenerme como un hermano y ayudarme en los momentos que más lo necesitaba.

A mi amigo Roberto Carlos por todo su apoyo y preocupación en todo momento.

A Pepe, Eliza, Elizita, Harold, Rey y Juan Miguel por permitirme ser parte de su familia y brindarme su apoyo y confianza en todo el tiempo que estuve en la universidad.

A mis compañeros y amigos David, Leo, Osmany, Heber, Jose, Alejo e Indira.

A Marianny, por prestarnos algo de su poco tiempo y sus sabios consejos.

A toda mi familia por confiar en mí y tenerme presente.

A la Revolución y en especial a la UCI por brindarme la posibilidad de prepararme y graduarme en esta profesión tan especial.

A todos los que directa e indirectamente nos ayudaron para que el presente trabajo saliera adelante, a todos muchas gracias.

Yuri

A:

Mi mamá y mi papá.

Yuri.

A:

Mis padres, mi hermanito, mi tati y mi abu.

Mireydis.

RESUMEN

La Universidad de las Ciencias Informáticas cuenta con un sistema automatizado para la Gestión Académica, Akademos, que permite llevar el control de toda la actividad docente del centro de estudios.

El presente trabajo se enfoca en dos de sus módulos, matrícula y profesor, encargados del control de los datos de los estudiantes y profesores de la institución. Los procesos realizados en los dos módulos anteriores, en realidad son parte de uno solo: la Gestión de Personal, por lo que muchas de las funcionalidades que brindan por separado se encuentran repetidas, perdiendo eficiencia el sistema.

Además está hecho completamente en software propietario, lo cual no se adecua a las políticas de software tanto de la Universidad como del país, ya que se ha optado por el software libre debido a las ventajas que proporciona su uso.

Con el objetivo de dar solución a los problemas antes planteados se comienza a desarrollar la segunda versión del sistema, surgiendo este trabajo, cuyo propósito fundamental es realizar el análisis y diseño del módulo Gestión de Personal para Akademos v2.0.

Para ello se utilizó RUP como metodología de desarrollo, UML como lenguaje de modelado, Visual Paradigm como herramienta CASE para el modelado, Ext JS y Zend Framework como frameworks de desarrollo, Doctrine como ORM y PHP como lenguaje de programación.

Se espera que se tome el diseño propuesto como punto de partida para la posterior implementación del módulo y que el nuevo sistema posea la flexibilidad y genericidad que se necesita para llevar a cabo con mayor eficiencia los procesos de Gestión de Personal en la Universidad.

ÍNDICE DE CONTENIDOS

INTRODUCCIÓN.....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	5
1.1 Introducción.....	5
1.2 Los Sistemas de Gestión Académica.	5
1.2.1 Los Sistemas de Gestión Académica en el Mundo.....	5
1.2.2 Los Sistemas de Gestión Académica en Cuba.....	7
1.2.3 Los Sistemas de Gestión Académica en la Universidad de las Ciencia Informáticas	10
1.2.4 Marco Teórico-Conceptual de la Investigación.....	12
1.3 Tendencias actuales en el desarrollo de Software.....	12
1.4 Metodologías, lenguajes y herramientas de modelado	14
1.4.1 Metodologías de desarrollo de software.....	14
1.4.2 Lenguaje de modelado.....	18
1.4.3 Herramientas CASE para el modelado.....	19
1.4.4 Fundamentación de la herramienta, lenguaje y metodología utilizadas para el modelado...	20
1.5 Arquitectura.....	21
1.5.1 Fundamentación de la arquitectura a utilizar	22
1.6 Herramientas, tecnologías y lenguajes para el desarrollo.....	22
1.6.1 Frameworks	22
1.6.2 Object Relational Mapper	25
1.6.3 Lenguajes para el desarrollo Web.....	25
1.6.4 Sistemas Gestores de Base de Datos.....	29
1.6.5 Fundamentación de las herramientas, tecnologías y lenguajes a utilizar para el desarrollo del sistema.	31
1.7 Conclusiones.....	32
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA	33
2.1 Introducción.....	33
2.2 Descripción del Negocio	33
2.3 Reglas del Negocio	36
2.4 Actores del Negocio	36
2.5 Trabajadores del Negocio.....	37
2.6 Diagrama de Casos de Uso del Negocio	38
2.7 Diagramas de Actividades de los Casos de Uso del Negocio	38
2.8 Descripción de los Casos de Uso del Negocio.	38

2.9 Modelo de Objetos	50
2.10 Especificación de Requerimientos del Sistema	51
2.10.1 Requerimientos Funcionales.....	51
2.10.2 Requerimientos no Funcionales.....	60
2.11 Definición de Actores del Sistema	62
2.12 Diagrama de Casos de Usos del Sistema.....	63
2.13 Descripciones de los Casos de Uso del Sistema.....	63
2.14 Estimación del Esfuerzo.....	68
2.15 Aportes y Beneficios.....	72
2.16 Conclusiones.....	73
CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA.....	74
3.1 Introducción.....	74
3.2 Análisis.....	74
3.2.1 Diagramas de Clases del Análisis	74
3.2.2 Diagramas de Colaboración.....	77
3.3 Diseño	77
3.3.1 Patrones de Diseño.....	77
3.3.2 Diagramas de Clases del Diseño.....	80
3.3.3 Diagramas de Secuencia	87
3.4 Diseño de la BD	87
3.4.1 Modelo Lógico.....	88
3.4.2 Modelo Físico.....	88
3.5 Diagrama de Despliegue	89
3.6 Conclusiones.....	89
CONCLUSIONES	90
RECOMENDACIONES.....	91
REFERENCIAS BIBLIOGRÁFICA	92
BIBLIOGRAFÍA.....	93
ANEXOS.....	95
GLOSARIO DE TÉRMINOS	166

ÍNDICE DE TABLAS

Tabla 2.1: Actores del Negocio.....	37
Tabla 2.2: Trabajadores del Negocio.....	37
Tabla 2.3: Descripción del CUN Prematricular	39
Tabla 2.4: Descripción del CUN Solicitar Matrícula.....	40
Tabla 2.5: Descripción del CUN Ratificar Matrícula	40
Tabla 2.6: Descripción del CUN Solicitar Reingreso.....	41
Tabla 2.7: Descripción del CUN Egresar	42
Tabla 2.8: Descripción del CUN Solicitar Ingreso	43
Tabla 2.9: Descripción de CUN Solicitar Traslado Estudiante	46
Tabla 2.10: Descripción de CUN Solicitar Traslado Profesor	46
Tabla 2.11: Descripción de CUN Solicitar Licencia Estudiante.....	47
Tabla 2.12: Descripción del CUN Solicitar Licencia Profesor	48
Tabla 2.13: Descripción del CUN Solicitar Baja Estudiante.....	49
Tabla 2.14: Descripción del CUN Solicitar Baja Profesor	50
Tabla 2.15: Definición de Actores del Sistema.....	62
Tabla 2.16: Descripción abreviada del CUS Gestionar Tipo Personal.....	64
Tabla 2.17: Descripción abreviada del CUS Gestionar Estados.....	64
Tabla 2.18: Descripción abreviada del CUS Gestionar Datos Persona.....	65
Tabla 2.19: Descripción abreviada del CUS Buscar Persona	66
Tabla 2.20: Descripción abreviada del CUS Realizar Movimiento	66
Tabla 2.21: Descripción abreviada del CUS Ubicar Persona en Estructura	66
Tabla 2.22: Descripción abreviada del CUS Asignar Asignaturas Profesor.....	67
Tabla 2.23: Descripción abreviada del CUS Asociar Plan de Estudio	67
Tabla 2.24: Descripción abreviada del CUS Consultar Información	68
Tabla 2.25: Factor de Peso de los Actores sin ajustar	69
Tabla 2.26: Factor de Peso de los Casos de Usos sin ajustar	69
Tabla 2.27: Factor de complejidad técnica.....	71
Tabla 2.28: Factor de ambiente.....	71
Tabla 2.29: Esfuerzo en Horas-Hombre	72

ÍNDICE DE FIGURAS

Figura 1: Diagrama de Casos de Uso del Negocio.....	38
Figura 2: Modelo de Objetos.....	50
Figura 3: Diagrama de Casos de Uso del Sistema.....	63
Figura 4: Diagrama de clases del análisis CUS Gestionar Tipo Personal	74
Figura 5: Diagrama de clases del análisis CUS Gestionar Estados	75
Figura 6: Diagrama de clases del análisis CUS Gestionar Datos Persona.....	75
Figura 7: Diagrama de clases del análisis CUS Buscar Persona	75
Figura 8: Diagrama de clases del análisis CUS Realizar Movimiento.....	76
Figura 9: Diagrama de clases del análisis CUS Ubicar Persona en Estructura	76
Figura 10: Diagrama de clases del análisis CUS Asignar Asignaturas Profesor.....	76
Figura 11: Diagrama de clases del análisis CUS Asociar Plan de Estudio	76
Figura 12: Diagrama de clases del análisis CUS Consultar Información	77
Figura 13: Diagrama de clases del diseño CUS Gestionar Tipo Personal.....	80
Figura 14: Diagrama de clases del diseño CUS Gestionar Estados.....	81
Figura 15: Diagrama de clases del diseño CUS Gestionar Datos Persona	82
Figura 16: Diagrama de clases del diseño CUS Buscar Persona.....	83
Figura 17: Diagrama de clases del diseño CUS Realizar Movimiento	84
Figura 18: Diagrama de clases del diseño CUS Ubicar Persona en Estructura.....	84
Figura 19: Diagrama de clases del diseño CUS Asignar Asignaturas Profesor	85
Figura 20: Diagrama de clases del diseño CUS Asociar Plan de Estudio	86
Figura 21: Diagrama de clases del diseño CUS Consultar Información	87
Figura 22: Modelo Lógico de Datos	88
Figura 23: Modelo Físico de Datos	88
Figura 24: Diagrama de Despliegue.....	89



INTRODUCCIÓN

La Gestión Académica constituye un elemento de gran importancia para el buen funcionamiento de cualquier centro educacional pues permite conocer, procesar y almacenar toda la información que se genera en el proceso docente.

Entre sus principales procesos se encuentra el control de los datos referentes al personal docente del centro de estudios. Este proceso incluye el registro de datos y ubicación del personal docente en la institución (facultades, grupos docentes, departamentos), así como la gestión de los movimientos a que es sometido este personal a su paso por la misma.

En los últimos años se ha producido un cambio radical en la forma en que se realizan los procesos docentes debido a la creciente incorporación de las nuevas tecnologías de la información. Los centros de estudios han comenzado a apostar por un modelo más personalizado, donde alumnos y profesores interactúen de forma directa con sistemas encargados de automatizar los procesos de gestión académica. Estos sistemas constituyen una poderosa herramienta de trabajo que permite agilizar los mismos y optimizar recursos.

La Universidad de las Ciencias Informáticas, desde sus inicios, también ha llevado a cabo la informatización de los procesos de gestión académica ya que cuenta con una inmensa cantidad de estudiantes, profesores, personal de secretaría y directivos. Para ello fue desarrollado el Sistema de Gestión Académica, Akademos, el cual posibilita una alta automatización de los procesos docentes en la Universidad, aportando una serie de facilidades al personal docente a través de varios servicios que brinda de manera dinámica.

Debido a las nuevas exigencias en la educación ha surgido la necesidad de llevar a cabo la informatización de los procesos de gestión académica de forma diferente, por lo que el sistema está sufriendo una serie de cambios y se pretende crear una nueva versión del mismo donde se solucionen los problemas de la gestión académica de un modo más genérico y permita lograr mayor eficiencia en todos los procesos involucrados. Esta versión contará con una serie de módulos: Gestión de Personal, Gestión de Carreras, Estructura y Composición, Registro y Control Docente, Reporte, Seguridad-Administración.

El presente trabajo abordará el análisis y diseño del módulo Gestión de Personal como parte de la nueva versión del sistema.

Situación Problemática

La versión del sistema Akademos que actualmente se utiliza, satisface muchas de las necesidades de la gestión académica de la Universidad, pero debido a la forma en que se implementó y la poca documentación existente se dificulta su mantenimiento y soporte.

Además está hecho completamente en software propietario, lo cual no se adecua a las políticas de software seguidas por la Universidad y el país, ya que se ha optado por el software libre debido a la serie de ventajas que proporciona su uso.

De los módulos que lo conforman, el módulo matrícula se encarga de la gestión de los datos y movimientos a que son sometidos los estudiantes en su paso por la Universidad y el módulo profesor gestiona los movimientos de los profesores en los grupos académicos, departamentos y facultades.

De acuerdo a las nuevas concepciones del sistema, los procesos que automatizan estos dos módulos por separado, en realidad forman parte de uno solo: la gestión del personal, por lo que muchas de las funcionalidades que brindan se encuentran repetidas, perdiendo eficiencia. Además, el módulo profesor no permite realizar la gestión de los datos de los profesores, los cuales juegan un papel muy importante en el proceso docente. Estos datos son importados de otro sistema sin permitir ninguna modificación en ellos, dependiendo en gran medida del sistema anterior.

La mayoría de estos problemas se deben a que el sistema no es un producto de software, sino una solución a los problemas de la gestión académica que existían en la Universidad en el momento de su desarrollo.

Teniendo en cuenta lo anteriormente expuesto, el **problema científico** queda formulado de la siguiente forma: ¿Cómo transformar los procesos de Gestión de Personal a un lenguaje entendible por los desarrolladores, que permita la posterior implementación?

El **objeto de estudio** de la presente investigación lo constituyen los procesos de Gestión de Personal en los Sistemas de Gestión Académica. Derivándose como **campo de acción** los procesos de Gestión de Personal en el Sistema Automatizado para la Gestión Académica – “Akademos”.

Se define como **objetivo** de la investigación: elaborar el análisis y diseño del módulo Gestión de Personal para Akademos v2.0. De ahí se derivan los siguientes **objetivos específicos**:

- Desarrollar el marco teórico de la investigación.
- Identificar los procesos relacionados con el módulo Gestión de Personal en la Universidad de las Ciencias Informáticas.
- Definir el análisis y diseño del módulo Gestión de Personal.

Para guiar la investigación se plantea la siguiente **idea a defender**: el análisis y diseño del módulo Gestión de Personal del sistema Automatizado para la Gestión Académica- “Akademos” facilitará su posterior implementación.

Para lograr el cumplimiento de los objetivos se proponen las siguientes **tareas investigativas**:

- Realizar la fundamentación teórica de todo el proceso.
- Realizar el estudio del estado del arte de los procesos de gestión académica.
- Fundamentar el uso de metodologías, lenguajes, tecnologías y herramientas a utilizar.
- Entrevistar a los clientes para estudiar los procesos del módulo Gestión de Personal.
- Estudiar la arquitectura propuesta para el desarrollo del sistema.

Métodos de Investigación Utilizados:

Histórico-lógico: En esta investigación dicho método se usa con el objetivo de estudiar todo lo referente a los procesos de gestión académica en el ámbito nacional e internacional, para de esta forma conocer más del tema, sus características e importancia. Este método ayuda a dar cumplimiento al estudio del estado del arte de los procesos de gestión académica.

Analítico-Sintético: Este método se usa para organizar y sintetizar toda la información obtenida del estudio del estado del arte, las tecnologías, metodologías, lenguajes y herramientas propuestas para la solución e identificar los principales problemas del sistema actual. La utilización de este método permite comprender mejor toda la información anterior, la cual será de gran utilidad para lograr un adecuado análisis y diseño del sistema.

Método de la Modelación: Partiendo de todo lo investigado y aprendido se realizan los modelos correspondientes al ciclo de vida del software, que ayudan a dar cumplimiento a las tareas de diseño de los procesos involucrados en la solución.

Método de observación: Este método sirve de apoyo para obtener información sobre el tema a tratar, y ayuda a dar solución a prácticamente todas las tareas. Haciendo uso de él se puede ver como funciona actualmente la gestión de personal, de qué manera fue desarrollado el sistema que



actualmente se explota en la Universidad, las ventajas y desventajas de las tecnologías, metodologías, lenguajes y herramientas a utilizar en el desarrollo del nuevo sistema. Además permite apreciar que es lo que realmente necesitan los clientes del sistema de acuerdo a las nuevas concepciones.

Entrevista: Se entrevistó al líder del proyecto y a algunos de los desarrolladores del sistema que actualmente se explota donde se obtuvo la mayor cantidad de información sobre los problemas que presenta dicho sistema, las posibles soluciones a estos problemas, los procesos que realizará el módulo de Gestión de Personal y todo lo nuevo que se le pretende incorporar. Además una secretaria docente y un cliente del sistema, brindaron información sobre cómo se llevan a cabo los procesos de Gestión de Personal. Este método tiene la ventaja de aportar conocimientos del tema que no se poseen.

El documento está estructurado en 3 capítulos:

Capítulo 1: Fundamentación Teórica: se muestra el resultado de la investigación bibliográfica sobre el objeto de estudio y los Sistemas de Gestión Académica, así como las tendencias que adquieren actualmente. Se detallan las tecnologías, metodologías, lenguajes y herramientas utilizadas, analizando sus características, desventajas y ventajas.

Capítulo 2: Características del Sistema: en dicho capítulo se describe el negocio, se describen las principales reglas del negocio, los actores y trabajadores del negocio, se realiza el modelo de casos de uso del negocio, se describen los requisitos funcionales y no funcionales que deberá cumplir el sistema y partiendo de los requisitos funcionales se realiza el diagrama de casos de uso del sistema, así como las descripción textual de dichos casos de uso. Además de que se realiza la estimación del esfuerzo por el método de análisis de Puntos de Casos de Uso.

Capítulo 3: Análisis y Diseño del Sistema: en el capítulo se modelan los diagramas de clases del análisis y del diseño, así como los diagramas de interacción correspondientes. Se hace referencia a la arquitectura y patrones utilizados para la construcción del diseño propuesto. Se muestra el modelo lógico y físico de la base de datos, así como el diagrama de despliegue.



CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.

1.1 Introducción

En este capítulo se muestra el resultado de la investigación bibliográfica sobre el objeto de estudio y los Sistemas de Gestión Académica en el ámbito nacional, internacional y la Universidad, así como las tendencias que adquieren actualmente. Se detallan las tecnologías, metodologías, lenguajes y herramientas utilizadas, analizando sus características, desventajas y ventajas.

1.2 Los Sistemas de Gestión Académica.

Realizar los procesos docentes de forma manual puede resultar una tarea compleja y tediosa debido al gran volumen de información involucrado. Afortunadamente, la incorporación de las tecnologías de la información ha dado la posibilidad de informatizar estos procesos mediante el empleo de sistemas automatizados para la gestión académica. Estos sistemas constituyen una poderosa herramienta de trabajo que permite hacer más eficientes los procesos docentes y optimizar sus recursos.

1.2.1 Los Sistemas de Gestión Académica en el Mundo.

Ágora

Ágora es un producto de software estándar de gran calidad para la gestión de Centros Docentes y Academias de todo tipo desarrollado por la empresa española Kherian Soft. Es un sistema completo, altamente configurable y flexible pues gestiona sin limitaciones los datos de alumnos, profesores, asignaturas, clases, aulas, grupos, matrículas, clientes. Brinda diferentes opciones de presentación y estética, se adapta fácilmente a las necesidades de sus usuarios y se adecua a cualquier tipo de centro o formación, ya sea esta de tipo oficial o de carácter libre (academias de enseñanza general, de idiomas, informática, música u oposiciones), ya se trate de un pequeño centro o una gran empresa de formación con una gestión centralizada de varias sucursales distantes geográficamente.

Ágora aporta soluciones en diversos sentidos, como son la gestión centralizada de datos a través de la cual se gestionan todas las bases de datos de alumnos, profesores, aulas y clases; la generación automática de la documentación que permite emitir todo tipo de documentación de entrega, como recibos y facturas, además de proporcionar listados de soporte de todo tipo.

Todos los datos manejados por la aplicación pueden filtrarse por cualquier criterio. Un sencillo asistente le permitirá construir la selección y guardarla para luego aplicarle el filtro que desee. Los



datos pueden igualmente ordenarse en cualquier momento por el campo que desee, del mismo modo, localizará inmediatamente un dato buscado con tan sólo introducir sus primeros caracteres.

Actualmente se encuentra disponible en 4 ediciones distintas que se adaptan a los distintos requerimientos o perfiles de sus usuarios: Edición Estándar, Edición Profesional, Edición Empresarial y Edición Corporativa. Cada una de ellas añade toda la potencia de la anterior e incluye nuevas funcionalidades y características.

Aún así, este sistema posee en general una apariencia compleja y poco amigable debido a la gran cantidad de funcionalidades que presenta y tiene como inconveniente el alto costo de licencia para su uso, mantenimiento y soporte, teniendo en cuenta que existen otros sistemas similares en el mundo que pueden obtenerse de forma gratuita o a más bajo precio.

SIGA (Sistema Integral de Gestión Académica)

Fue desarrollado por el Servicio de Informática de la Universidad de Córdoba, España, se encuentra en continua evolución, ampliando nuevas funcionalidades y adaptándose a las nuevas tecnologías, actualmente se utiliza en esta Universidad para la automatización de la gestión académica. La versatilidad y flexibilidad del sistema permite organizar la información de diversas instituciones en un espectro que abarca a conservatorios, academias, colegios, institutos y universidades, así como centros de formación de empresas, y eventos especiales como maestrías y cursos de postgrado.

Las funcionalidades soportadas por SIGA son muy amplias, abarcan desde la gestión del acceso a la Universidad hasta la tramitación de los títulos pasando por la matrícula, actas de examen, expedientes académicos, estadísticas, informes, gestión de becas, convalidaciones (1).

Posee varios módulos que se encargan de realizar todas estas funcionalidades, entre los que se encuentran: acceso, matrícula, automatrícula, planes de estudio, becas, cursos de verano, estudios propios, programas de intercambio, expedientes, actas, doctorado, convalidaciones y títulos.

El módulo matrícula permite la inscripción del estudiante en las asignaturas que desee y le estén permitidas, conforme a su expediente académico y a la normativa existente. Además permite actualizar datos estadísticos, solicitar grupos de clase, solicitud y renovación de la tarjeta universitaria, aplicar deducciones de matrícula y domiciliación del pago.

El módulo automatrícula es un nuevo servicio que la Universidad de Córdoba pone en marcha, que permite que los alumnos se matriculen fácilmente vía Internet sin necesidad de hacer cola, utilizando



para ello las aulas de informática de cada centro o incluso desde el ordenador de sus casas. Por supuesto, los alumnos que lo deseen podrán acudir en cualquier momento a la secretaría de su centro para matricularse como siempre.

Tiene como desventaja que su interfaz no es muy amigable lo cual puede provocar serios problemas de utilización teniendo en cuenta la diversidad de usuarios que interactúan con el sistema: alumnos, profesores, personal del centro, entre otros.

Akademios .Net

Esta aplicación está específicamente diseñada para el Centro de la Universidad Nacional de Educación a Distancia (UNED) de Barbastro y sirve para llevar un control completo de la organización académica del mismo. Fue desarrollada por el Departamento de Informática de este centro e incorpora varios módulos de gestión: asignaturas, alumnos, tutores y tutorías, así como un módulo para la gestión de las actividades de extensión universitaria de cada centro.

Módulo Alumnos:

Posibilita la gestión de los alumnos que obran en la bases de datos central. Se han incorporado filtros para poder observar todo tipo de estados, especialmente en el período de la matrícula. Dispone de un sistema de comunicaciones que permite el envío de correos electrónicos, correo telefónico (sms) o bien la confección de etiquetas postales.

Módulo Tutores:

Dispone de los datos de los tutores propios del centro con los datos personales, académicos, profesionales y de representatividad, proporcionado directamente de la base de datos central. Se incorpora la posibilidad de actualizar los datos de los tutores directamente desde los terminales autorizados de los centros.

Módulo Tutorías:

Permite la generación de las sesiones y su control. Así como la posibilidad de remitir la información de incidencias del profesorado a los alumnos del centro mediante correo electrónico o postal.

El sistema está desarrollado en el entorno .Net que proporciona la estructura Framework de Microsoft, presentado el inconveniente de ser software propietario, por tanto es necesario contar con una licencia para su uso. Además la aplicación no puede ser utilizada fuera del marco del centro para el cual fue desarrollada.



1.2.2 Los Sistemas de Gestión Académica en Cuba

GESTACAD

GESTACAD es un sistema informativo automatizado, desarrollado por el Departamento de Informática de la Universidad de Matanzas "Camilo Cienfuegos". Permite actualizar y mantener la información sobre estudiantes y profesores de una Universidad y obtener determinados resultados propios del trabajo de las áreas implicadas, aunque el grueso de las informaciones se obtiene mediante el acceso al sitio WEB de la Universidad.

En la actualidad el sistema es capaz de realizar diversas acciones y brindar numerosos reportes, los cuales son fruto de los requisitos funcionales del mismo. Estos son la búsqueda de un alumno, el listado de estudiantes por grupo, reportes dinámicos de la información existente, reportes de notas por asignatura y grupo, tablas con los resultados docentes de un grupo en un semestre, reportes de los resultados académicos de un estudiante en toda su carrera, actas de exámenes de las diferentes asignaturas, registros de características de un grupo de estudiantes, así como dar baja a los estudiantes.

El sistema consta de los siguientes módulos:

Administración:

Para la gestión de las tablas del sistema vía Web así como agregar nuevas consultas al sitio oficial y establecer los distintos niveles de acceso a estas.

Web para las Secretarías Docentes:

Para la Gestión de Estudiantes que permite hasta el momento la realización de acciones generales comunes en una Secretaría Docente así como la obtención de reportes oficiales.

Web para los Jefes de Departamentos Docentes:

Donde se incluyen acciones relativas como la asignación de la carga docente y el control sobre los profesores del Dpto.

Web para la Gestión de la Matrícula.

Donde se incluyen acciones relativas para llevar el control de los datos de todos sus estudiantes.

Web para los Profesores:

Donde estos pueden llevar el control docente de sus estudiantes, el control de las evaluaciones así como reportes relativos a su carga docente.

Un Sitio Web con Reportes en Línea:



Con la utilidad del registro docente para los profesores además de la búsqueda de estudiantes la cual devuelve, algunos datos personales del estudiante y su ubicación según el horario docente detallando aula, asignatura y tipo de clases que está recibiendo. Además de su estado y si se ha pasado asistencia en el turno de clase.

Dicho sistema tiene como desventajas que no brinda la posibilidad de contar con más de un plan de estudio y tampoco permite dar un seguimiento a las evaluaciones de una asignatura.

SIGENU

El Sistema de Gestión de la Nueva Universidad (SIGENU) es un software desarrollado en el Instituto Superior Politécnico “José Antonio Echeverría” (CUJAE) en el 2007 con el fin de ser una herramienta que permita la gestión de toda la información académica vinculada con la educación superior en Cuba. En correspondencia con su carácter nacional y la gran diversidad de sistemas de enseñanza superior con que cuenta la Universidad Cubana, este sistema ha sido concebido de manera tal que sea capaz de brindar gran seguridad e integridad de la información, y a la vez, tan flexible que permita ser adaptado a todos los centros de educación superior del país con sus diversas particularidades y distintas maneras de realizar determinados procedimientos.

Está compuesto por cuatro elementos fundamentales: Base de Datos, Servidor de Aplicaciones, Aplicación Cliente de Administración y Aplicación Cliente (SIGENU).

Esta última es una aplicación de escritorio que constituye el elemento que esencialmente permite la inserción y actualización de toda la información que se registre en el sistema. Además, permite obtener un conjunto importante de reportes muy usados cotidianamente en el mundo de la educación superior. Consta con los siguientes módulos: Codificadores, Matrícula, Control de estudiantes, Plan de Estudio, Evaluaciones y Reportes.

El módulo Matrícula se encarga de realizar el proceso a través del cual los nuevos estudiantes pasarán a ser registrados en el sistema como estudiantes de la educación superior. Tiene como particularidad que cuando una persona arriba a una secretaría para ser matriculada, se registra inicialmente como estudiante de nuevo ingreso y así es considerada hasta que finalice todo el proceso de matrícula con la operación cierre de matrícula. A partir de este momento, todos los nuevos ingresos pasan a ser estudiantes del sistema.



El módulo control de estudiantes permite buscar un estudiante registrado en el sistema, modificar sus datos tanto personales como docentes, ubicarlo en un grupo o cambiarlo de grupo, definir los que serán repitentes así como darle baja del centro ya sea por licencia de matrícula, resolución o traslado.

El sistema se limita a hacer la gestión de los datos de los estudiantes sin permitir la gestión de ningún dato referente a los profesores, ni las relaciones de estos con los estudiantes y grupos académicos. Su interfaz principal necesita ser instalada en cada computadora que se va a utilizar por ser esta una aplicación de escritorio, lo que ocasiona pérdidas de tiempo y limitaciones en su uso.

1.2.3 Los Sistemas de Gestión Académica en la Universidad de las Ciencia Informáticas

El Sistema para la Gestión Académica (GESTACAD) fue usado en la UCI en el curso 2002-2003, cuando la Universidad contaba con una matrícula de 2000 estudiantes y sólo un plan de estudio. Tenía muchas funcionalidades que en el comienzo fueron de mucha ayuda a la Universidad, sin embargo, el sistema no obedecía a la tónica del sistema docente empleado en esta, por lo que en el próximo curso este sistema dejó de tener validez, ya que la matrícula creció y se comenzó a usar más de un plan de estudio, además, se necesitaba seguir una trayectoria en las evaluaciones de los estudiantes.

UCIMAT

Fue el primer sistema que se utilizó en la Universidad para controlar algunos procesos docentes como la matrícula (en el primer curso se matricularon más de dos mil estudiantes), permitía consultar algunos reportes pero sólo los que traía por defecto de lo contrario había que ir directamente a la base de datos.

Akademos v1.0

En el año 2003 surge Akademos, el Sistema de Gestión Académica de la Universidad, el cual tiene como objetivo permitir la gestión automatizada de los elementos que intervienen en la labor académica de un centro de estudios, que pueda enfrentar los cambios de forma natural, adaptándose a las nuevas condiciones y formas de hacer con el menor costo. El mismo fue desarrollado en Visual Studio .NET 2003 utilizando SQL Server 2000 como gestor de base de datos y se diseñó teniendo en cuenta los siguientes principios:

- El dinamismo del proceso de gestión académica constituye la principal fuente de riesgos para un sistema que intente automatizarlo.



- Un sistema que automatice la gestión académica debe lograr que todos los involucrados (directivos, personal de secretaría, profesores y estudiantes) tengan un papel activo en el proceso.
- El plan de estudios es la entidad fundamental del proceso de gestión académica y rige todos sus subprocesos (matrícula, control, planificación).

El sistema está dividido en siete módulos, que abarcan la mayoría de los procesos involucrados en la gestión docente. Los mismos interactúan entre sí para llevar a cabo cada una de las tareas que automatiza el sistema.

A continuación se nombra y describen algunos de los módulos:

Módulo Matrícula:

Permite el control de los datos de los estudiantes que van a comenzar a ser parte de la Universidad, así como la gestión de los movimientos a que estos son sometidos en su paso por la misma. En los centros que aceptan cada año miles de estudiantes, la tarea de matricularlos puede convertirse en una labor titánica, si no se involucra a una gran cantidad de personas en la misma. Este módulo permite la descentralización del proceso de ingreso en un conjunto de personas tan grande como se decida y también es el encargado de definir la estructura del centro de estudios, la cual se concibe como un conjunto de estructuras de diferentes tipos: facultades, grupos docentes, que se organizan jerárquicamente para agrupar por un criterio determinado, tanto a los estudiantes como a los demás implicados en la labor docente, sean profesores, personal de secretaría o directivos.

Módulo Profesor:

Da la posibilidad de planificar la carga docente de los profesores pues es el encargado de asignar al mismo a un departamento y además asignarle grupos. Los profesores son la materia fundamental con que cuenta un centro de estudios para desarrollar con éxito su labor formativa.

Módulo Estudiantes:

Brinda a los estudiantes, principales protagonistas del proceso docente, un espacio destinado a brindarles información referida a su desempeño académico en su paso por la Universidad.

Sin embargo, el sistema está desarrollado completamente en software propietario, por lo que no se adecua a las políticas de software tanto de la Universidad como del país, ya que se ha optado por el software libre debido a la serie de ventajas que proporciona su uso. Posee muy poca documentación,



lo que hace muy difícil su mantenimiento y soporte. Además, muchas de las funcionalidades de los módulos matrícula y profesor se encuentran repetidas, perdiendo eficiencia.

1.2.4 Marco Teórico-Conceptual de la Investigación.

Después de realizar un estudio de los sistemas de gestión académica utilizados en el mundo y en el país, se pudo apreciar que a pesar de las amplias posibilidades que brindan en cuanto a la gestión académica, en general fueron diseñados según las particularidades de los centros donde son aplicados.

En el caso de los sistemas Akademos.net, SIGA, Ágora, se puede señalar que están desarrollados completamente en software propietario lo que no se adecua a las políticas de software seguidas por la universidad y el país, lo que trae consigo que algunos como Ágora tengan como inconveniente su alto costo.

Otros como SIGENU se limitan a hacer la gestión de los datos de los estudiantes sin permitir la gestión de ningún dato referente a los profesores. Y el sistema Akademos que actualmente se explota en la universidad, realiza parte de la gestión de los datos de los profesores del centro pero carece de una serie de funcionalidades que permitan llevar el control detallado de los mismos, además posee muy poca documentación por lo que se hace difícil su mantenimiento y soporte.

1.3 Tendencias actuales en el desarrollo de Software

“El software libre, es un movimiento tecnológico que ha revolucionado la sociedad. Presenta características especiales que han permitido la experimentación de nuevas formas de desarrollo y mantenimiento de programas, nuevos modelos económicos, y nuevas normas legales. Es un asunto de libertad, no de precio. Para entender el concepto, se debe pensar en “libre” como en “libertad de expresión” (2).

Software Libre se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. De modo más preciso, se refiere a cuatro libertades de los usuarios del software: La libertad de usar el programa, con cualquier propósito. La libertad de estudiar cómo funciona el programa, y adaptarlo a tus necesidades. La libertad de distribuir copias, con lo que puedes ayudar a otros. La libertad de mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie.



Para que las libertades de hacer modificaciones y de publicar versiones mejoradas tengan sentido, se debe tener acceso al código fuente del programa. Por lo tanto, la posibilidad de acceder al código fuente es una condición necesaria para el Software Libre.

Esto da la medida de la viabilidad económica de este sistema libre, que por ser libre no es necesariamente gratuito, sino que da la posibilidad de comercializarlo, regalarlo, prestarlo con total libertad y protegerlo legalmente. Evidentemente es la alternativa para los países subdesarrollados.

Ventajas:

- Costo: el costo total de propiedad del sistema operativo libre Linux es menos de la mitad que el de Windows. Gran parte del ahorro proviene de no tener que pagar licencia y de sus menores costos de administración.
- Innovación tecnológica: el desarrollo en comunidad de este sistema y el conocimiento del código fuente, propician que a cada instante, un desarrollador necesite nuevas actualizaciones y las realice él mismo, proponiendo nueva funcionalidad al programa.
- Escrutinio público: el proceso de revisión pública al que está sometido el desarrollo del software libre imprime un gran dinamismo al proceso de corrección de errores. Cada mejora es socializada libremente, la comunidad puede cambiar la realidad de las innovaciones.
- Independencia del proveedor: el Software Libre garantiza una independencia con respecto al proveedor gracias a la disponibilidad del código fuente.
- Desarrollo de la industria local: en el Software Libre al disponer del código fuente de la aplicación, es posible desarrollar internamente las mejoras o las modificaciones necesarias. De este modo, se contribuye a la formación de profesionales en nuevas tecnologías, al desarrollo local y de la industria nacional de software.
- Privacidad y seguridad: el Software Libre, por su carácter abierto, dificulta la introducción de código malicioso, espía o de control remoto, debido a que el código lo revisan muchos usuarios y desarrolladores que pueden detectar posibles puertas traseras. En el mundo del software libre, cualquier programador puede realizar una auditoría para comprobar que no se ha introducido ningún código malicioso, y, a su vez, cualquier entidad puede añadir libremente encriptación adicional a la aplicación que utilice para proteger sus datos.

Desventajas:

- La curva de aprendizaje es mayor: No es fácil aprender a trabajar con el software libre si ya se ha usado un software propietario. Dos personas que nunca han tocado una computadora



posiblemente tarden lo mismo en aprender a utilizar el software libre, que el software propietario, pero si ya ha usado un software propietario, generalmente se tarda más en aprender a usar el software libre.

- Se necesita dedicar recursos a la reparación de errores: El software libre se adquiere sin garantías explícitas, se vende tal cual, aunque pueden existir garantías específicas para determinadas situaciones, por lo que hay que dedicar recursos para reparar cualquier daño del software, sin embargo en el software propietario es imposible reparar errores, hay que esperar a obtener una nueva versión.

1.4 Metodologías, lenguajes y herramientas de modelado

1.4.1 Metodologías de desarrollo de software

Con el objetivo de crear y mantener las aplicaciones de software, aplicando las tecnologías y prácticas computacionales, surge la Ingeniería de Software. El desarrollo y evolución constante experimentada por los procesos de Ingeniería de Software, ha traído consigo la realización de varias tareas en este campo, como son: análisis de requisitos, especificación, diseño y arquitectura, programación, prueba, documentación y mantenimiento.

El proceso de desarrollo del software, define el conjunto de actividades precisas para convertir los requisitos de los usuarios en el conjunto seguro y resistente de artefactos que componen un producto de software. Las tendencias presentes, luego del perfeccionamiento de los procesos del software durante años, han llevado a cabo dos corrientes significativas: los llamados métodos ligeros y métodos pesados. Aunque ambos están enfocados a beneficiar la labor de aquellas personas que intervienen en el proceso de desarrollo.

Los métodos ligeros o ágiles, proponen mejorar la calidad del software teniendo como premisa la comunicación inmediata y directa, mientras que los métodos pesados obtienen sus resultados a través de orden y documentación.

Se hace necesario definir metodologías para guiar el proceso de desarrollo de un producto de software.

Las metodologías se definen por pasos a seguir para el cumplimiento de un objetivo. El objetivo dentro del desarrollo del software es producir un producto de alta calidad que cumpla con los requerimientos del cliente.



Entre las metodologías de desarrollo de software se pueden encontrar:

- XP (eXtreme Programming)
- FDD (Feature Driven Development)
- MSF (Microsoft Solution Features)
- RUP (Rational Unified Process)
- METRICA3
- SCRUM
- Crystal
- DSDM(Dynamic Systems Development Method)

Entre las características generales que presentan dichas metodologías se encuentran las siguientes:

- No pueden aplicarse a todo tipo de proyectos.
- Están orientadas en función de los nuevos principios de desarrollo del software.
- Pueden ser ajustables de acuerdo a las características del proyecto.

RUP

RUP (Proceso Unificado de Desarrollo de Software) es una metodología pesada. Describe detalladamente todas las actividades, roles, responsabilidades, productos de trabajo y herramientas para definir el quién, cómo, qué y cuándo en un proyecto de desarrollo de software.

Esta metodología unifica los mejores elementos de las anteriores, por lo que se encuentra preparada para el desarrollo de grandes y complejos proyectos. RUP, representa un ideal de referencia para todo el equipo de desarrollo. Es además, un proceso de desarrollo de software que contiene un conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software, se puede especializar para gran variedad de sistemas de software, distintas áreas de aplicación, tipos de organizaciones, niveles de actitud y tamaños de proyecto.

Se caracteriza por estar dividido en fases. En cada una de estas se producirán una o varias iteraciones, cuyo tamaño varía según la complejidad del proyecto. Dentro de cada una de ellas se seguirá un modelo de cascada en los flujos de trabajo que lo requieran.

Las fases de RUP concluyen con un hito bien definido, en cada uno de estos se deben tomar acuerdos y decisiones, garantizando el cumplimiento de los objetivos y metas antes de la transición a la nueva fase (3) .



Los hitos por cada una de las fases son:

- Inicio: visión de los objetivos.
- Elaboración: prototipo de la arquitectura.
- Construcción: capacidad operacional inicial.
- Transición: liberación del producto.

XP

Basándose en la simplicidad, la comunicación y la reutilización del código, surge la Programación Extrema (XP), siendo así una metodología ágil de desarrollo de software. Como requisito para alcanzar el éxito del proyecto se tiene al usuario final como parte del equipo. Es una metodología con reconocido éxito y se utiliza en proyectos con entregas a cortos plazos.

XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en la realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios (4).

Las características fundamentales de XP son:

- Desarrollo iterativo e incremental: que permite pequeñas mejoras, unas tras otras, consecutivamente.
- Pruebas unitarias: estas son continuas, frecuentemente repetidas y automatizadas, incluyendo pruebas de regresión.
- Programación en parejas: recomienda que las tareas de desarrollo se lleven a cabo por dos personas en un mismo puesto. Permitiendo que el código sea revisado al mismo tiempo que se escribe.
- Frecuente interacción del equipo de programación con el cliente o usuario: recomendando que un representante del cliente trabaje junto al equipo de desarrollo.
- Corrección de todos los errores antes de añadir una nueva funcionalidad: haciendo entregas frecuentes.
- Refactorización del código: reescribir ciertas partes del código para aumentar su legibilidad y mantenibilidad.
- Propiedad del código compartida: no se dividen las responsabilidades en el desarrollo de cada módulo. Este método promueve que todo el personal pueda corregir y extender cualquier parte



del proyecto. Las frecuentes pruebas de regresión garantizan que los posibles errores serán detectados.

- Simplicidad en el código: es más sencillo hacer algo simple y tener un poco de trabajo extra para cambiarlo si se requiere, que realizar algo complicado y quizás nunca utilizarlo.

Lo esencial en este proceso de desarrollo es lograr la comunicación entre desarrolladores y usuarios, la retroalimentación entre ellos, con los usuarios finales y la simplicidad en el código.

SCRUM

SCRUM define un marco para la gestión de proyectos, que se ha utilizado con éxito durante los últimos 10 años. Sus principales características se pueden resumir en dos. El desarrollo de software se realiza mediante iteraciones, denominadas sprint, con una duración de 30 días. El resultado de cada sprint es un incremento ejecutable que se muestra al cliente. La segunda característica importante son las reuniones a lo largo del proyecto, entre ellas destaca la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración. Aunque presenta otra serie de características como:

Características:

- Equipos auto dirigidos.
- Utiliza reglas para crear un entorno ágil de administración de proyectos.
- No prescribe prácticas específicas de ingeniería.
- Los requerimientos se capturan como ítems de la lista reserva del producto.
- El producto se construye en una serie de sprint de un mes de duración.
- Usado para proyectos complejos con requerimientos cambiantes.
- Basado en un control de proceso empírico.

Prácticas:

- Iteración (Sprint).
- Reunión de planificación de las iteraciones.
- Reuniones diarias.
- Reuniones de revisión de las iteraciones.
- Reuniones de revisión del diseño.
- Estabilización de las iteraciones.
- Meta Scrum.



Los proyectos que utilizan Scrum, generalmente poseen las siguientes características:

- Rápido cambio de requisitos.
- No mayor de 10 ingenieros.

SXP

Esta metodología surgió en Cuba, en la Universidad de las Ciencias informáticas (UCI) y no es más que la unión de XP y Scrum, para el logro de un buen desarrollo de software, propuesta en el 2007 por la ingeniera Malay Rodríguez Villar, y probada en los proyectos que trabajan con Software Libre, obteniendo buenos resultados. Esta metodología surgió con el nombre metodología ágil Gladys Marsi Peñalver Romero UNICORNIOS revisión 2 (MA-GMPR-UR2) la cual fue renombrada poco tiempo después.

La metodología está dividida en cuatro fases, que son precisamente la base de la estructura del nuevo expediente de proyecto, estas son:

- Planificación-Definición.
- Desarrollo.
- Entrega.
- Mantenimiento.

Cada una de estas fases está compuesta por una serie de actividades tales como escribir la visión y escribir la reserva del producto que son las que generan los artefactos como la plantilla de concepción del sistema y la lista de reserva del producto que quedan incluidos en el nuevo expediente de proyecto. Estas actividades están recogidas en el guión de la metodología.

Para la definición de los artefactos que se generan en cada una de las fases se tiene en cuenta como elemento fundamental, las características de las metodologías ágiles, las cuales tienen como premisa la no duplicación de esfuerzos, así como la integración del cliente en el equipo de desarrollo, esto garantiza que no haya necesidad de documentaciones extensas, sólo se documenta lo necesario para una futura reutilización.

1.4.2 Lenguaje de modelado

UML

El lenguaje unificado de modelado (UML) prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos. Es capaz de describir la semántica general de los diagramas y los significados de los símbolos utilizados. Actualmente es el más conocido y utilizado.



UML desarrolla un lenguaje gráfico para visualizar, especificar y documentar las variadas partes que incluye el desarrollo de software (5).

Las características más generales de UML son:

- Lenguaje de modelado orientación a objetos.
- Viabilidad en la corrección de errores.

1.4.3 Herramientas CASE para el modelado

Las aplicaciones informáticas que facilitan el trabajo dentro del ciclo de desarrollo del software son conocidas como herramientas CASE (Ingeniería de Software Asistida por Computadora) y se emplean para aumentar la productividad del desarrollo del software disminuyendo los tiempos de construcción y el costo de los mismos. La tecnología CASE supone la automatización del desarrollo del software, contribuyendo a mejorar la calidad en el desarrollo de sistemas de información.

Rational Rose

Rational Rose, es una herramienta para realizar el modelado visual, forma parte de un conjunto más amplio de herramientas que todas juntas abarcan el ciclo de vida del desarrollo de software. Permite completar diferentes disciplinas (flujos fundamentales) de RUP; incluye, a su vez, herramientas de ingeniería inversa y generación de código que facilitan el tránsito hasta el producto final.

Rational Rose, es considerado una de las mejores herramientas para traducir requisitos de alto nivel a una arquitectura basada en componentes. Se encuentra en la avanzada en cuanto al desarrollo con UML por lo que se ha convertido en una de las mejores opciones por la notación estándar que provee para especificar, visualizar y construir productos de software.

Visual Paradigm

Es un software privativo, para modelado en UML. Esta herramienta tiene unas características gráficas muy cómodas, que facilitan la realización de los diagramas de modelado que sigue el estándar de UML, los mismo son: diagramas de clase, casos de uso, comunicación, secuencia, estado, actividad, componentes.

Ventajas de Visual Paradigm:

- Entorno de creación de diagramas para UML 2.0.
- Diseño centrado en casos de uso y enfocado al negocio que genera un software de mayor calidad.



- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Capacidades de ingeniería directa e inversa.
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- Disponibilidad de múltiples versiones, para cada necesidad.
- Disponibilidad en múltiples plataformas.
- Producto de calidad.
- Soporta aplicaciones Web.
- Varios idiomas.
- Fácil de instalar y actualizar.
- Compatibilidad entre ediciones.

1.4.4 Fundamentación de la herramienta, lenguaje y metodología utilizadas para el modelado.

Dentro de las metodologías de desarrollo de software se considera que RUP es la más óptima para modelar el sistema debido a que representa grandes potencialidades para el análisis, implementación y documentación de sistemas orientados a objetos. Su característica de ser iterativo e incremental permitirá ir perfeccionando el software en cualquier momento del desarrollo teniendo en cuenta la variabilidad de las necesidades del cliente. Además de que permite mantener desde sus inicios una documentación profunda y detallada de todo el proceso, lo cual es de gran importancia a la hora de realizar proyectos complejos.

Como herramienta a utilizar para el modelado fue seleccionado Visual Paradigm, que aunque es un software propietario la universidad posee la licencia para su uso. Dicha herramienta tiene características gráficas muy cómodas, que facilitan la realización de los diagramas de modelado que sigue el estándar de UML. También posee cualidades como la generación automática de diagramas a partir de descripciones de casos de usos, permitiendo la agilidad en el trabajo del analista. Además usa un lenguaje estándar común para todo el equipo de desarrollo que facilita la comunicación.

Como lenguaje a utilizar para el modelado se escogió UML, ya que este permite visualizar, construir y documentar los artefactos de un proyecto de software. Además es el lenguaje que utiliza la herramienta escogida para realizar el modelado.



1.5 Arquitectura

3 capas

La programación por capas es un estilo de programación en el que el objetivo primordial es la separación de la lógica del negocio de la lógica de diseño. Su principal ventaja es que el desarrollo se puede llevar a cabo en varios niveles y, en caso de que sobrevenga algún cambio, sólo se ataca al nivel requerido sin tener que revisar entre código mezclado y permite distribuir el trabajo de creación de una aplicación por niveles; de este modo, cada grupo de trabajo está totalmente abstraído del resto de los niveles, de forma que basta con conocer la API que existe entre niveles.

Capa de presentación: es la que presenta el sistema al usuario, comunica y captura la información de este en un mínimo de procesos. Esta capa se comunica únicamente con la capa de negocio. También es conocida como interfaz gráfica y debe tener la característica de ser entendible y fácil de usar para el usuario.

Capa de negocio: Es donde residen las funcionalidades que debe realizar la aplicación, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio (e incluso de lógica del negocio) porque es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos almacenar o recuperar datos de él.

Capa de datos: Es donde residen los datos y es la encargada de acceder a los mismos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

Modelo Vista Controlador (MVC)

El modelo Vista Controlador 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. El patrón MVC se ve frecuentemente en aplicaciones web, donde la vista son las páginas HTML y el código que provee de datos dinámicos a la página; el modelo es la representación de los datos con que trabaja la aplicación y la lógica de negocio; y el controlador es el responsable de recibir los eventos de entrada desde la vista.



Modelo: es el objeto que representa los datos del programa. Maneja los datos y controla todas sus transformaciones. El modelo no tiene conocimiento específico de los controladores o de las vistas, ni siquiera contiene referencias a ellos. Es el propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre el modelo y sus vistas, y notificar a las vistas cuando cambia el modelo.

Vista: es el objeto que maneja la presentación visual de los datos representados por el modelo. Genera una representación visual del modelo y muestra los datos al usuario. Interactúa con el modelo a través de una referencia al propio modelo.

Controlador: es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el modelo. Cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información del modelo o por alteraciones de la vista. Interactúa con el modelo a través de una referencia al propio modelo.

1.5.1 Fundamentación de la arquitectura a utilizar

Para realizar el diseño del sistema se escogió el patrón de arquitectura MVC ya es el ideal para aplicaciones que manejan gran cantidad de datos y transacciones complejas. Al separar la lógica de los datos y la presentación será mucho más sencillo realizar múltiples representaciones de los mismos datos o agregar nuevos tipos de datos según sea requerido. Además, el código de la aplicación será más legible, lo que permitirá un mantenimiento más sencillo y ofrecerá maneras más simples de probar el correcto funcionamiento de esta.

1.6 Herramientas, tecnologías y lenguajes para el desarrollo

1.6.1 Frameworks

Este no es más que una estructura de soporte definida mediante la cual otro proyecto de software puede ser desarrollado y organizado. Puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto. Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio. Provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio.

Son diseñados con el intento de facilitar el desarrollo de software, permitiendo a los diseñadores y programadores pasar más tiempo identificando requerimientos de software que tratando con los



tediosos detalles de bajo nivel de proveer un sistema funcional. Es el esqueleto sobre el cual varios objetos son integrados para una solución dada. No es más que una base de programación que atiende a sus descendientes (manejado de una forma estructural y/o en cascada) posibilitando cualquier respuesta ante las necesidades de sus miembros, o secciones de una aplicación web.

Symfony

Symfony es un completo framework diseñado para optimizar el desarrollo de las aplicaciones web mediante algunas de sus principales características. 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 Microsoft SQL Server. Se puede ejecutar tanto en plataformas Unix, Linux, como en plataformas Windows. Este posee una serie de características que se enuncian a continuación:

Características:

- Fácil de instalar y configurar en la mayoría de las plataformas.
- Independiente del sistema gestor de bases de datos.
- Sencillo de usar en la mayoría de los casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos
- Basado en la premisa de “convenir en vez de configurar”, en la que el desarrollador solo debe configurar aquello que no es convencional.
- Sigue la mayoría de las mejores prácticas y patrones de diseño para la web.
- Preparado para aplicaciones empresariales y adaptables a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo.
- Código fácil de leer que incluye comentarios de phpDocumentor y que permite un mantenimiento muy sencillo.
- Fácil de extender, lo que permite su integración con las librerías de otros fabricantes.



Zend framework

Es un framework de código abierto orientado a componentes para desarrollar aplicaciones y servicios web con PHP 5.0. Está implementado utilizando código totalmente orientado a objetos. Su estructura de componentes es algo única; cada componente se diseña con pocas dependencias en otros componentes. Esta arquitectura débilmente acoplada permite que los desarrolladores utilicen componentes individualmente los cuales forman una potente y extensible aplicación web al ser combinados, todas estas ventajas que brinda y las demás que a continuación se mencionan hacen que dicho framework sea el más adecuado para programar la capa de negocio del sistema.

Ventajas:

- Acoplable a variadas tecnologías de desarrollo (adición de múltiples plugins externos).
- Cuenta con módulos para manejar archivos PDF, canales RSS, Servicios Web.
- Completa documentación y pruebas de alta calidad.
- Un buscador compatible con Lucene.
- Robustas clases para autenticación y filtrado de entrada.
- Clientes para servicios web, incluidos Google Data APIs y Strikelron.
- Marco de trabajo oficial de la comunidad PHP.
- Orientado a componentes.

Ext JS

Es un framework para JavaScript muy utilizado en el desarrollo de aplicaciones Web con AJAX. Tiene una librería inmensa que permite configurar las interfaces Web de manera semejante a aplicaciones escritorio. Tiene incluidos la mayoría de los controles de los formularios Web incluyendo Grids para mostrar datos y elementos semejantes a la programación de escritorio como los formularios, paneles, barras de herramientas, menús y muchos otros.

Dentro de su librería de componentes incluye componentes para el manejo de datos, lectura de XML, lectura de datos JSON e implementaciones basadas en AJAX. Presenta el uso de JavaScript con una programación orientada a objetos.

Actualmente se está haciendo muy popular en el desarrollo de aplicaciones web pues permite crear interfaces de usuarios muy amigables y funcionales de una manera intuitiva.



1.6.2 Object Relational Mapper

Doctrine

Doctrine es un potente y completo sistema Object Relational Mapper (ORM) para PHP 5.2 o superior, con un Database Abstraction layer (DBAL) incorporado. Entre muchas otras cosas tiene la posibilidad de exportar una base de datos existente a sus clases correspondientes y también a la inversa, es decir convertir clases (convenientemente creadas siguiendo las pautas del ORM) a tablas de una base de datos.

Su principal ventaja radica en poder acceder a la base de datos utilizando la programación Orientada a Objetos (POO) debido a que doctrine utiliza el patrón Active Record para manejar la base de datos, tiene su propio lenguaje de consultas y trabaja de manera rápida y eficiente. Es fácilmente integrable a los principales framework de desarrollo utilizados actualmente (6).

1.6.3 Lenguajes para el desarrollo Web

El desarrollo de aplicaciones Web ha tenido un auge en todo el mundo gracias a las ventajas que las mismas ofrecen a empresas e instituciones. Para el desarrollo de estas existen numerosos lenguajes informáticos que se dividen en dos grupos, el primer grupo abarca los lenguajes que corren en el lado del cliente y el segundo los lenguajes que corren en el lado del servidor. Las técnicas de desarrollo Web y lenguajes del lado del cliente más utilizados son HTML y JavaScript, mientras que los lenguajes de programación del lado del servidor más usados en software libre son Java, PHP y Perl.

PHP

PHP es un lenguaje de programación interpretado, el cual fue diseñado originalmente para la creación de páginas web dinámicas aunque últimamente también ha intervenido en la creación de aplicaciones con interfaces gráficas usando bibliotecas específicas.

PHP es un acrónimo recursivo que significa PHP Hypertext Pre-processor. Fue creado originalmente por Rasmus Lerdorf en 1994; sin embargo la implementación principal de PHP es producida ahora por The PHP Group y sirve como el estándar de facto para PHP al no haber una especificación formal. Publicado bajo la Licencia PHP, la fundación de software libre considera esta licencia como software libre.

Es capaz de combinarse con servidores de bases de datos tales como MySQL, PostgreSQL, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird y SQLite; lo cual permite la creación de aplicaciones Web robustas y, arroja resultados muy interesantes y prometedores para aquellas páginas que deseen ser



activas y dinámicas. La ejecución e interpretación de PHP es completamente en el servidor Web: en este se encuentra almacenado el script por lo que el cliente solo puede recibir el resultado de la ejecución. PHP puede ser ejecutado en la mayoría de los sistemas operativos, tales como UNIX, Linux, Mac OS X y Windows, a la vez es capaz de interactuar con servidores Web muy populares como ISAPI y Apache por existir también en versión CGI.

El uso de PHP presenta ventajas como:

- Es capaz de leer y manipular datos desde diversas fuentes, incluyendo datos que pueden ingresar los usuarios desde formularios HTML.
- Cuenta con una biblioteca sumamente amplia por defecto de funciones.
- No requiere definición de tipos de variables ni manejo detallado del bajo nivel.
- Tiene capacidad de conexión con la mayoría de los manejadores de base de datos utilizados en la actualidad.
- Posee una amplia documentación en su Web oficial de Internet. En la misma se encuentran muy bien explicadas todas las funciones del sistema, contando con ejemplos detallados.
- Pertenece a la alternativa de código abierto (Open Source), por lo que se presenta como una elección de fácil acceso para todos.
- Permite las técnicas de Programación Orientada a Objetos.
- Permite crear formularios para la Web.

JAVA

Es un lenguaje de programación sencillo, orientado a objetos, de propósito general e independiente de la plataforma de desarrollo, cuya sintaxis es muy parecida al lenguaje C o C++, fue creado por James Gosling de la compañía Sun Microsystems principios de los años 90.

Entre noviembre de 2006 y mayo de 2007, Sun Microsystems liberó la mayor parte de sus tecnologías Java bajo la licencia GNU GPL, de acuerdo con las especificaciones del Java Community Process, de tal forma que prácticamente todo el Java de Sun es ahora software libre.

Su principal ventaja es la independencia de plataforma. La cual se crea, y después de compilada una vez sirve en cualquier computador o máquina con el software de interpretación de Java adecuado, llamado Máquina Virtual Java.



Características.

- Dinámico
- Multihilo
- Produce Applets
- Portable
- Alto rendimiento
- Indiferente a la arquitectura
- Robusto
- Interpretado y compilado a la vez
- Seguro
- Distribuido

XML

Extensible Markup Language (XML) es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium. Es una simplificación y adaptación del Lenguaje de Marcado Generalizado (SGML) y permite definir la gramática de lenguajes específicos. Por lo tanto XML no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades.

Esta tecnología propone un estándar de datos que puede codificar el contenido, la semántica y los esquemas de una gran variedad de casos, desde los más simples a los más complejos. Se puede usar en bases de datos, editores de texto, hojas de cálculo y casi cualquier cosa imaginable. Aquí se busca dar solución al problema de expresar información estructurada de la manera más abstracta y reutilizable posible. Que la información sea estructurada quiere decir que se compone de partes bien definidas, y que esas partes se componen a su vez de otras partes.

Es una tecnología sencilla que tiene a su alrededor otras que la complementan y la hacen mucho más grande y con unas posibilidades mucho mayores. Tiene un papel muy importante en la actualidad ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil.

Ventajas:

- Es extensible: después de diseñado y puesto en producción, es posible extender XML con la adición de nuevas etiquetas, de modo que se pueda continuar utilizando sin complicación alguna.



- El analizador es un componente estándar, no es necesario crear un analizador específico para cada versión de lenguaje XML. Esto posibilita el empleo de cualquiera de los analizadores disponibles. De esta manera se evitan bugs y se acelera el desarrollo de aplicaciones.
- Si un tercero decide usar un documento creado en XML, es sencillo entender su estructura y procesarla. Mejora la compatibilidad entre aplicaciones.

HTML

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

Características: (7)

- Información por hipertexto: diversos elementos (texto o imágenes) de la información que se muestra en la pantalla están vinculados con otras informaciones que pueden ser de otras fuentes. Para mostrar en pantalla esta otra información bastará con hacer clic sobre ellos.
- Gráfico: en la pantalla aparece simultáneamente texto, imágenes e incluso sonidos.
- Global: se puede acceder a él desde cualquier tipo de plataforma, usando cualquier navegador y desde cualquier parte del mundo.
- Pública: toda su información está distribuida en miles de ordenadores que ofrecen su espacio para almacenarla. Toda esta información es pública y toda puede ser obtenida por el usuario.
- Dinámica: la información, aunque esta almacenada, puede ser actualizada por el que la publicó sin que el usuario deba actualizar su soporte técnico.
- Independiente: dada la inmensa cantidad de fuentes, es independiente y libre.

JavaScript

Fue creado en la empresa Netscape Communications, específicamente para su uso en el desarrollo de sitios Web. Es un lenguaje interpretado ya que no necesita de ningún compilador, solamente de un navegador, es orientado a objetos. JavaScript es sumamente utilizado en el mundo del desarrollo Web por ser muy versátil y potente, tanto para la realización de pequeñas tareas como para la gestión de



complejas aplicaciones. Con su utilización se puede evitar un poco la sobrecarga del lado del servidor ya que funciones como la validación de formularios, que antes se hacían del lado del servidor, se pueden realizar con este lenguaje y de una manera rápida.

Ajax

El término AJAX se presentó por primera vez en el artículo "Ajax: A New Approach to Web Applications" publicado por Jesse James Garrett el 18 de Febrero de 2005. Hasta ese momento, no existía un término normalizado que hiciera referencia a un nuevo tipo de aplicación web que estaba apareciendo.

“Ajax no es una tecnología en sí misma. En realidad, se trata de varias tecnologías independientes que se unen de formas nuevas y sorprendentes” (8).

Las tecnologías que forman AJAX son:

- XHTML y CSS, para crear una presentación basada en estándares.
- DOM, para la interacción y manipulación dinámica de la presentación.
- XML, XSLT y JSON, para el intercambio y la manipulación de información.
- XMLHttpRequest, para el intercambio asíncrono de información.
- JavaScript, para unir todas las demás tecnologías.

AJAX permite mejorar completamente la interacción del usuario con la aplicación, evitando las recargas constantes de la página, ya que el intercambio de información con el servidor se produce en un segundo plano.

Las aplicaciones construidas con AJAX eliminan la recarga constante de páginas mediante la creación de un elemento intermedio entre el usuario y el servidor. La nueva capa intermedia de AJAX mejora la respuesta de la aplicación, ya que el usuario nunca se encuentra con una ventana del navegador vacía esperando la respuesta del servidor.

1.6.4 Sistemas Gestores de Base de Datos.

Los sistemas de gestión de base de datos son un tipo de software muy específico, dedicado a servir de interfaz entre la Base de Datos, el usuario y las aplicaciones utilizadas. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. El propósito general de los sistemas de gestión de base de datos es el de manejar de manera clara,



sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante, para un buen manejo de datos. Entre los más utilizados están: PostgreSQL y MySQL.

MYSQL

El mismo es un sistema de bases de datos relacional, cuenta con más de seis millones de instalaciones, es multihilo, multiusuario, rápido y muy robusto. Inicialmente desarrollado por la compañía sueca MySQL AB la cual desde enero de 2008 es una subsidiaria de Sun Microsystems. Esta última desarrolla MYSQL como software libre en un esquema de licenciamiento dual.

MySQL es muy utilizado en los sistemas Web. Tiene gran popularidad por estar grandemente ligado a PHP. Es capaz de funcionar sobre múltiples plataformas, incluyendo AIX, BSD, FreeBSD, HP-UX, GNU/Linux, Mac OS X, NetBSD, Novell Netware, OpenBSD, OS/2 Warp, QNX, SGI IRIX, Solaris, SunOS, SCO OpenServer, SCO UnixWare, Tru64, eBD, Windows 95, Windows 98, Windows NT, Windows 2000, Windows XP, Windows Vista y otras versiones de Windows.

Características:

- Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
- Soporta gran cantidad de tipos de datos para las columnas.
- Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP).
- Gran portabilidad entre sistemas.
- Soporta hasta 32 índices por tabla.
- Gestión de usuarios y contraseñas manteniendo un muy buen nivel de seguridad en los datos.

PostgreSQL

Es un servidor de base de datos relacional orientado a objetos de software libre. Posee las características de los más potentes sistemas comerciales como Oracle o SQL Server, con la ventaja de que su licencia es gratuita. Como muchos otros proyectos open source, el desarrollo de PostgreSQL no es manejado por una sola compañía sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales. Dicha comunidad es denominada PostgreSQL Global Development Group (PGDG). Tiene más de 15 años de desarrollo activo y se ha ganado la reputación de ser confiable y mantener la integridad de los datos.

Ventajas:

- Alta concurrencia.
- Amplia variedad de tipos nativos



- Multiplataforma
- Extensible
- Estabilidad y Confiabilidad
- Instalación ilimitada
- Diseñado para ambientes de alto volumen

1.6.5 Fundamentación de las herramientas, tecnologías y lenguajes a utilizar para el desarrollo del sistema.

Entre los proyectos que lleva a cabo la dirección de informatización de la universidad se encuentra Akademos, dicha dirección tiene como objetivo fundamental estandarizar el uso de las herramientas, tecnologías y lenguajes en los diferentes proyectos que desarrolla.

Para el desarrollo se utilizará como lenguaje de programación PHP, pues está diseñado principalmente para el desarrollo web, controla el manejo de excepciones, posee una biblioteca nativa de funciones sumamente amplia e incluida y es uno de los lenguajes más utilizados actualmente. Además posee una amplia comunidad de desarrollo, la cual implementa constantemente mejoras en su código.

El ORM doctrine se utilizará como capa de abstracción de base de datos en la aplicación pues presenta un gran rendimiento en ejecución y permite escribir de forma concisa consultas muy complejas. Esto brindará a los desarrolladores una alternativa de gran alcance para SQL muy flexible sin requerir la duplicación innecesaria del código.

Para el desarrollo del sistema se utilizarán varios frameworks, entre ellos: Ext JS para programar la capa de presentación ya que permite crear interfaces de usuarios muy amigables y funcionales de una manera intuitiva, y Zend Framework para programar la capa de negocio, ya que su estructura de componentes es algo única, pues posibilita diseñar dicha estructura con pocas dependencias entre los componentes, esta arquitectura débilmente acoplada permite que los desarrolladores utilicen los diferentes componentes de forma individual, los cuales al ser combinados forman una potente y extensible aplicación Web.

Como sistema gestor de base de datos se utilizará PostgreSQL, ya que es uno de los gestores de bases de datos relacional orientado a objetos de software libre más utilizado en el mundo. Es capaz de manejar una enorme cantidad de datos, permitiendo el acceso simultáneo de un conjunto de usuarios, además brinda seguridad y estabilidad a los mismos y facilita el trabajo con procedimientos almacenados y consultas.



1.7 Conclusiones

En dicho capítulo se desarrolló una amplia investigación sobre el estado del arte de los sistemas de gestión académica y su comportamiento, tanto en el ámbito nacional como internacional, llegando a la conclusión de que ninguno de ellos puede ser usado en la Universidad ya que en general fueron diseñados según las particularidades de los centros donde son aplicados, y están desarrollados en software propietario lo que no se adecua a las políticas de software seguidas por la universidad y el país. Además de hacer un análisis profundo de todas las tecnologías, lenguajes, metodologías y herramientas a utilizar para el desarrollo del sistema. Para dar cumplimiento a estas tareas se tuvo en cuenta algunos métodos de la investigación como el histórico-lógico y el analítico-sintético. Después de realizados estos análisis se llega a la conclusión que se utilizará como lenguaje para el desarrollo: PHP; como lenguaje de modelado: UML; como sistema gestor de bases de datos: PostgreSQL; como metodología de desarrollo: RUP; como herramienta de modelado: Visual Paradigm; como frameworks para el desarrollo: Ext JS y Zend framework y como ORM: Doctrine.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

2.1 Introducción:

En el presente capítulo se realizará la descripción del negocio actual, se describen las principales reglas del negocio a tener en cuenta, se realiza el modelo de casos de uso del negocio, la descripción de los actores y trabajadores del negocio, se describen los requisitos funcionales y no funcionales que deberá cumplir el sistema, y a partir de la descripción de los requisitos funcionales se realiza el modelo de casos de uso del sistema, así como la descripción de los mismos. Además de que se realizará la estimación del esfuerzo mediante el método de análisis de Puntos de Casos de Uso.

2.2 Descripción del Negocio:

Debido a las nuevas exigencias de la educación superior se da la necesidad de desarrollar una nueva versión del Sistema de Gestión Académica (Akademos), este nuevo sistema contará con varios módulos entre ellos Gestión de Personal, el cual se encargará de llevar el control de todo el personal docente de la Universidad tales como estudiantes y profesores.

Todos los procesos que se explican a continuación tienen algo en común que no es más que para realizar cualquiera de ellos hay que llenar una serie de documentos oficiales que luego se archivan en el expediente académico del estudiante y en el caso del profesor en el expediente laboral.

Todo aquel que desee estudiar en una Universidad por lo general se presenta a las pruebas de ingreso, de actitud o cualquier otro tipo de examen que requiera hacer, si aprueba estos exámenes pasa a ser parte de los estudiantes de nuevo ingreso de la Universidad, cuando se presentan en el centro y se les toma la foto pasan a ser pre matrícula de la misma, en esta etapa solo se conocen los datos básicos del futuro estudiante, tales como nombre y apellidos, sexo y datos del centro de procedencia. Después que es pre matrícula, se procede a ubicar en una estructura, es decir la facultad y el grupo en el cual estará ubicado.

Luego se presenta a hacer oficial su matrícula, en esta etapa es donde la Universidad completa todos los datos que le hace falta conocer como dirección particular, carnet de identidad y datos familiares. En caso de que un estudiante que sea pre matrícula no se presente a hacer la matrícula ya no se considera estudiante del centro y se quitará de la estructura en la cual se ubicó, en el caso de que llegue después del plazo fijado la secretaria general podrá realizarle la matrícula y ubicarlo en una estructura.

Todo estudiante que desee continuar en el centro debe presentarse en la secretaría docente de su facultad a ratificar matrícula en el período que el centro universitario lo estime pertinente, deberá presentar su carnet de identidad para que la secretaria docente de su facultad llene el modelo de ratificación de matrícula, en el cual se recogen los datos básicos como nombre, apellidos, dirección particular y carnet de identidad.

En el caso de los traslados el estudiante debe presentarse en la facultad y realizar la solicitud, estos pueden ser internos y externos, en el caso de que sea interno, que es hacer un cambio de facultad o de grupo dentro de la facultad a la cual pertenece; debe presentarse en la facultad y solicitar el cambio de grupo, si el decano de la facultad autoriza el cambio la secretaria docente lo realiza.

En el caso de que el traslado sea de facultad, debe presentarse en la misma y realizar la solicitud, presentando para ello una carta donde expone los motivos por los cuales quiere realizar el cambio, la carta debe ser aprobada por los decanos de las dos facultades, en caso de que estos aprueben la solicitud, el decano de la facultad a la cual pertenece el estudiante emite una resolución aprobando el traslado, la secretaria docente informa a la secretaria general y ella es la encargada de realizar el cambio.

En el caso de que el traslado sea externo, después de que el estudiante se presente en la facultad y solicite el traslado, presentando una carta donde expone los motivos por los cuales realiza dicha solicitud; el decano es el encargado de aprobar el traslado, para esto debe comprobar una serie de datos del estudiante como son las notas, su comportamiento y las causas por las cuales solicita el traslado; si el decano aprueba el traslado, la secretaria docente de la facultad se lo hace saber a la secretaría general y ella es la encargada de presentárselo al rector; si el rector aprueba dicho traslado emite una resolución rectoral, este documento es entregado a la secretaria general que es la encargada de la gestión que se realiza en otra Universidad, es decir, el centro le presenta a la otra Universidad una copia del expediente del estudiante y estos atendiendo a una serie de factores como la capacidad de la facultad para donde se va a realizar el traslado y la trayectoria lo aceptan o no; en el caso de que la otra Universidad acepte el traslado, la secretaria general archiva la resolución que aprueba el traslado en el expediente, el cual se envía para la otra Universidad.

En el caso de la licencia, el estudiante debe presentarse en la facultad y solicitar la licencia, para ello debe presentar una carta en la cual expone las causas por las cuales realiza dicha solicitud, si el decano aprueba la licencia, la secretaria le entrega al estudiante el modelo de tramitación de baja para que este entregue todos los medios que posee de la Universidad, cuando llena dicho documento lo

entrega a secretaría y la secretaria docente es la encargada de enviarle a la secretaria general todos los documentos del estudiante.

En el caso de las bajas, el estudiante se presenta en la facultad y solicita la baja, presentado una carta donde expone los motivos por los cuales realiza la solicitud, si el decano aprueba la baja, la secretaria le hace saber al estudiante que sí se le aprobó la baja y le entrega el documento de tramitación de baja para que este entregue todos los medios que posee de la Universidad, cuando llena dicho documento lo entrega a secretaría y la secretaria docente informa a la secretaria general de la baja, y es la secretaria general quien le da baja al estudiante.

En el caso de que el estudiante desee reingresar al centro de estudios después de haber causado baja del mismo, se debe presentar en secretaría general y solicitar su reingreso, la secretaria general informa al rector y este es el encargado de aprobar el reingreso, verificando una serie de aspectos antes como son el expediente, el comportamiento antes de darle la baja y los motivos por los cuales causó baja del centro, en caso que se apruebe el reingreso el rector emite una resolución rectoral y la secretaria general lo matricula y lo ubica en una facultad y en un grupo. Este puede ser ubicado en la misma facultad en la que estaba antes de darle la baja.

Puede darse el caso de que no se le apruebe al estudiante la baja, la licencia, el traslado o el reingreso, en estos casos la secretaria le informaría al estudiante que no se le aprobó lo solicitado y las causas por las cuales no se le aprueba.

Cuando el estudiante egresa, es decir culmina sus estudios se presenta en la secretaria docente y la secretaria le entrega el documento de tramitación de baja para que este entregue todos los medios que posee de la Universidad, cuando llena dicho documento lo entrega a secretaría y la secretaria es la encargada de entregarle su certificación de notas.

En el caso de los profesores, estos se presentan en la Dirección de Recursos Humanos (RRHH) del centro universitario a hacer su contrato laboral, y posteriormente estos solicitan la ubicación. Los decanos de cada una de las facultades reciben el listado de los profesores que fueron ubicados en dicha facultad, y procede a realizarle una entrevista a cada uno de ellos para conocer una serie de datos que le hace falta tener en cuenta a la hora de ubicarlos, luego el profesor es ubicado en un departamento. Posteriormente el profesor se presenta en el departamento que fue ubicado para que le informen que asignatura va a impartir.

En el caso de que un profesor desee trasladarse de facultad, se dirige al decano de la facultad y solicita el traslado, presentando para ello una carta donde expone las causas por las cuales desea trasladarse, si el decano aprueba el traslado y el decano de la otra facultad aprueba que el profesor pertenezca a su facultad, el jefe de departamento es el encargado de asignarle la asignatura que ese profesor impartía a otro profesor de su departamento.

En el caso que el profesor solicite la baja debe presentarle la carta de solicitud a su jefe inmediato superior y este la eleva a los superiores de la estructura a la cual pertenece el profesor, el máximo responsable de la estructura, en el caso de las facultades el decano, es quién aprueba la baja del profesor y el jefe de departamento es el encargado de asignarle la asignatura que ese profesor impartía a otro profesor de su departamento. Después el profesor se presenta en la Dirección de Recursos Humanos (RRHH) de la Universidad y acaba de tramitar su baja del centro.

En el caso de que un profesor solicite licencia por motivos específicos deberá entregarle una carta al jefe inmediato superior donde le exponga el motivo de la licencia, este es el encargado de aprobar la licencia o no; si aprueba la licencia se le informa al profesor que se le aprobó la misma y se asigna otro profesor a impartir la asignatura que él impartía.

2.3 Reglas del Negocio

Para garantizar las restricciones que existen en el negocio se debe tener en cuenta, a la hora de automatizar los procesos de Gestión de Personal las siguientes reglas:

- Para realizarle la matrícula a un estudiante debe estar prematriculado.
- Para darle baja, licencia o traslado a un estudiante el mismo debe estar matriculado.
- La ubicación del personal ocurre al comenzar cada período del curso escolar.
- El Jefe de Departamento es el encargado de asignarle al profesor la asignatura que va a impartir.
- Para que un estudiante pueda solicitar el reingreso debe haber sido baja anteriormente.

2.4 Actores del Negocio

Actores	Descripción
Estudiante	Representa a todos los estudiantes de la Universidad, y puede solicitar matrícula, baja, traslado, reingreso o licencia del centro de estudios. Además de ser el mayor beneficiado con los procesos de pre matrícula y egreso.
Profesor	Representa a todos los profesores de la Universidad, y puede solicitar



	ingreso, traslado de facultad o baja del centro de estudios.
--	--

Tabla 2.1: Actores del Negocio

2.5 Trabajadores del Negocio

Trabajadores	Descripción
Secretaria General	Es la encargada de realizar el proceso de pre matrícula y reingreso de los estudiantes, así como parte de los procesos de baja, traslado y licencia de los mismos.
Secretaria docente	Es la encargada de realizar parte de los procesos de matrícula, baja, traslado, licencia, egreso, así como de realizar el proceso de ratificación de matrícula de los estudiantes. Además de que realiza parte de los procesos de ingreso, baja, licencia y traslado de los profesores.
Jefe de departamento	Es el encargado de asignarle a un profesor la asignatura que va a impartir.
Decano	Es el encargado de analizar y aprobar las solicitudes de baja, traslado y licencia de los estudiantes y profesores de su facultad.
Rector	Es el encargado de analizar y aprobar las solicitudes de reingreso a la Universidad y traslado hacia otra Universidad de los estudiantes.

Tabla 2.2: Trabajadores del Negocio

2.6 Diagrama de Casos de Uso del Negocio

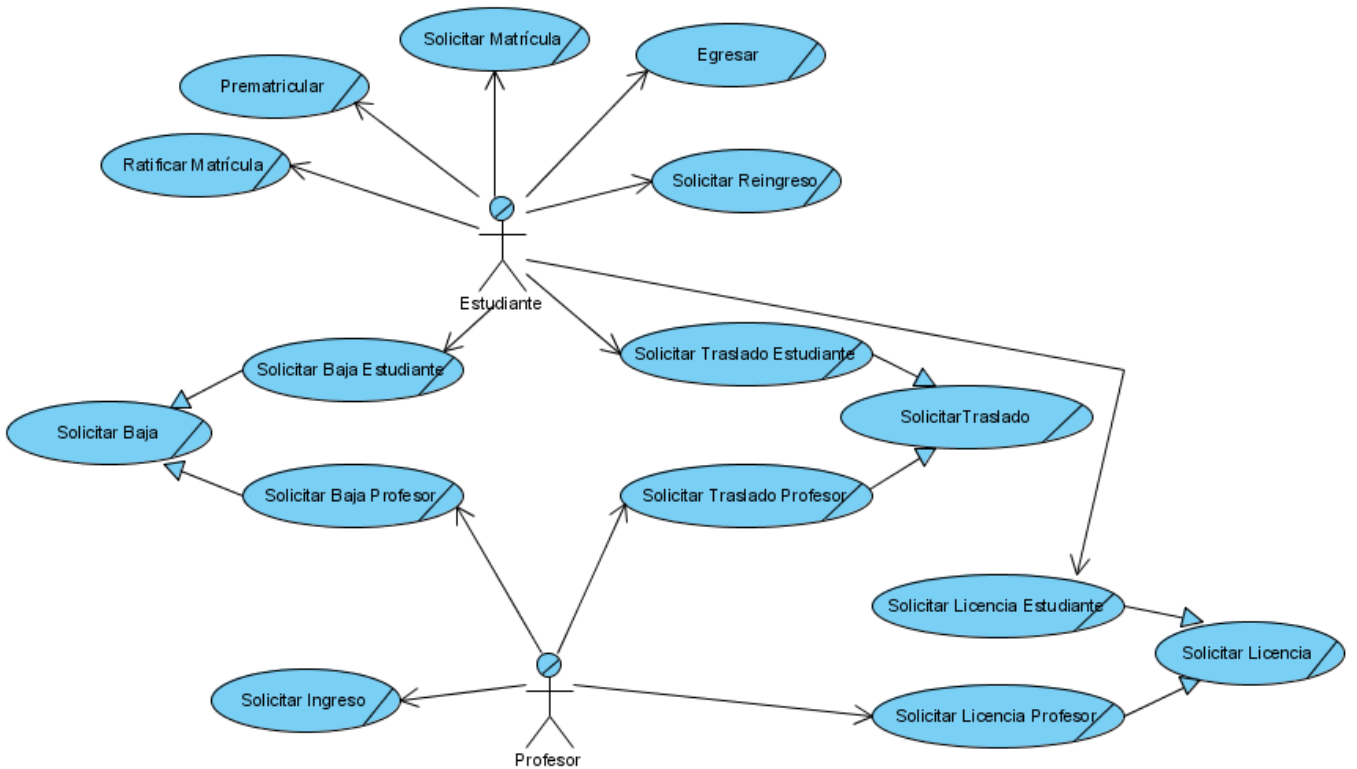


Figura 1: Diagrama de Casos de Uso del Negocio

2.7 Diagramas de Actividades de los Casos de Uso del Negocio

Los diagramas de actividades de los Casos de Uso del Negocio se encuentran en el Anexo #1.

2.8 Descripción de los Casos de Uso del Negocio.

CUN: Prematricular

Caso de Uso:	Prematricular
Actores:	Estudiante
Trabajadores:	Secretaria General.

Resumen:	Inicialmente el estudiante se somete a una prueba (Test) en la cual debe poner el nombre, Carné de Identidad, Centro de Estudio, posteriormente el examen realizado es llevado a la Universidad donde se revisa y se determina si el estudiante puede o no ingresar al centro, mediante un escalafón que se conforma teniendo en cuenta las notas del test. En caso de que el estudiante sea aprobado para entrar a la Universidad sus datos son recogidos en una plantilla de Ingreso que acredita al estudiante como ingreso al centro pero no como matrícula aún, luego que el estudiante llega al centro, se presenta a tirarse la foto y entonces pasa a ser pre matrícula del centro, donde se ubica en una estructura.	
Precondiciones:	Test realizado	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Negocio	
1. El estudiante realiza el test de ingreso al centro.	2. Si el estudiante aprueba el test, se realiza el escalafón. 3. Si el estudiante forma parte del escalafón, se llena la hoja de ingreso.	
4. El estudiante se presenta a tirarse la foto.	5. Se realiza la pre matrícula. 6. Se ubica en una facultad y en un grupo.	
Flujos Alternos		
Acción del Actor	Respuesta del Negocio	
	2.1. Si el estudiante no aprueba el test, se le informa que no puede ser ingreso del centro.	
	3.1. Si el estudiante no forma parte del escalafón se le informa que no puede ser ingreso del centro.	
Poscondiciones		

Tabla 2.3: Descripción del CUN Prematricular

CUN: Solicitar Matrícula

Caso de Uso:	Solicitar Matrícula.
Actores:	Estudiante
Trabajadores:	Secretaria Docente.
Resumen:	El estudiante después que es pre matrícula del centro se presenta en la secretaría docente de su facultad y solicita la matrícula, aquí se le llena la hoja de matrícula, el estudiante firma su hoja de matrícula y la secretaria archiva sus documentos, después de terminada la matrícula se pueden reorganizar los grupos

	administrativos.
Precondiciones:	El estudiante tiene que ser pre matrícula del centro.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Negocio
1. El estudiante se presenta y solicita matrícula.	2. La secretaria comprueba que sea pre matrícula del centro y le llena la hoja de matrícula.
3. El estudiante firma dicho documento.	4. La secretaria docente archiva los documentos.
Flujos Alternos	
Acción del Actor	Respuesta del Negocio
	2.1. Si el estudiante no es pre matricula del centro, la secretaria le informa que no puede realizarle la matrícula.
Poscondiciones	

Tabla 2.4: Descripción del CUN Solicitar Matrícula

CUN: Ratificar Matrícula

Caso de Uso:	Ratificar Matrícula
Actores:	Estudiante
Trabajadores:	Secretaria Docente.
Resumen:	El estudiante se presenta en la secretaría docente de su facultad una vez por curso, en el tiempo que la Universidad estime a ratificar su matrícula, debe presentar el carnet de identidad para que la secretaria llene el modelo de ratificación de matrícula, luego la secretaria archiva el modelo en el expediente del estudiante.
Precondiciones:	El estudiante tiene que ser matrícula de la Universidad.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Negocio
1. El estudiante se presenta a ratificar matrícula.	2. Se llena el modelo de ratificación de matrícula.
3. Revisa modelo de ratificación de matrícula.	4. La secretaria docente archiva el documento.
Flujos Alternos	
Acción del Actor	Respuesta del Negocio
3.1. Le comunica a la secretaria que el modelo está incorrecto.	3.2. Rectifica el modelo y vuelve al paso 3 del flujo normal de eventos.
Poscondiciones	

Tabla 2.5: Descripción del CUN Ratificar Matrícula

CUN: Solicitar Reingreso

Caso de Uso:	Solicitar Reingreso	
Actores:	Estudiante	
Trabajadores:	Secretaria General, Rector.	
Resumen:	Inicialmente el estudiante se presenta en la secretaría general de la Universidad y solicita el reingreso, presentando para ello una carta, la secretaria presenta dicha carta al rector y este aprueba o no el reingreso. Si le aprueban el reingreso al estudiante la secretaria general le llena la hoja de matrícula y lo ubica en una estructura.	
Precondiciones:	Tiene que haber sido matrícula del centro anteriormente.	
Flujo Normal de Eventos		
	Acción del Actor	Respuesta del Negocio
	1. El estudiante se presenta a solicitar reingreso.	2. La secretaria general le entrega al rector la solicitud de reingreso. 3. El rector aprueba el reingreso. 4. El rector emite resolución rectoral. 5. El rector entrega documentos a la secretaria general. 6. La secretaria general le llena la hoja de matrícula al estudiante.
	7. El estudiante firma la hoja de matrícula.	8. La secretaria general ubica al estudiante en una estructura. 9. La secretaria general manda los documentos del estudiante para la estructura donde fue ubicado.
Flujos Alternos		
	Acción del Actor	Respuesta del Negocio
		3.1. Si el rector no aprueba el reingreso informa a la secretaria general y esta informa al estudiante.
Poscondiciones		

Tabla 2.6: Descripción del CUN Solicitar Reingreso

CUN: Egresar

Caso de Uso:	Egresar
Actores:	Estudiante

Trabajadores:	Secretaria Docente.	
Resumen:	Es cuando el estudiante termina sus estudios, es decir ya ha vencido todas las asignaturas y discutió su tesis. El estudiante se presenta en la secretaria docente de su facultad a recoger el modelo de tramitación de baja para entrega los medios que posee de la Universidad, después de llenado el modelo se lo entrega a la secretaria y esta le entrega su certificación de notas.	
Precondiciones:	Tiene que haber terminado estudios.	
Flujo Normal de Eventos		
Acción del Actor		Respuesta del Negocio
1. El estudiante se presenta a recoger el modelo de tramitación de baja.	2. La secretaria docente le entrega dicho modelo.	
3. El estudiante entrega a la secretaria el modelo.	4. La secretaria docente archiva el modelo. 5. La secretaria docente imprime la certificación de notas del estudiante y se la entrega.	
6. Recibe la certificación de notas.		
Flujos Alternos		
Acción del Actor		Respuesta del Negocio
Poscondiciones		

Tabla 2.7: Descripción del CUN Egresar

CUN: Solicitar Ingreso

Caso de Uso:	Solicitar Ingreso	
Actores:	Profesor	
Trabajadores:	Decano, Jefe de Departamento.	
Resumen:	El profesor después de haber realizado su contrato en la dirección de recursos humanos se presenta en la facultad para que lo ubiquen, el decano le realiza una entrevista para saber donde lo va a ubicar, el caso de uso termina cuando el profesor es ubicado en un departamento y se informa que asignatura va a impartir.	
Precondiciones:	Tiene que haber realizado su contrato laboral.	
Flujo Normal de Eventos		
Acción del Actor		Respuesta del Negocio
1. El profesor se presenta donde fue contratado y solicita su ubicación.	2. El decano le realiza una entrevista al profesor con el objetivo de conocer su especialidad y experiencia laboral.	

<p>3. El profesor presenta su título y currículum.</p>	<p>4. El decano verifica el título del profesor y lee su currículum. 5. El decano consulta el listado de profesores por departamento y ubica al profesor en un departamento dependiendo de los resultados de la entrevista y de las necesidades de cada departamento. 6. El decano actualiza el listado de profesores por departamentos. 7. El decano le orienta al profesor que vea al jefe de departamento donde fue ubicado.</p>
<p>8. El profesor se presenta en el departamento donde fue ubicado para que le informen que asignaturas va a impartir.</p>	<p>9. El jefe de departamento consulta el listado de profesores por asignaturas y le asigna las asignaturas a impartir. 10. El jefe de departamento actualiza el listado de profesores por asignaturas.</p>
Flujos Alternos	
Acción del Actor	Respuesta del Negocio
Poscondiciones	

Tabla 2.8: Descripción del CUN Solicitar Ingreso

CUN: Solicitar Traslado Estudiante

Caso de Uso:	Solicitar Traslado Estudiante
Actores:	Estudiante.
Trabajadores:	Decano, Secretaria General, Secretaria Docente, Rector.
Resumen:	Inicialmente el estudiante realiza una solicitud en la cual expone los motivos por los cuales desea trasladarse. Luego se realizaran un flujo de actividades en dependencia del traslado que solicite el estudiante. El caso de uso termina cuando se realiza el traslado.
Precondiciones:	Es estudiante tiene que ser matrícula de la Universidad.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Negocio
<p>1. El estudiante entrega la solicitud de traslado.</p>	<p>2. El decano de la facultad recibe la solicitud de traslado y la lee. 2.1 En caso de que el traslado sea de grupo ver sección: "Solicitar Traslado Estudiante Grupo".</p>

	<p>2.2 En caso de que el traslado sea de facultad ver sección: “Solicitar Traslado Estudiante Facultad”.</p> <p>2.3 En caso de que el traslado sea de Universidad ver sección: “Solicitar Traslado Estudiante Universidad”.</p>
Flujo normal de eventos	
Sección: “Solicitar Traslado Estudiante Grupo”.	
Acción del Actor	Respuesta del Negocio
1. El estudiante entrega la solicitud de traslado.	<p>2. El decano de la facultad recibe la solicitud de traslado y la lee.</p> <p>3. El decano aprueba la solicitud de traslado.</p> <p>4. La secretaria docente cambia al estudiante de grupo.</p> <p>5. La secretaria docente actualiza el expediente del estudiante.</p> <p>6. La secretaria docente actualiza el listado de traslados de la facultad.</p>
Flujo Alterno 1:	
Acción del Actor	Respuesta del Negocio
	3.1. El decano informa al estudiante la no aprobación de la solicitud de traslado.
Flujo normal de eventos	
Sección: “Solicitar Traslado Estudiante Facultad”.	
Acción del Actor	Respuesta del Negocio
1. El estudiante entrega la solicitud de traslado.	<p>2. Los decanos reciben la solicitud de traslado y la leen.</p> <p>3. Los decanos aprueban la solicitud de traslado.</p> <p>4. La secretaria docente envía los documentos del estudiante a la secretaria general.</p> <p>5. La secretaria general ubica al estudiante en la facultad que solicitó.</p> <p>6. La secretaria general actualiza el expediente del estudiante.</p> <p>7. La secretaria general actualiza el listado de traslados de la Universidad.</p> <p>8. La secretaria general le informa a la secretaria</p>

	<p>docente del traslado.</p> <p>9. La secretaria actualiza el listado de traslados de la facultad.</p>
Flujo Alterno 1:	
Acción del Actor	Respuesta del Negocio
	3.1. Los decanos le informan al estudiante de la no aprobación de la solicitud de traslado.
Flujo normal de eventos	
Sección: "Solicitar Traslado Estudiante Universidad".	
Acción del Actor	Respuesta del Negocio
1. El estudiante entrega la solicitud de traslado.	<p>2. El decano de la facultad recibe la solicitud de traslado y la lee.</p> <p>3. El decano aprueba la solicitud de traslado.</p> <p>4. La secretaria docente envía los documentos del estudiante a la secretaria general.</p> <p>5. La secretaria general envía la solicitud y la resolución del decano aprobando el traslado al rector.</p> <p>6. El rector aprueba la solicitud de traslado.</p> <p>7. El rector emite una resolución aprobando la solicitud de traslado.</p> <p>8. El rector le entrega a la secretaria general los documentos del estudiante.</p> <p>9. La secretaria general informa a la secretaria docente.</p> <p>10. La secretaria docente le entrega al estudiante el modelo de tramitación de baja para que este entregue todos los medios que posee de la Universidad.</p>
11. El estudiante entrega los medios y le entrega el modelo a la secretaria docente.	<p>12. La secretaria general envía todos los documentos del estudiante para otra Universidad.</p> <p>13. La secretaria general actualiza el listado de traslados de la Universidad y le informa a la secretaria docente.</p> <p>14. La secretaria docente actualiza el listado de traslados de la facultad.</p>
Flujo Alterno 1:	
Acción del Actor	Respuesta del Negocio

	3.1. El decano informa al estudiante de la no aprobación de la solicitud de traslado.
	6.1. El rector informa a la secretaria general de que no aprobó el traslado y esta informa al estudiante.
Poscondiciones:	

Tabla 2.9: Descripción de CUN Solicitar Traslado Estudiante

CUN: Solicitar Traslado Profesor

Caso de Uso:	Solicitar Traslado Profesor	
Actores:	Profesor.	
Trabajadores:	Decano, Jefe de Departamento.	
Resumen:	Inicialmente el profesor realiza una solicitud en la cual expone los motivos por los cuales desea trasladarse. Luego se realizaran un flujo de actividades para realizar el mismo. El caso de uso termina cuando se realiza el traslado.	
Precondiciones:	El profesor tiene que ser ingreso de la Universidad.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Negocio	
1. El profesor entrega la solicitud de traslado.	2. Los decanos reciben la solicitud de traslado y la leen. 3. Los decanos aprueban la solicitud de traslado. 4. El jefe de departamento le elimina al profesor la o las asignaturas que impartía.	
5. El profesor se presenta en la otra facultad para ser ubicado.		
Flujos Alternos		
Acción del Actor	Respuesta del Negocio	
	3.1. Los decanos informan al profesor de la no aprobación de la solicitud de traslado.	
Poscondiciones		

Tabla 2.10: Descripción de CUN Solicitar Traslado Profesor

CUN: Solicitar Licencia Estudiante

Caso de Uso:	Solicitar Licencia Estudiante
Actores:	Estudiante.
Trabajadores:	Decano, Secretaria General, Secretaria Docente.

Resumen:	El caso de uso inicia cuando el estudiante se presenta en el decanato solicitando licencia, este es atendido por el decano que luego de valorar el caso decide si se le otorga o no. El caso de uso termina cuando se le da baja temporal al estudiante del centro.	
Precondiciones:	El estudiante tiene que ser matrícula del centro.	
Flujo normal de eventos		
	Acción del Actor	Respuesta del Negocio
	1. El estudiante solicita la licencia presentándose en el decanato.	2. El decano analiza el caso del estudiante y aprueba la licencia. 3. La secretaria docente le entrega el modelo de tramitación de baja.
	4. El estudiante entrega todos los medios básicos que posee de la Universidad.	5. La secretaria docente verifica que el estudiante haya entregado todos los medios. 6. La secretaria docente llena el modelo de licencia. 7. El decano firma el modelo de licencia. 8. La secretaria docente archiva el modelo de licencia en el expediente del estudiante y se lo envía a la secretaria general. 9. La secretaria docente actualiza el listado de licencias de la facultad. 10. La secretaria general actualiza el listado de licencias del centro.
Flujo Alterno 1:		
	Acción del Actor	Respuesta del Negocio
		2.1.El decano informa al estudiante que no fue aprobada la licencia.
Poscondiciones:		

Tabla 2.11: Descripción de CUN Solicitar Licencia Estudiante

CUN: Solicitar Licencia Profesor

Caso de Uso:	Solicitar Licencia Profesor
Actores:	Profesor.
Trabajadores:	Decano, Jefe de Departamento.

Resumen:	El caso de uso inicia cuando el profesor se presenta en el decanato solicitando licencia, este es atendido por el decano que luego de valorar el caso decide si se le otorga o no. El caso de uso termina cuando se le da baja temporal al profesor del centro.	
Precondiciones:	El profesor tiene que tener actualizado su contrato laboral.	
Flujo normal de eventos		
Acción del Actor		Respuesta del Negocio
1. El profesor solicita la licencia presentándose en el decanato.	2. El decano analiza el caso del profesor y aprueba la licencia. 3. El jefe de departamento busca que asignatura(s) imparte el profesor. 4. Revisa el listado de profesores de esa(s) asignatura(s) y elimina al profesor del listado. 5. El decano elabora el documento de baja y lo firma.	
Flujo Alterno 1:		
Acción del Actor		Respuesta del Negocio
	2.1. El decano informa al profesor que no fue aprobada la licencia.	
Poscondiciones		

Tabla 2.12: Descripción del CUN Solicitar Licencia Profesor

CUN: Solicitar Baja Estudiante

Caso de Uso:	Solicitar Baja Estudiante	
Actores:	Estudiante.	
Trabajadores:	Decano, Secretaria General, Secretaria Docente.	
Resumen:	El caso de uso se inicia cuando el estudiante se presenta en el decanato solicitando la baja, este es atendido por el decano que luego de valorar el caso decide si aprueba la solicitud de baja o no. El caso de uso termina con la baja del estudiante del centro.	
Precondiciones:	El estudiante tiene que ser matrícula del centro.	
Flujo normal de eventos		
Acción del Actor		Respuesta del Negocio
1. El estudiante solicita la baja presentándose en el decanato.	2. El decano analiza las causas de la baja del estudiante y aprueba la baja.	

	3. La secretaria docente le entrega el modelo de tramitación de baja.
4. El estudiante entrega todos los medios básicos que posee de la Universidad.	5. La secretaria docente verifica que el estudiante haya entregado todos los medios. 6. La secretaria docente llena el modelo de baja. 7. El decano firma el modelo de baja. 8. La secretaria docente archiva el modelo de baja en el expediente del estudiante. 9. La secretaria docente actualiza el listado de bajas de la facultad. 10. La secretaria docente informa a la secretaria docente general y le envía el expediente del estudiante. 11. La secretaria docente general da baja al estudiante del centro.
Flujo Alterno 1:	
Acción del Actor	Respuesta del Negocio
	2.1. El decano informa al estudiante que no fue aprobada la baja.
Poscondiciones	

Tabla 2.13: Descripción del CUN Solicitar Baja Estudiante

CUN: Solicitar Licencia Profesor

Caso de Uso:	Solicitar Baja Profesor
Actores:	Profesor
Trabajadores:	Decano, Jefe de Departamento.
Resumen:	El caso de uso se inicia cuando el profesor se presenta en el decanato solicitando la baja, este es atendido por el decano que luego de valorar el caso decide si aprueba la solicitud baja o no. El caso de uso termina con la baja del profesor del centro.
Precondiciones:	El profesor tiene que tener actualizado su contrato laboral.
Flujo normal de eventos	
Acción del Actor	Respuesta del Negocio
1. El profesor solicita la baja presentándose en el decanato.	2. El decano analiza el motivo de la baja del profesor y aprueba la baja. 3. El jefe de departamento busca que asignatura(s)

	imparte el profesor. 4. Revisa el listado de profesores de esa(s) asignatura(s) y elimina al profesor del listado. 5. El decano elabora y firma el modelo de baja.
Flujo Alternativo 1:	
Acción del Actor	Respuesta del Negocio
	2.1. El decano informa al profesor que no fue aprobada la baja.
Poscondiciones	

Tabla 2.14: Descripción del CUN Solicitar Baja Profesor

2.9 Modelo de Objetos

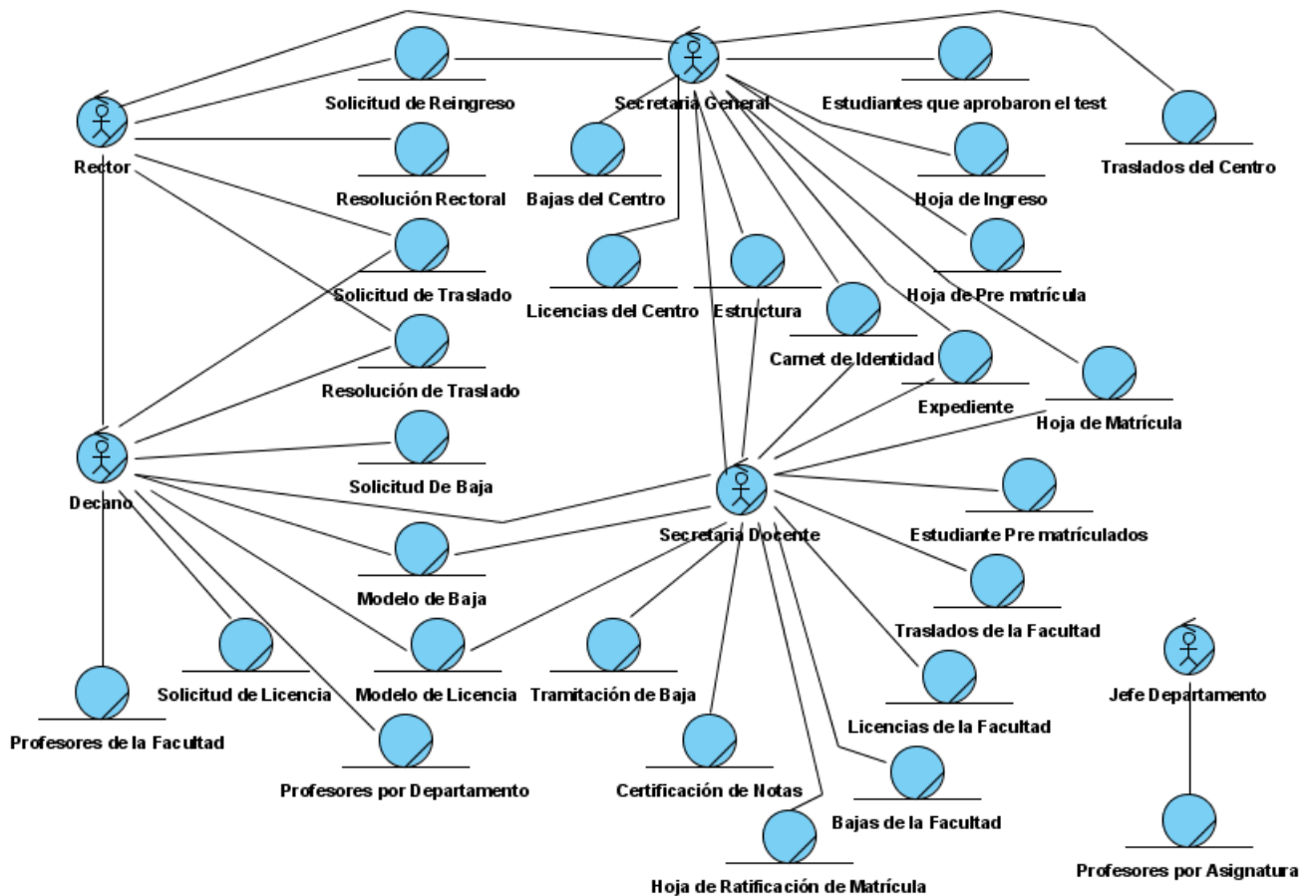


Figura 2: Modelo de Objetos

2.10 Especificación de Requerimientos del Sistema

2.10.1 Requerimientos Funcionales.

A continuación se muestra el listado de los requerimientos funcionales del módulo Gestión de Personal:

R1. Crear tipo de personal.

1. Si el usuario cuenta con los permisos necesarios, mostrar una interfaz con un listado de todos los tipos de personal definidos anteriormente y realizar los próximos pasos.
2. Mostrar la opción “Crear Tipo de Personal” y permitir seleccionarla.
3. Mostrar una interfaz con formulario para la creación de un tipo de personal.
4. Permitir introducir los datos correspondientes al tipo de personal a crear: nombre, descripción, categoría, estado inicial, y documentos asociados por cada uno de los estados definidos anteriormente.
5. Validar los datos introducidos:
 - 5.1. Si existen datos incorrectos, mostrar mensaje de error y permitir volver a introducirlos.
6. Comprobar que no exista un tipo de personal con el mismo nombre:
 - 6.1. Si existe un tipo de personal con el mismo nombre, mostrar el mensaje de error correspondiente y permitir volver a introducir los datos.
7. Almacenar los datos del tipo de personal creado.
8. Registrar incidencias.

R2. Modificar tipo de personal.

1. Si el usuario cuenta con los permisos necesarios, mostrar una interfaz con un listado de todos los tipos de personal definidos anteriormente y realizar los próximos pasos.
2. Permitir seleccionar tipo de personal a modificar.
3. Mostrar los datos correspondientes al tipo de personal seleccionado.
4. Permitir modificar los datos del tipo de personal seleccionado: nombre, descripción, categoría, estado inicial y documentos asociados por cada uno de los estados definidos anteriormente.
5. Validar los datos modificados:
 - 5.1. Si existen datos incorrectos, mostrar mensaje de error y permitir volver a modificarlos.

6. Comprobar que no exista un tipo de personal con el mismo nombre:
 - 6.1. Si existe un tipo de personal con el mismo nombre, mostrar mensaje de error y permitir volver a modificar los datos.
7. Almacenar los datos del tipo de personal modificado.
8. Registrar incidencias.

R3. Eliminar tipo de personal.

1. Si el usuario cuenta con los permisos necesarios, mostrar una interfaz con un listado de todos los tipos de personal definidos anteriormente y realizar los próximos pasos.
2. Permitir seleccionar el tipo de personal a eliminar.
3. Mostrar opción "Eliminar Tipo de Personal" y permitir seleccionarla.
4. Comprobar que el tipo de personal seleccionado no esté asignado a alguna persona:
 - 4.1. Si está asignado a alguna persona, mostrar mensaje de error indicando que no puede ser eliminado.
5. Eliminar el tipo de personal seleccionado.
6. Registrar incidencias.

R4. Crear estado

1. Si el usuario cuenta con los permisos necesarios, mostrar interfaz con listado de todos los estados definidos anteriormente y realizar los próximos pasos.
2. Mostrar la opción "Crear Estado" y permitir seleccionarla.
3. Mostrar una interfaz con un formulario con los campos para la creación de un estado.
4. Permitir introducir los datos correspondientes al estado a crear: nombre, descripción, color, nombre del movimiento asociado al estado, estados precedentes y documentos asociados por cada uno de los tipos de personal definidos anteriormente.
5. Validar los datos introducidos:
 - 5.1. Si existen datos incorrectos, mostrar mensaje de error y permitir volver a introducirlos.
6. Comprobar que no exista un estado con el mismo nombre:
 - 6.1. Si existe un estado con el mismo nombre, mostrar mensaje de error y permitir volver a introducir los datos.
7. Almacenar los datos del estado creado.
8. Registrar incidencias.

R5. Modificar estado.

1. Si el usuario cuenta con los permisos necesarios, mostrar una interfaz con un listado de todos los estados definidos anteriormente y realizar los próximos pasos.
2. Permitir seleccionar el estado a modificar.
3. Mostrar los datos correspondientes al estado seleccionado.
4. Permitir modificar los datos del estado seleccionado: nombre, descripción, color, nombre del movimiento asociado, documentos asociados por cada uno de los tipos de personal definidos y estados precedentes.
5. Validar los datos modificados:
 - 5.1. Si existen datos incorrectos, mostrar mensaje de error y permitir volver a modificarlos.
6. Comprobar que no exista un estado con el mismo nombre:
 - 6.1. Si existe un estado con el mismo nombre: mostrar mensaje de error y permitir volver a modificar los datos.
7. Almacenar los datos del estado modificado.
8. Registrar incidencias.

R6. Eliminar estado.

1. Si el usuario cuenta con los permisos necesarios, mostrar una interfaz con un listado de todos los estados definidos anteriormente y realizar los próximos pasos.
2. Permitir seleccionar el estado a eliminar.
3. Mostrar la opción “Eliminar Estado” y permitir seleccionarla.
4. Comprobar que no existan personas que tengan asociado el estado seleccionado.
 - 4.1. Si existen personas con ese estado asociado, mostrar mensaje de error indicando que no puede ser eliminado.
5. Comprobar que no existan tipos de personal que tengan asociado el estado seleccionado como estado inicial.
 - 5.1. Si existen tipos de personal con ese estado asociado, mostrar mensaje de error indicando que no puede ser eliminado.
6. Eliminar las relaciones de precedencia de este estado con los demás estados.
7. Eliminar el estado seleccionado.
8. Registrar incidencias.

R7. Registrar datos de la persona manual.

1. Si el usuario cuenta con los permisos necesarios, mostrar la opción “Registrar Datos Manual” y realizar los próximos pasos.
2. Mostrar una interfaz con las opciones para registrar los datos de una persona de forma manual.
3. Permitir seleccionar el tipo de personal de la persona que se desea registrar sus datos.
4. Permitir seleccionar la foto de la persona.
5. Permitir seleccionar la opción “Ubicar en estructura automáticamente”.
6. Según el tipo de personal seleccionado, mostrar los documentos asociados al estado inicial correspondiente.
7. Permitir seleccionar los documentos para llenar sus datos:
 - 7.1. Mostrar formulario con los campos correspondientes al documento seleccionado.
 - 7.2. Permitir introducir los datos que solicita el documento seleccionado.
 - 7.3. Validar los datos introducidos:
 - 7.3.1. Si existen datos incorrectos, mostrar mensaje de error y permitir volver a introducirlos.
8. Verificar que todos los documentos hayan sido llenados correctamente:
 - 8.1. Si quedan documentos con datos sin llenar, mostrar mensaje de error y permitir llenar los documentos que faltan.
9. Asociar a la persona el tipo de personal seleccionado.
10. Asociar a la persona el estado inicial correspondiente al tipo de personal seleccionado.
11. Si fue seleccionada la opción “Ubicar en estructura automáticamente” en el paso 4, ubicar la persona en una estructura de acuerdo a un conjunto de reglas definidas.
12. Almacenar todos los datos correspondientes a la persona.
13. Registrar incidencias.

R8. Registrar datos de la persona automático.

1. Si el usuario cuenta con los permisos necesarios, mostrar la opción “Registrar Datos Automático” y realizar los próximos pasos.
2. Mostrar interfaz con las opciones para importar los datos de las personas de otro sistema.
3. Permitir seleccionar el tipo de personal de las personas que se desea importar sus datos.
4. Permitir seleccionar el sistema externo de donde se desea importar los datos.
5. Permitir seleccionar la opción “Ubicar en estructura automáticamente”.

6. Según el tipo de personal seleccionado, mostrar los documentos asociados al estado inicial correspondiente al tipo de personal seleccionado.
7. Mostrar un listado de todos los campos que poseen en conjunto todos los documentos asociados al tipo de personal seleccionado en su estado inicial.
8. Permitir seleccionar estos campos.
9. Permitir seleccionar, de los campos que brinda el sistema externo, cuáles se utilizarán para importar los datos de las personas.
10. Permitir seleccionar la opción “Importar Datos”.
11. Para toda persona registrada en el sistema externo, importar los datos de los campos seleccionados.
12. Asociar a las personas el tipo de personal seleccionado.
13. Asociar a las personas el estado inicial correspondiente al tipo de personal seleccionado.
14. Si fue seleccionada la opción “Ubicar en estructura automáticamente” en el paso 4, ubicar a las personas en una estructura de acuerdo a un conjunto de reglas definidas.
15. Almacenar todos los datos correspondientes a las personas.
16. Registrar incidencias.

R9. Modificar datos de la persona.

1. Buscar la persona a modificar sus datos mediante la interfaz de búsqueda.
2. Permitir seleccionar la persona a modificar sus datos.
3. Si el usuario cuenta con los permisos necesarios, mostrar la opción “Modificar Datos” y realizar los próximos pasos.
4. Mostrar foto y datos correspondientes a la persona seleccionada.
5. Mostrar los documentos asociados a la persona desde su registro.
6. Mostrar listado con todos los estados por los que ha pasado la persona desde su registro.
7. Si el usuario cuenta con los permisos necesarios permitir seleccionar documentos del listado para ver y/o modificar sus datos:
 - 7.1. Mostrar formulario con los campos correspondientes al documento seleccionado.
 - 7.2. Permitir modificar los datos del documento seleccionado para esa persona.
 - 7.3. Validar los datos modificados.
 - 7.3.1. Si existen datos incorrectos, mostrar mensaje de error y permitir volver a modificarlos.
 - 7.4. Almacenar los datos modificados de la persona.

8. Registrar incidencias.

R10. Buscar persona.

1. Mostrar la interfaz de búsqueda de personas.
2. Permitir seleccionar diferentes criterios de búsqueda.
3. Buscar de acuerdo a los criterios seleccionados.
4. Listar todas las personas que cumplan con los criterios seleccionados. Los resultados mostrarán la foto, tipo persona, estructura y otros datos por cada persona encontrada.
5. Permitir seleccionar una persona.
6. Si el usuario cuenta con los permisos necesarios, mostrar las opciones:
 - Modificar datos de la persona.
 - Realizar movimiento a la persona.
 - Ubicar personas en estructura.
 - Asignar asignaturas en caso de ser un profesor.
 - Asignar planes de estudio en caso de ser estudiante.
7. Registrar incidencia.

R11. Realizar movimientos.

1. Buscar la persona a realizar movimiento mediante la interfaz de búsqueda.
2. Permitir seleccionar la persona a realizarle movimiento.
3. Si el usuario cuenta con los permisos necesarios mostrar la opción “Realizar Movimiento” y realizar los próximos pasos.
4. Mostrar una interfaz con los datos de la persona seleccionada y las opciones para realizar movimiento.
5. Mostrar los movimientos que pueden ser aplicados a la persona a partir del estado actual.
6. Permitir seleccionar uno de estos movimientos.
7. Mostrar documentos asociados al tipo de personal que pertenece la persona seleccionada y al estado al que cambiará esta al realizar el movimiento.
8. Permitir seleccionar los documentos para llenar sus datos:
 - 8.1. Mostrar formulario con los campos correspondientes al documento seleccionado.
 - 8.2. Introducir los datos que solicita el documento seleccionado para esa persona en el nuevo estado.
 - 8.3. Validar los datos introducidos:

- 8.3.1. Si existen datos incorrectos, mostrar mensaje de error y permitir volver a introducirlos.
9. Verificar que todos los documentos hayan sido llenados correctamente.
 - 9.1. Si queda algún documento con datos sin llenar, mostrar mensaje de error y permitir llenar los datos que le faltan.
10. Cambiar el estado actual de la persona por el estado seleccionado.
11. Almacenar los datos de los documentos asociados al nuevo estado para esa persona.
12. Registrar incidencias.

R12. Ubicar personas en estructura.

1. Buscar las personas a ubicar en estructura mediante la interfaz de búsqueda.
2. Permitir seleccionar una de las personas encontradas.
3. Si el usuario cuenta con los permisos necesarios, mostrar la opción “Ubicar en Estructura” y realizar los próximos pasos.
4. Mostrar una interfaz con las opciones para realizar la ubicación en las estructuras.
5. Mostrar listado de las personas que se desean ubicar:
 - 5.1. Si el usuario seleccionó una persona en la interfaz de búsqueda, se mostrará sólo la persona seleccionada.
 - 5.2. Si el usuario no seleccionó alguna persona en la interfaz de búsqueda, se mostrarán todas las personas que se fueron encontradas en la búsqueda.
6. Permitir seleccionar una o varias personas del listado.
7. Mostrar campo para seleccionar una nueva estructura.
8. Mostrar y permitir seleccionar las opciones: “Ubicar Seleccionados” y “Quitar Ubicación”.
 - 8.1. Si el usuario selecciona la opción “Ubicar Seleccionados”, todas las personas seleccionadas en el listado se ubicarán en la estructura seleccionada.
 - 8.2. Si el usuario selecciona la opción “Quitar Ubicación”, todas las personas seleccionadas en el listado se les quitará su ubicación actual y quedarán sin ubicación.
 - 8.3. Almacenar los cambios realizados.
9. Registrar incidencias.

R13. Asignar asignaturas al profesor.

1. Buscar la persona mediante la interfaz de búsqueda.

2. Permitir seleccionar una de las personas encontradas.
3. Mostrar los datos de la persona seleccionada.
4. Si el usuario cuenta con los permisos necesarios y la persona es profesor, mostrar opción “Asignar Asignaturas” y realizar los próximos pasos.
5. Permitir seleccionar la opción “Asignar Asignaturas”.
6. Mostrar interfaz con las opciones para realizar la asignación de asignaturas al profesor.
7. Mostrar listado de las asignaturas asignadas.
8. Mostrar listado de todas las asignaturas disponibles.
9. Permitir agregar o quitar asignaturas en el listado de asignaturas asignadas a partir del listado de asignaturas disponibles.
10. Asociar al profesor las asignaturas que estén en el listado de asignaturas asignadas. Si el profesor tiene asignadas asignaturas que no estén en este listado, quitar estas asociaciones.
11. Guardar los cambios realizados.
12. Registrar incidencias.

R14. Asociar plan de estudio.

1. Buscar personas a asociar plan de estudio mediante la interfaz de búsqueda.
2. Permitir seleccionar una de estas personas.
3. Si el usuario cuenta con los permisos necesarios y las personas son estudiantes, mostrar opción “Asociar Plan de Estudio” y realizar los próximos pasos.
4. Mostrar interfaz con las opciones para la asociación de planes de estudio.
5. Mostrar listado de los estudiantes que se desea asociar a un plan de estudio:
 - 5.1. Si el usuario seleccionó un estudiante en la interfaz de búsqueda, se mostrará sólo el estudiante seleccionado.
 - 5.2. Si el usuario no seleccionó ningún estudiante en la interfaz de búsqueda, se mostrarán todos los estudiantes que fueron encontrados en la búsqueda.
6. Permitir seleccionar uno o varios estudiantes de ese listado.
7. Mostrar campo con listado de todos los planes de estudios vigentes.
8. Permitir seleccionar planes de estudio.
9. Asociar los planes de estudio seleccionados en este listado a los estudiantes seleccionados.
10. Guardar los cambios realizados.

11. Registrar incidencias.

R15. Consultar información.

1. Si el usuario cuenta con los permisos necesarios, mostrar interfaz con todas las opciones de reportes que se pueden consultar:
 - 1.1. Personas por períodos:
 - 1.1.1. Mostrar listado con todos los períodos definidos.
 - 1.1.2. Permitir seleccionar uno o varios períodos.
 - 1.2. Personas por tipo de personal:
 - 1.2.1. Mostrar listado de los tipos de personal definidos.
 - 1.2.2. Permitir seleccionar uno o varios tipos de personal.
 - 1.3. Personas por estados:
 - 1.3.1. Mostrar listado de todos los estados definidos.
 - 1.3.2. Permitir seleccionar uno o varios estados.
 - 1.4. Personas por estructuras.
 - 1.4.1. Mostrar las estructuras definidas.
 - 1.4.2. Permitir seleccionar una o varias estructuras.
 - 1.5. Profesores por asignaturas (Sólo se mostrará esta opción si los tipos de personal seleccionados son todos profesores):
 - 1.5.1. Mostrar listado de las asignaturas definidas.
 - 1.5.2. Permitir seleccionar una o varias asignaturas.
 - 1.6. Campos a mostrar:
 - 1.6.1. Mostrar listado de todos los campos que estén presentes a la vez en los tipos de personal y estados seleccionados.
 - 1.6.2. Permitir seleccionar uno o varios campos.
2. Permitir seleccionar de estas opciones cuáles desea incluir en la información que desea obtener en el listado de reportes.
3. Mostrar interfaz con listado de todas las personas que sus datos cumplan con los criterios antes seleccionados. Sólo se mostrará de las personas, la información de los campos que fueron seleccionados en el paso 1.6 y de las opciones seleccionadas en el paso 2.
4. Registrar incidencias.

2.10.2 Requerimientos no Funcionales.

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

1. Usabilidad:

- 1.1. Facilidad de uso por parte de los usuarios: el sistema debe presentar una interfaz amigable que permita la fácil interacción con el mismo y llegar de manera rápida y efectiva a la información buscada. Debe, además, ser una interfaz de manejo cómodo que posibilite a los usuarios sin experiencia una rápida adaptación al mismo.
- 1.2. Especificación de la terminología utilizada: el sistema debe adaptarse al lenguaje y términos utilizados por los clientes en la rama abordada con vista a una mayor comprensión por parte del cliente de la herramienta de trabajo.
- 1.3. Emplear perfiles de usuario: diferenciar las interfaces y opciones para los usuarios que accedan al sistema con diferentes roles (secretaria, administrador, estudiantes).
- 1.4. Estándares de usabilidad:
 - Hacer uso de gráficos e imágenes claras y nítidas.
 - Limitar el número de enlaces en una página.
 - Respetar el espacio entre los enlaces y los botones.
 - Evitar el uso de los menús en cascada.
 - Proporcionar a los formularios un título que muestre claramente su función.
 - Utilizar con los textos una nomenclatura clara y familiar.
 - Evitar uso de preguntas complejas.
 - Hacer corresponder el tamaño visible de los campos de texto con la longitud del contenido que ha de introducir el usuario.
 - Establecer en los campos, siempre que sea posible, una opción por defecto.
 - Identificar claramente los campos obligatorios y opcionales.
 - Informar al usuario el resultado de una acción cuando esta se ejecute.

2. Fiabilidad:

- 2.1. El sistema debe estar disponible las 24 horas del día y los 7 días de la semana.
- 2.2. Hacer transparente para el usuario el mantenimiento del sistema.

- 2.3. Las salidas del sistema deben ser de alta precisión y exactitud de acuerdo a las necesidades del usuario.

3. Seguridad:

- 3.1. Políticas de seguridad por usuarios y roles: el sistema debe contar con un grupo de políticas de accesibilidad a las diferentes funcionalidades del mismo en dependencia del nivel de autorización que presente un usuario determinado.
- 3.2. Autenticación basada en dos niveles: el sistema debe tener la posibilidad de, en caso que no exista un Directorio Activo, gestionar usuarios y roles, así como las prioridades y niveles de acceso en los mismos; debe además permitir conectarse a un Directorio Activo y gestionar parte de la seguridad según las especificaciones del mismo.
- 3.3. Servicios Web restringidos: los servicios Web que brinde el sistema deben estar restringidos a grupos de usuarios definidos y aprobados previamente.
- 3.4. Seguridad de la Base de Datos: la base de datos deberá estar fraccionada en esquemas que permitan un mejor uso de la información y la división de forma lógica de las funcionalidades del sistema, trayendo consigo además la protección de la información al ocurrir un incidente sobre una parte de la base de datos. El SGBD escogido debe presentar facilidades de administración de roles y usuarios restringiendo el acceso a los datos.
- 3.5. Registro sistemáticos de incidencias: el sistema debe ser capaz de registrar el accionar del usuario sobre el mismo, así como permitir auditorias y exámenes de las trazas tanto en tiempo real como en históricos. Se precisa un monitor de incidencia para la visualización y tratamiento de las mismas.
- 3.6. Alta protección de los datos: al estar trabajando con información sensible, se hace necesario una alta protección de los datos a nivel de aplicación y de tráfico por la red, para tal fin se ha definido además la seguridad en varios niveles dentro de la aplicación (Nivel de Interfaz, Nivel de Acceso a Datos y Nivel de Base de Datos), así como el uso del protocolo seguro HTTPS para el tráfico de la información por la red.

4. Rendimiento:

- 4.1. Escalabilidad: el sistema debe ser capaz de mantener un rendimiento y una estabilidad adecuado al gestionar amplios volúmenes de datos, así como de permitir la incorporación de nuevas funcionalidades adaptándose de manera natural a los procesos de gestión que se requieran.

4.2. Hardware:

Para el desarrollo: PC Intel Pentium 4 o superior, CPU 3GHZ o superior, 512 MB RAM o superior, 160 GB HDD o superior.

Para explotación del cliente: PC Pentium 3 o superior, CPU 133 MHZ o superior, 128 RAM mínimo 512 RAM recomendada o superior.

Para explotación del servidor: CPU: Dual Core 2.0 GHZ o superior, RAM: 4 GB (Recomendado 6 GB), 250 GB HDD.

4.3. Software:

Para el Cliente: sistema operativo con interfaz gráfica y conexión a red. Navegador Web (Mozilla FireFox_Recomendado).

Para el Servidor: sistema operativo Linux: Ubuntu Server 7.10.

Servidor Web: Apache 2.2.9

Gestor de Base de Datos: PostgreSQL 8.3.5

2.11 Definición de Actores del Sistema

Trabajadores	Descripción
Administrador	Es el encargado de crear, modificar o eliminar los estados y los tipos de personal.
Secretaria	Es el encargado de registrar o modificar los datos de la persona, realizar movimientos, ubicar personas en estructuras y asociar planes de estudio a los estudiantes.
Jefe de departamento	Es el encargado de asignar asignaturas a los profesores.
Usuario	A este actor se generalizan el administrador, el jefe de departamento y la secretaria, cada uno de ellos a través del usuario pueden buscar una persona o consultar todos los reportes que le sean necesarios.

Tabla 2.15: Definición de Actores del Sistema

2.12 Diagrama de Casos de Usos del Sistema.

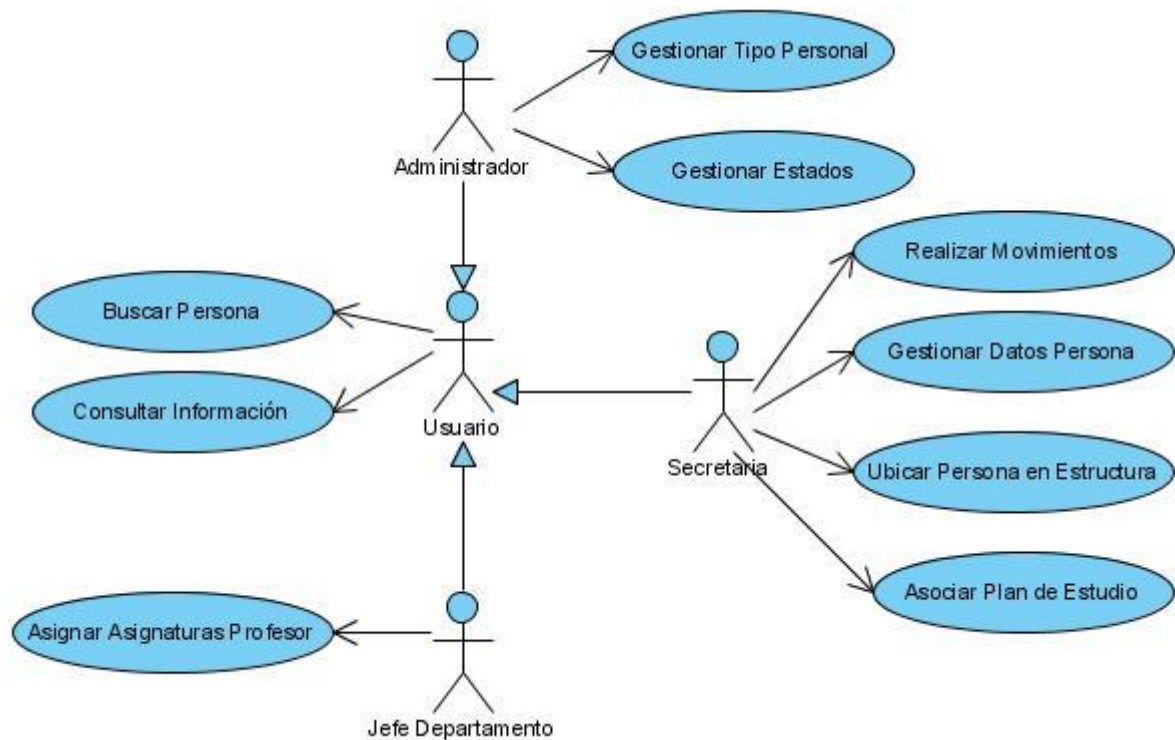


Figura 3: Diagrama de Casos de Uso del Sistema

2.13 Descripciones de los Casos de Uso del Sistema.

A continuación se muestran las descripciones abreviadas de los Casos de Uso del Sistema, las descripciones detalladas se encuentran en el Anexo #2.

CUS: Gestionar Tipo Personal

CUS-1	Gestionar Tipo Personal
Propósito	Permitir realizar las acciones Crear Tipo de Personal, Modificar Tipo de Personal y Eliminar Tipo de Personal.
Actores	Administrador
Resumen:	El caso de uso inicia cuando el Administrador decide gestionar los tipos de personal que presentará el sistema. Este le permitirá crear un tipo de personal, modificar un tipo de personal o eliminar un tipo de personal.
Precondiciones	El usuario debe estar registrado y tener los permisos necesarios para realizar las acciones.

	<p>Para eliminar o modificar un Tipo de Personal, este debe haber sido creado con anterioridad.</p> <p>Para eliminar un Tipo de Personal este no debe haber sido asignado a alguna persona.</p>
Referencias	R1, R2, R3.
Prioridad	Crítico
Poscondiciones	<p>Cuando se crea un tipo de personal los datos del mismo quedan guardados en la base de datos.</p> <p>Cuando se modifica un tipo de personal los datos del mismo quedan modificados en la base de datos.</p> <p>Cuando se elimina un tipo de personal los datos del mismo se eliminan de la base de datos.</p>

Tabla 2.16: Descripción abreviada del CUS Gestionar Tipo Personal

CUS: Gestionar Estados

CUS-2	Gestionar Estados
Propósito	Permitir realizar las acciones Crear Estado, Modificar Estado y Eliminar Estado.
Actores	Administrador
Resumen:	El caso de uso inicia cuando el administrador decide gestionar los estados por los que puede pasar una persona. El sistema le permitirá crear, modificar o eliminar un estado.
Precondiciones	<p>El usuario debe estar registrado y tener los permisos necesarios para realizar las acciones.</p> <p>Para eliminar o modificar estado, este debe haber sido creado con anterioridad.</p> <p>Para eliminar un estado esto no debe haber sido asignado a alguna persona.</p>
Referencias	R4, R5, R6.
Prioridad	Crítico
Poscondiciones	<p>Cuando se crea un estado los datos del mismo quedan guardados en la base de datos.</p> <p>Cuando se modifica un estado los datos del mismo quedan modificados en la base de datos.</p> <p>Cuando se elimina un estado los datos del mismo se eliminan de la base de datos.</p>

Tabla 2.17: Descripción abreviada del CUS Gestionar Estados

CUS: Gestionar Datos Persona

CUS-3	Gestionar Datos Persona
Propósito	Permitir realizar las acciones Registrar Datos de la Persona Manual, Registrar Datos de la Persona Automático o Modificar Datos de la Persona.
Actores	Secretaria
Resumen:	El caso de uso inicia cuando la secretaria decide gestionar los datos de la persona. El sistema le permitirá registrar o modificar los datos de la persona.
Precondiciones	El usuario debe estar registrado y tener los permisos necesarios para realizar las acciones. Para registrar los datos automático es necesaria la presencia de un sistema externo de donde importar los datos. Para modificar los datos de una persona, esta debe haber sido registrada en el sistema.
Referencias	R7, R8.R9
Prioridad	Crítico
Poscondiciones	Cuando se registran una persona los datos de la misma quedan guardados en la base de datos, así como el estado en que se encuentra. Cuando se modifican los datos de una persona estos quedan modificados en la base de datos.

Tabla 2.18: Descripción abreviada del CUS Gestionar Datos Persona

CUS: Buscar Persona

CUS-4	Buscar Persona
Propósito	Buscar personas en el sistema.
Actores	Usuario
Resumen:	El caso de uso inicia cuando un usuario decide buscar personas en el sistema. Este le permitirá buscarlas de acuerdo a diferentes criterios de búsqueda y una vez encontradas las personas permitirá visualizar los datos de las mismas así como realizar otras operaciones.
Precondiciones	El usuario debe estar registrado y tener los permisos necesarios para realizar las acciones.
Referencias	R10
Prioridad	Crítico
Poscondiciones	Se mostrarán las personas que cumplan con los criterios de búsqueda

	seleccionados.
--	----------------

Tabla 2.19: Descripción abreviada del CUS Buscar Persona

CUS: Realizar Movimiento

CUS-5	Realizar Movimiento
Propósito	Aplicar un movimiento a una persona. Esta acción consiste en realizar un cambio de estado a la persona a partir de otro estado y llenar los documentos asociados al nuevo estado.
Actores	Secretaria
Resumen:	El caso de uso inicia cuando la secretaria decide aplicar un movimiento a una persona. El sistema realizará el cambio de estado pertinente.
Precondiciones	El usuario debe estar registrado y tener los permisos necesarios para realizar la acción.
Referencias	R11.
Prioridad	Crítico
Poscondiciones	Se cambia el estado en que se encuentra la persona en ese momento por el estado final del movimiento.

Tabla 2.20: Descripción abreviada del CUS Realizar Movimiento

CUS: Ubicar Persona en Estructura

CUS-6	Ubicar Persona en Estructura
Propósito	Permitir ubicar una o varias personas en las diferentes estructuras definidas.
Actores	Secretaria
Resumen:	El caso de uso inicia cuando la secretaria decide ubicar una o varias personas en las diferentes estructuras definidas. El sistema ubicará las personas en la estructura seleccionada o les quitará la ubicación anterior.
Precondiciones	El usuario debe estar registrado y tener los permisos necesarios para realizar las acciones.
Referencias	R12.
Prioridad	Crítico
Poscondiciones	

Tabla 2.21: Descripción abreviada del CUS Ubicar Persona en Estructura

CUS: Asignar Asignaturas Profesor

CUS-7	Asignar Asignaturas Profesor
Propósito	Permitir asignar a un profesor las asignaturas que deberá impartir.
Actores	Jefe de Departamento
Resumen	El caso de uso inicia cuando el jefe de departamento decide asignar las asignaturas que deberá impartir un profesor. El sistema le permitirá seleccionar estas asignaturas a partir de un listado de todas las asignaturas disponibles.
Precondiciones	El usuario debe estar registrado y tener los permisos necesarios para realizar la acción.
Referencias	R13
Prioridad	Crítico
Poscondiciones	Todos los cambios realizados son almacenados en la bases de datos.

Tabla 2.22: Descripción abreviada del CUS Asignar Asignaturas Profesor

CUS: Asociar Plan de Estudio

CUS-8	Asociar Plan de Estudio.
Propósito	Permitir asociar uno o varios estudiantes a un plan de estudio.
Actores	Secretaria
Resumen:	El caso de uso inicia cuando la secretaria decide asociar un plan de estudio a uno o varios estudiantes. El sistema asociará los estudiantes al plan de estudio.
Precondiciones	El usuario debe estar registrado y tener los permisos necesarios para realizar las acciones.
Referencias	R14.
Prioridad	Crítico
Poscondiciones	Todos los cambios realizados son almacenados en la bases de datos.

Tabla 2.23: Descripción abreviada del CUS Asociar Plan de Estudio

CUS: Consultar Información

CUS-9	Consultar información
Propósito	Realizar diferentes reportes con la información sobre el personal existente en el sistema.
Actores	Usuario

Resumen	El caso de uso inicia cuando un usuario decide consultar información sobre el personal registrado en el sistema. El sistema permitirá ver esta información de acuerdo a diferentes criterios.
Precondiciones	El usuario debe estar registrado y tener los permisos necesarios para realizar la acción.
Referencias	R15.
Prioridad	Crítico
Poscondiciones	

Tabla 2.24: Descripción abreviada del CUS Consultar Información

2.14 Estimación del Esfuerzo.

Definidos los casos de uso se puede predecir una estimación del tiempo de duración del proyecto mediante el análisis de Puntos de Casos de Uso.

La estimación mediante el análisis de Puntos de Casos de Uso es un método propuesto originalmente por Gustav Karner de Objectory AB, y posteriormente refinado por muchos otros autores.

“La estimación mediante el Análisis de Puntos de Casos de Uso se trata de un método de estimación del tiempo de desarrollo de un proyecto mediante la asignación de "pesos" a un cierto número de factores que lo afectan, para finalmente, contabilizar el tiempo total estimado para el proyecto a partir de esos factores” (9).

A continuación, se detallan los pasos a seguir para la aplicación de éste método.

1. Cálculo de Puntos de Casos de Uso sin ajustar

Se calcula a partir de la siguiente ecuación:

UUCP = UAW + UUCW, donde:

UUCP: Puntos de Casos de Uso sin ajustar

UAW: Factor de Peso de los Actores sin ajustar

UUCW: Factor de Peso de los Casos de Uso sin ajustar

2. Factor de Peso de los Actores sin ajustar (UAW):

Este valor se calcula mediante un análisis de la cantidad de Actores presentes en el sistema y la complejidad de cada uno de ellos.

Actores del Sistema	Tipo de Actor	Factor de Peso
Usuario	Complejo	3
Administrador		
Secretaria		
Jefe de Departamento		

Tabla 2.25: Factor de Peso de los Actores sin ajustar

Cantidad de Actores de tipo complejo: 4

UAW= Sumatoria de la multiplicación de la cantidad de actores de un tipo específico con su factor de peso.

$$UAW = \sum (\text{actores} * \text{Peso})$$

$$UAW = 4 \times 3$$

$$UAW = 12$$

3. Factor de Peso de los Casos de Usos sin ajustar (UUCW):

Este valor se calcula mediante un análisis de la cantidad de Casos de Uso presentes en el sistema y la complejidad de cada uno de ellos.

Casos de Uso	Cantidad de Transacciones	Tipo de Casos de Uso	Factor de Peso
Gestionar Tipo de Personal	4 transacciones	Medio	10
Buscar Persona	2 transacciones	Simple	5
Gestionar Estados	4 transacciones	Medio	10
Gestionar Datos del Personal	7 transacciones	Medio	10
Realizar Movimientos	3 transacciones	Simple	5
Ubicar Persona en Estructura	3 transacciones	Simple	5
Asignar Asignaturas al Profesor	3 transacciones	Simple	5
Asociar Plan de Estudio	3 transacciones	Simple	5
Consultar Información	2 transacciones	Simple	5

Tabla 2.26: Factor de Peso de los Casos de Usos sin ajustar

Cantidad de casos de uso simples: 6

Cantidad de casos de uso medios: 3

UUCW = Sumatoria de la multiplicación de la cantidad de Casos de Uso de un tipo específico con su factor de peso.

$$UUCW = \sum CU * \text{Peso}$$

$$UUCW = (6 \times 5) + (3 \times 10)$$

$$UUCW = 30 + 30$$

$$UUCW = 60$$

Al ser $UAW = 12$ y $UUCW = 60$

$UUCP = UAW + UUCW$, sustituyendo, tenemos:

$$UUCP = 12 + 60$$

$$UUCP = 72$$

4. Cálculo de los puntos de Casos de Uso ajustados

Se calcula mediante la siguiente ecuación:

$$UCP = UUCP \times TCF \times EF \text{ donde:}$$

UUCP: Puntos de Casos de Uso sin ajustar.

UCP: Puntos de Casos de Uso ajustados.

TCF: Factor de complejidad técnica.

EF: Factor de ambiente.

5. Factor de complejidad técnica (TCF):

Este coeficiente se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema.

Factor	Descripción	Peso	Valor Asignado	Peso*Valor Asignado
T1	Sistema Distribuido	2	0	0
T2	Objetivos de Performance o tiempo de respuesta	1	4	4
T3	Eficiencia del usuario final	1	2	2
T4	Procesamiento interno complejo	1	3	3
T5	El código debe ser reutilizable	1	5	5
T6	Facilidad de instalación	0.5	4	2
T7	Facilidad de uso	0.5	3	1.5
T8	Portabilidad	2	4	8
T9	Facilidad de cambio	1	3	3
T10	Concurrencia	1	5	5
T11	Incluye objetivos especiales de seguridad	1	5	5
T12	Provee acceso directo a terceras partes	1	0	0

T13	Se requieren facilidades especiales de entrenamiento a usuarios.	1	4	3
-----	--	---	---	---

Tabla 2.27: Factor de complejidad técnica

El Factor de Complejidad Técnica se calcula mediante la siguiente ecuación:

$$TCF = 0.6 + 0.01 \times \Sigma (\text{Peso } i \times \text{Valor asignado } i)$$

$$TCF = 0.6 + 0.01 \times \Sigma (0+4+2+3+5+2+1.5+8+3+5+5+0+3)$$

$$TCF = 0.6 + 0.01 \times 41.5$$

$$TCF = 0.6 + 0.415$$

$$TCF = 1.015$$

6. Factor de ambiente (EF):

Las habilidades y el entrenamiento del grupo involucrado en el desarrollo tienen un gran impacto en las estimaciones de tiempo.

Factor	Descripción	Peso	Valor Asignado	Peso*Valor
E1	Familiaridad con el modelo de proyecto utilizado.	1.5	0	0
E2	Experiencia en la aplicación	0.5	3	1.5
E3	Experiencia en orientación a objetos	1	4	4
E4	Capacidad del analista líder	0.5	5	2.5
E5	Motivación	1	4	4
E6	Estabilidad de los requerimientos	2	2	4
E7	Personal part-time	-1	0	0
E8	Dificultad del lenguaje de programación	-1	2	-2

Tabla 2.28: Factor de ambiente

El Factor de ambiente se calcula mediante la siguiente ecuación:

$$EF = 1.4 - 0.03 \times \Sigma (\text{Peso } i \times \text{Valor asignado } i)$$

$$EF = 1.4 - 0.03 \times \Sigma (0+1.5+4+2.5+4+4+0-2)$$

$$EF = 1.4 - (0.03 \times 14)$$

$$EF = 0.98$$

Puntos de Casos de Uso ajustados:

Al ser **UUCP** = 72, **TCF** = 1.015 y **EF** = 0.98

UCP = **UUCP** x **TCF** x **EF**, sustituyendo, tenemos:

$$UCP = 72 \times 1.015 \times 0.98$$

UCP = 73.995

Calcular el esfuerzo del Flujo de trabajo de implementación

El esfuerzo en horas-hombre viene dado por:

E = UCP x CF, donde:

E: Esfuerzo

UCP: Puntos de Casos de Uso ajustados.

CF: Factor de conversión.

Para calcular CF:

CF = 20 horas-hombre (si Total EF ≤ 2)

CF = 28 horas-hombre (si Total EF = 3 ó Total EF = 4)

CF = abandonar o cambiar proyecto (si Total EF ≥ 5)

Total EF = Cant EF < 3 (entre E1 –E6) + Cant EF > 3 (entre E7, E8)

Como **Total EF = 2 + 0**

Total EF = 2

CF = 20 horas-hombre (porque Total EF ≤ 2)

Al tener **UCP = 73.995** y **CF = 20 horas-hombre/Punto de Casos de Uso**

E = UCP x CF, sustituyendo, tenemos:

E = 73.995 x 20 horas-hombre/Punto de Casos de Uso

E = 1479.9 horas-hombre

Actividad	Porcentaje	Esfuerzo
Análisis	10 %	370.0 horas-hombre
Diseño	20 %	740.0 horas-hombre
Implementación	40 %	1479.9 horas-hombre
Prueba	15 %	555.0 horas-hombre
Sobrecarga	15 %	555.0 horas-hombre
Total	100 %	3699.9 horas-hombre

Tabla 2.29: Esfuerzo en Horas-Hombre

2.15 Aportes y Beneficios

El sistema propuesto permitirá gestionar los datos del personal docente de la Universidad de una manera muy flexible, ya que brindará la posibilidad de configurar y adaptar, según las necesidades, qué información se gestionará para cada tipo de personal, sea estudiante o profesor, en los diferentes estados por los que pueden pasar durante su estancia en el centro de estudios. Además, se podrá tener acceso en todo momento a esta información, incluso cuando las personas registradas en el sistema ya no sean parte del personal docente de la institución.

También se unificarán, en un solo proceso, todos los procesos que automatizan por separado los módulos matrícula y profesor en la versión actual del sistema, lo que eliminará las redundancias existentes y permitirá llevar a cabo estos procesos de una forma más eficiente. De esta manera se logrará un mejor control de la información del personal docente y se facilitará el trabajo de los diferentes usuarios del sistema, ya que no tendrán que realizar operaciones similares en diferentes módulos.

2.16 Conclusiones

En este capítulo se realizó un profundo análisis de los procesos que intervienen en la Gestión de Personal en la Universidad, lo que permitió obtener el conjunto de funcionalidades que debe cumplir el módulo, expresados en los requisitos funcionales y agrupados finalmente en los casos de uso del sistema. Además de que en el capítulo quedan descritos los requisitos no funcionales que deberá cumplir el sistema. También se realiza la estimación del esfuerzo lo cual sirve para predecir el tiempo de duración del proyecto.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

3.1 Introducción:

En el presente capítulo se efectuarán los modelos de análisis y diseño de los casos de usos críticos del sistema, los cuales son de suma importancia para el desarrollo del software. Además se realizan los diagramas de colaboración del análisis y de secuencia del diseño de dichos casos de uso, así como los modelos lógico y físico de la base de datos y el diagrama de despliegue.

3.2 Análisis

“Durante el análisis, se analizan los requisitos que se describen en la captura de requerimientos, refinándolos y estructurándolos. El objetivo de hacerlo es conseguir una comprensión más precisa de los requisitos y una descripción de los mismos que sea fácil de mantener y que ayude a estructurar el sistema entero, incluyendo su arquitectura” (10).

El análisis abarca las abstracciones primarias de clases y objetos, así como los mecanismos presentes en el dominio del problema, o sea del mundo real, y no se consideran clases que definen detalles y soluciones en el sistema de software. Las clases que se modelan refinan los requisitos funcionales obtenidos anteriormente, son identificadas con sus relaciones y descritas en un diagrama de clases.

3.2.1 Diagramas de Clases del Análisis

Un Diagrama de clases del análisis es un artefacto en el que se representan los conceptos en un dominio del problema. En este tipo de diagrama se representan las clases y sus relaciones. Ellos representan una vista estática del sistema.

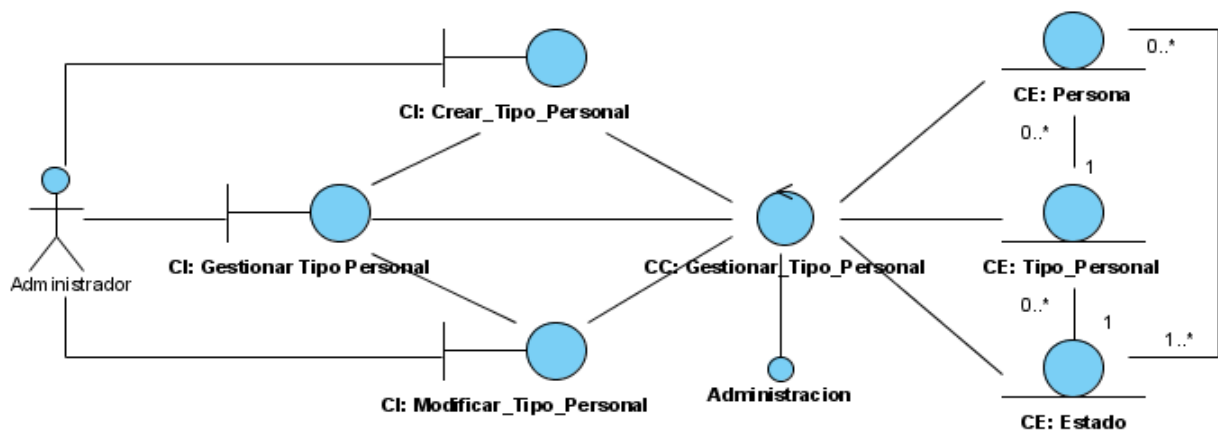


Figura 4: Diagrama de clases del análisis CUS Gestionar Tipo Personal

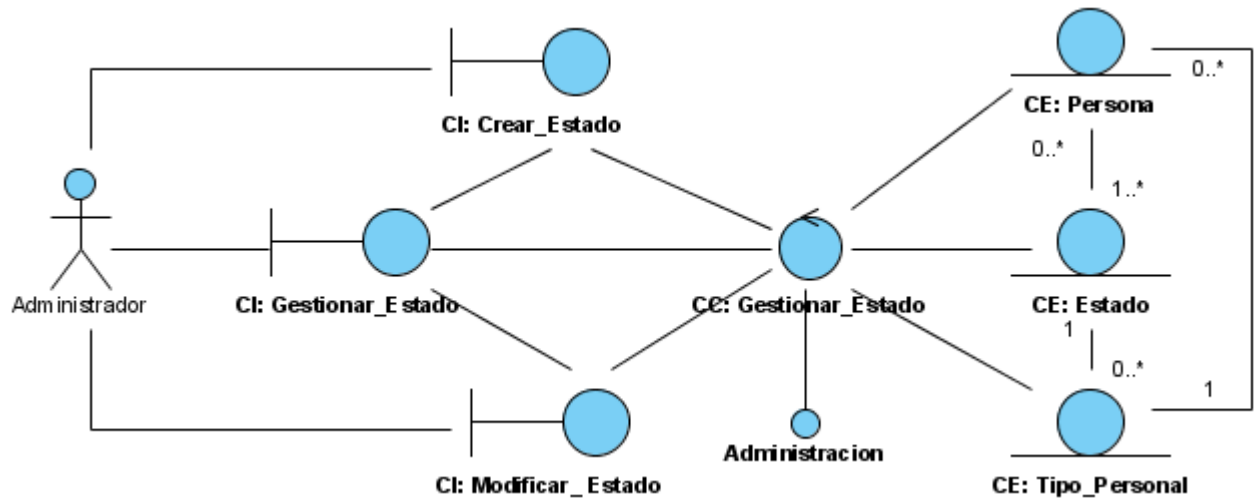


Figura 5: Diagrama de clases del análisis CUS Gestionar Estados

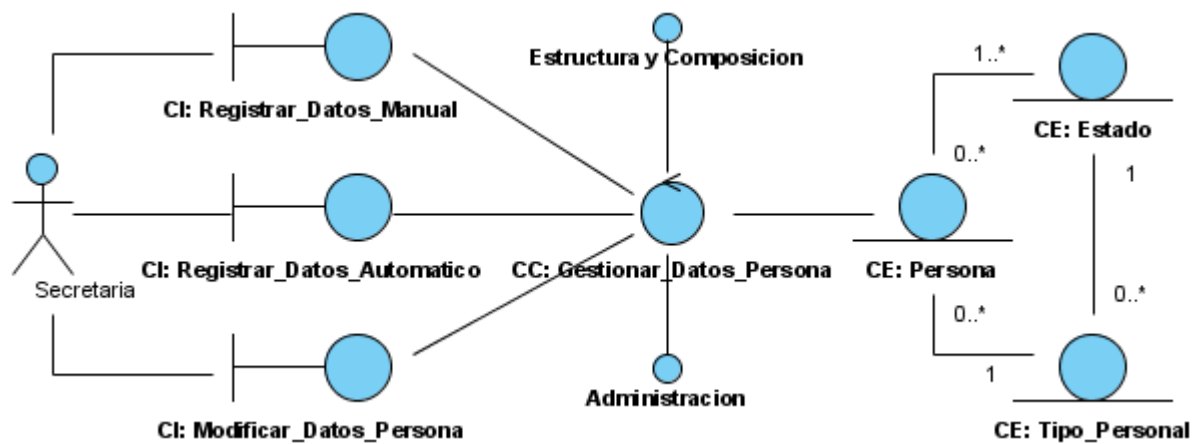


Figura 6: Diagrama de clases del análisis CUS Gestionar Datos Persona

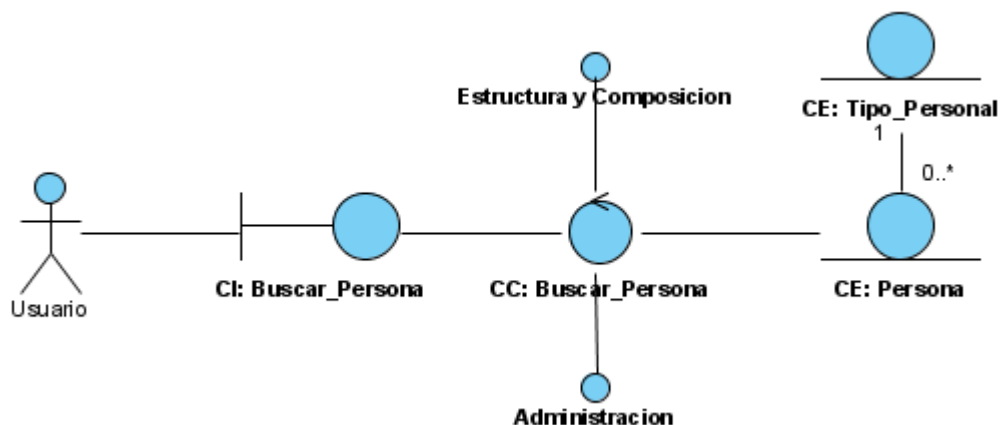


Figura 7: Diagrama de clases del análisis CUS Buscar Persona

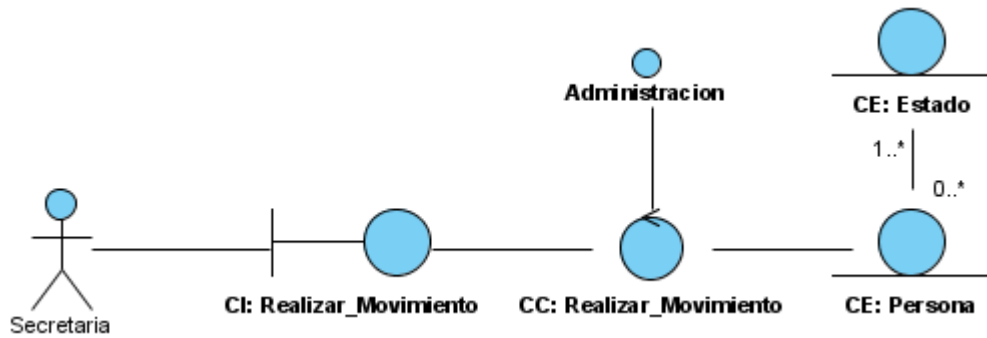


Figura 8: Diagrama de clases del análisis CUS Realizar Movimiento

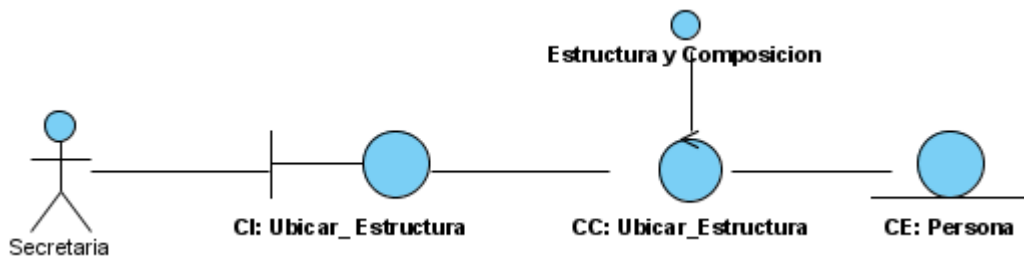


Figura 9: Diagrama de clases del análisis CUS Ubicar Persona en Estructura

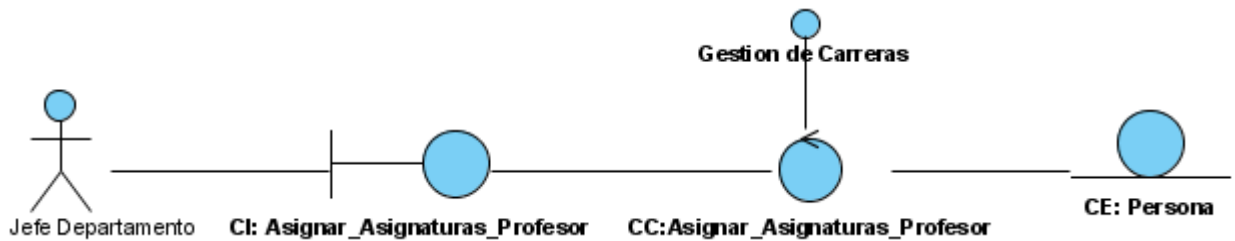


Figura 10: Diagrama de clases del análisis CUS Asignar Asignaturas Profesor

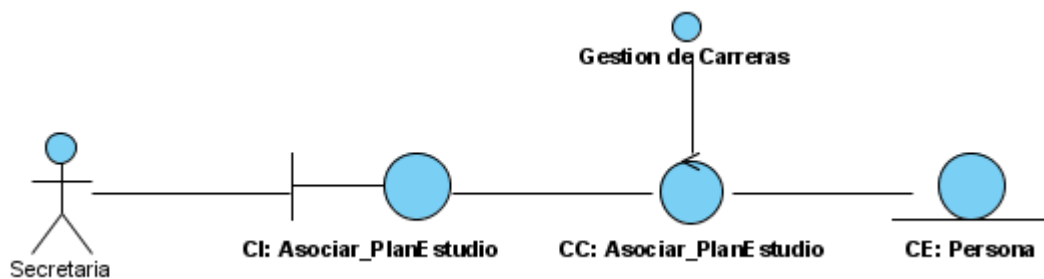


Figura 11: Diagrama de clases del análisis CUS Asociar Plan de Estudio

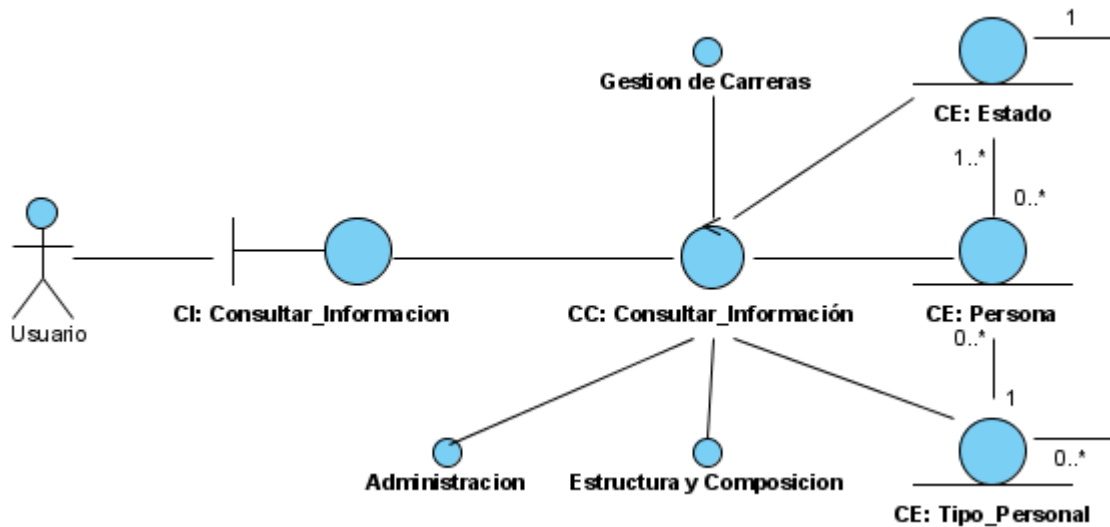


Figura 12: Diagrama de clases del análisis CUS Consultar Información

3.2.2 Diagramas de Colaboración.

Los diagramas de colaboración de los Caso de Uso del Sistema se pueden encontrar en el Anexo #3.

3.3 Diseño

En el diseño se modela el sistema y se encuentra su forma (incluida la arquitectura) para que soporte todos los requisitos, incluyendo los no funcionales y las restricciones que se le suponen. Una entrada esencial en el diseño es el resultado del análisis, que proporciona una comprensión detallada de los requisitos.

El diseño del módulo se realizará teniendo en cuenta el patrón de arquitectura Modelo Vista Controlador (MVC), los frameworks Ext JS y Zend framework, el Object Relational Mapper (ORM) Doctrine y como lenguaje de programación PHP.

3.3.1 Patrones de Diseño

Un patrón de diseño es una solución a un problema de diseño. Para que una solución sea considerada un patrón debe poseer ciertas características. Una de ellas es que debe haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores. Otra es que debe ser reusable, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias.



Patrones GOF:

Decorator: este patrón permite añadir funcionalidad a una clase dinámicamente. Zend framework implementa dicho patrón en la clase Zend_View, encargada de asignarle responsabilidades a objetos de manera dinámica y configurarlos con nuevos atributos.

Singleton: este patrón garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia. Zend Framework tiene una instancia única del controlador frontal disponible mediante este patrón para lograr una vía de entrada única a las solicitudes.

Facade: este patrón simplifica los accesos a las clases de la capa de acceso a datos proporcionando un objeto que todas las clases de capas superiores utilizarán para acceder a las clases contenidas en la capa del modelo. Define una interface de más alto nivel que permite usar el sistema más fácil. El objetivo de la aplicación de este patrón es reducir la dependencia entre clases. Se utilizará una clase intermediaria entre las clases controladoras de Zend Framework y las de acceso a datos de Doctrine, la que brindará, de las operaciones de acceso a datos, sólo las que necesiten los controladores para su funcionamiento, lo que reduce la dependencia de estos entre las múltiples clases de acceso a datos existentes en el sistema.

Factory: este patrón proporciona una interfaz para crear familias de objetos relacionados o dependientes sin especificar su clase concreta. Permite configurar en tiempo de ejecución un sistema con una familia u otra de objetos. Además garantiza que un conjunto de clases se usen a la vez. Zend Framework provee una API para el acceso a datos conformada por un conjunto de clases que implementa dicho patrón.

Patrones de acceso a datos:

Active Record: este patrón representa de forma Orientada a Objetos los datos de una Base de Datos Relacional, definiendo interfaces sencillas para acceder y manipular esos datos. Es un enfoque al problema de acceder a los datos de una base de datos. Una fila en la tabla de la base de datos (o vista) se envuelve en una clase, de manera que se asocian filas únicas de la base de datos con objetos del lenguaje de programación usado. Cuando se crea uno de estos objetos, se añade una fila a la tabla de la base de datos. Cuando se modifican los atributos del objeto, se actualiza la fila de la base de datos. Doctrine usa este patrón para manejar la base de datos.



Row Data Gateway: este patrón asume el comportamiento de un objeto que actúa como puerta de enlace a una fila de una tabla de la base de datos. Tiene propiedades que reflejan las columnas de la tabla, y métodos de actualización en la base. Zend Framework provee una API para el acceso a datos conformada por un conjunto de clases que implementa dicho patrón.

Table Data Gateway: similar a Row Data Gateway, pero a diferencia de este define la estructura de acceso por registros en las entidades de la base de datos, específica su acceso a nivel de tabla, proponiendo un objeto que se comporte como puerta de enlace a cada tabla de la base de datos. Contiene una interface que permite actualizar, buscar, borrar e insertar en la tabla y puede retornar un registro, un grupo de registro, y hasta un objeto del dominio. Zend Framework provee una API para el acceso a datos conformada por un conjunto de clases que implementa dicho patrón.

Patrones de asignación de responsabilidades (GRASP)

Controlador: este patrón se tiene en cuenta para realizar las asignaciones en cuanto al manejo de los eventos del sistema y definir sus operaciones. Zend Framework contribuye a la utilización de este patrón ya que define un Controlador Frontal (Front Controller) que implica que todas las solicitudes son dirigidas a un único script PHP que se encarga de instanciar al controlador frontal y redirigir las llamadas.

Experto: este patrón propone como solución asignar la responsabilidad a la clase que cuenta con la información necesaria para cumplir la responsabilidad. Las clases que brinda el framework Ext JS se encargarán de visualizar las interfaces ya que cuentan con la información para crear los diferentes componentes visuales, las clases controladoras del Zend Framework manejarán las peticiones del cliente, y las clases que genera y utiliza el Doctrine serán las encargadas del acceso a datos pues contienen y representan los datos que manejará el sistema. Permitiendo que se conserve el encapsulamiento, soportando un bajo acoplamiento y una alta cohesión.

Creador: este patrón se tiene en cuenta para la asignación de responsabilidades a las clases relacionadas con la creación de objetos, de forma tal que una instancia de un objeto solo pueda ser creada por el objeto que contiene la información necesaria para ello. Existe un único script PHP que se encarga de instanciar al controlador frontal, este último es el encargado de instanciar las clases controladoras y estas, a su vez, instancian objetos de la clase Zend_View. El uso de este patrón permite crear las dependencias mínimas necesarias entre las clases, favoreciendo al mantenimiento del sistema.

Bajo Acoplamiento: este patrón brinda como solución asignar responsabilidades de manera que las clases no dependan fuertemente de otras. Ofreciendo como beneficio que son fáciles de entender por separadas, fáciles de reutilizar y no se afectan por cambios de otros componentes. Dicho patrón se tiene en cuenta debido a la importancia de realizar un diseño de clases independientes que soporten los cambios.

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.

3.3.2 Diagramas de Clases del Diseño.

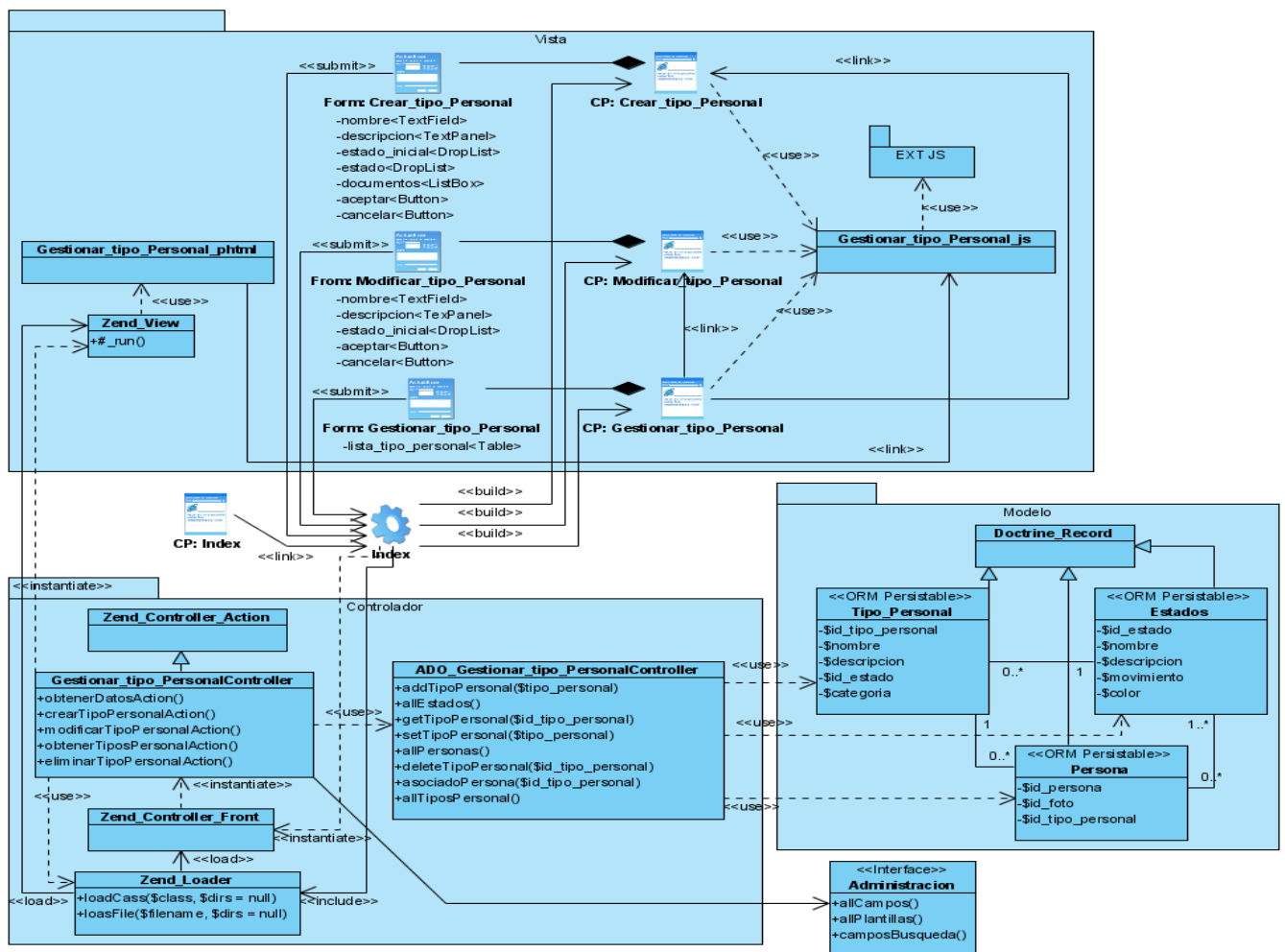


Figura 13: Diagrama de clases del diseño CUS Gestionar Tipo Personal

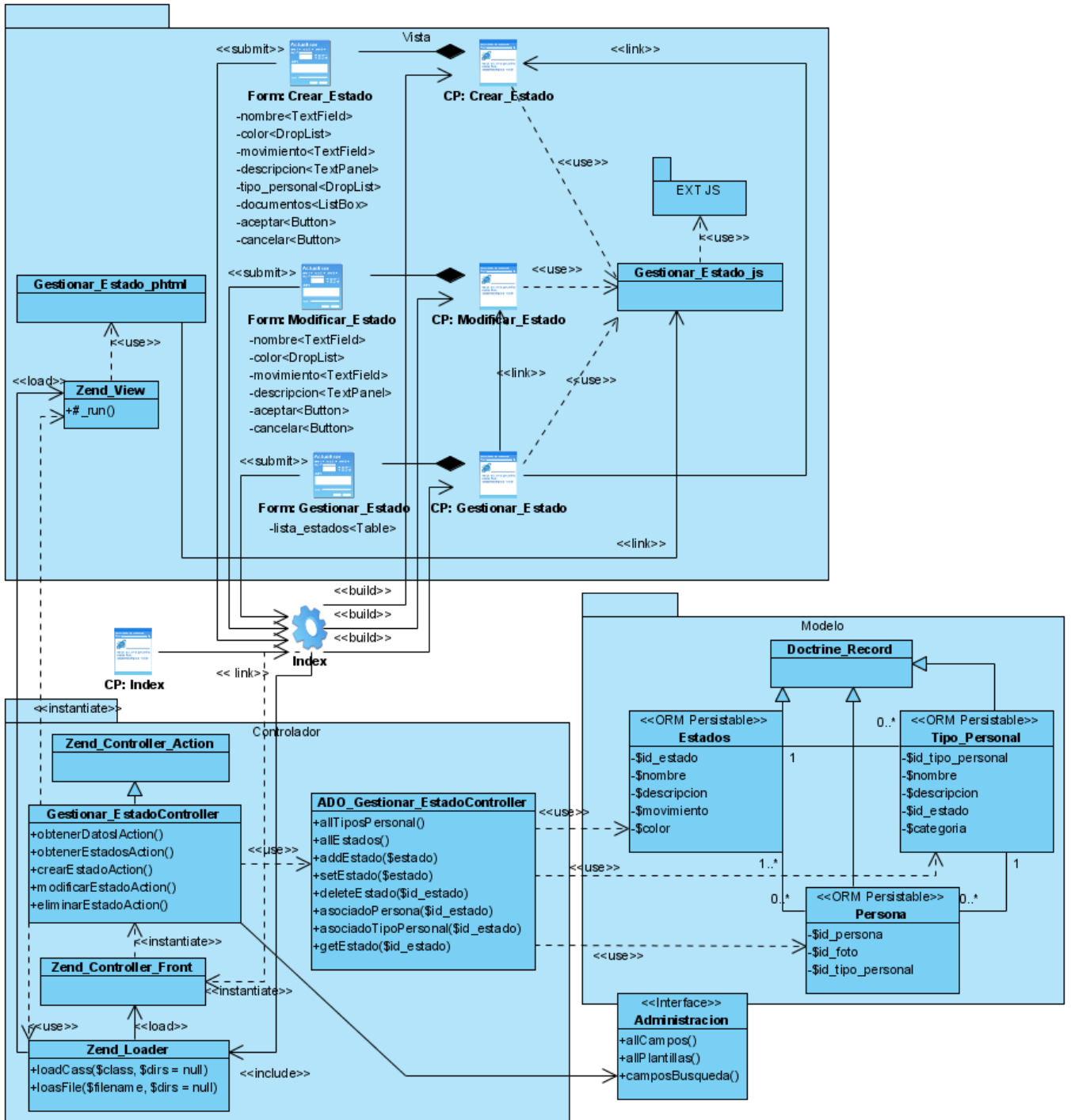


Figura 14: Diagrama de clases del diseño CUS Gestionar Estados

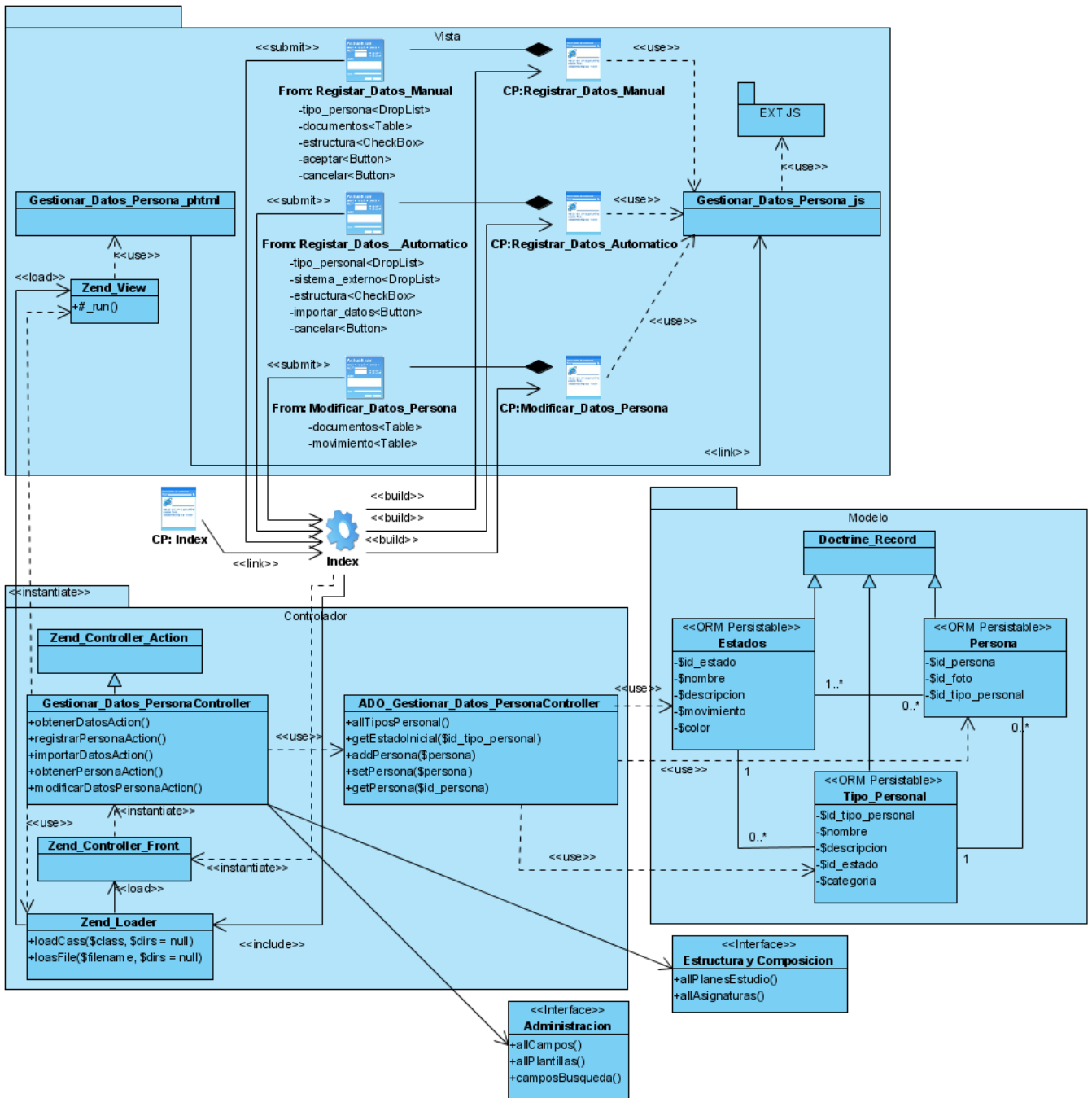


Figura 15: Diagrama de clases del diseño CUS Gestionar Datos Persona

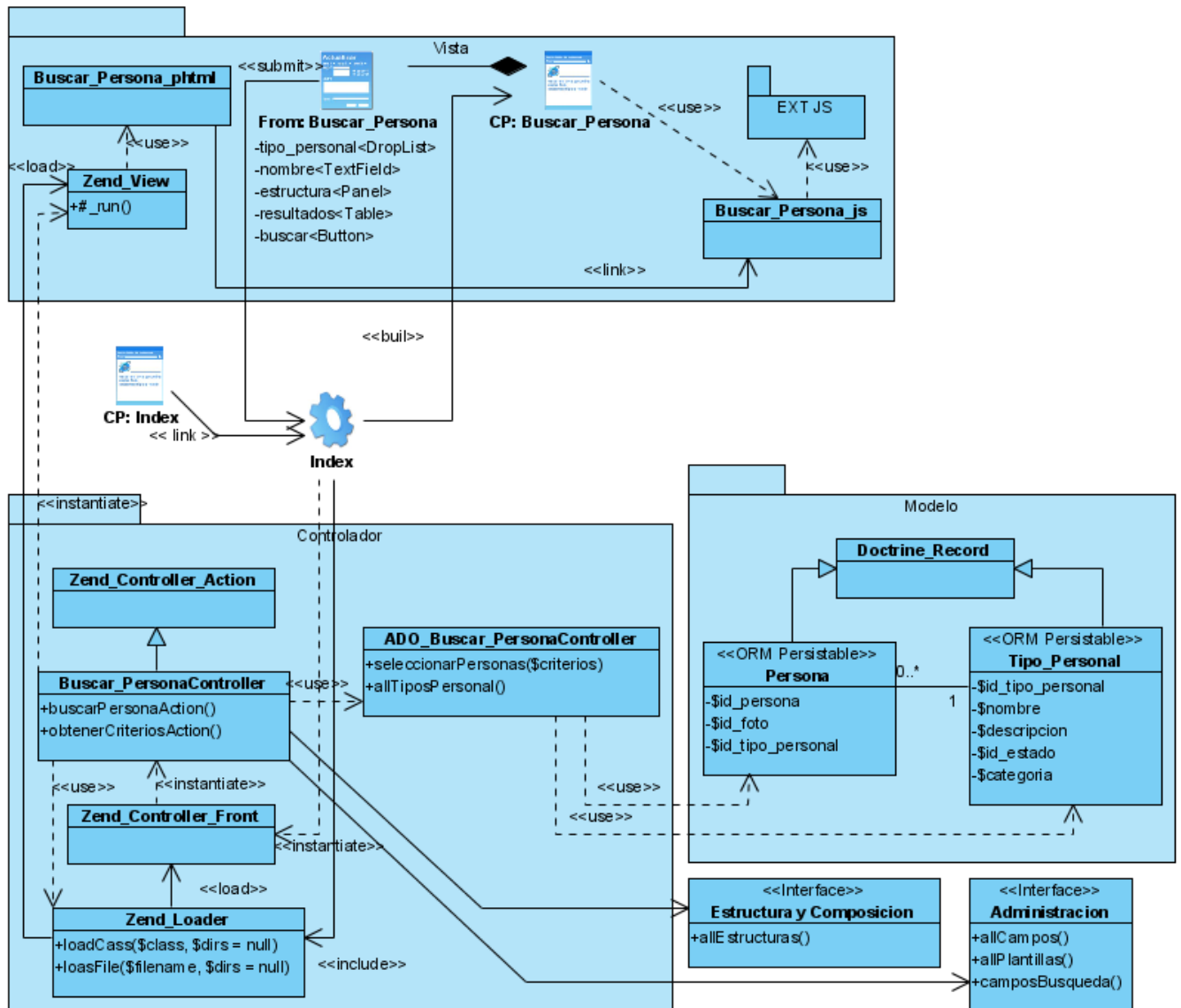


Figura 16: Diagrama de clases del diseño CUS Buscar Persona

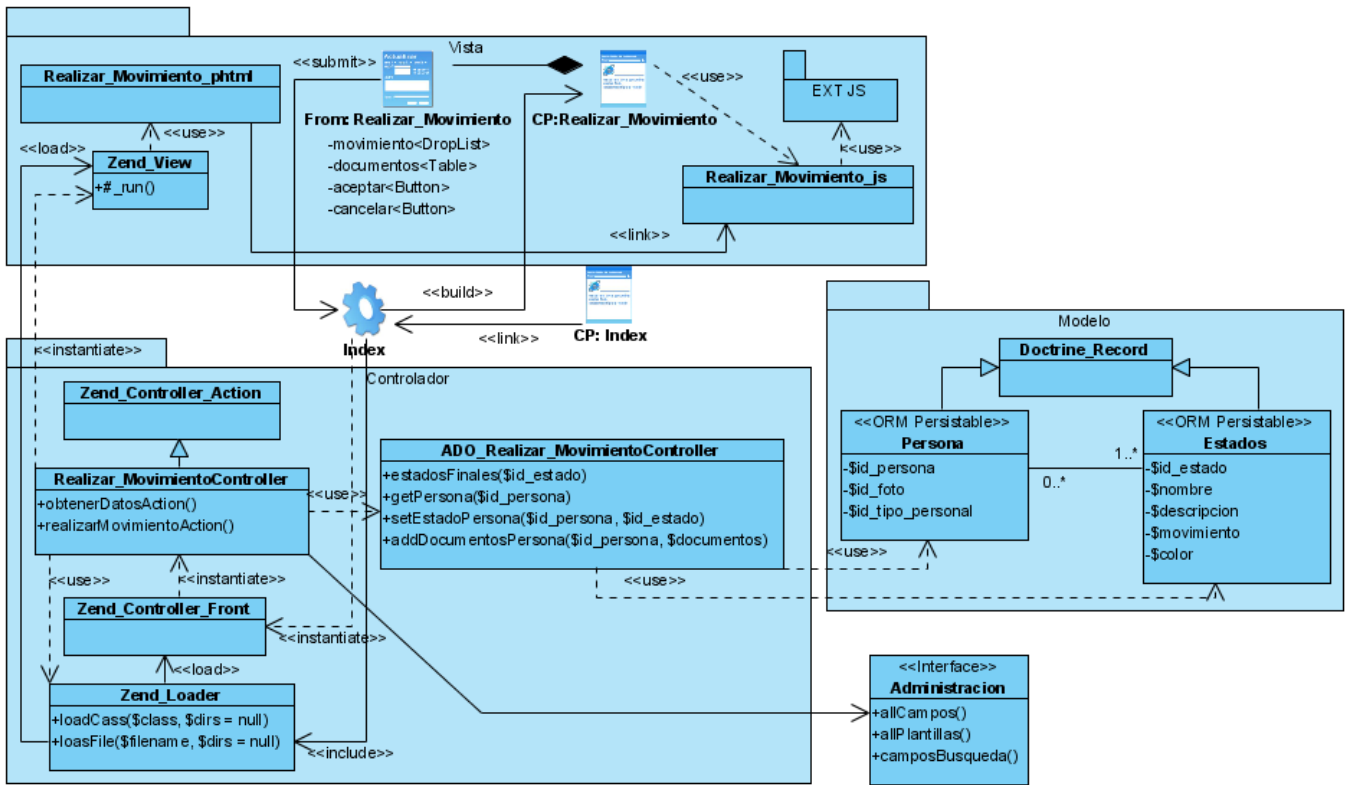


Figura 17: Diagrama de clases del diseño CUS Realizar Movimiento

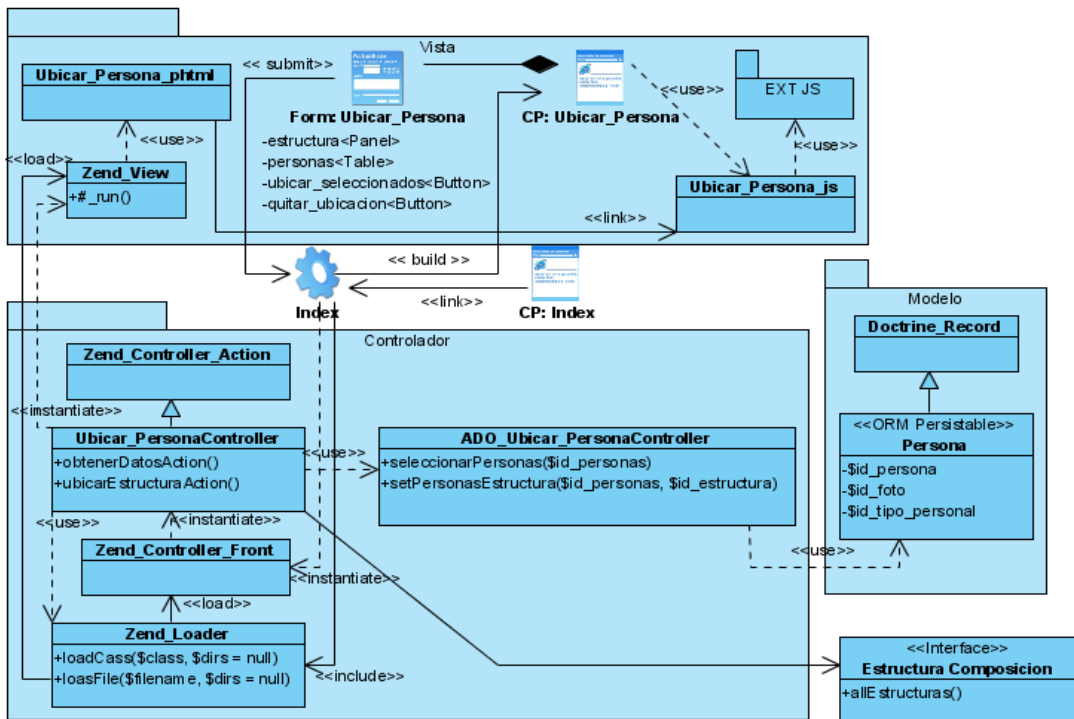


Figura 18: Diagrama de clases del diseño CUS Ubicar Persona en Estructura

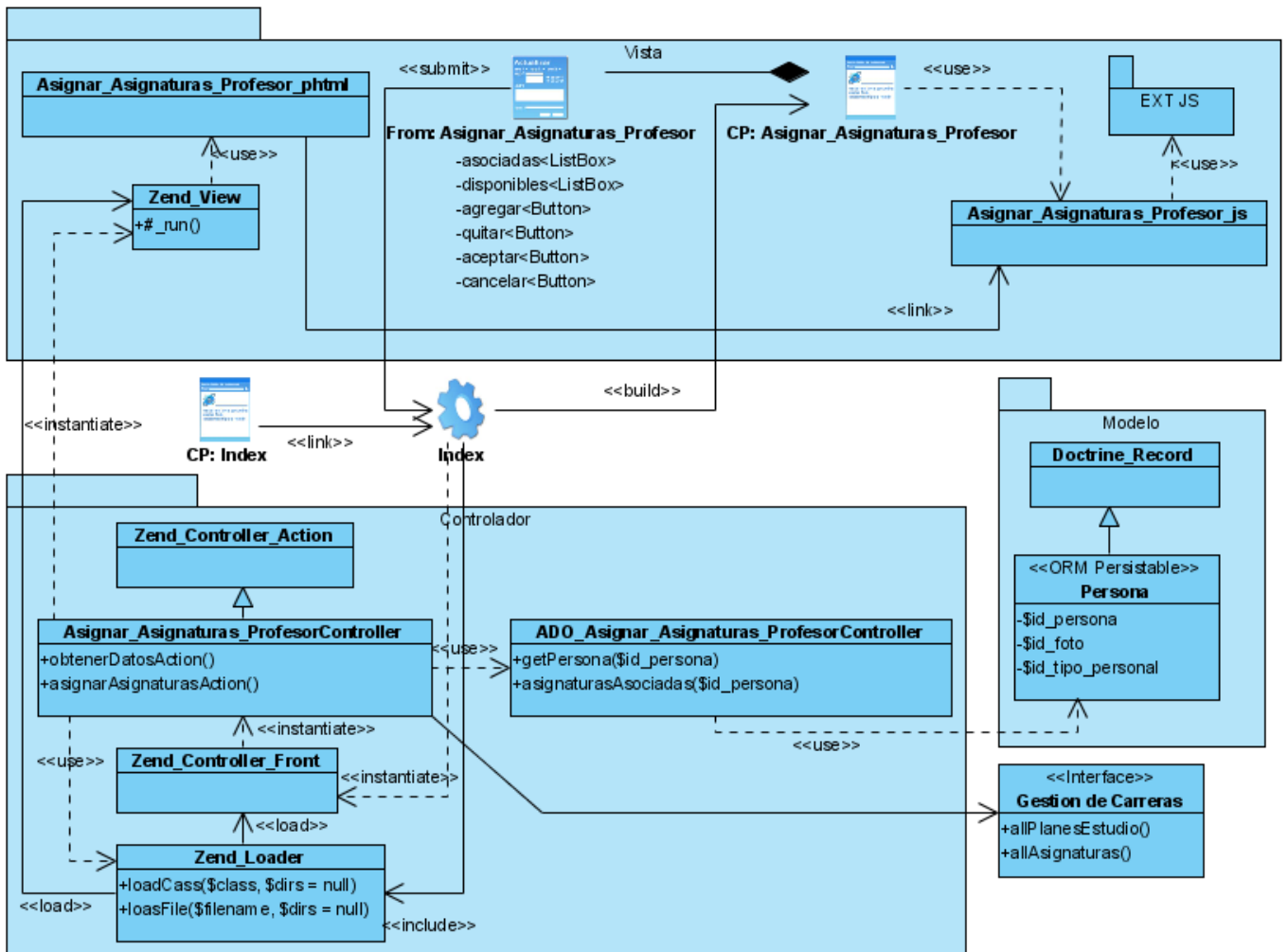


Figura 19: Diagrama de clases del diseño CUS Asignar Asignaturas Profesor

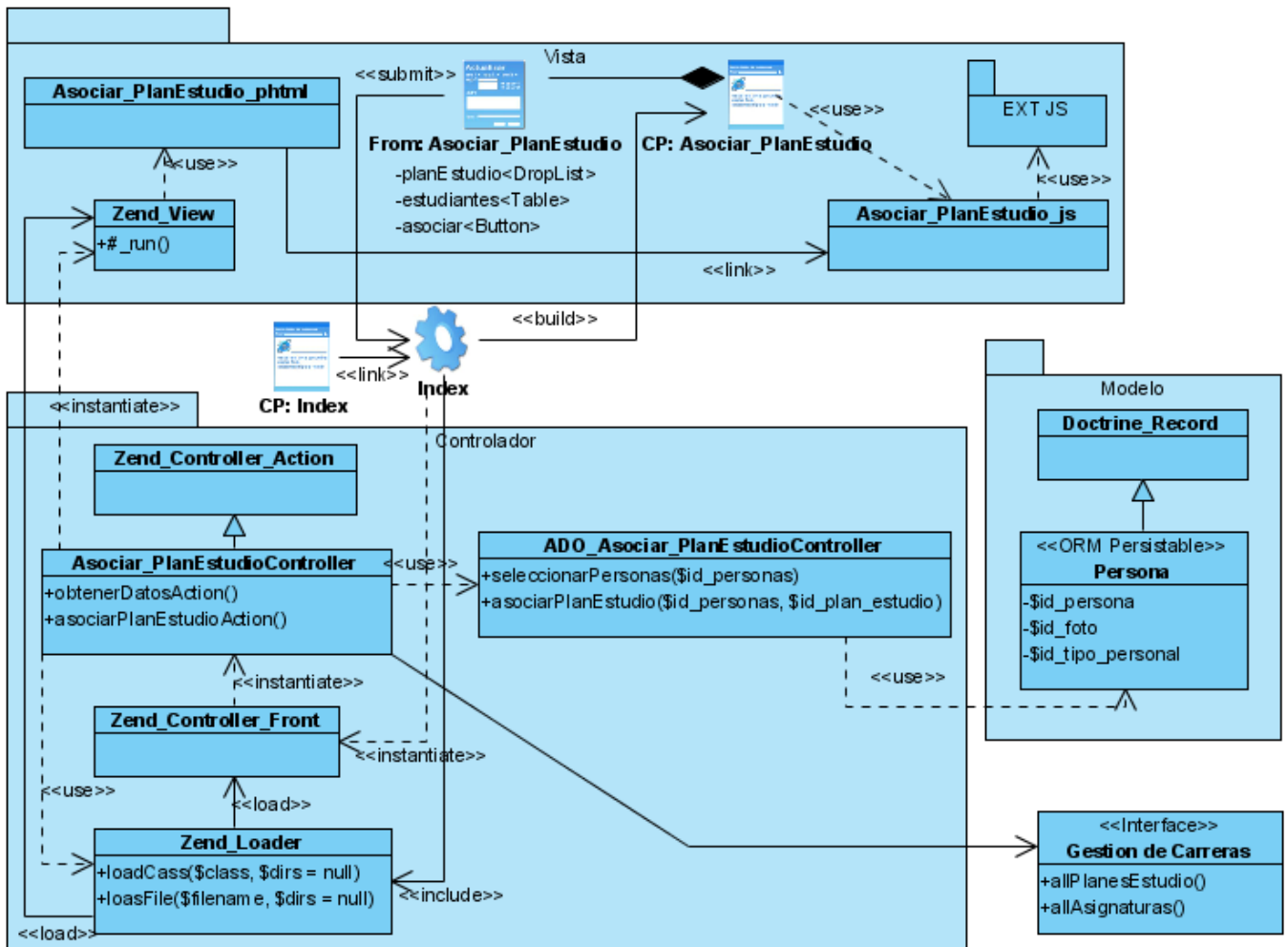


Figura 20: Diagrama de clases del diseño CUS Asociar Plan de Estudio

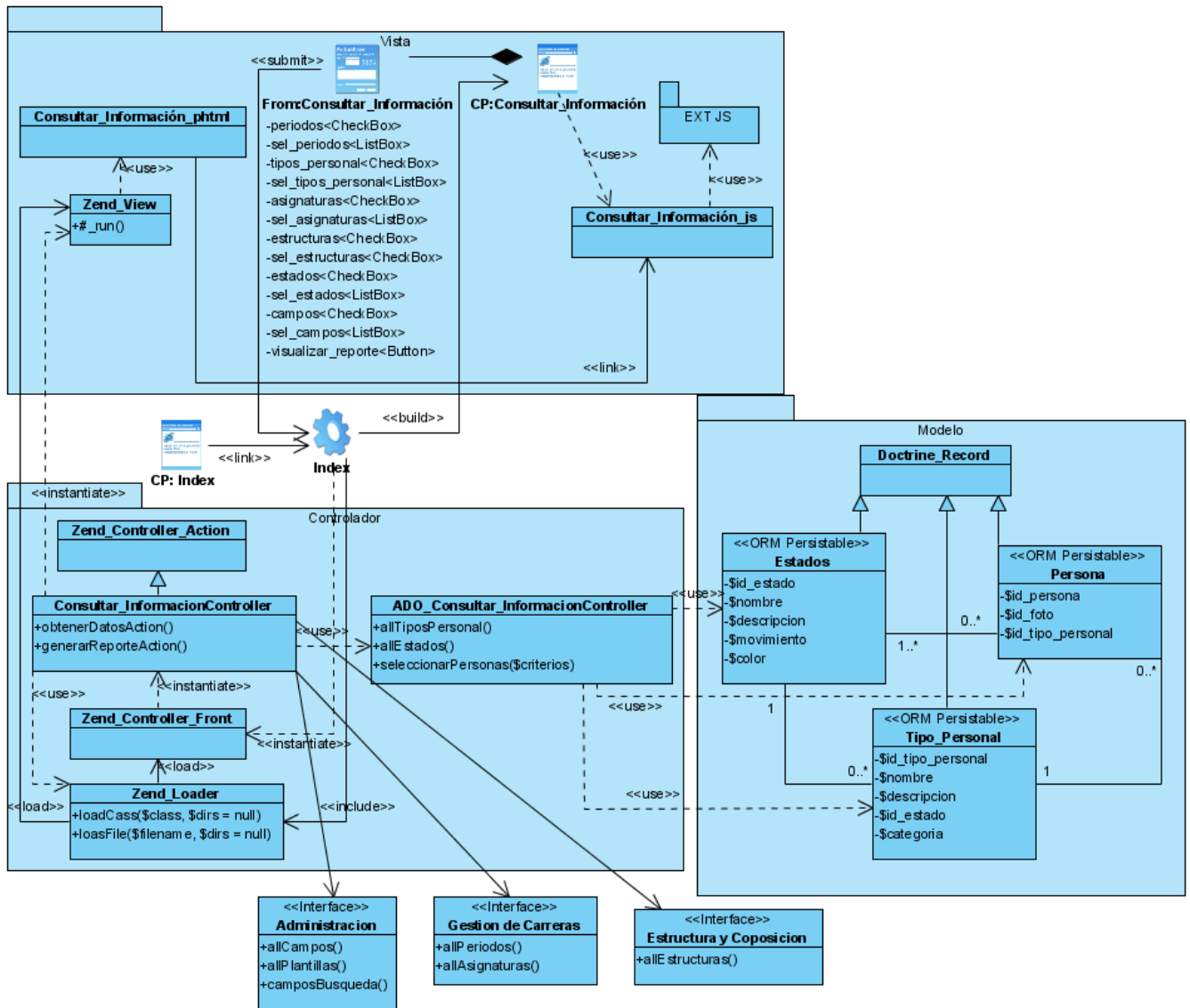


Figura 21: Diagrama de clases del diseño CUS Consultar Información

3.3.3 Diagramas de Secuencia

Los diagramas de secuencia de los Casos de Uso del Sistema se pueden encontrar en el Anexo #4.

3.4 Diseño de la BD

En el desarrollo de un sistema informático, el diseño de la BD es de gran importancia, ya que en ella se almacenan todos los datos que son necesarios en la modelación del problema que se desea resolver, además ésta es la fuente de obtención de toda la información que se quiera recuperar del sistema. Las bases de datos necesitan de una definición de su estructura que le permitan almacenar datos, reconocer el contenido, y recuperar la información.

3.4.1 Modelo Lógico

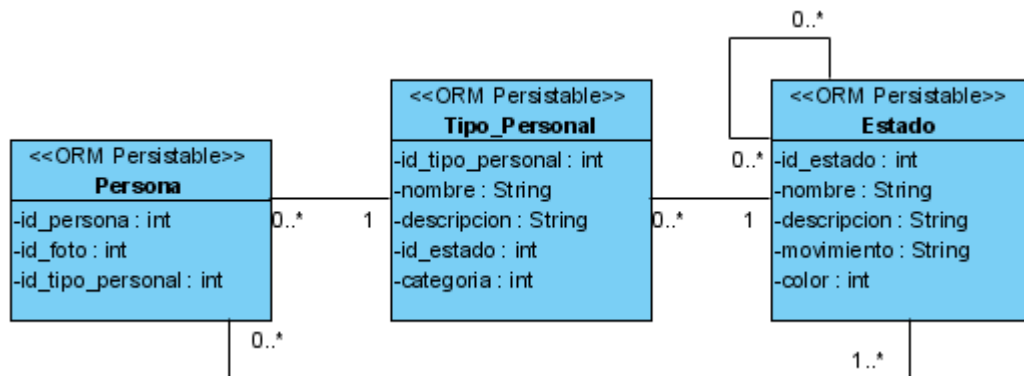


Figura 22: Modelo Lógico de Datos

3.4.2 Modelo Físico

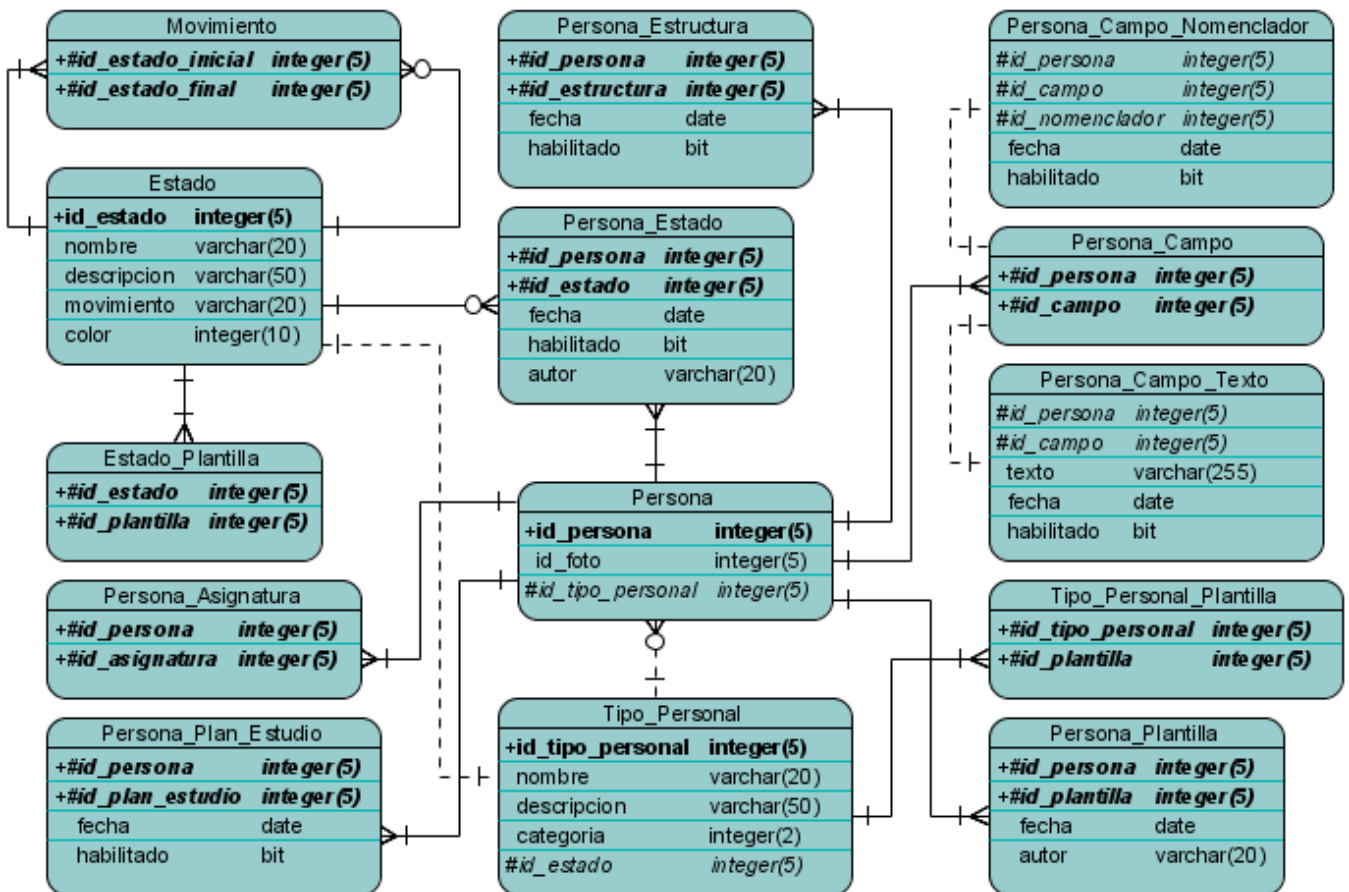


Figura 23: Modelo Físico de Datos

3.5 Diagrama de Despliegue

A continuación se muestra la distribución física del sistema teniendo en cuenta la arquitectura del software y del hardware.

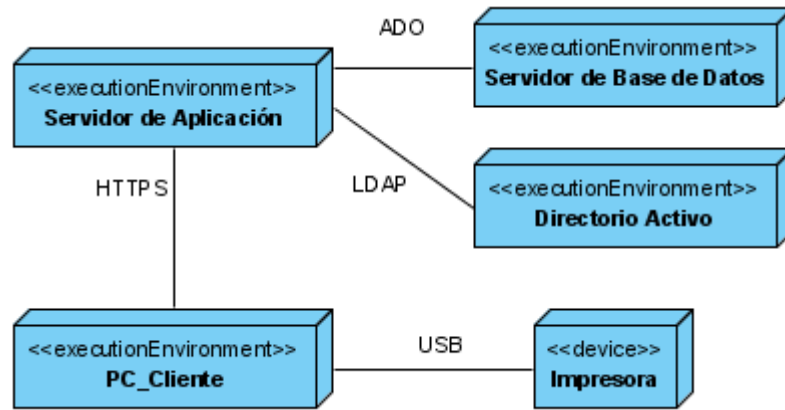


Figura 24: Diagrama de Despliegue

3.6 Conclusiones

En el presente capítulo se abordaron los aspectos básicos relacionados con el análisis y diseño del sistema, además se mostraron los diagramas correspondientes: los de clases del análisis y de clases del diseño así como los diagramas de colaboración y secuencia para cada una de las funcionalidades descritas. Se describe la arquitectura y los patrones que se tuvieron en cuenta para el diseño de los casos de uso. Además, se muestran los modelos físico y lógico de la base de datos los cuales son de gran importancia para la implementación del sistema, así como el diagrama de despliegue.

CONCLUSIONES

Con el desarrollo de este trabajo se arriba a las siguientes conclusiones:

- Se llevó a cabo un profundo estudio de los sistemas de gestión académica en Cuba y en el mundo.
- Se realizaron entrevistas para conocer cómo se llevan a cabo los procesos de Gestión de Personal en la Universidad.
- Se estudiaron las tecnologías, metodologías, lenguajes y herramientas necesarias para realizar el análisis, diseño y posterior desarrollo del sistema.
- Se realizó un estudio de la arquitectura propuesta.
- Se obtuvieron los artefactos necesarios, según la metodología de desarrollo de software seleccionada (RUP), para implementar la propuesta de solución.

Por todo lo antes expuesto se concluye que se le ha dado cumplimiento al objetivo general de la investigación: Elaborar el análisis y diseño del módulo Gestión de Personal para Akademos v2.0, y que está todo listo para pasar a la próxima fase de desarrollo.

RECOMENDACIONES

A partir del estudio realizado en la presente investigación y teniendo en cuenta las ideas que surgieron durante el desarrollo del análisis y diseño del módulo, se realizan las siguientes recomendaciones:

- Continuar investigando sobre los sistemas de gestión académica en Cuba y el mundo, sobre todo enfatizar en cómo se lleva a cabo la Gestión de Personal en dichos sistemas, para garantizar mejoras en las próximas fases de desarrollo.
- Realizar la implementación del módulo teniendo en cuenta la propuesta de diseño obtenida en la presente investigación.
- Estudiar con más profundidad los frameworks y el ORM que se utilizarán en el desarrollo del sistema, para de esta manera lograr una exitosa implementación del módulo.

REFERENCIAS BIBLIOGRÁFICA

1. Servicio de Informática, aplicaciones corporativas, SIGA. Servicio de Informática. [En línea] [Citado el: 01 de febrero de 2009.] <http://www.gestion.uco.es/gestion/aplicaciones/siga>.
2. Rodríguez, Carlos Moreno. Las esferas socioculturales del Software Libre. [En línea] 2004. [Citado el: 05 de febrero de 2009.] <http://www.derecho-internet.org/node/293>.
3. Departamento de Sistemas Informáticos, Universidad Politécnica de Valencia. Portal de Desarrollo de Software. [En línea] [Citado el: 08 de febrero de 2009.] <https://pid.dsic.upv.es/C1/Material/Documentos%20Disponibles/Introducci%C3%B3n%20a%20RUP.doc>.
4. Beck, K. Extreme Programming Explained. Embrace Change. Pearson Education, 1999. *Traducido al español como: "Una explicación de la programación extrema. Aceptar el cambio", Addison Wesley, 2000.*
5. Popkin Software and Systems. Modelado de sistemas con UML. Sitio Web de TLDP-ES. [En línea] Servicios editoriales para la documentación libre en español... [Citado el: 20 de enero de 2009.] <http://es.tldp.org/Tutoriales/doc-modelado-sistemas-UML/multiple-html/>.
6. Arias, Yuniel Eliades Proenza. *Diseño Avanzado de Aplicaciones Web. EXT, Zen Framework y Doctrine*. nueve, Ciudad de La Habana, Cuba. : s.n., 9 de marzo de 2009.
7. Álvarez, M. A. Lenguaje de Marcas: HTML. [En línea] 2006. [Citado el: 15 de marzo de 2009.] <http://www.webestilo.com/html/cap1a.phtml>.
8. Pérez, Javier Eguíluz. Introducción a AJAX. [En línea] 7 de junio de 2008. [Citado el: 22 de febrero de 2009.] <http://www.librosweb.es/ajax>.
9. Clase Teórico-Práctica #6. Fase de Inicio. Flujo de Trabajo Gestión de proyecto. [En línea] [Citado el: 03 de mayo de 2009.] <http://teleformacion.uci.cu/mod/resource/view.php?id=12606>.
10. Ivar Jacobson, G. B., James Rumbaugh. *El Proceso Unificado de Desarrollo del Software*. s.l. : Félix Varela 2004.

BIBLIOGRAFÍA

1. Departamento de Informática del Centro de la UNED de Barbastro. *Akados.net. UNED Barbastro*. [En línea] [Citado el: 20 de enero de 2009.] <http://www.barbastro.unedragon.org/proyectos/default.aspx>.
2. Desarrollo Web. Lenguajes del lado servidor. [En línea] [Citado el: 15 de diciembre de 2008.] <http://www.desarrolloweb.com/articulos/243.php>.
3. Ingeniería de Software y su relación con las Herramientas Case. [En línea] [Citado el: 30 de enero de 2009.] http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/rea_c_ji/capítulo2.pdf.
4. Kherian Soft. *Soluciones informáticas para el sector docente*. [En línea] [Citado el: 08 de enero de 2009.] http://www.kherian.com/agora_ediciones.asp.
5. PosgretSQL. Advantages. [En línea] [Citado el: 05 de enero de 2009.] <http://www.postgresql.org/about/advantages>.
6. Soluciones Informáticas para el Sector Docente: AGORA. [En línea] 2008. [Citado el: 01 de marzo de 2009.] <http://www.kherian.com/mexico/default.asp>.
7. Wikipedia. Proceso Unificado de Rational. [En línea] 2008. [Citado el: 10 de enero de 2009.] http://es.wikipedia.org/wiki/Proceso_Unificado_de_Rational.
8. Álvarez, M. A. Lenguajes del lado del cliente. [En línea] 2007. [Citado el: 20 de diciembre de 2008.] <http://www.desarrolloweb.com>.
9. Crescencio García Rubio, Félix Oscar y Bravo Santos. Metodologías de Desarrollo de Software. [En línea] 2007 de diciembre de 2007. [Citado el: 20 de febrero de 2009.] http://alarcos.inf-cr.uclm.es/per/fgarcia/isoftware/doc/tema9_2xh.pdf.
10. International, V. P. Visual Paradigm for UML. [En línea] 2008. [Citado el: 25 de enero de 2009.] <http://www.visual-paradigm.com/product/vpuml/>.
11. Larman, C. *UML Y Patrones. Introducción al análisis y diseño orientado a objetos*. México : Félix Varela 2004.
12. Latina, O. ¿Porqué es importante UML? [En línea] 2008. [Citado el: 07 de marzo de 2009.] <http://www.osmosislatina.com/lenguajes/uml/basico.htm>.
13. Marrero, D. P. C. J. L. A. O. Y. S. R. H. E. Manual de Usuario de SIGENU Sistema de Gestión de la Nueva Universidad. [En línea] abril de 2007.
14. Pressman, R. S. *Ingeniería de Software. Un enfoque práctico*. cuarta. Madrid : s.n., 1998. pág. 171.
15. Pavón, Eduardo León. Visual Paradigm, una herramienta de lo más útil. [En línea] 2007. [Citado el: 03 de mayo de 2009.] <http://slion2000.blogspot.com>.



16. Jesús García Molina, M. José Ortín, Begoña Moros, Joaquín Nicolás, Ambrosio Toval. De los Procesos del Negocio a los Casos de Uso. [En línea] [Citado el: 25 de febrero de 2009.] http://www.willydev.net/descargas/willydev_modeladodenegocio.pdf.
17. Gracia, Joaquín. Patrones de diseño. *Diseño de Software Orientado a Objetos*. [En línea] 27 de mayo de 2005. [Citado el: 03 de mayo de 2009.] <http://www.ingenierossoftware.com/analisisydiseno/patrones-diseno.php>.
18. Molpeceres, Alberto. Procesos de desarrollo: RUP, XP y FDD. [En línea] [Citado el: 20 de abril de 2009.] [http://www.javahispano.org/contenidos/es/procesos_de_desarrollo/..](http://www.javahispano.org/contenidos/es/procesos_de_desarrollo/)
19. Leopoldo, Carlos. Zend Framework, una introducción. *Un simple blog personal*. [En línea] 2007 de noviembre de 27. [Citado el: 10 de mayo de 2009.] <http://www.carlosleopoldo.com/post/zend-framework-una-introduccion/>.
20. Marcello Visconti, Hernán Astudillo. Fundamentos de Ingeniería de Software. [En línea] [Citado el: 05 de abril de 2009.] <http://www.inf.utfsm.cl/~visconti/ili236/Documentos/08-Patrones.pdf>.
21. Guerrero, Hanz Cocchi. Libro UML y patrones de diseno-h@nz ...el Geek. [En línea] [Citado el: 27 de febrero de 2009.] <http://hancocchi.net/>.
22. Allen, Rob. Desarrollo de Aplicaciones con Zend Framework. [En línea] 2006. [Citado el: 15 de mayo de 2009.] www.akrobat.com.
23. Zend Framework Documentation. [En línea] 2006. [Citado el: 12 de abril de 2009.] <http://framework.zend.com/manual/>.

Anexo #1: Diagramas de Actividades de los Casos de Uso del Negocio

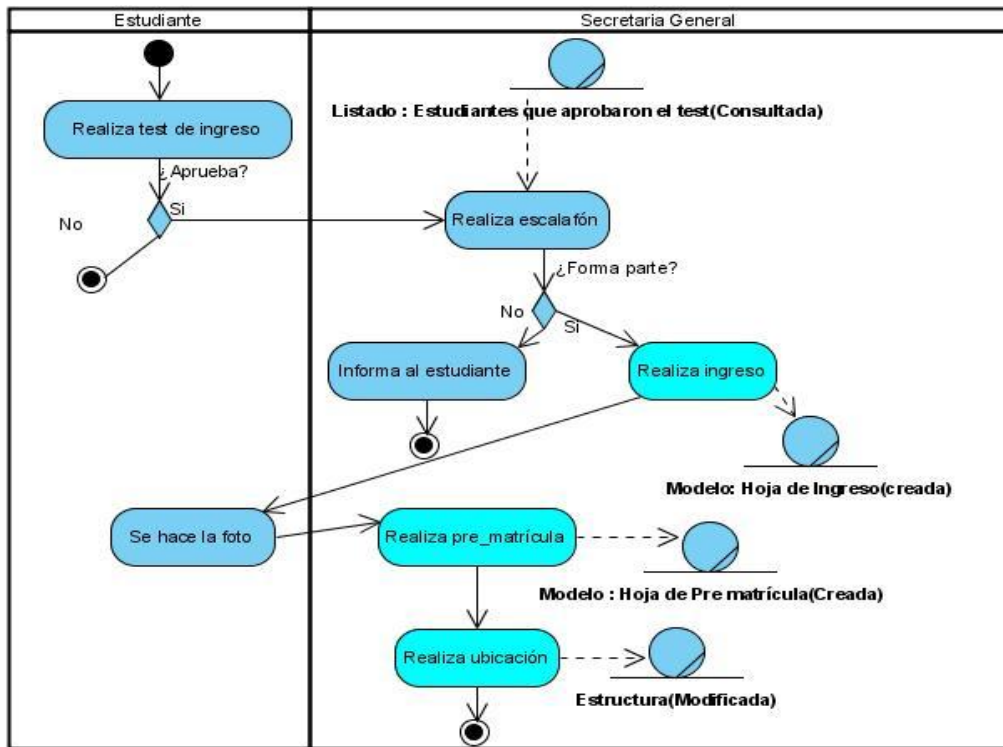


Figura 25: DA CUN_Prematricular

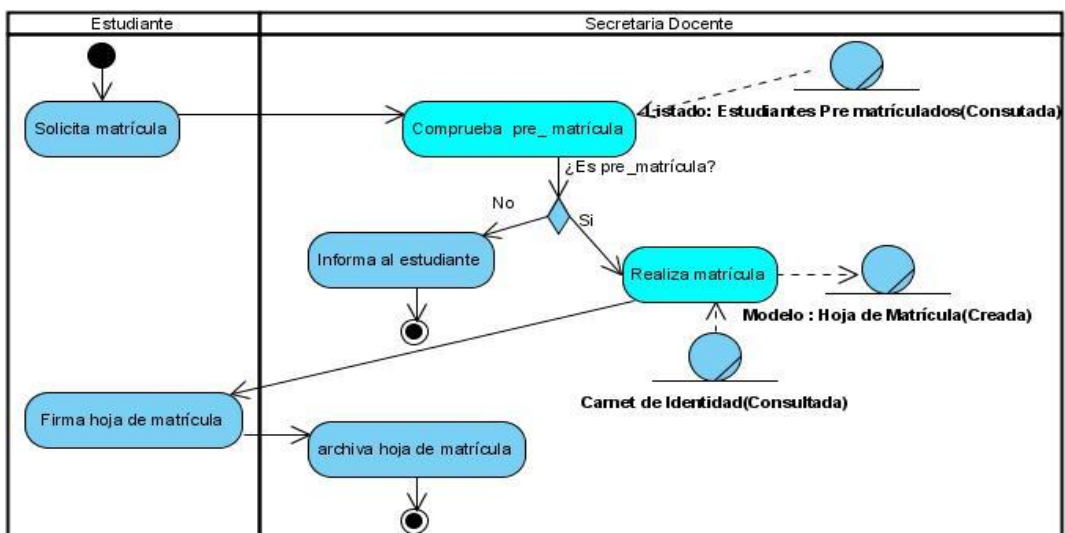


Figura 26: DA CUN_Solicitar Matricula

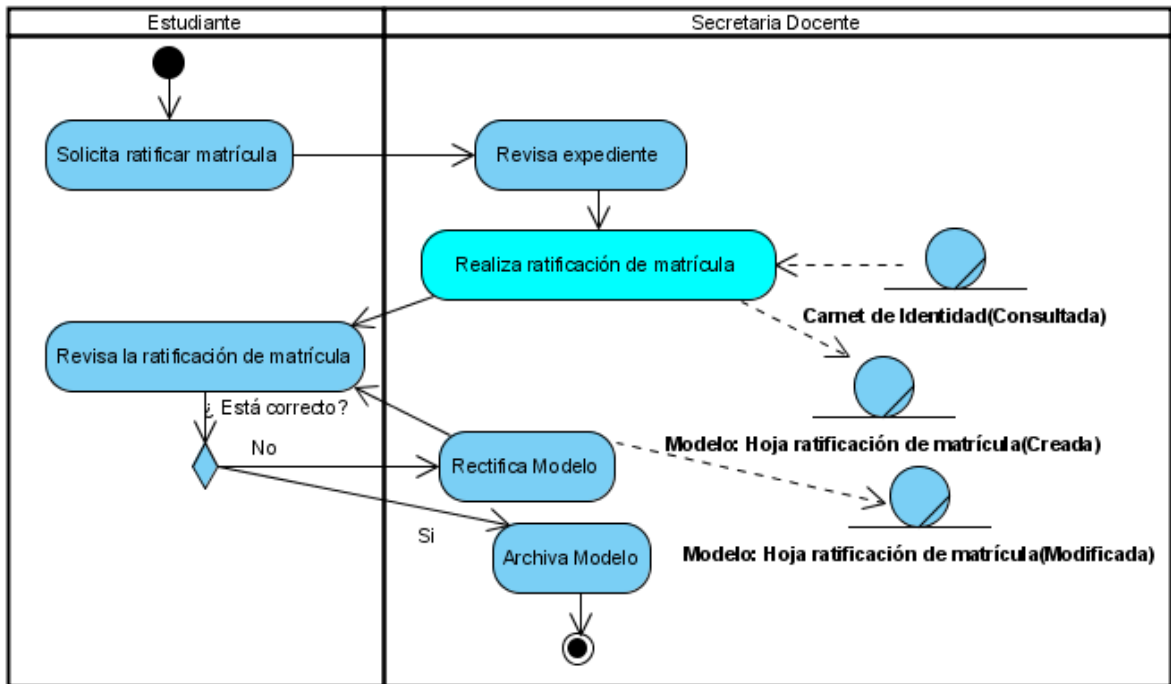


Figura 27: DA CUN_Ratificar Matrícula

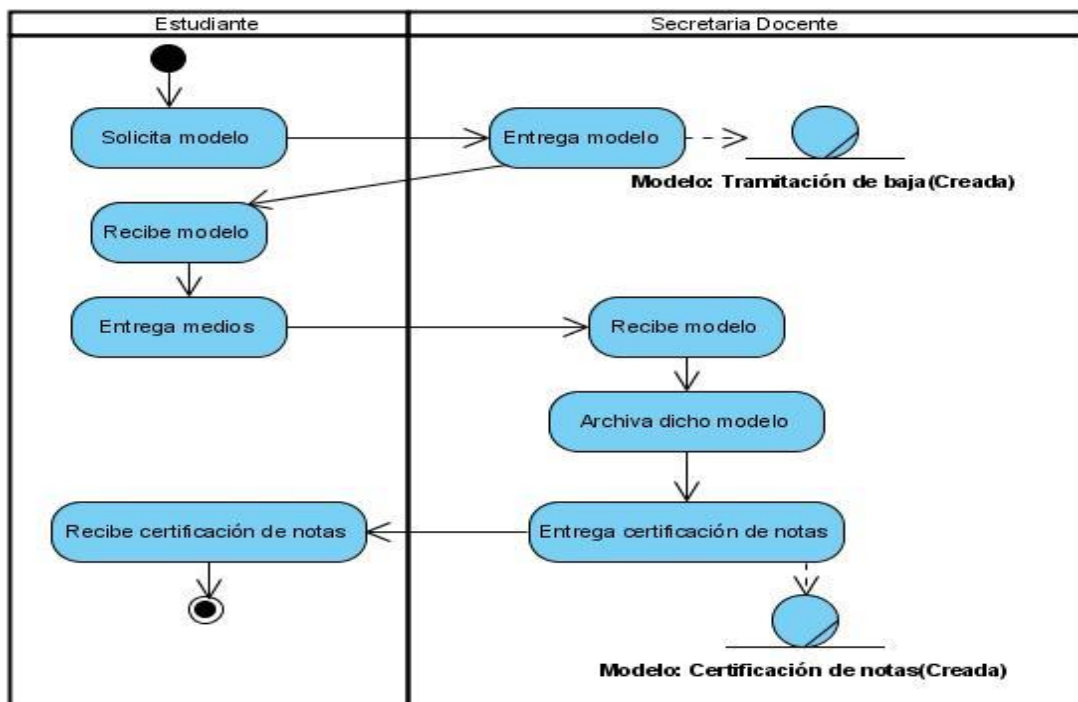


Figura 28: DA CUN_Egresar

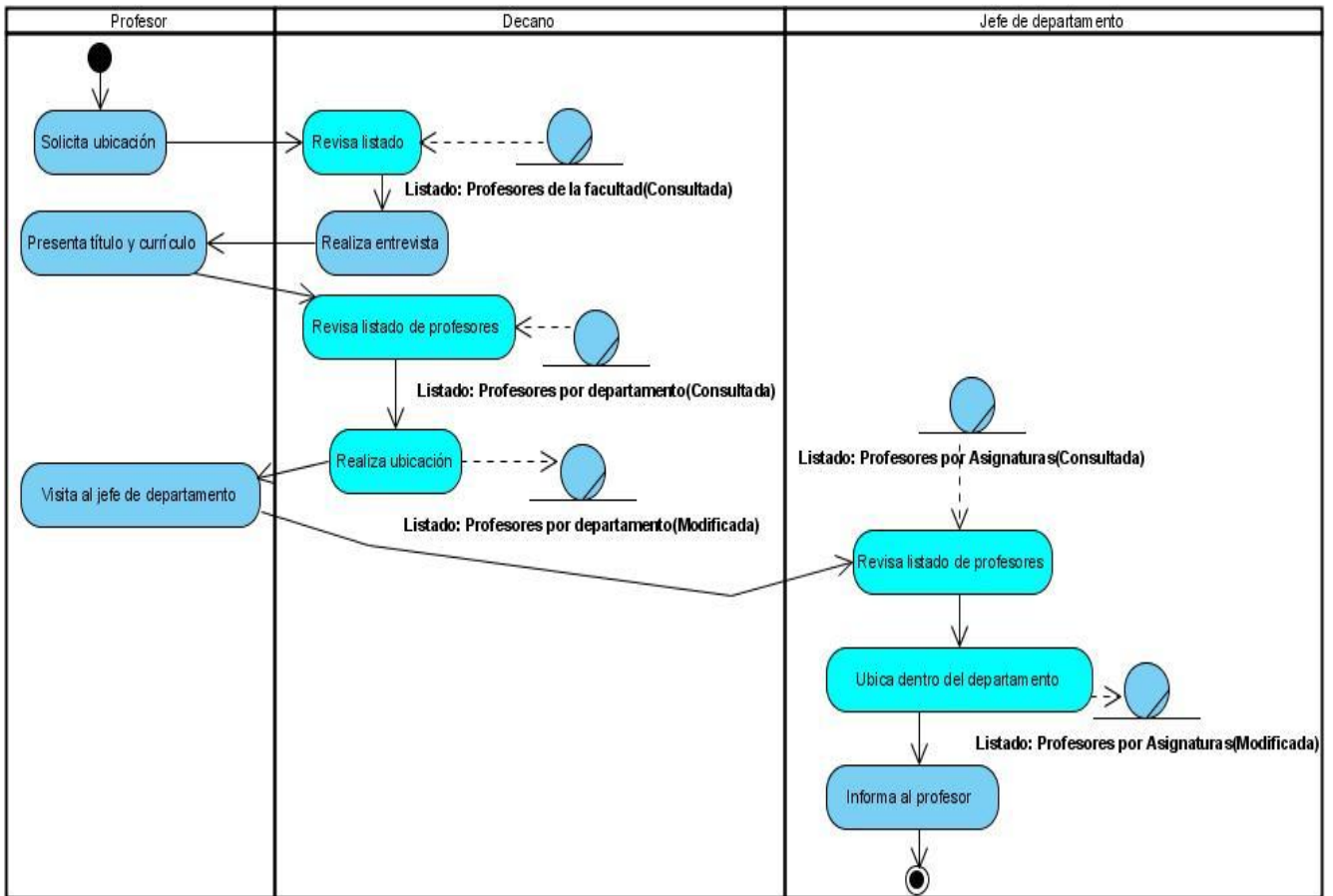


Figura 29: DA CUN _Solicitar Reingreso

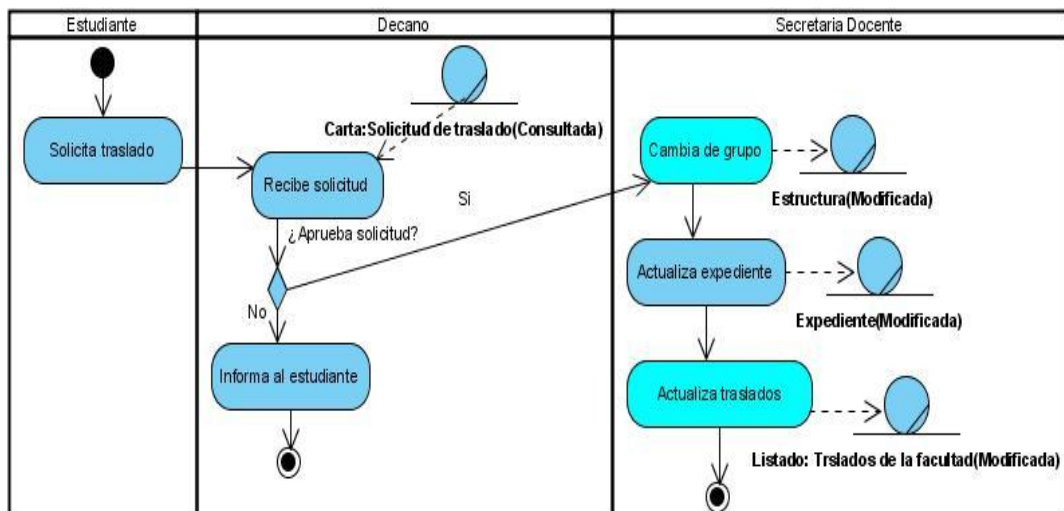


Figura 30: DA CUN_Solicitar Traslado Estudiante Sección: Solicitar Traslado Estudiante Grupo

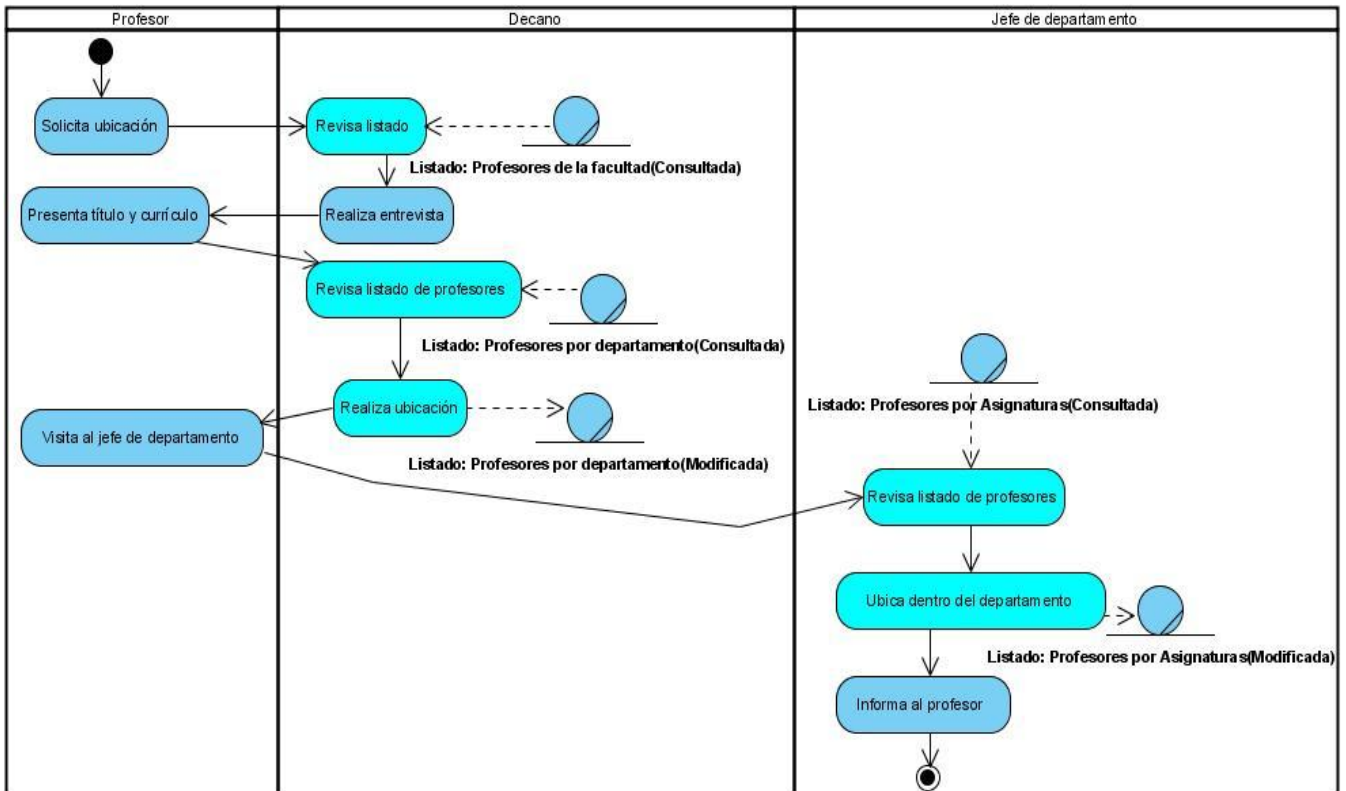


Figura 31: DA CUN_Solicitar Ingreso

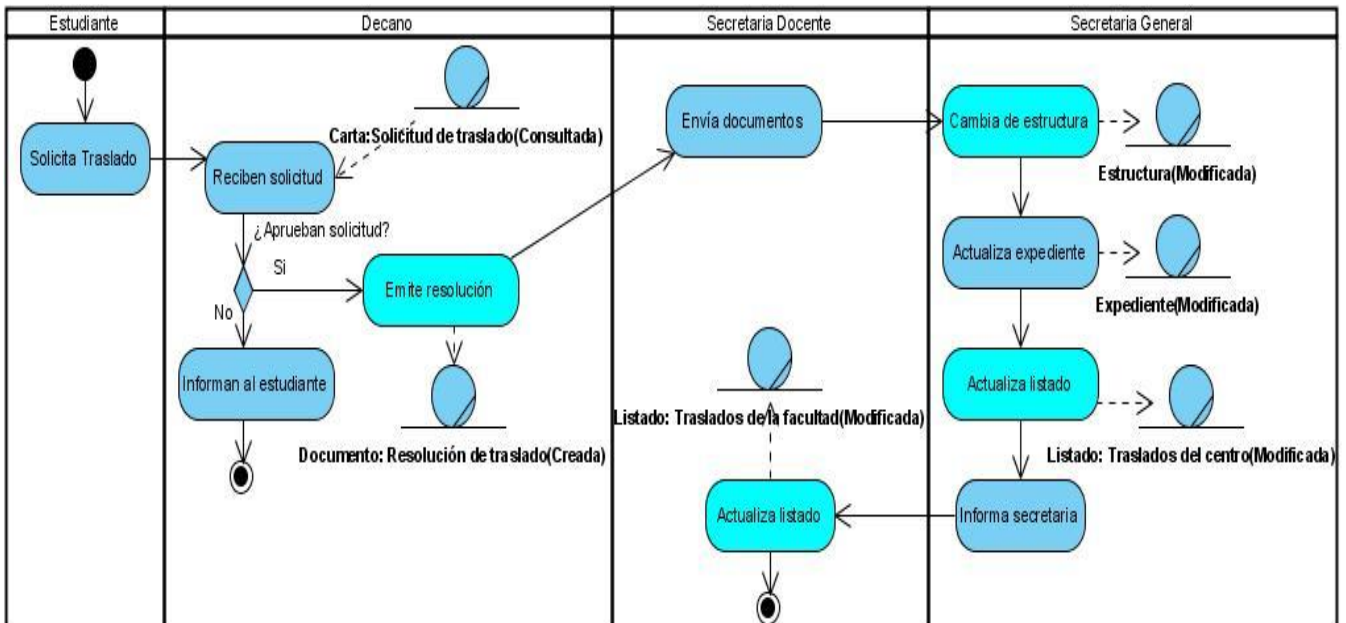


Figura 32: DA CUN_Solicitar Traslado Estudiante Sección: Solicitar Traslado Estudiante Facultad

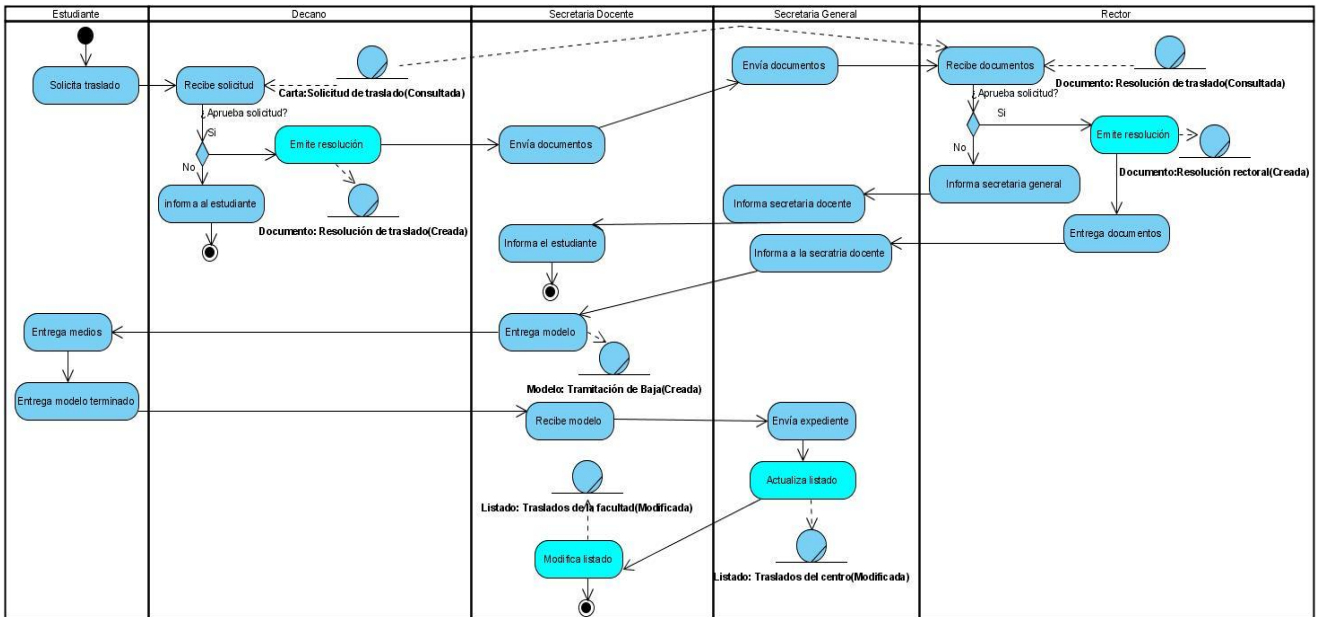


Figura 33: DA CUN_ Solicitar Traslado Estudiante Sección: Solicitar Traslado Estudiante Universidad

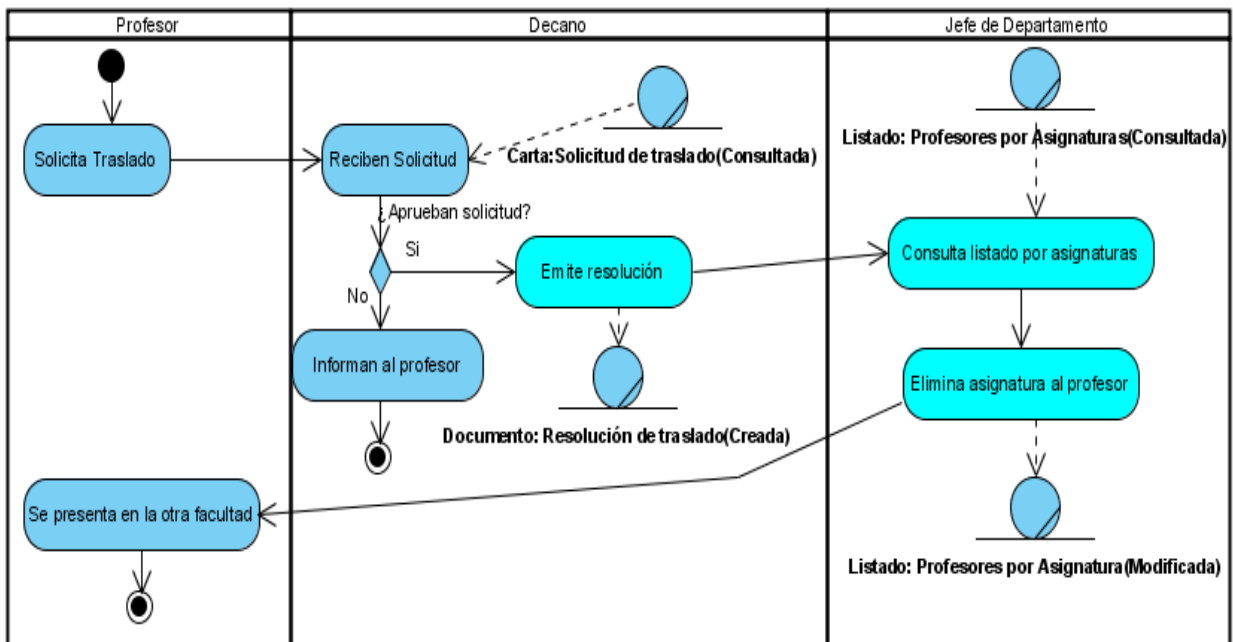


Figura 34: DA CUN_ Solicitar Traslado Profesor

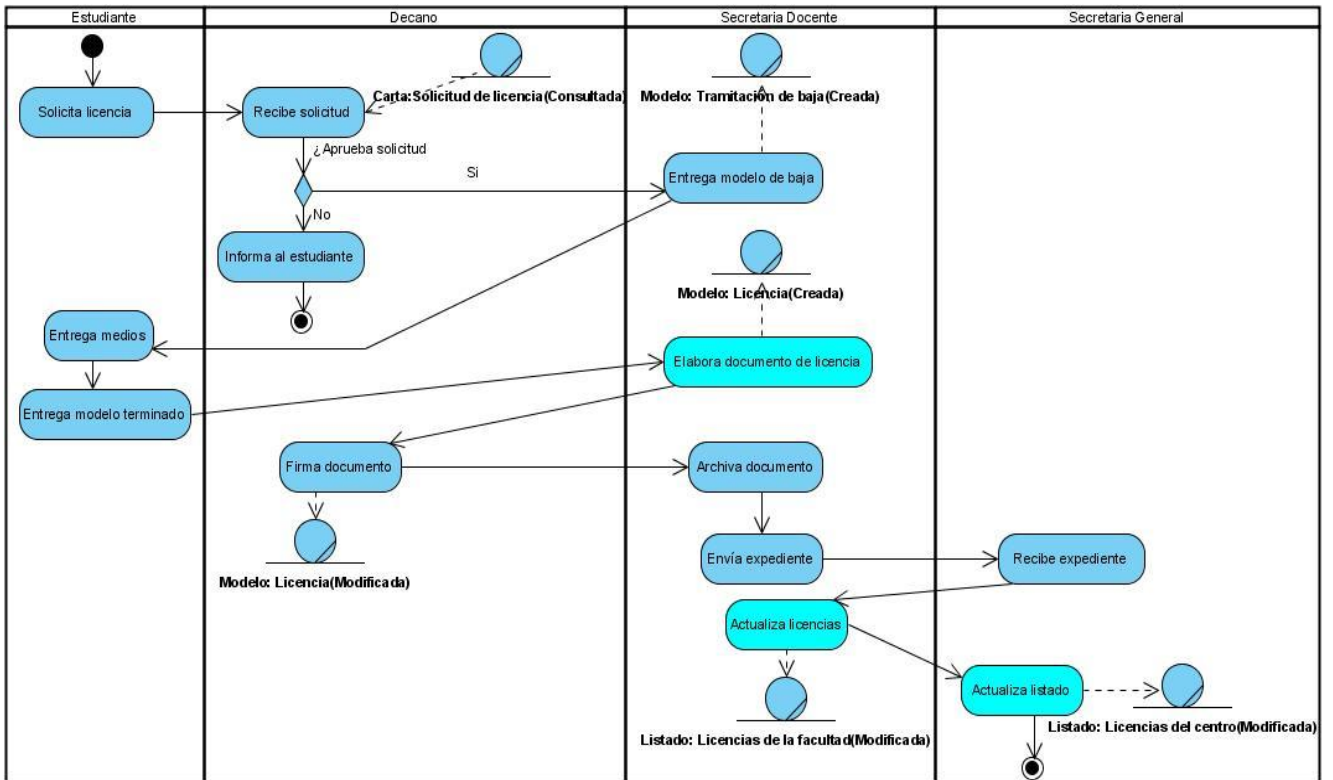


Figura 35: DA CUN _ Solicitar Licencia Estudiante

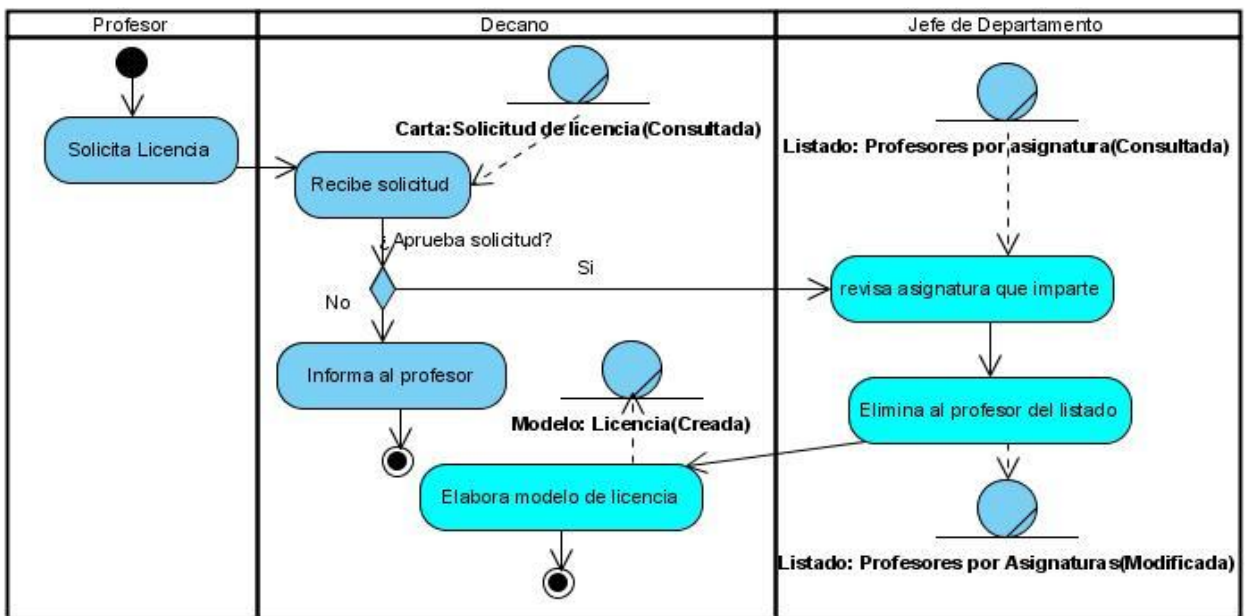


Figura 36: DA CUN_ Solicitar Licencia Profesor

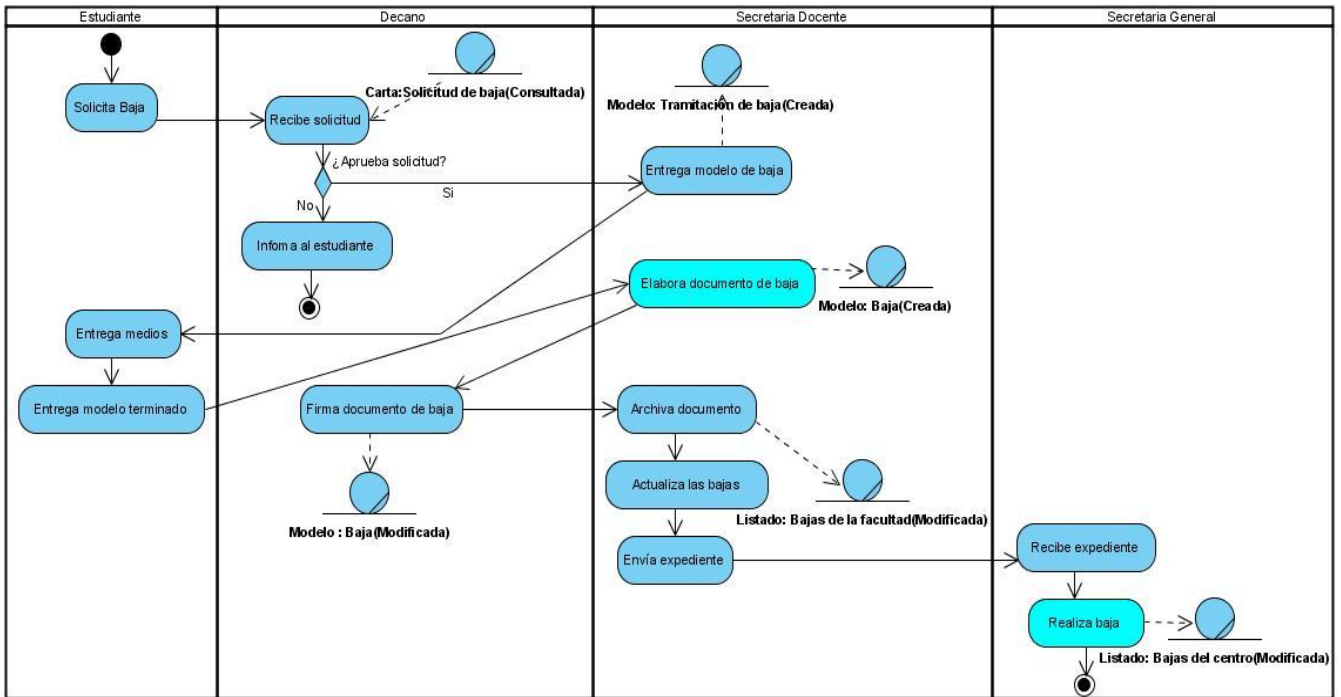


Figura 37: DA CUN_ Solicitar Baja Estudiante

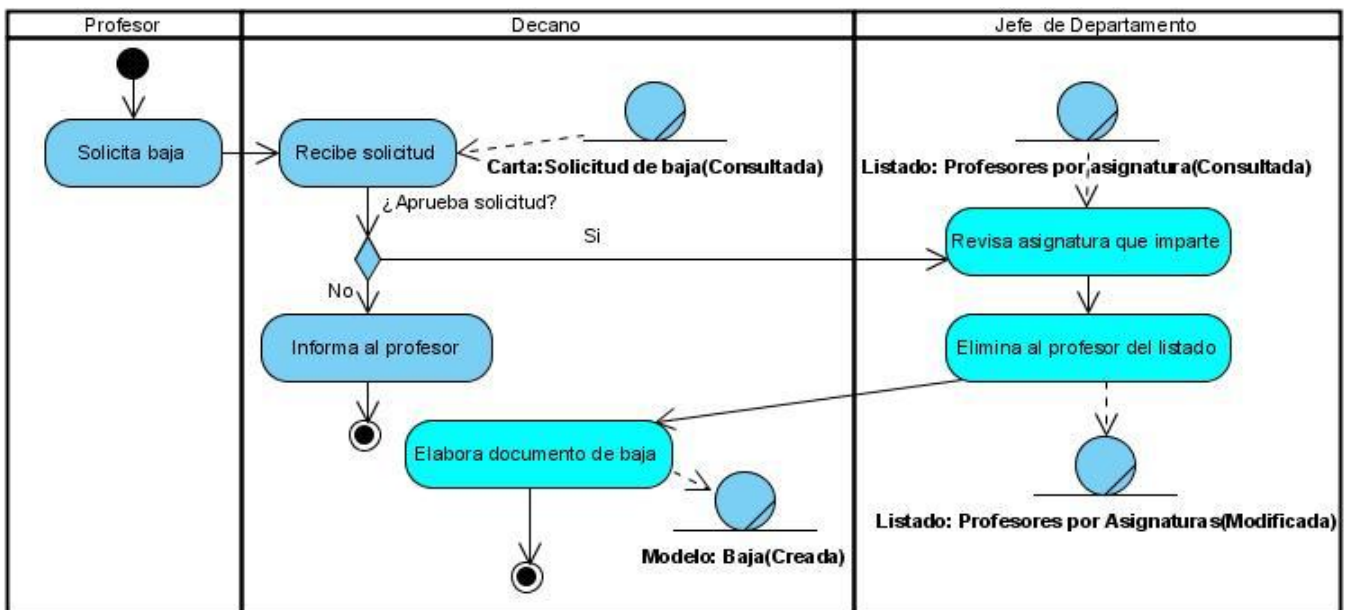
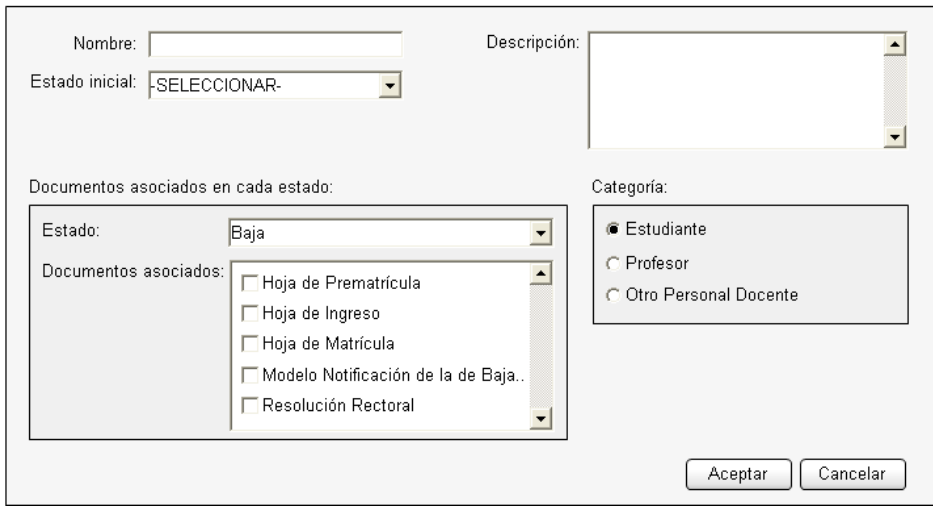


Figura 38: DA CUN_ Solicitar Baja Profesor

Anexo #2: Descripción detallada de los Casos de Uso del Sistema

CUS-1	Gestionar Tipo Personal	
Propósito	Permitir realizar las acciones Crear Tipo de Personal, Modificar Tipo de Personal y Eliminar Tipo de Personal.	
Actores	Administrador	
Resumen:	El caso de uso inicia cuando el Administrador decide gestionar los tipos de personal que presentará el sistema. Este le permitirá crear un tipo de personal, modificar un tipo de personal o eliminar un tipo de personal.	
Precondiciones	<p>El usuario debe estar registrado y tener los permisos necesarios para realizar las acciones.</p> <p>Para eliminar o modificar un Tipo de Personal, este debe haber sido creado con anterioridad.</p> <p>Para eliminar un Tipo de Personal este no debe haber sido asignado a alguna persona.</p>	
Referencias	R1, R2, R3.	
Prioridad	Crítico	
Poscondiciones	<p>Cuando se crea un tipo de personal los datos del mismo quedan guardados en la base de datos.</p> <p>Cuando se modifica un tipo de personal los datos del mismo quedan modificados en la base de datos.</p> <p>Cuando se elimina un tipo de personal los datos del mismo se eliminan de la base de datos.</p>	
Sección: "General"		
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El administrador solicita gestionar un tipo de personal.	2. El sistema muestra listado de todos los tipos de personal definidos y las opciones: <ul style="list-style-type: none"> • Crear Tipo de Personal. • Modificar Tipo de Personal. • Eliminar Tipo de Personal. 	
3. El administrador selecciona una de las opciones mostradas.	4. Según la selección realizada: <ol style="list-style-type: none"> 4.1. Si selecciona "Crear Tipo de Personal" ver sección: "Crear Tipo de Personal". 4.2. Si selecciona "Modificar Tipo de Personal" ver 	

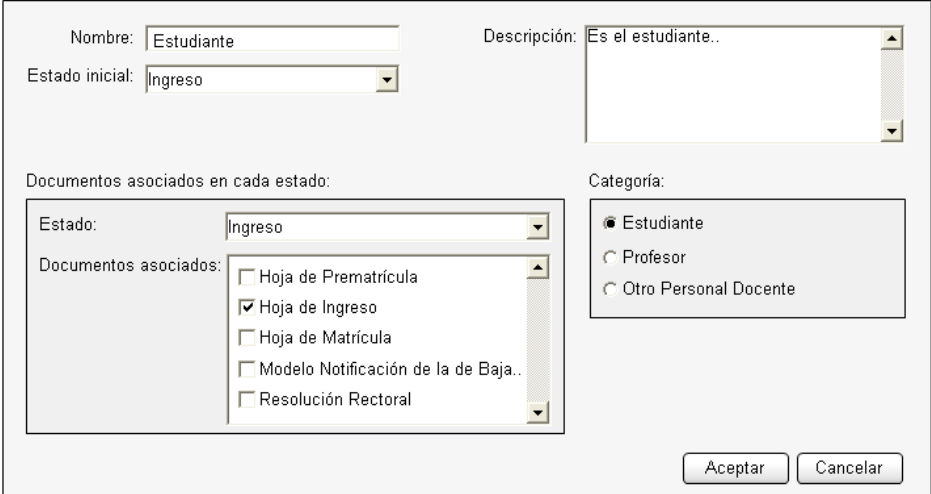
	<p>sección: “Modificar Tipo de Personal”.</p> <p>4.3. Si selecciona “Eliminar Tipo de Personal” ver sección: “Eliminar Tipo de Personal”.</p>
Sección: “Crear Tipo de Personal”	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El administrador selecciona la opción Crear Tipo de Personal.	2. El sistema muestra una interfaz con un formulario para la creación de un tipo de personal.
3. El administrador introduce los datos correspondientes al tipo de personal a crear: nombre, descripción, categoría, estado inicial y documentos asociados por cada estado.	<p>4. El sistema valida los datos introducidos.</p> <p>5. El sistema comprueba que no exista un tipo de personal con el mismo nombre.</p> <p>6. El sistema almacena los datos del tipo de personal creado.</p> <p>7. El sistema registra las incidencias generadas.</p>
Flujos Alternativos	
Acción del Actor	Respuesta del Sistema
	4.1. Si existen datos incorrectos, el sistema muestra mensaje de error correspondiente y regresa al paso 3.
	5.1. Si existe un tipo de personal con el mismo nombre, el sistema muestra mensaje de error indicando el problema y regresa al paso 3.
Prototipo de Interfaz de Usuario	
	
Sección: “Modificar Tipo de Personal”	

Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El administrador selecciona el tipo de personal a modificar.	2. El sistema muestra los datos correspondientes al tipo de personal seleccionado.
3. El administrador modifica los datos del tipo de personal seleccionado: nombre, descripción, categoría, estado inicial y documentos asociados por cada estado.	4. El sistema valida los datos modificados. 5. El sistema comprueba que no exista ningún tipo de personal con el mismo nombre. 6. El sistema almacena los datos del tipo de personal modificado. 7. El sistema registra las incidencias generadas.

Flujos Alternativos	
Acción del Actor	Respuesta del Sistema
	4.1. Si existen datos incorrectos, el sistema muestra mensaje de error y regresa al paso 3.
	5.1 Si existe un tipo de personal con el mismo nombre, el sistema muestra mensaje de error correspondiente y regresa al paso 3.

Prototipo de Interfaz de Usuario

Modificar tipo de personal



Nombre: Descripción:

Estado inicial:

Documentos asociados en cada estado:

Estado:

Documentos asociados:

- Hoja de Prematrícula
- Hoja de Ingreso
- Hoja de Matrícula
- Modelo Notificación de la de Baja..
- Resolución Rectoral

Categoría:

- Estudiante
- Profesor
- Otro Personal Docente

Sección: "Eliminar Tipo de Personal"

Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El administrador selecciona el tipo de personal	2. El sistema comprueba que no existan personas que

a eliminar.	tengan asignado el tipo de personal seleccionado. 3. El sistema elimina el tipo de personal seleccionado. 4. El sistema registra las incidencias generadas.
Flujos Alternativos	
Acción del Actor	Respuesta del Sistema
	2.1 Si existen personas con ese tipo de personal asignado el sistema muestra mensaje de error indicando que no puede ser eliminado y regresa al paso 1.
Prototipo de Interfaz de Usuario	
Listado de tipos de personal << < > >>	
Nombre	Descripción Crear.. Eliminar
Estudiante	Es el estudiante que normalmente esta en la institucion.. <input checked="" type="checkbox"/>
Profesor	Es el profesor.. <input type="checkbox"/>
Profesor Adjunto	Es el profesor externo.. <input type="checkbox"/>
Profesor CPT	Es el profesor.. <input checked="" type="checkbox"/>

Tabla 2.1: Descripción detallada del CUS Gestionar Tipo Personal

CUS-2	Gestionar Estado
Propósito	Permitir realizar las acciones Crear Estado, Modificar Estado y Eliminar Estado.
Actores	Administrador
Resumen:	El caso de uso inicia cuando el administrador decide gestionar los estados por los que puede pasar una persona. El sistema le permitirá crear, modificar o eliminar un estado.
Precondiciones	El usuario debe estar registrado y tener los permisos necesarios para realizar las acciones. Para eliminar o modificar estado, este debe haber sido creado con anterioridad. Para eliminar un estado esto no debe haber sido asignado a alguna persona.
Referencias	R4, R5, R6.
Prioridad	Crítico
Poscondiciones	Cuando se crea un estado los datos del mismo quedan guardados en la base de datos. Cuando se modifica un estado los datos del mismo quedan modificados en la base de datos. Cuando se elimina un estado los datos del mismo se eliminan de la base de datos.

Sección: "General"	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El administrador solicita Gestionar Estados.	2. El sistema muestra un listado de todos los estados disponibles y las opciones: <ul style="list-style-type: none"> • Crear Estado. • Modificar Estado. • Eliminar Estado.
3. El administrador selecciona una de las opciones mostradas.	4. Según la selección realizada: <ol style="list-style-type: none"> 4.1. Si selecciona "Crear Estado" ver sección: "Crear Estado". 4.2. Si selecciona "Modificar Estado" ver sección: "Modificar Estado". 4.3. Si selecciona "Eliminar Estado" ver sección: "Eliminar Estado".
Sección: "Crear Estado"	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El administrador selecciona la opción Crear Estado.	2. El sistema muestra una interfaz con un formulario para la creación de un estado.
3. El administrador introduce los datos correspondientes al estado a crear: nombre, descripción, color, nombre del movimiento asociado, estados precedentes y documentos asociados por cada tipo de personal.	4. El sistema valida los datos introducidos. 5. El sistema comprueba que no exista un estado con el mismo nombre. 6. El sistema almacena los datos del estado creado. 7. El sistema registra las incidencias generadas.
Flujos Alternativos	
Acción del Actor	Respuesta del Sistema
	4.1. Si existen datos incorrectos, el sistema muestra mensaje de error y regresa al paso 3.
	5.1 Si existe un estado con el mismo nombre, el sistema muestra mensaje de error y regresa al paso 3.
Prototipo de Interfaz de Usuario	

Crear estado

Nombre: * Descripción:

Color: *

Movimiento asociado: *

Documentos asociados para cada tipo de personal:

Tipo de personal:

Documentos asociados:

- Hoja de Prematrícula
- Hoja de Ingreso
- Hoja de Matrícula
- Modelo Notificación de la de Baja..
- Resolución Rectoral

Estados precedentes:

- Ingreso
- Prematriculado
- Matriculado
- Traslado
- Baja

Sección: "Modificar Estado"

Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
1. El administrador selecciona el estado a modificar.	2. El sistema muestra los datos correspondientes al estado seleccionado.
3. El administrador modifica los datos del estado seleccionado: nombre, descripción, color, nombre del movimiento asociado, estados precedentes y documentos asociados por cada tipo de personal.	4. El sistema valida los datos modificados. 5. El sistema comprueba que no exista ningún estado con el mismo nombre. 6. El sistema almacena los datos del estado modificado. 7. El sistema registra las incidencias generadas.

Flujos Alternativos

Acción del Actor	Respuesta del Sistema
	4.1. Si existen datos incorrectos, el sistema muestra mensaje de error y regresa al paso 3.
	5.1. Si existe un estado con el mismo nombre el sistema muestra mensaje de error y regresa al paso 3.

Prototipo de Interfaz de Usuario

Modificar estado

Nombre: * Descripción:

Color: *

Movimiento asociado: *

Documentos asociados para cada tipo de personal:

Tipo de personal:

Documentos asociados:

- Hoja de Prematrícula
- Hoja de Ingreso
- Hoja de Matrícula
- Modelo Notificación de la de Baja..
- Resolución Rectoral

Estados precedentes:

- Ingreso
- Prematriculado
- Matriculado
- Traslado
- Baja

Sección: "Eliminar Estado"

Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
1. El administrador selecciona el estado a eliminar.	2. El sistema comprueba que no existan personas que tengan asignado el estado seleccionado. 3. El sistema comprueba que no existan tipos de personal que tengan asociado el estado seleccionado como estado inicial. 4. El sistema elimina las relaciones de precedencia con otros estados. 5. El sistema elimina el estado seleccionado. 6. El sistema registra las incidencias generadas.

Flujos Alternativos

Acción del Actor	Respuesta del Sistema
	2.1. Si existen personas con ese estado asignado el sistema muestra mensaje de error indicando que no puede ser eliminado y regresa al paso 1.
	3.1. Si existen tipos de personal con ese estado asignado el sistema muestra mensaje de error indicando que no puede ser eliminado y regresa al paso 1.

Prototipo de Interfaz de Usuario

Listado de estados		
Nombre	Descripción	Crear.. Eliminar
Ingreso	El el estado..	<input checked="" type="checkbox"/>
Prematriculado	El el estado..	<input type="checkbox"/>
Matriculado	El el estado..	<input type="checkbox"/>
Baja	El el estado..	<input type="checkbox"/>
Licencia	El el estado..	<input type="checkbox"/>

Tabla 2.2: Descripción detallada del CUS Gestionar Estado

CUS-3	Gestionar Datos Persona
Propósito	Permitir realizar las acciones Registrar Datos de la Persona Manual, Registrar Datos de la Persona Automático o Modificar Datos de la Persona.
Actores	Secretaria
Resumen:	El caso de uso inicia cuando la secretaria decide gestionar los datos de la persona. El sistema le permitirá registrar o modificar los datos de la persona.
Precondiciones	El usuario debe estar registrado y tener los permisos necesarios para realizar las acciones. Para registrar los datos de manera automática es necesaria la presencia de un sistema externo de donde importar los datos. Para modificar los datos de una persona, esta debe haber sido registrada en el sistema.
Referencias	R7, R8.R9
Prioridad	Crítico
Poscondiciones	Cuando se registran una persona los datos de la misma quedan guardados en la base de datos, así como el estado en que se encuentra. Cuando se modifican los datos de una persona estos quedan modificados en la base de datos.
Sección: "General"	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. La secretaria solicita Gestionar Datos de la persona.	2. El sistema muestra las opciones: <ul style="list-style-type: none"> • Registrar Datos Manual. • Registrar Datos Automático.

	<ul style="list-style-type: none"> • Modificar Datos de la Persona.
3. La secretaria selecciona una de las opciones mostradas.	<p>4. Según la selección realizada:</p> <p>4.1. Si selecciona “Registrar Datos de la Persona Manual” ver sección: “Registrar Datos de la Persona Manual”.</p> <p>4.2. selecciona “Registrar Datos de la Persona Automático” ver sección: “Registrar Datos de la Persona Automático”.</p> <p>4.3. Si selecciona “Modificar Datos de la Persona” ver sección: “Modificar Datos de la Persona”.</p>
Sección: “Registrar Datos de la Persona Manual”	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. La secretaria selecciona la opción Registrar Datos de una Persona Manual.	<p>2. El sistema muestra una interfaz con las opciones para registrar los datos de una persona de forma manual.</p> <p>3. El sistema muestra campos para seleccionar el tipo de personal a registrar, foto y la opción “Ubicar en estructura automáticamente”.</p>
4. La secretaria selecciona el tipo de personal a registrar sus datos, foto y la opción “Ubicar en estructura automáticamente”.	5. Según el tipo de personal seleccionado, el sistema muestra los documentos a llenar de ese tipo de personal en su estado inicial.
6. La secretaria selecciona un documento.	7. El sistema muestra formulario con los campos correspondientes al documento seleccionado.
8. La secretaria introduce los datos correspondientes al documento seleccionado.	<p>9. El sistema valida los datos introducidos.</p> <p>10. El sistema verifica que todos los documentos hayan sido llenados correctamente.</p> <p>11. El sistema asocia a la persona el tipo de personal seleccionado.</p> <p>12. El sistema asocia a la persona el estado inicial correspondiente al tipo de personal seleccionado.</p> <p>13. Si fue seleccionada la opción “Ubicar en estructura automáticamente”, el sistema ubicará a la persona en una estructura de acuerdo a un conjunto de reglas definidas con anterioridad.</p> <p>14. El sistema almacena los datos de la persona.</p>

	15. El sistema registra las incidencias generadas.
Flujos Alternativos	
Acción del Actor	Respuesta del Sistema
	9.1. Si existen datos incorrectos el sistema muestra mensaje de error y regresa al paso 8.
	10.1 Si existen documentos sin completar, el sistema muestra mensaje de error y regresa al paso 9.

Prototipo de Interfaz de Usuario

Registrar datos de la persona manual

Tipo de persona:

Foto

Documentos oficiales asociados:

Nombre	Prematricular
<input checked="" type="checkbox"/> Hoja de prematrícula	Sin completar

[Seleccionar foto..](#)

Ubicar en estructura automáticamente

HOJA DE PREMATRÍCULA

Carnet de Identidad: <input type="text"/>	Primer nombre: <input type="text"/>
Numero de Serie del: <input type="text"/>	Segundo nombre: <input type="text"/>
Sexo: <input type="text" value="-SELECCIONAR-"/>	Primer apellido: <input type="text"/>
Nombre del padre: <input type="text"/>	Segundo apellido: <input type="text"/>
Nombre de la madre: <input type="text"/>	

Sección: “Registrar Datos de la Persona Automático”

Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
1. La secretaria selecciona la opción Registrar Datos de una Persona Automático.	2. El sistema muestra una interfaz con las opciones para registrar los datos de una persona de forma automática. 3. El sistema muestra campo para seleccionar el tipo de personal a registrar y la opción “Ubicar en estructura automáticamente”.

<p>4. La secretaria selecciona el tipo de personal a registrar sus datos, foto y la opción: “Ubicar en estructura automáticamente”</p>	<p>5. Según el tipo de personal seleccionado, el sistema muestra los documentos asociados a ese tipo de personal en su estado inicial.</p> <p>6. El sistema muestra un listado de todos los campos que poseen en conjunto todos los documentos asociados.</p>
<p>7. La secretaria selecciona los campos donde desea poner la información importada.</p>	<p>8. Para cada campo seleccionado, el sistema muestra los campos del sistema externo de donde se importará la información.</p>
<p>9. La secretaria selecciona el campo externo correspondiente a cada campo seleccionado.</p> <p>10. La secretaria selecciona la opción “Importar datos”</p>	<p>11. Para toda persona registrada en el sistema externo, el sistema importa los datos de los campos seleccionados.</p> <p>12. Si fue seleccionada la opción “Ubicar en estructura automáticamente”, el sistema ubicará a las personas en una estructura de acuerdo a un conjunto de reglas definidas con anterioridad.</p> <p>13. El sistema almacena los datos importados de las personas.</p> <p>14. El sistema registra las incidencias generadas.</p>

Prototipo de Interfaz de Usuario

Registrar datos de la persona automático

Tipo de persona: Seleccionar sistema externo:

Documentos oficiales:	Registrar
<input checked="" type="checkbox"/> Hoja de Registro	Sin completar

Ubicar en estructura automáticamente

Campos a importar	Campos del sistema externo	Tipo de campo
<input checked="" type="checkbox"/> Carnet de Identidad	<input type="text" value="-SELECCIONAR-"/>	Texto
<input checked="" type="checkbox"/> Primer nombre	<input type="text" value="-SELECCIONAR-"/>	Texto
<input checked="" type="checkbox"/> Segundo nombre	<input type="text" value="-SELECCIONAR-"/>	Texto
<input checked="" type="checkbox"/> Primer apellido	<input type="text" value="-SELECCIONAR-"/>	Texto
<input checked="" type="checkbox"/> Segundo apellido	<input type="text" value="-SELECCIONAR-"/>	Texto
<input checked="" type="checkbox"/> Sexo	<input type="text" value="-SELECCIONAR-"/>	Nom
<input type="checkbox"/> Teléfono	<input type="text" value="-SELECCIONAR-"/>	Texto
<input type="checkbox"/> Centro de procedencia	<input type="text" value="-SELECCIONAR-"/>	Texto

Sección: "Modificar Datos de la Persona"	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. La secretaria selecciona la persona que desea modificar sus datos.	2. El sistema muestra la foto y los datos correspondientes a la persona seleccionada. 3. El sistema muestra los documentos asociados a la persona desde su registro. 4. El sistema muestra todos los estados por los que ha pasado la persona desde su registro.
5. La secretaria selecciona un documento del listado.	6. El sistema muestra formulario con los datos correspondientes al documento seleccionado.
7. La secretaria modifica los datos del documento seleccionado.	8. El sistema valida los datos modificados. 9. El sistema almacena los datos modificados de la persona. 10. El sistema registra las incidencias generadas.
Flujos Alternativos	
Acción del Actor	Respuesta del Sistema
	8.1 Si existen datos incorrectos el sistema muestra mensaje de error y regresa al paso 7.
Prototipo de Interfaz de Usuario	

Expediente Personal



Foto

Tipo de persona: Estudiante
CI: 86101215819
Nombre: Mireydis Orellana León
Estructura: 1502
Provincia: Matanzas
Municipio: Colón
Correo: morellana@estudiantes.uci.cu

MATRICULADO

Documentos asociados:

Nombre	Autor	Última actualización
<input checked="" type="checkbox"/> Hoja de prematrícula	ybeaton	09/05/2009
<input checked="" type="checkbox"/> Hoja de matrícula	ybeaton	09/05/2009

Movimientos realizados

Estados	Autor	Fecha de realización
<input type="checkbox"/> Ingreso	ybeaton	09/05/2009
<input type="checkbox"/> Prematriculado	ybeaton	09/05/2009
<input type="checkbox"/> Matriculado	ybeaton	09/05/2009

Tabla 2.3: Descripción detallada del CUS Gestionar Datos Persona

CUS-4	Buscar persona
Propósito	Buscar personas en el sistema.
Actores	Usuario
Resumen:	El caso de uso inicia cuando un usuario decide buscar personas en el sistema. Este le permitirá buscarlas de acuerdo a diferentes criterios de búsqueda y una vez encontradas las personas permitirá visualizar los datos de las mismas así como realizar otras operaciones.
Precondiciones	El usuario debe estar registrado y tener los permisos necesarios para realizar las acciones.
Referencias	R10
Prioridad	Crítico
Poscondiciones	Se mostrarán las personas que cumplan con los criterios de búsqueda seleccionados.
Sección: "General"	
Flujo Normal de Eventos	

Acción del Actor	Respuesta del Sistema
1. El usuario selecciona la opción buscar persona.	2. El sistema muestra la interfaz de búsqueda de personas. 3. El sistema muestra diferentes criterios de búsqueda.
4. El usuario selecciona uno o varios de los criterios de búsqueda.	5. El sistema busca de acuerdo a los criterios seleccionados. 6. El sistema muestra interfaz con listado de todas las personas que cumplan con los criterios seleccionados. 7. El sistema da la posibilidad de seleccionar una de estas personas.
8. El usuario selecciona una persona.	9. El sistema muestra los algunos datos de la persona seleccionada. 10. De acuerdo a los permisos del usuario, el sistema muestra algunas de las opciones: <ul style="list-style-type: none"> • Modificar datos de la persona. • Realizar movimiento a la persona. • Ubicar personas en estructura. • Asignar asignaturas (Si la persona es profesor). • Asignar planes de estudio. (Si las personas son estudiantes).
Prototipo de Interfaz de Usuario	

<p>Buscar persona:</p> <p>Tipo de personal: <input type="text" value="-SELECCIONAR-"/></p> <p>Nombre: <input type="text"/></p> <p>Seleccionar estructura: <div style="border: 1px solid black; padding: 5px; text-align: center;">Árbol de estructuras</div></p> <p>Búsqueda avanzada..</p> <p><input type="button" value="Buscar persona"/></p>	<p>Resultados de la búsqueda (5 personas encontradas) << < > >></p> <table border="1"> <thead> <tr> <th>Nombre</th> <th>Tipo de persona</th> <th>Estructura</th> <th></th> </tr> </thead> <tbody> <tr> <td> Yurisleis Alonso Beatón</td> <td>Estudiante</td> <td>1502</td> <td>Expediente..</td> </tr> <tr> <td colspan="4" style="text-align: center;"> Tipo de persona: Estudiante CI: 86101215819 Nombre: Mireydis Orellana León Estructura: 1502 Provincia: Matanzas Municipio: Colón Correo: morellana@estudiantes.uci.cu </td> </tr> <tr> <td> Yisel Ávila Portales</td> <td>Estudiante</td> <td>1502</td> <td>Expediente..</td> </tr> <tr> <td colspan="4" style="text-align: center;">MATRICULADO</td> </tr> <tr> <td> Ramón Valdez Domínguez</td> <td>Profesor</td> <td>Informatización</td> <td>Expediente..</td> </tr> <tr> <td> Enmanuel Azaharez Reytez</td> <td>Estudiante</td> <td>1502</td> <td>Expediente..</td> </tr> </tbody> </table> <p style="text-align: right;"><< < > >></p>	Nombre	Tipo de persona	Estructura		 Yurisleis Alonso Beatón	Estudiante	1502	Expediente..	Tipo de persona: Estudiante CI: 86101215819 Nombre: Mireydis Orellana León Estructura: 1502 Provincia: Matanzas Municipio: Colón Correo: morellana@estudiantes.uci.cu				 Yisel Ávila Portales	Estudiante	1502	Expediente..	MATRICULADO				 Ramón Valdez Domínguez	Profesor	Informatización	Expediente..	 Enmanuel Azaharez Reytez	Estudiante	1502	Expediente..
Nombre	Tipo de persona	Estructura																											
 Yurisleis Alonso Beatón	Estudiante	1502	Expediente..																										
Tipo de persona: Estudiante CI: 86101215819 Nombre: Mireydis Orellana León Estructura: 1502 Provincia: Matanzas Municipio: Colón Correo: morellana@estudiantes.uci.cu																													
 Yisel Ávila Portales	Estudiante	1502	Expediente..																										
MATRICULADO																													
 Ramón Valdez Domínguez	Profesor	Informatización	Expediente..																										
 Enmanuel Azaharez Reytez	Estudiante	1502	Expediente..																										

Tabla 2.4: Descripción detallada del CUS Buscar Persona

CUS-5	Realizar movimiento
Propósito	Aplicar un movimiento a una persona. Esta acción consiste en realizar un cambio de estado a la persona a partir de otro estado y llenar los documentos asociados al nuevo estado.
Actores	Secretaria
Resumen:	El caso de uso inicia cuando la secretaria decide aplicar un movimiento a una persona. El sistema realizará el cambio de estado pertinente.
Precondiciones	El usuario debe estar registrado y tener los permisos necesarios para realizar la acción.
Referencias	R11.
Prioridad	Crítico
Poscondiciones	Se cambia el estado en que se encuentra la persona en ese momento por el estado final del movimiento. Los datos de los documentos asociados al nuevo estado quedan almacenados en la base de datos.
Sección: "General"	

Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. La secretaria selecciona una persona para realizarle movimiento.	2. El sistema muestra la opción "Realizar Movimiento".
3. La secretaria selecciona la opción "Realizar Movimiento".	4. El sistema muestra interfaz con los datos de la persona seleccionada y las opciones para realizar movimiento. 5. El sistema muestra los posibles movimientos que pueden ser aplicados a la persona a partir del estado actual.
6. La secretaria selecciona un movimiento.	7. El sistema muestra listado con los documentos asociados al estado final del movimiento y al tipo de personal a que pertenece la persona seleccionada.
8. La secretaria llena los documentos asociados.	9. El sistema comprueba que hayan sido llenados correctamente todos los documentos. 10. El sistema cambia el estado actual de la persona por el estado final del movimiento. 11. El sistema almacena los documentos asociados al nuevo estado de la persona. 12. El sistema almacena los cambios realizados. 13. El sistema registra las incidencias generadas.
Flujos Alternativos	
Acción del Actor	Respuesta del Sistema
	9.1 Si existen documentos que no han sido llenados correctamente, mostrar mensaje de error y regresar al paso 8.
Prototipo de Interfaz de Usuario	

Realizar movimiento

Foto

Tipo de persona: Estudiante
Nombre: Mireydis Orellana León
Estructura: 1502

Seleccionar movimiento:

MATRICULADO

Documentos oficiales asociados:

Nombre	Estado de la información
<input checked="" type="checkbox"/> Modelo Notificación de la Baja	Sin completar

Modelo Notificación de la Baja

Carnet de Identidad:

Numero de Serie del:

Sexo:

Primer nombre:

Segundo nombre:

Primer apellido:

Segundo apellido:

Tabla 2.5: Descripción detallada del CUS Realizar Movimiento

CUS-6	Ubicar Personas en Estructura
Propósito	Permitir ubicar una o varias personas en las diferentes estructuras definidas.
Actores	Secretaria
Resumen:	El caso de uso inicia cuando la secretaria decide ubicar una o varias personas en las diferentes estructuras definidas. El sistema ubicará las personas en la estructura seleccionada o les quitará la ubicación anterior.
Precondiciones	El usuario debe estar registrado y tener los permisos necesarios para realizar las acciones.
Referencias	R12.
Prioridad	Crítico
Poscondiciones	
Sección: "General"	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. La secretaria busca las personas a ubicar mediante la interfaz de búsqueda.	2. El sistema muestra la opción "Ubicar Persona en Estructura".

3. La secretaria selecciona la opción “Ubicar Persona en estructura”.	4. El sistema muestra interfaz con las opciones para la ubicación en las estructuras. 5. El sistema muestra listado de las personas encontradas en la interfaz de búsqueda (ver paso 1).
6. La secretaria selecciona una o varias personas del listado.	7. El sistema muestra una interfaz para la selección de una nueva estructura a partir de las estructuras definidas.
8. La secretaria selecciona una nueva estructura.	9. El sistema muestra las opciones: <ul style="list-style-type: none"> • Ubicar Seleccionados. • Quitar Ubicación.
10. El usuario selecciona la opción “Ubicar Seleccionados”.	11. El sistema ubica todas las personas seleccionadas en la nueva estructura seleccionada en el paso 8.
12. El usuario selecciona la opción “Quitar Ubicación”	13. El sistema le quita la ubicación actual a todas las personas seleccionadas en el listado y las deja sin ubicaciones. 14. El sistema almacena los cambios realizados. 15. El sistema registra las incidencias generadas.

Flujos Alternativos

Acción del Actor	Respuesta del Sistema
	5.1. Si la secretaria seleccionó una persona en el paso 1, el sistema sólo mostrará en el listado la persona seleccionada.

Prototipo de Interfaz de Usuario

Ubicar personas en estructura

Seleccionar nueva estructura: Personas a realizar ubicación:

Seleccionar aquí una estructura para ubicar en ella las personas seleccionadas en el listado	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">Nombre</th> <th style="text-align: left;">Estructura actual</th> <th style="text-align: center;">checkbox</th> </tr> </thead> <tbody> <tr> <td>Yurisleis Alonso Beatón</td> <td>1502</td> <td style="text-align: center;"><input type="checkbox"/></td> </tr> <tr> <td>Enmanuel Azhares Reyes</td> <td>1502</td> <td style="text-align: center;"><input checked="" type="checkbox"/></td> </tr> <tr> <td>Mireydis Orellana León</td> <td>1502</td> <td style="text-align: center;"><input type="checkbox"/></td> </tr> <tr> <td>Dariel Enrique Tamayo Palma</td> <td>1502</td> <td style="text-align: center;"><input checked="" type="checkbox"/></td> </tr> <tr> <td>Yisel Ávila Portales</td> <td>1502</td> <td style="text-align: center;"><input type="checkbox"/></td> </tr> <tr> <td>Acralys Ferriol Ortiz</td> <td>1502</td> <td style="text-align: center;"><input checked="" type="checkbox"/></td> </tr> </tbody> </table>	Nombre	Estructura actual	checkbox	Yurisleis Alonso Beatón	1502	<input type="checkbox"/>	Enmanuel Azhares Reyes	1502	<input checked="" type="checkbox"/>	Mireydis Orellana León	1502	<input type="checkbox"/>	Dariel Enrique Tamayo Palma	1502	<input checked="" type="checkbox"/>	Yisel Ávila Portales	1502	<input type="checkbox"/>	Acralys Ferriol Ortiz	1502	<input checked="" type="checkbox"/>
Nombre	Estructura actual	checkbox																				
Yurisleis Alonso Beatón	1502	<input type="checkbox"/>																				
Enmanuel Azhares Reyes	1502	<input checked="" type="checkbox"/>																				
Mireydis Orellana León	1502	<input type="checkbox"/>																				
Dariel Enrique Tamayo Palma	1502	<input checked="" type="checkbox"/>																				
Yisel Ávila Portales	1502	<input type="checkbox"/>																				
Acralys Ferriol Ortiz	1502	<input checked="" type="checkbox"/>																				

Tabla 2.6: Descripción detallada del CUS Ubicar Personas en Estructura

CUS-7	Asignar Asignaturas Profesor	
Propósito	Permitir asignar a un profesor las asignaturas que deberá impartir.	
Actores	Jefe de Departamento	
Resumen	El caso de uso inicia cuando el jefe de departamento decide asignar las asignaturas que deberá impartir un profesor. El sistema le permitirá seleccionar estas asignaturas a partir de un listado de todas las asignaturas disponibles.	
Precondiciones	El usuario debe estar registrado y tener los permisos necesarios para realizar la acción.	
Referencias	R13	
Prioridad	Crítico	
Poscondiciones	Todos los cambios realizados son almacenados en la bases de datos.	
Sección: "General"		
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El jefe de departamento busca una persona. 2. El jefe de departamento selecciona una persona.	3. El sistema muestra los datos de la persona seleccionada. 4. Si la persona es profesor el sistema muestra la opción "Asignar Asignaturas"	
5. El jefe de departamento selecciona la opción "Asignar Asignaturas".	6. El sistema muestra interfaz con las opciones para asignar asignaturas al profesor. 7. El sistema muestra un listado de las asignaturas asignadas. En caso de no tener asignada ninguna asignatura se mostrará el listado en blanco. 8. El sistema muestra listado de las asignaturas disponibles.	
9. El jefe de departamento agrega o quita asignaturas en el listado	10. El sistema asocia al profesor las asignaturas que estén en el listado de asignaturas asignadas. Si el profesor tiene asignadas asignaturas que no estén en este listado, el sistema quitará estas asociaciones. 11. El sistema guarda los cambios realizados. 12. El sistema registra las incidencias generadas.	
Prototipo de Interfaz de Usuario		

Asignar asignaturas al profesor

Foto

Tipo de persona: Profesor

Nombre: Alexey Peña Paz

Estructura: Facultad 1

REGISTRADO

Asignaturas asociadas:

Administración de Empresas

<

-

Asignaturas disponibles:

Comercio Electrónico
Contabilidad y Finanzas

Tabla 2.7: Descripción detallada del CUS Asignar Asignaturas Profesor

CUS-8	Asociar Plan de Estudio.
Propósito	Permitir asociar uno o varios estudiantes a un plan de estudio.
Actores	Secretaria
Resumen:	El caso de uso inicia cuando la secretaria decide asociar un plan de estudio a uno o varios estudiantes. El sistema asociará los estudiantes al plan de estudio.
Precondiciones	El usuario debe estar registrado y tener los permisos necesarios para realizar las acciones.
Referencias	R14.
Prioridad	Crítico
Poscondiciones	Todos los cambios realizados son almacenados en la bases de datos.
Sección: "General"	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. La secretaria busca las personas que desea asociar a un plan de estudio mediante la interfaz de búsqueda.	2. Si las personas son estudiantes, el sistema muestra la opción "Asociar Plan de Estudio".
3. La secretaria selecciona la opción "Asociar Plan de Estudio".	4. El sistema muestra interfaz con las opciones para la asociación de un plan de estudio. 5. El sistema muestra listado de todos los estudiantes encontrados en la interfaz de búsqueda (ver paso 1).
6. La secretaria selecciona uno o varios	7. El sistema muestra campo con listado de los todos

estudiantes del listado.	planes de estudio vigentes.
8. La secretaria selecciona un plan de estudio.	9. El sistema asocia el plan de estudio seleccionado a los estudiantes seleccionados en el paso 6. 10. El sistema almacena los cambios realizados. 11. El sistema registra las incidencias generadas.

Flujos Alternativos

Acción del Actor	Respuesta del Sistema
------------------	-----------------------

	5.1. Si la secretaria seleccionó un estudiante en el paso 1, el sistema sólo mostrará en el listado el estudiante seleccionado.
--	---

Prototipo de Interfaz de Usuario

Asociar estudiantes a plan de estudio

Seleccionar plan de estudio: Estudiantes a asociar plan de estudio: Asociar seleccionados

SELECCIONAR- ▼

Nombre	Plan de estudio actual	☐
Yurisleis Alonso Beatón	Ninguno	☐
Enmanuel Azhares Reyes	Ninguno	☑
Mireydis Orellana León	Ninguno	☐
Dariel Enrique Tamayo Palma	Ninguno	☑
Yisel Ávila Portales	Ninguno	☐
Acralys Ferriol Ortiz	Ninguno	☑

Tabla 2.8: Descripción detallada del CUS Asociar Plan de Estudio

CUS-9	Consultar información
Propósito	Realizar diferentes reportes con la información sobre el personal existente en el sistema.
Actores	Usuario
Resumen	El caso de uso inicia cuando un usuario decide consultar información sobre el personal registrado en el sistema. El sistema permitirá ver esta información de acuerdo a diferentes criterios.
Precondiciones	El usuario debe estar registrado y tener los permisos necesarios para realizar la acción.

Referencias	R15.
Prioridad	Crítico
Poscondiciones	
Sección: "General"	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El usuario selecciona la opción Reportes.	2. El sistema muestra una interfaz con las posibles opciones de reportes que se pueden consultar: <ul style="list-style-type: none"> • Seleccionar Períodos. • Seleccionar Tipos de Personal. • Seleccionar Estados. • Seleccionar Estructuras. • Seleccionar Campos a Mostrar
3. El usuario selecciona en las opciones mostradas los diferentes criterios que cumplirán las personas que se visualizarán en el reporte.	6. El sistema muestra un formulario con un listado de todas las personas que cumplan con los criterios seleccionados. En el mismo sólo estará la información de los campos seleccionados en la opción "Seleccionar Campos a Mostrar" así como la información de las opciones seleccionadas en el paso 4.
4. El usuario selecciona cuáles de estas opciones incluirán su información en la visualización del reporte.	
5. El usuario selecciona la opción "Visualizar Reporte"	
7. El sistema registra las incidencias generadas.	
Flujos Alternativos	
Acción del Actor	Respuesta del Sistema
	3.1 Si los tipos de personal seleccionados son profesores, el sistema mostrará la opción Seleccionar Asignaturas.
Prototipo de Interfaz de Usuario	

Generador de Reportes

<input checked="" type="checkbox"/> Seleccionar períodos: <div style="border: 1px solid gray; height: 100px; width: 100%;"></div>	<input type="checkbox"/> Seleccionar tipos de personal: <input type="checkbox"/> Estudiante <input checked="" type="checkbox"/> Profesor <input type="checkbox"/> Profesor Adjunto <input type="checkbox"/> Profesor CPT	<input type="checkbox"/> Seleccionar asignaturas: <input checked="" type="checkbox"/> Matemática I <input type="checkbox"/> Matemática II <input type="checkbox"/> Teleinformática I <input type="checkbox"/> Teleinformática II <input type="checkbox"/> Programación I <input type="checkbox"/> Inglés I
<input type="checkbox"/> Seleccionar estructuras: <div style="border: 1px solid gray; height: 100px; width: 100%;"></div>	<input type="checkbox"/> Seleccionar estados: <input type="checkbox"/> Ingreso <input type="checkbox"/> Prematriculado <input type="checkbox"/> Matriculado <input type="checkbox"/> Traslado <input checked="" type="checkbox"/> Baja	<input checked="" type="checkbox"/> Campos a mostrar <input checked="" type="checkbox"/> Nombre <input checked="" type="checkbox"/> Apellidos <input type="checkbox"/> Número ID <input checked="" type="checkbox"/> Provincia <input checked="" type="checkbox"/> Municipio <input type="checkbox"/> Sexo

Tabla 2.9: Descripción detallada del CUS Consultar Información

Anexo #3: Diagramas de Colaboración de los Casos de Uso del Sistema

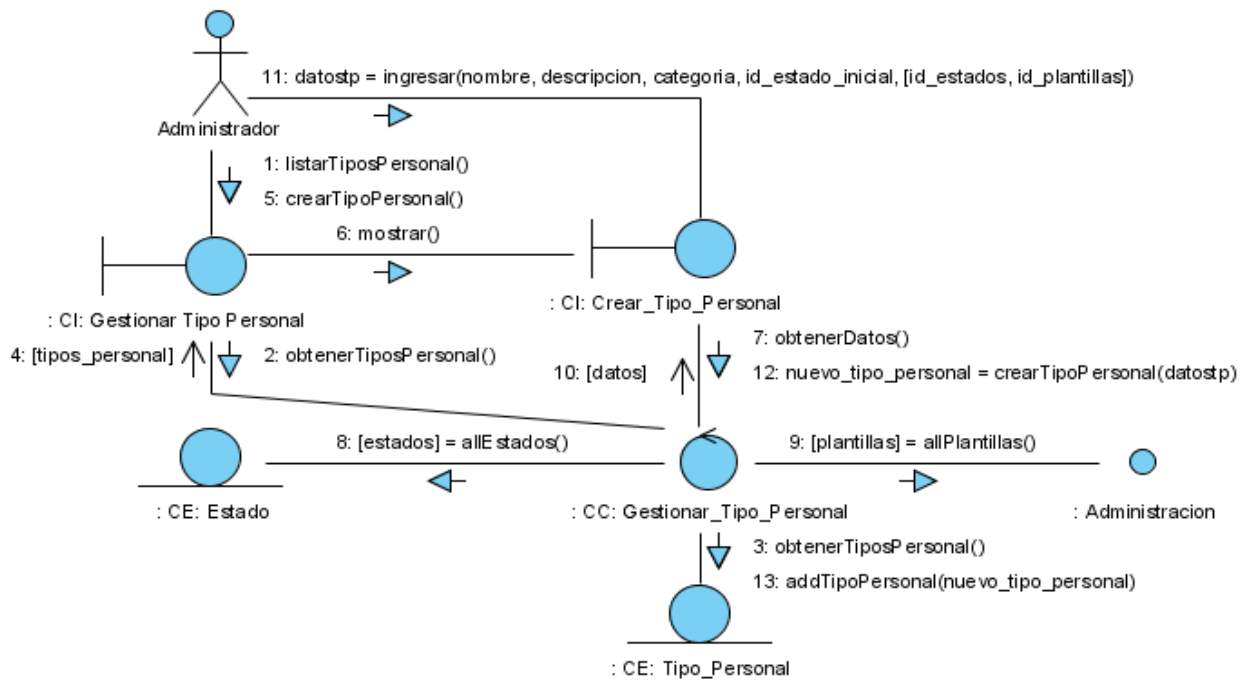


Figura 39: DC CUS_ Gestionar Tipo Personal Sección Crear Tipo Personal

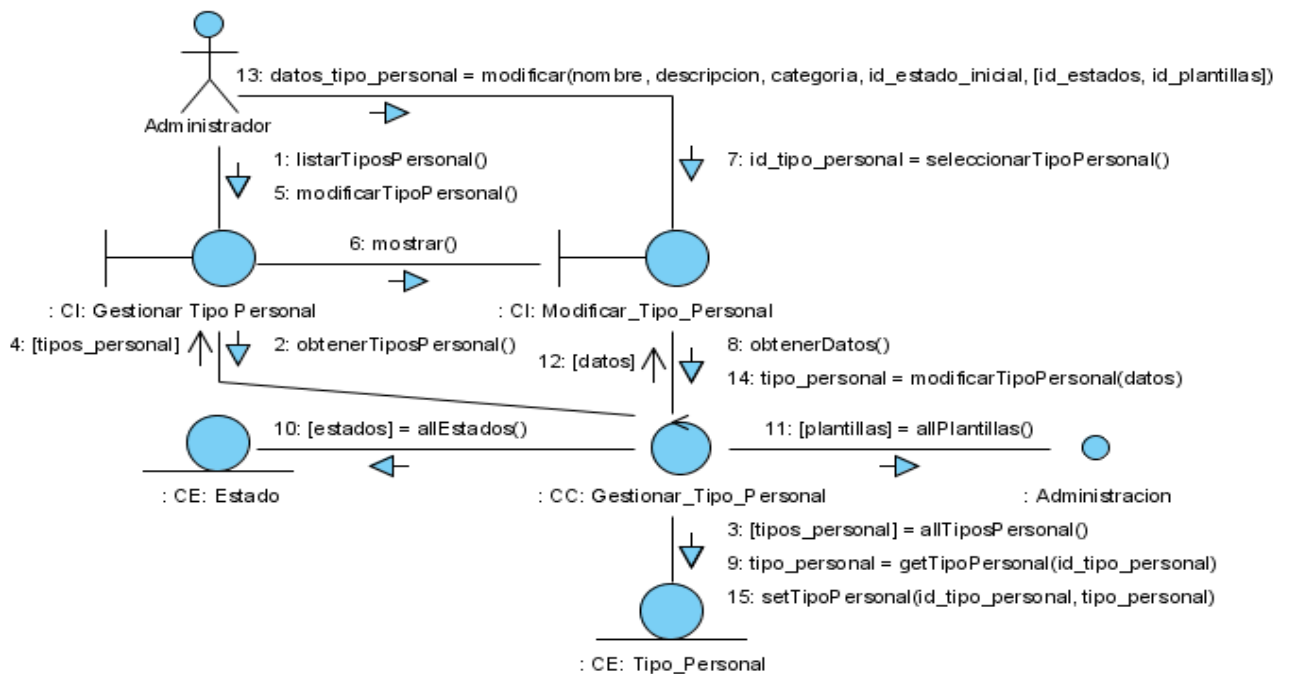


Figura 40: DC CUS_ Gestionar Tipo Personal Sección Modificar Tipo Personal

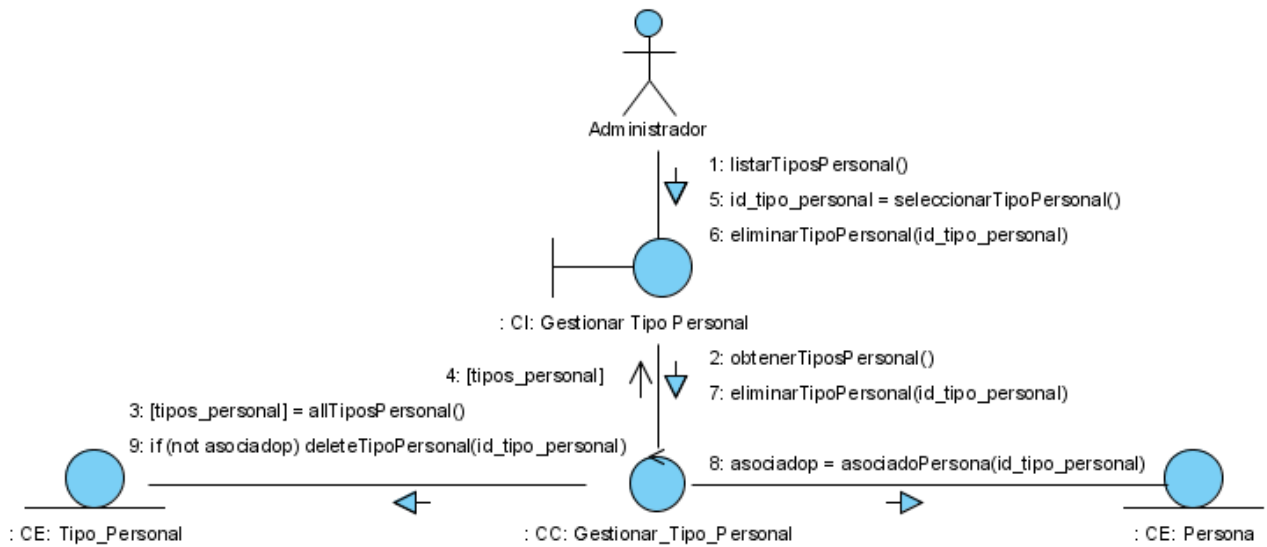


Figura 41: DC CUS_ Gestionar Tipo Personal Sección Eliminar Tipo Personal

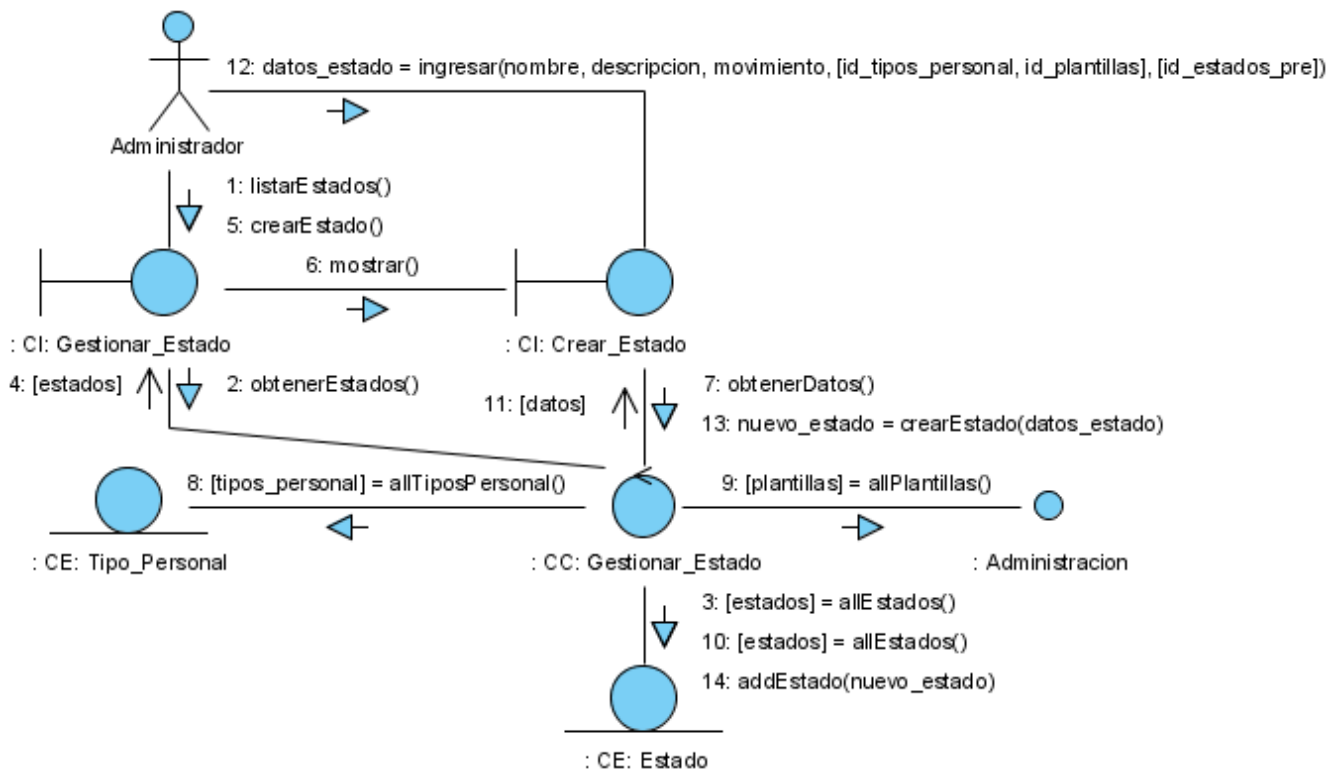


Figura 42: DC CUS_ Gestionar Estado Sección Crear Estado

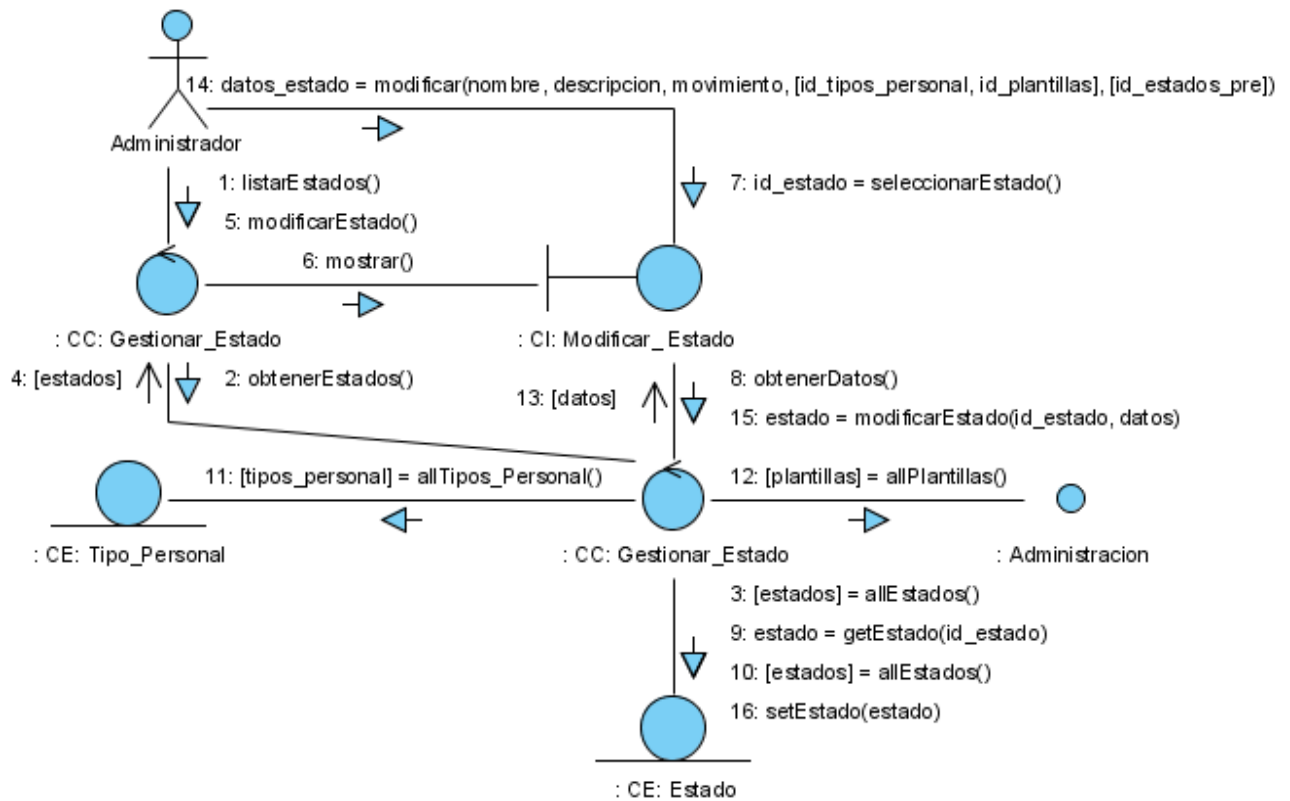


Figura 43: DC CUS _ Gestionar Estado Sección Modificar Estado

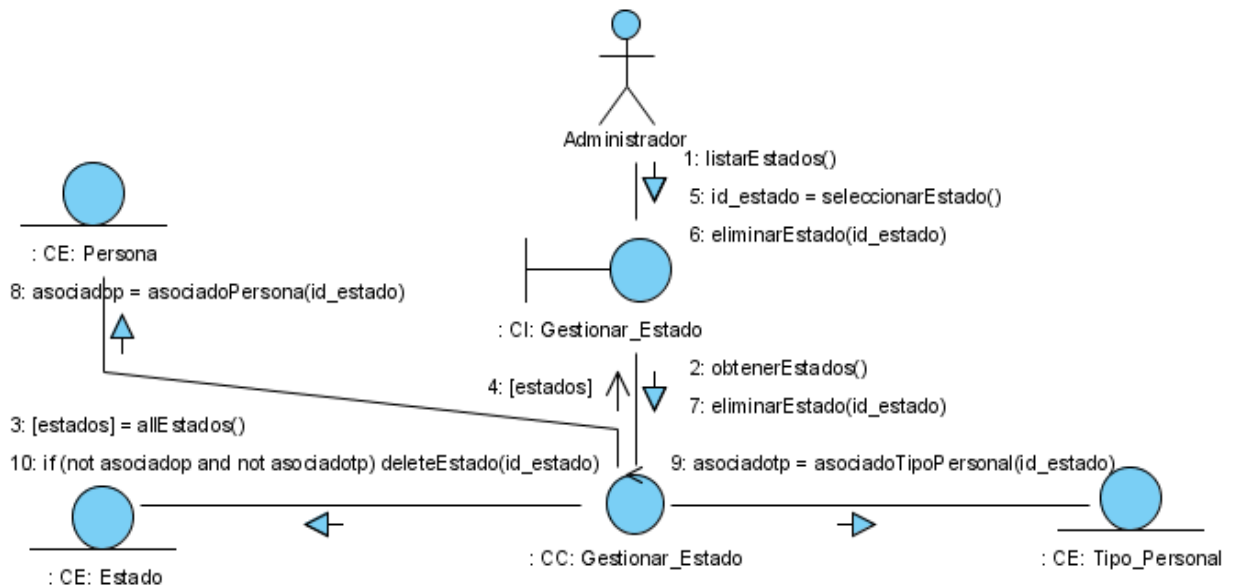


Figura 44: DC CUS _ Gestionar Estado Sección Eliminar Estado

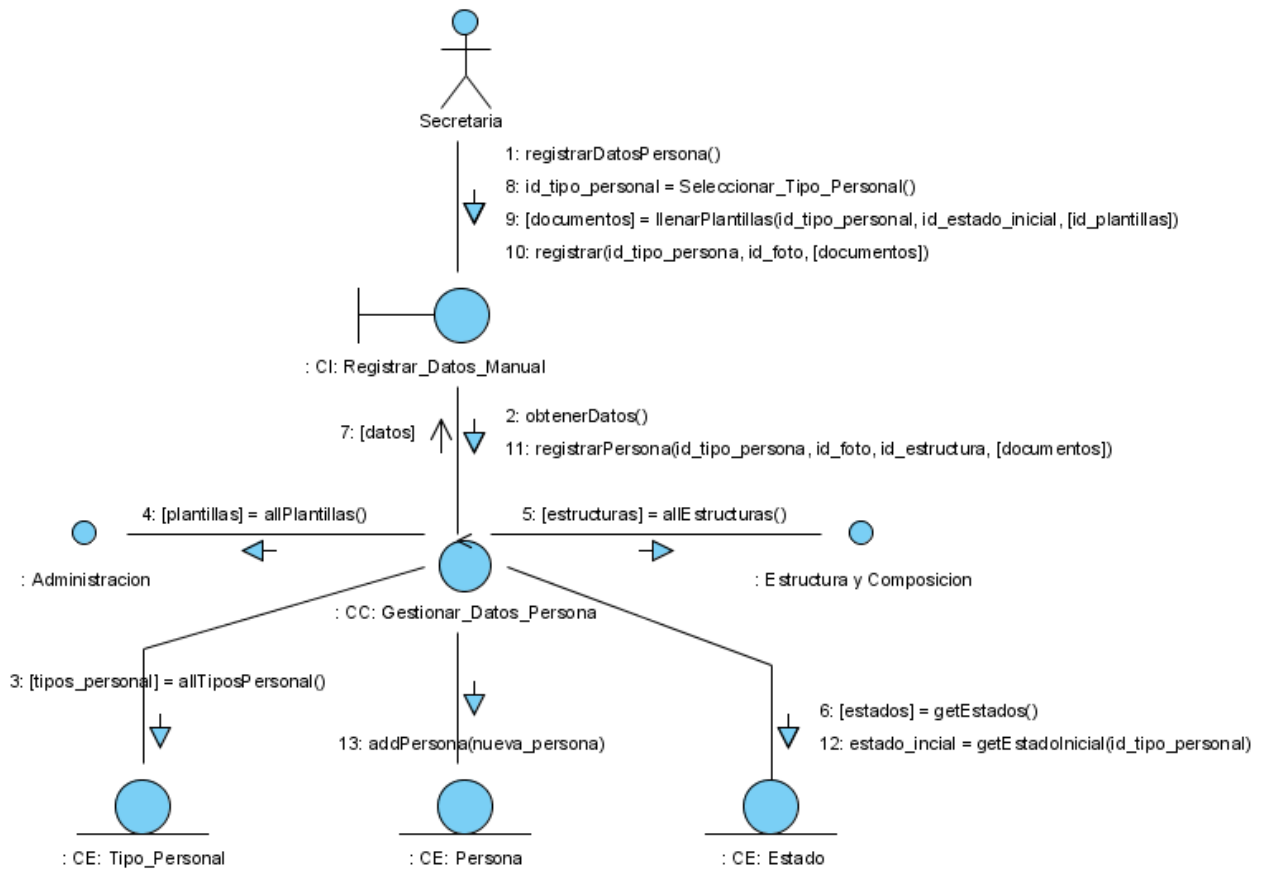


Figura 46: DC CUS _ Gestionar Datos Persona Sección Registrar Datos Persona Manual

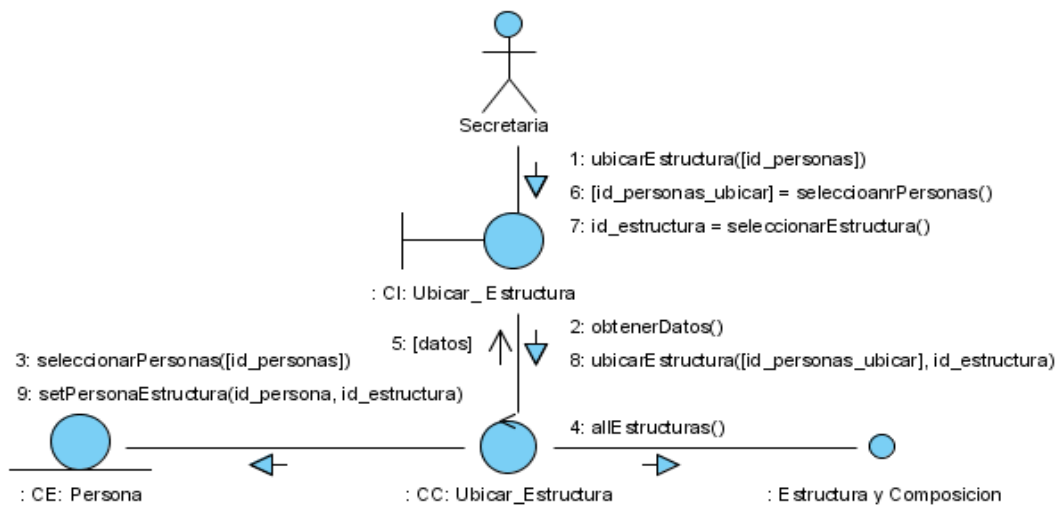


Figura 47: DC CUS _ Ubicar Persona en Estructura

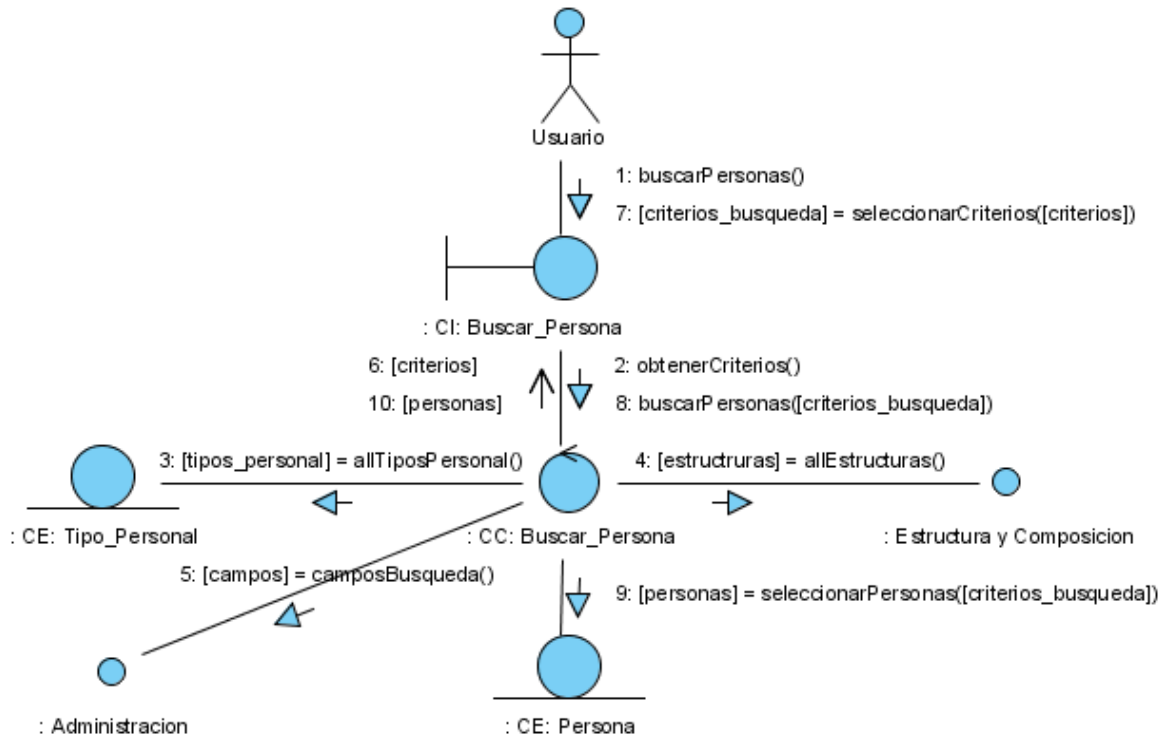


Figura 48: DC CUS _ Buscar Persona

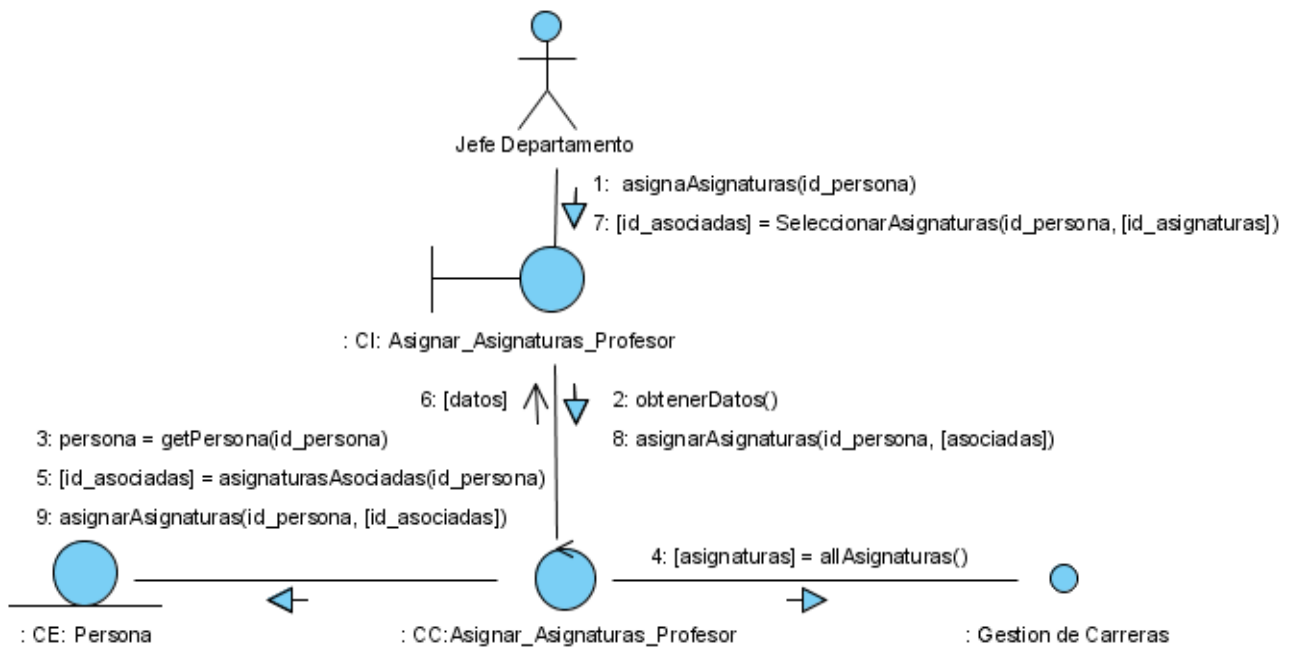


Figura 49: DC CUS _ Asignar Asignaturas Profesor

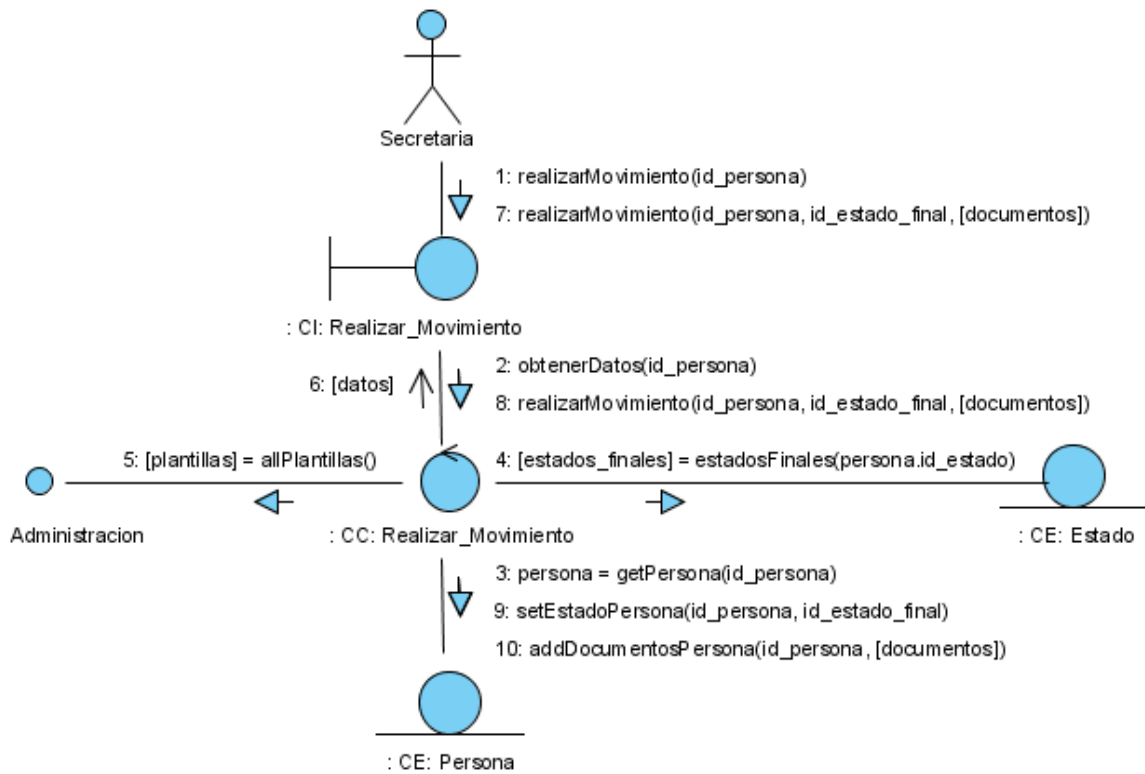


Figura 50: DC CUS _ Realizar Movimiento

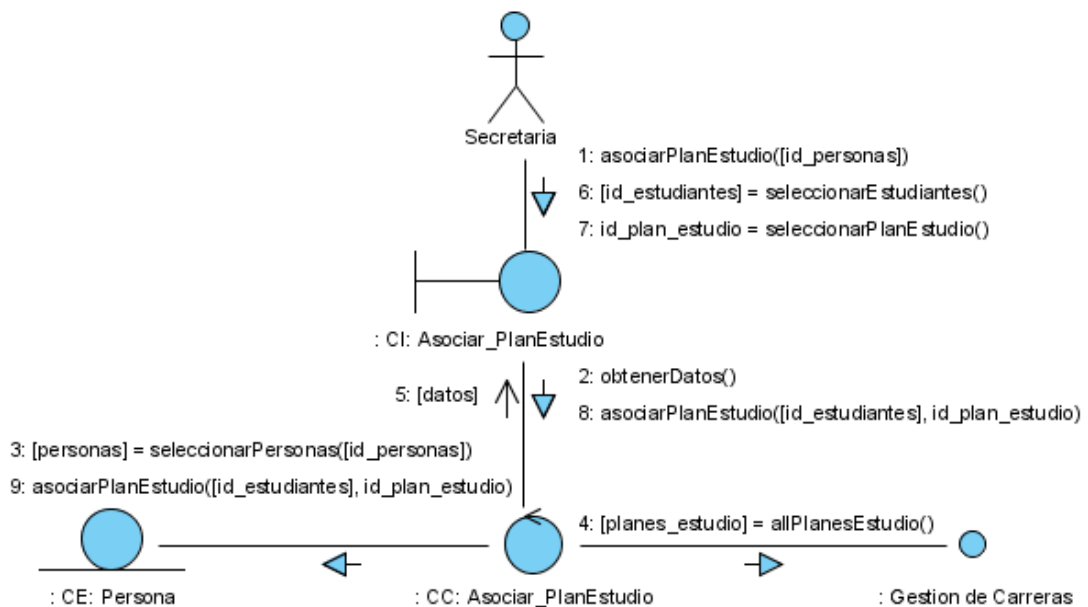


Figura 51: DC CUS _ Asociar Plan de Estudio

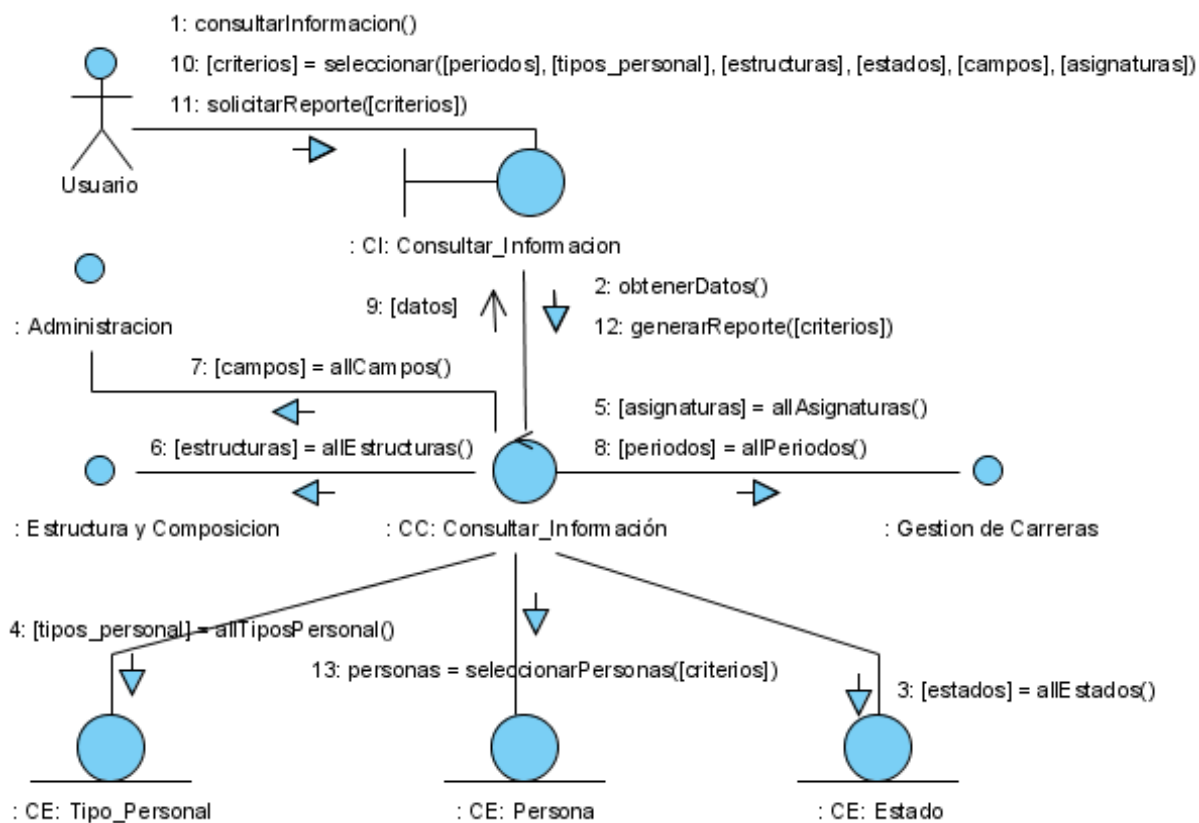


Figura 52: DC CUS_ Consultar Información

Anexo #4: Diagramas de Secuencia de los Casos de Uso del Sistema.

A

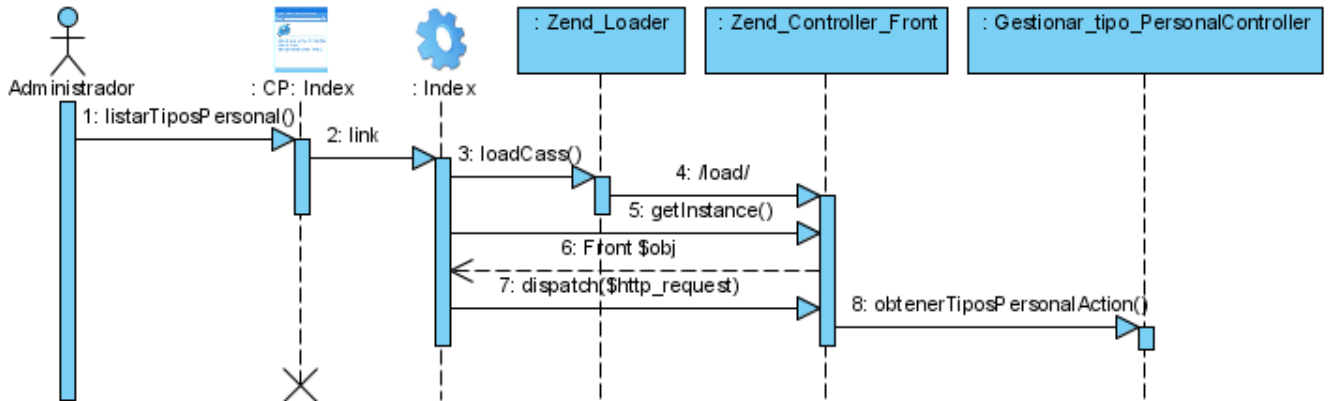


Figura 53: DS CUS_ Gestionar Tipo Personal: Sección Crear Tipo Personal A

B

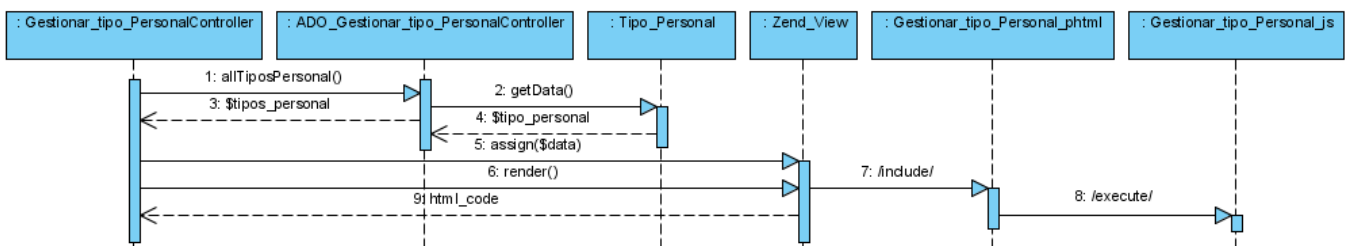


Figura 53: DS CUS_ Gestionar Tipo Personal: Sección Crear Tipo Personal B

C

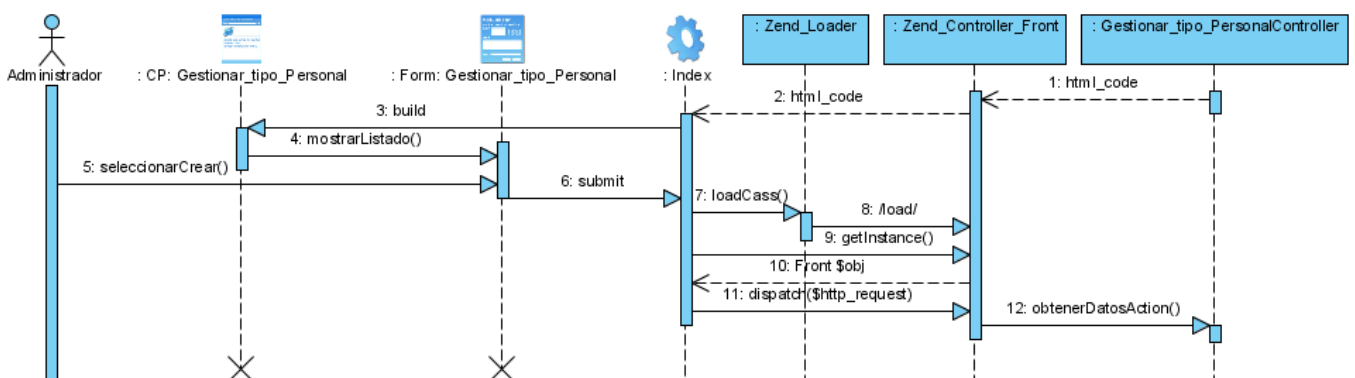


Figura 53: DS CUS_ Gestionar Tipo Personal: Sección Crear Tipo Personal C

D

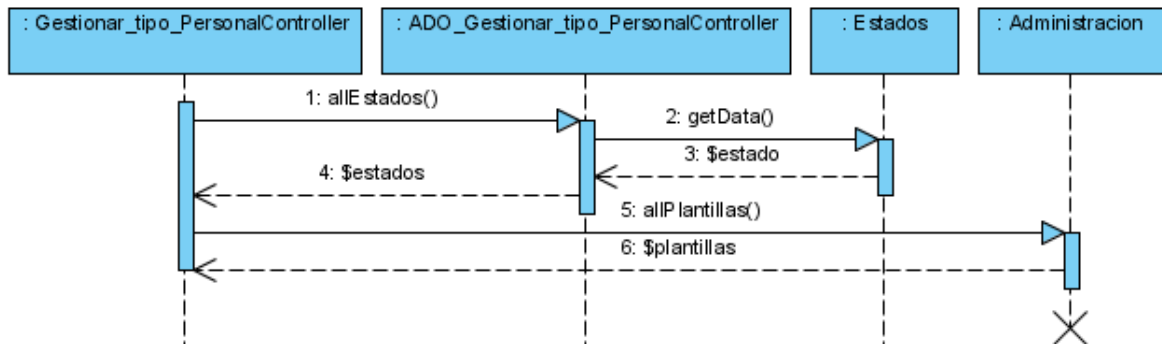


Figura 53: DS CUS _ Gestionar Tipo Personal: Sección Crear Tipo Personal D

E

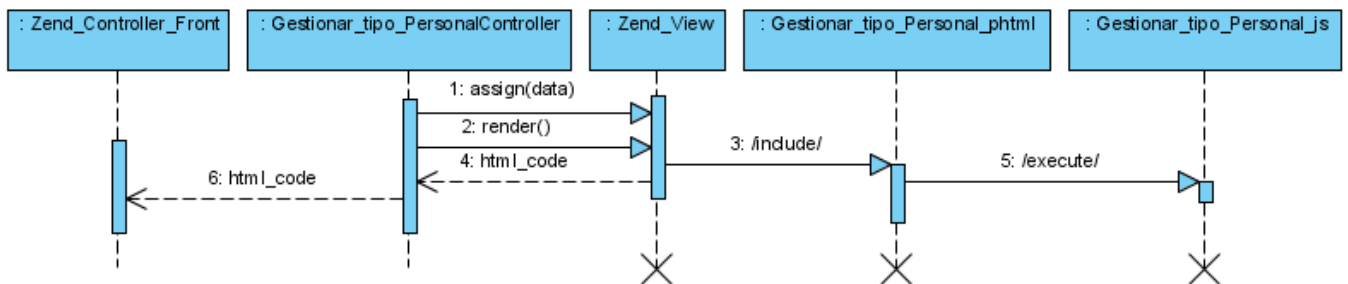


Figura 53: DS CUS _ Gestionar Tipo Personal: Sección Crear Tipo Personal E

F

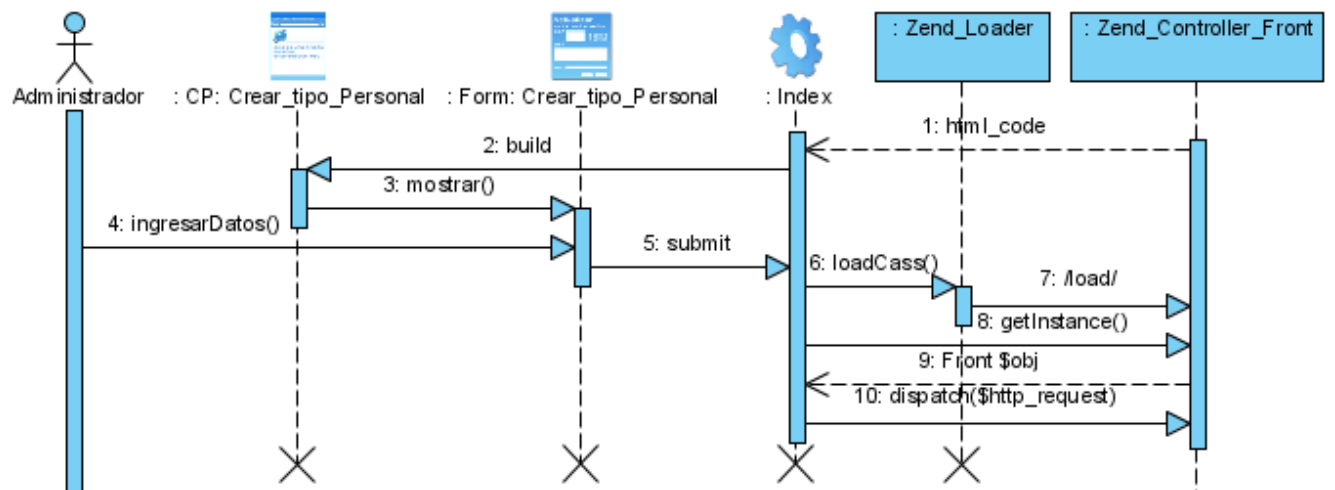


Figura 53: DS CUS _ Gestionar Tipo Personal: Sección Crear Tipo Personal F

G

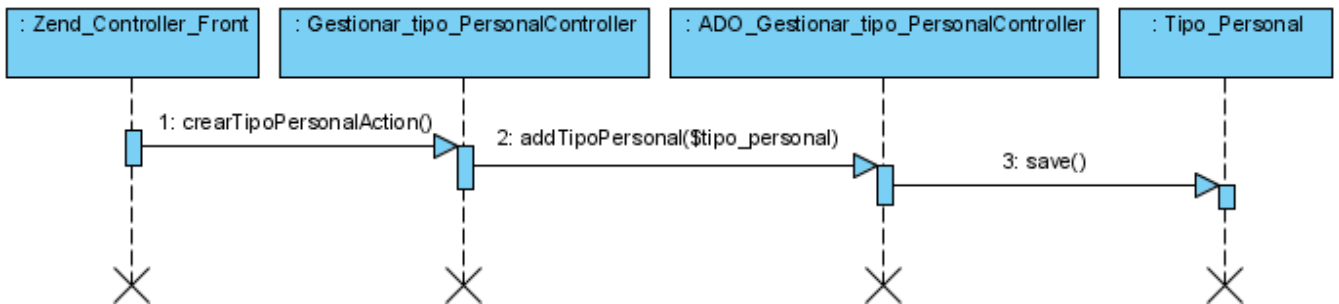


Figura 53: DS CUS _ Gestionar Tipo Personal: Sección Crear Tipo Personal G

CUS: Gestionar Tipo Personal: Sección Modificar Tipo Personal

A

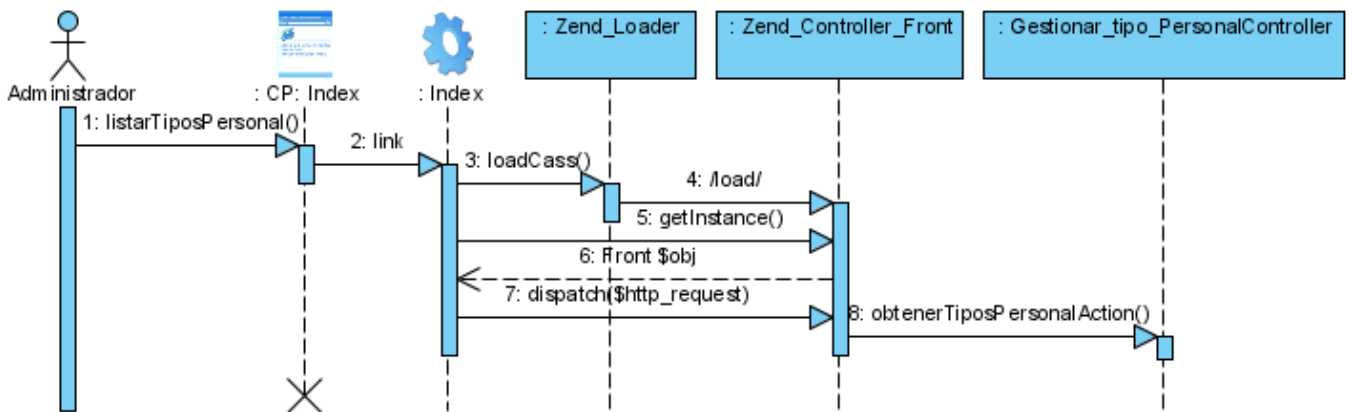


Figura 54: DS CUS _ Gestionar Tipo Personal: Sección Modificar Tipo Personal A

B

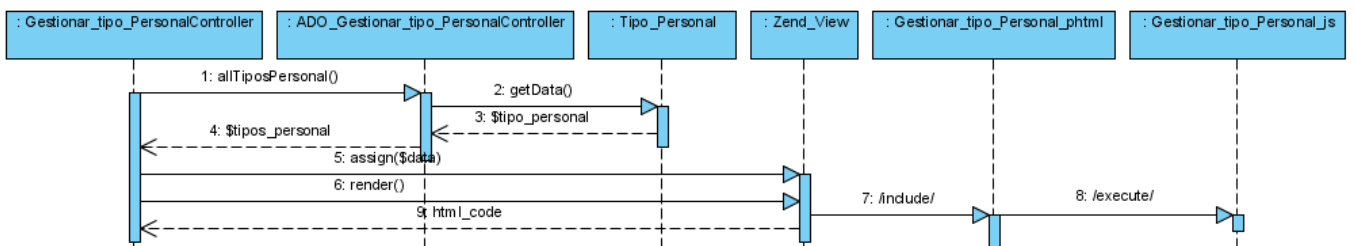


Figura 54: DS CUS _ Gestionar Tipo Personal: Sección Modificar Tipo Personal B

C

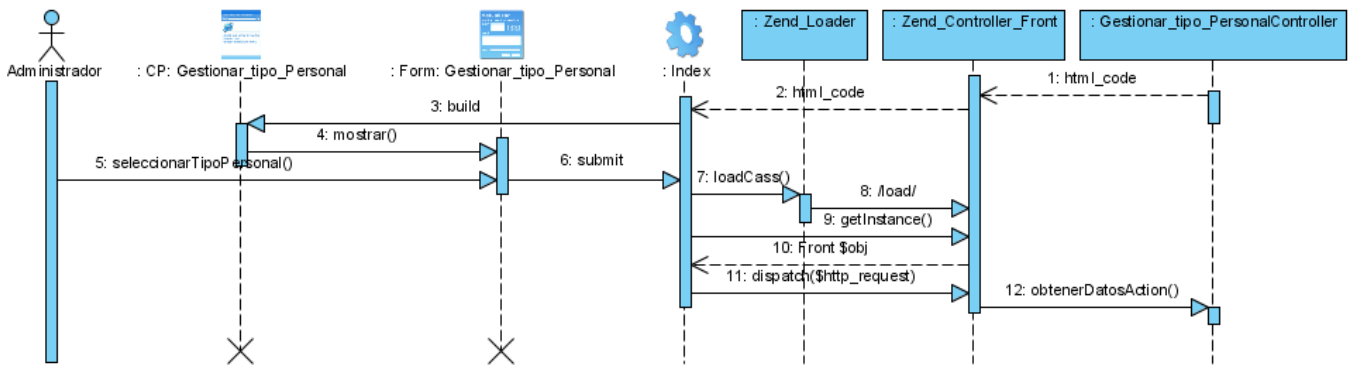


Figura 54: DS CUS _ Gestionar Tipo Personal: Sección Modificar Tipo Personal C

D

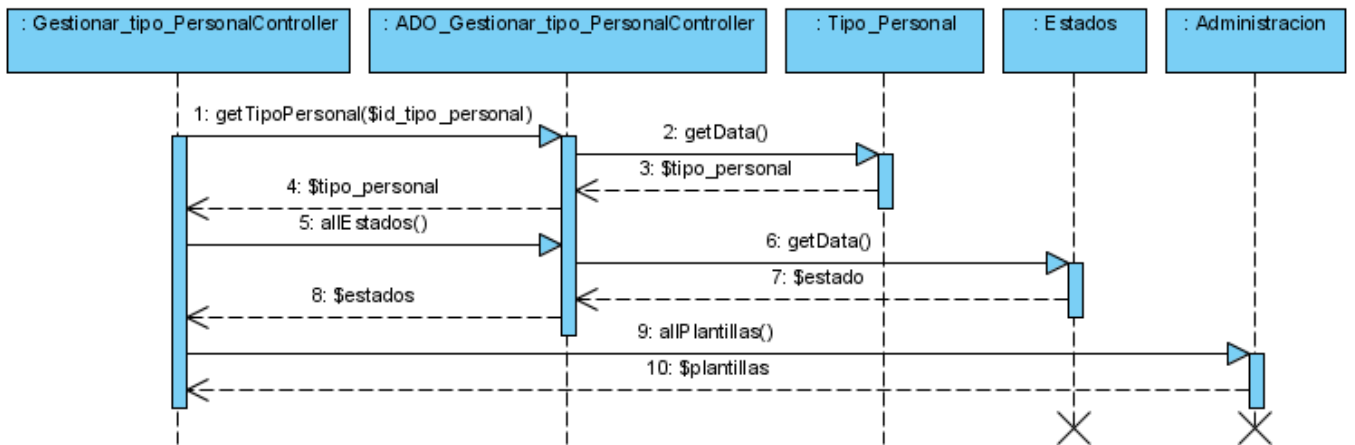


Figura 54: DS CUS _ Gestionar Tipo Personal: Sección Modificar Tipo Personal D

E

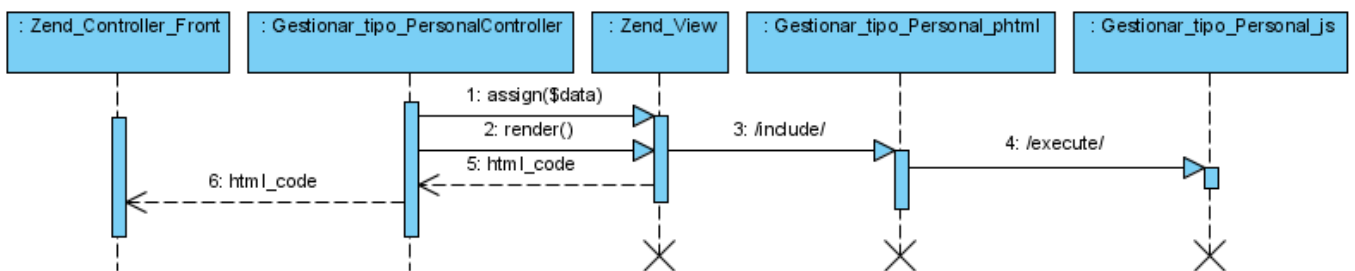


Figura 54: DS CUS _ Gestionar Tipo Personal: Sección Modificar Tipo Personal E



F

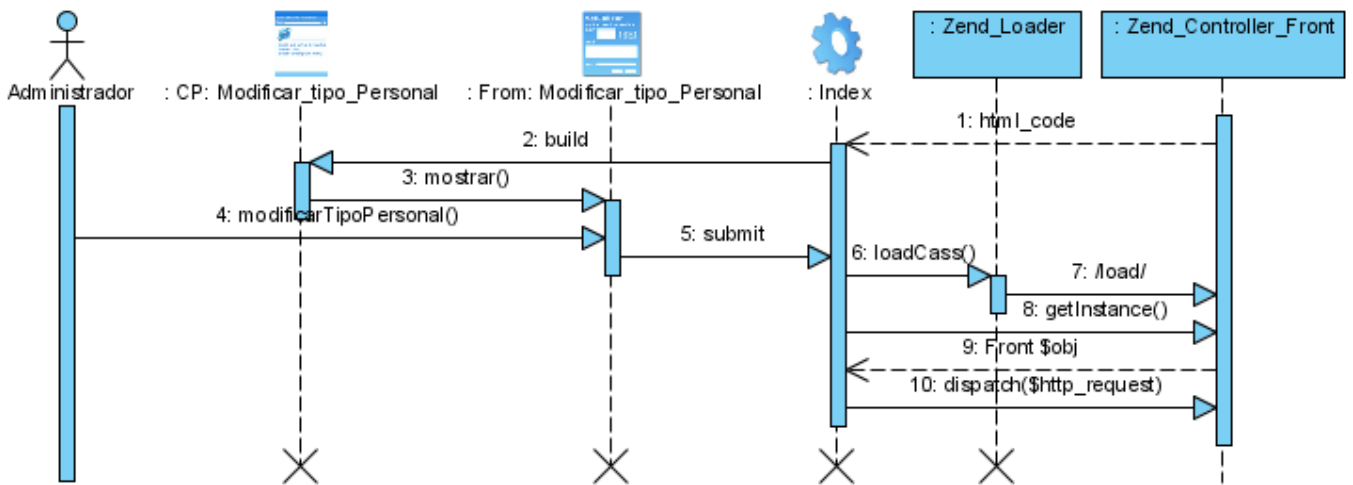


Figura 54: DS CUS _ Gestionar Tipo Personal: Sección Modificar Tipo Personal F

G

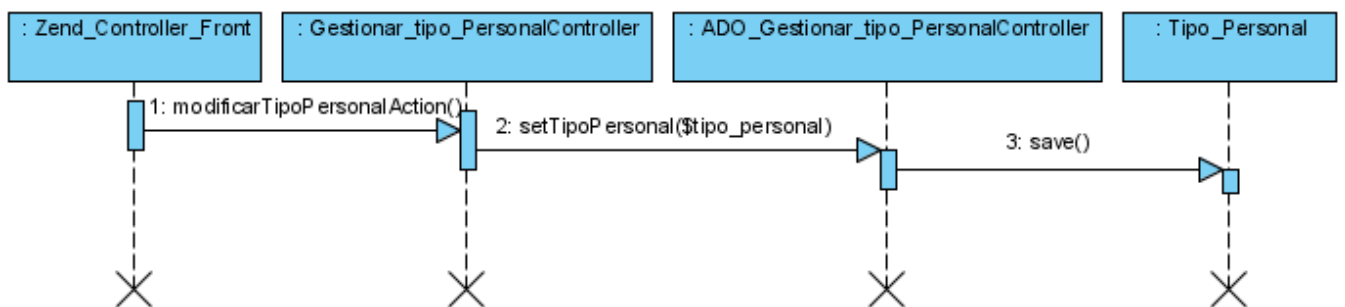


Figura 54: DS CUS _ Gestionar Tipo Personal: Sección Modificar Tipo Personal G

CUS: Gestionar Tipo Personal: Sección Eliminar Tipo Personal

A

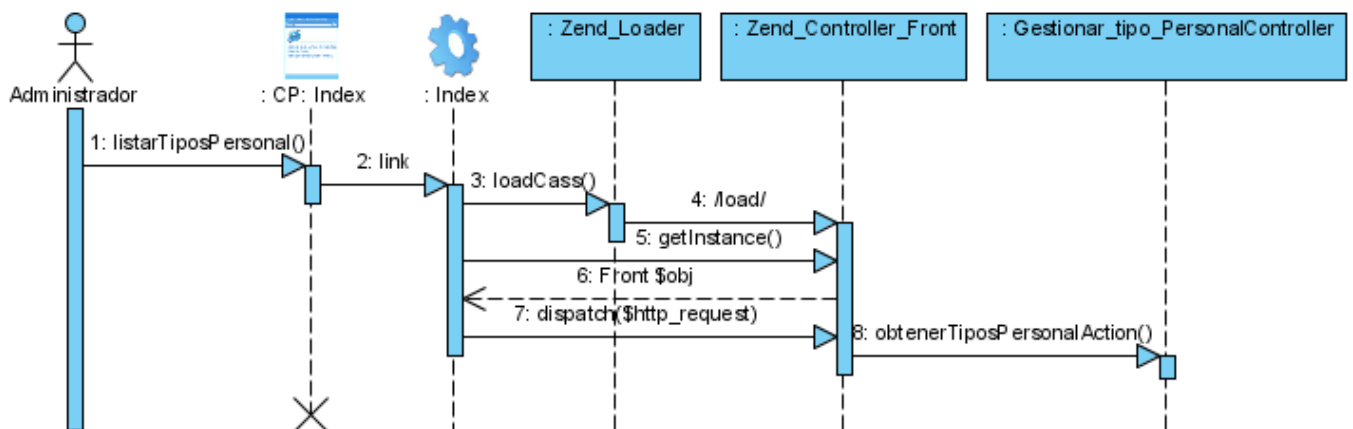


Figura 55: DS CUS _ Gestionar Tipo Personal: Sección Eliminar Tipo Personal A

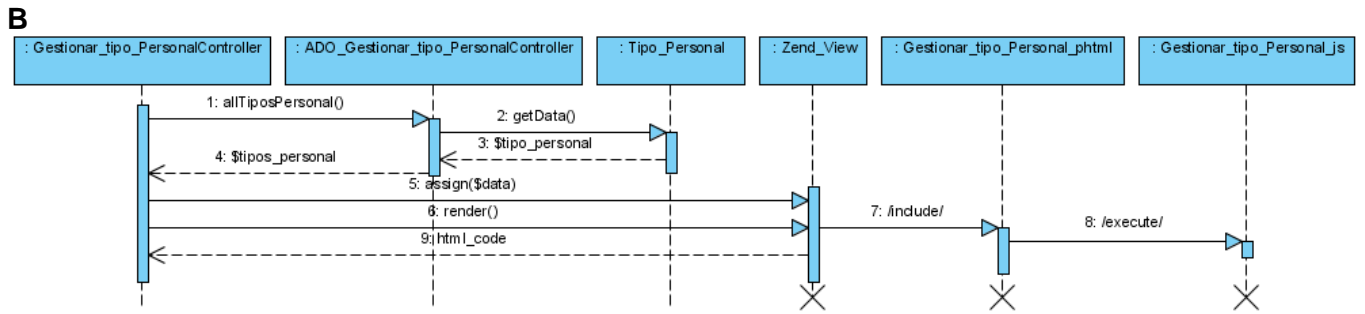


Figura 55: DS CUS _ Gestionar Tipo Personal: Sección Eliminar Tipo Personal B

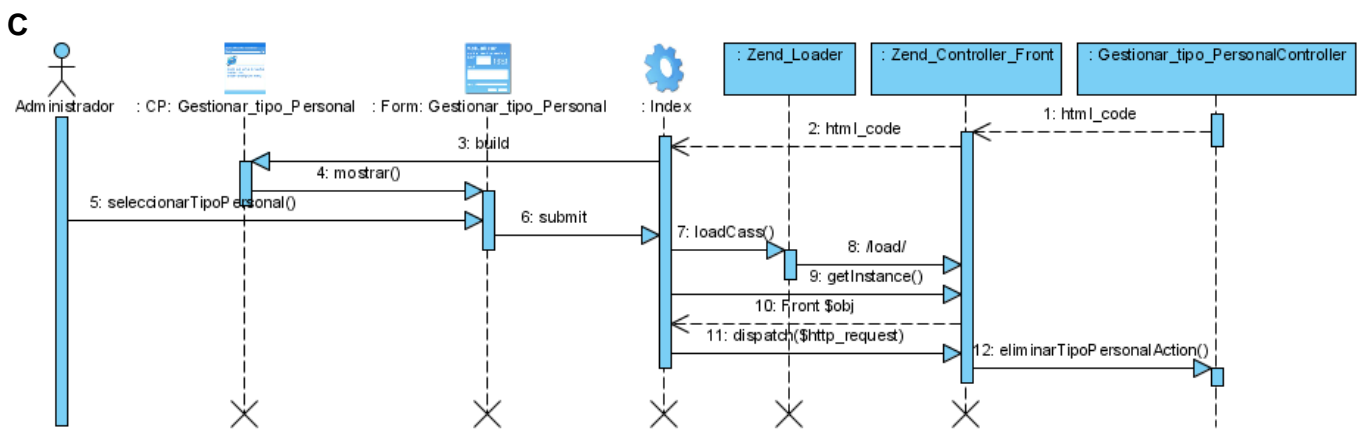


Figura 55: DS CUS _ Gestionar Tipo Personal: Sección Eliminar Tipo Personal C

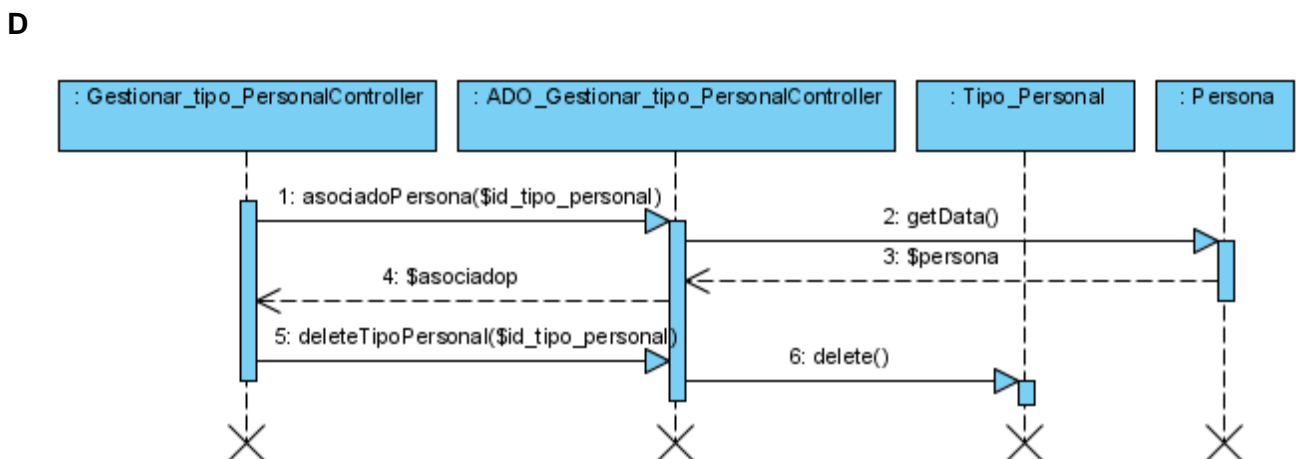


Figura 55: DS CUS _ Gestionar Tipo Personal: Sección Eliminar Tipo Personal D

CUS: Gestionar Estado: Sección Crear Estado

A

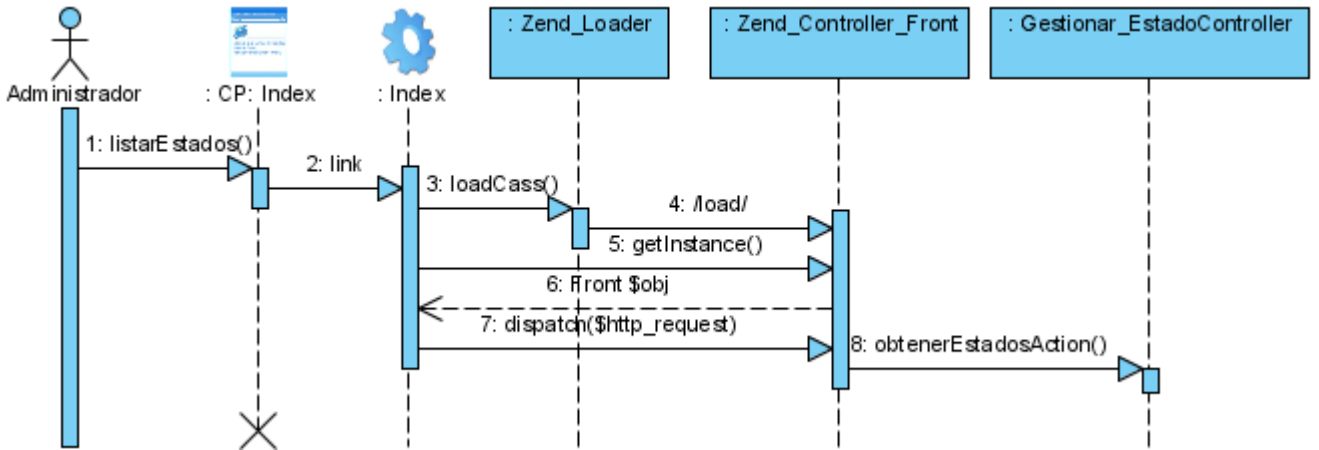


Figura 56: DS CUS _ Gestionar Estado: Sección Crear Estado A

B

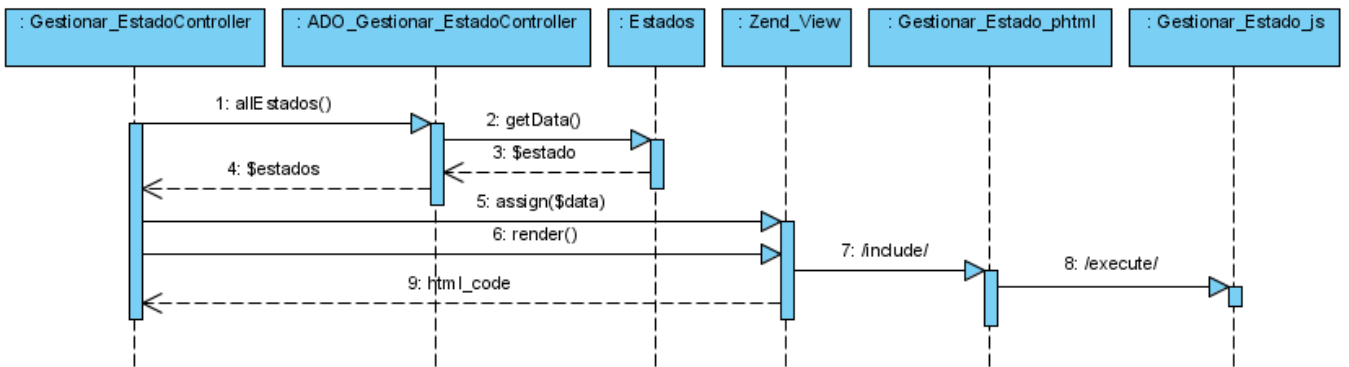


Figura 56: DS CUS _ Gestionar Estado: Sección Crear Estado B

C

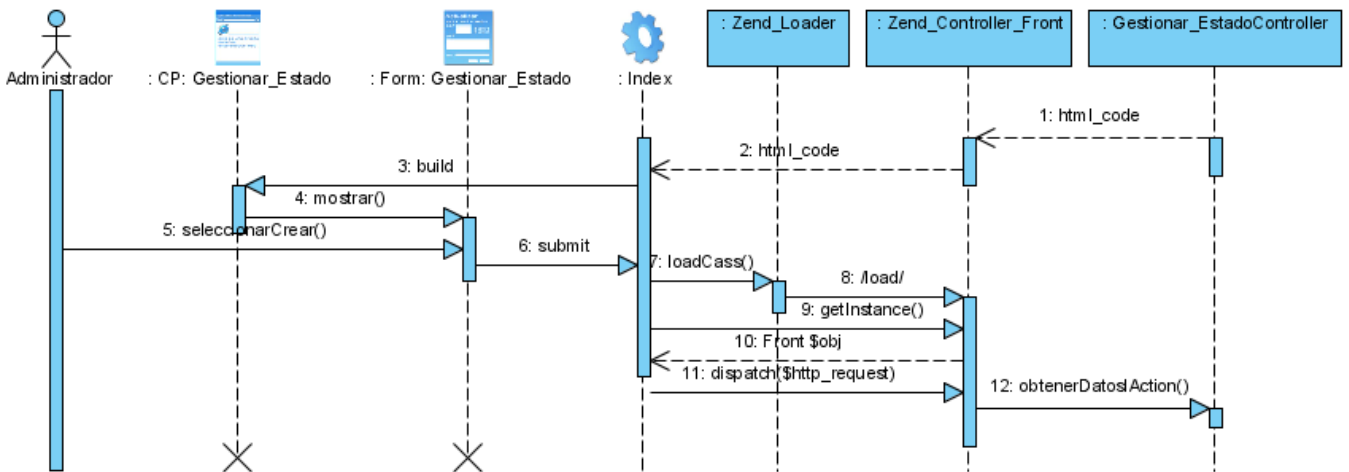


Figura 56: DS CUS _ Gestionar Estado: Sección Crear Estado C

D

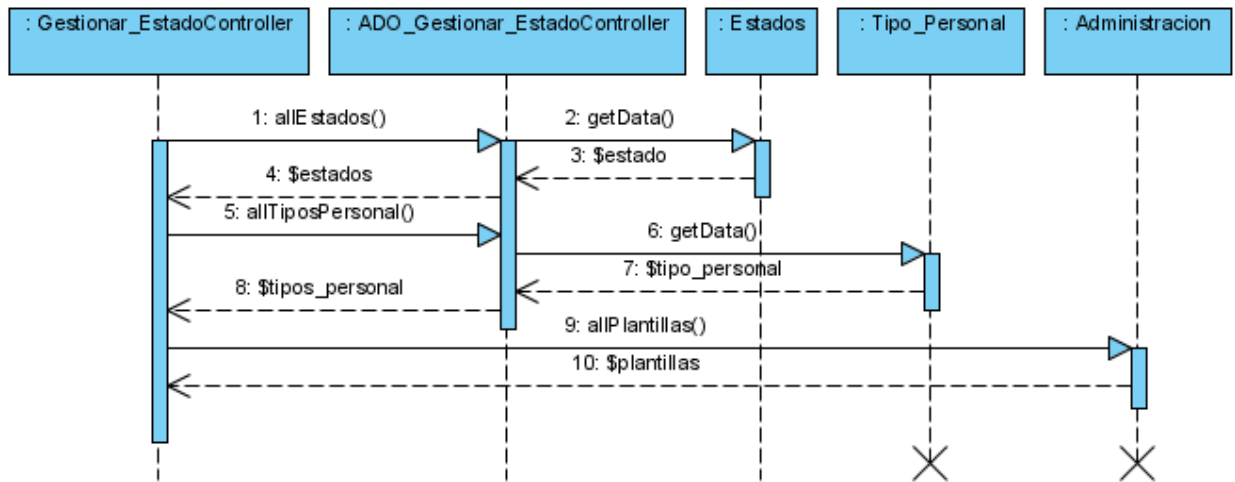


Figura 56: DS CUS _ Gestionar Estado: Sección Crear Estado D

E

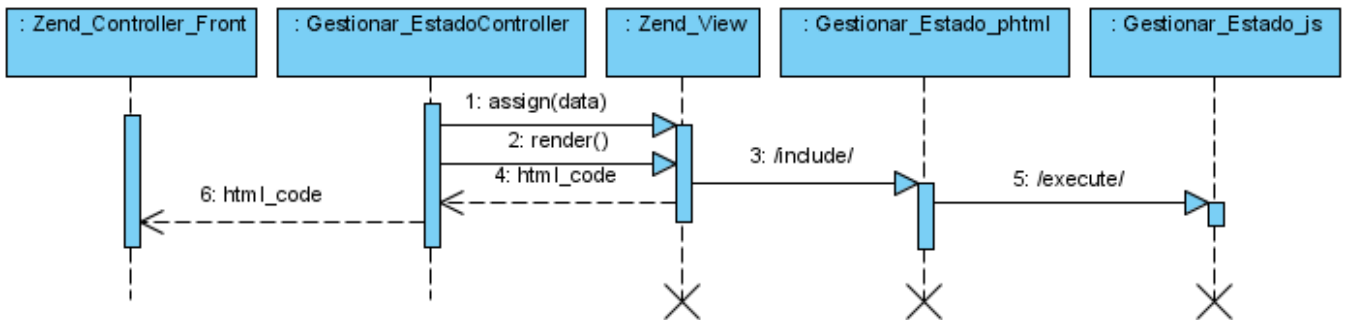


Figura 56: DS CUS _ Gestionar Estado: Sección Crear Estado E

F

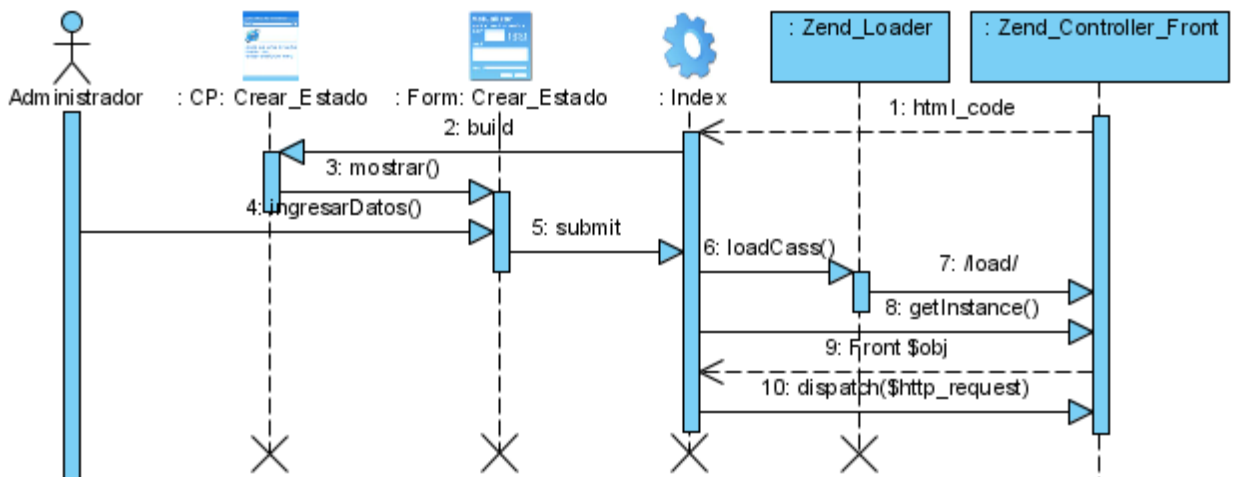


Figura 56: DC CUS _ Gestionar Estado: Sección Crear Estado F

G

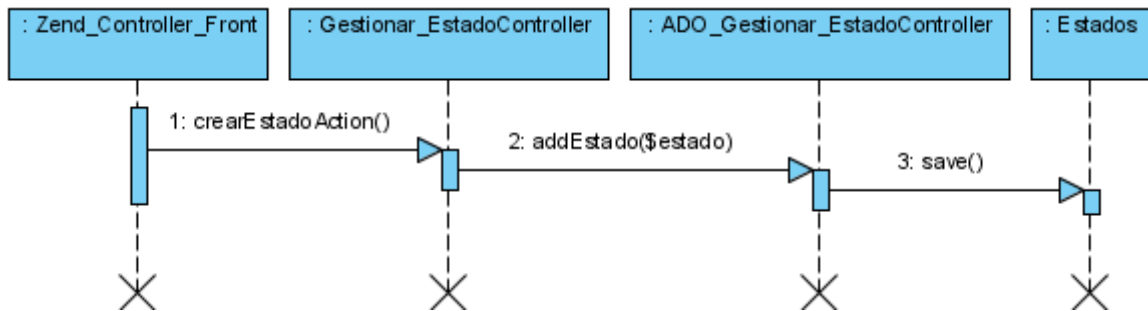


Figura 56: DC CUS _ Gestionar Estado: Sección Crear Estado G

CUS: Gestionar Estado: Sección Modificar Estado

A

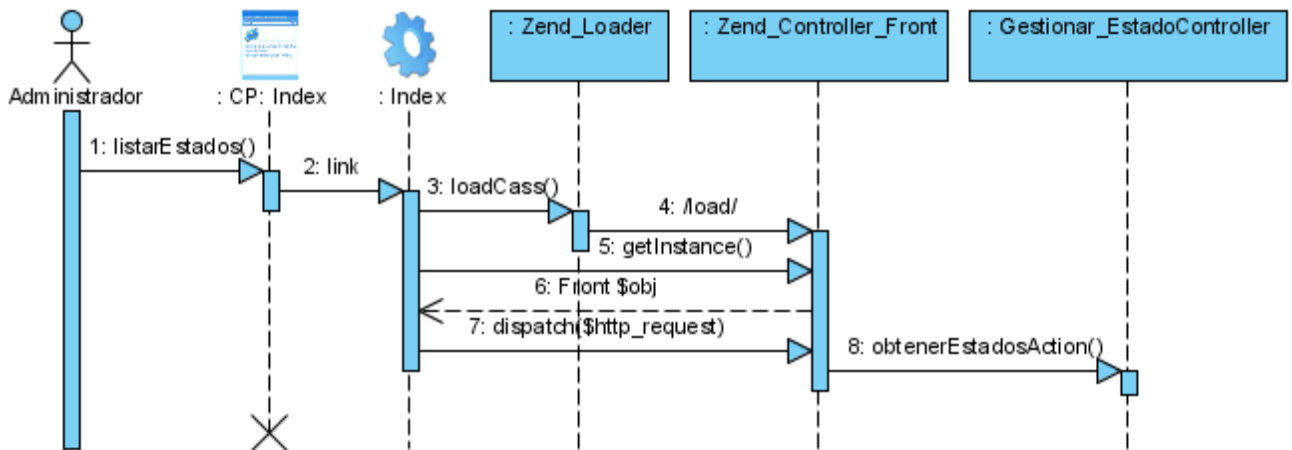


Figura 57: DC CUS _ Gestionar Estado: Sección Modificar Estado A

B

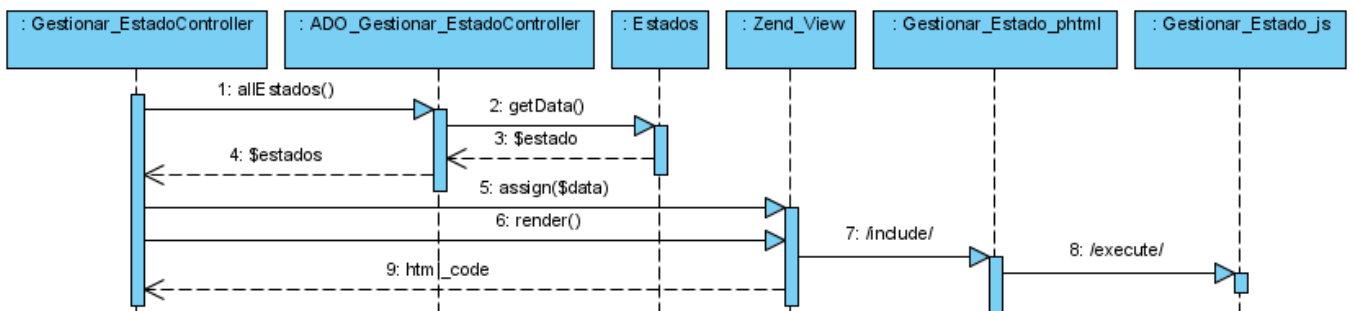


Figura 57: DC CUS _ Gestionar Estado: Sección Modificar Estado B

C

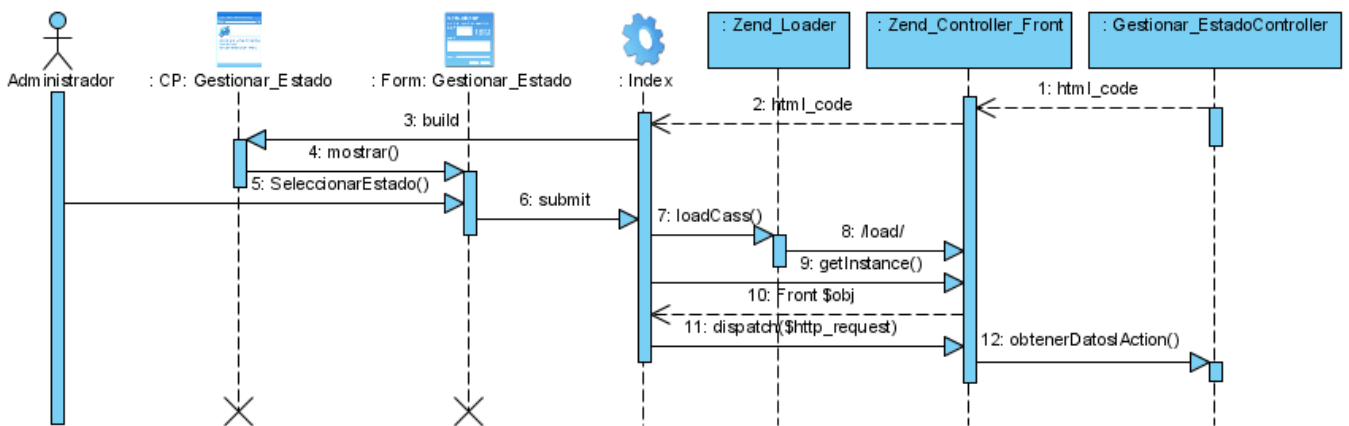


Figura 57: DS CUS _ Gestionar Estado: Sección Modificar Estado C

D

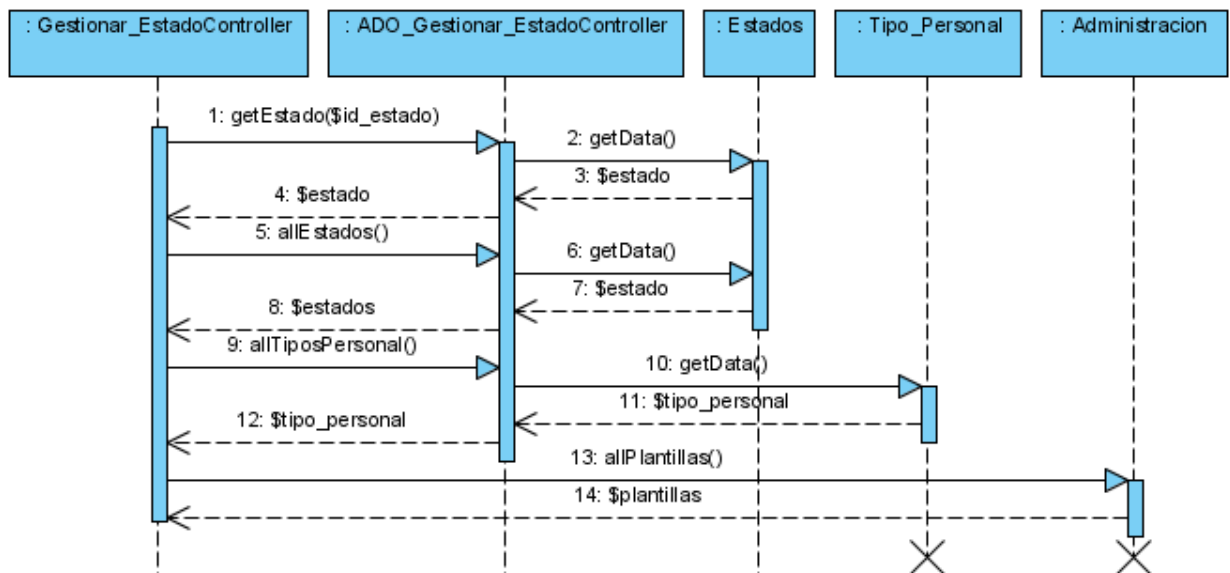


Figura 57: DS CUS _ Gestionar Estado: Sección Modificar Estado D

E

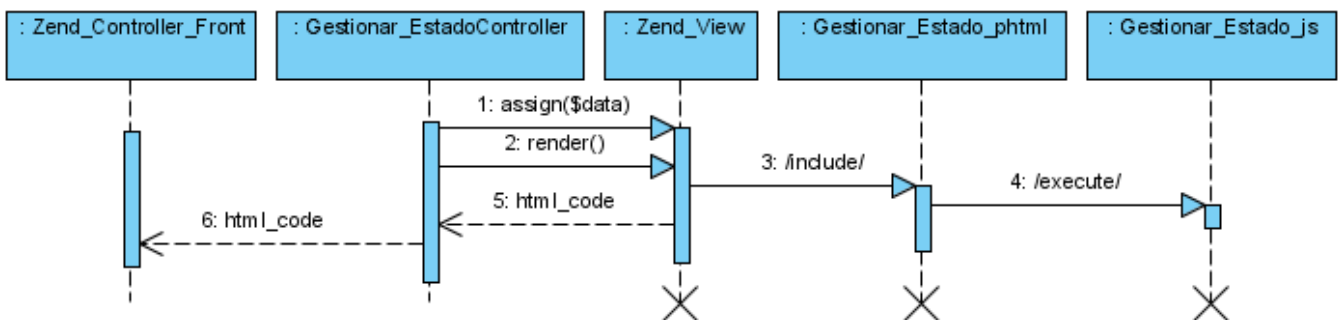


Figura 57: DS CUS _ Gestionar Estado: Sección Modificar Estado E

F

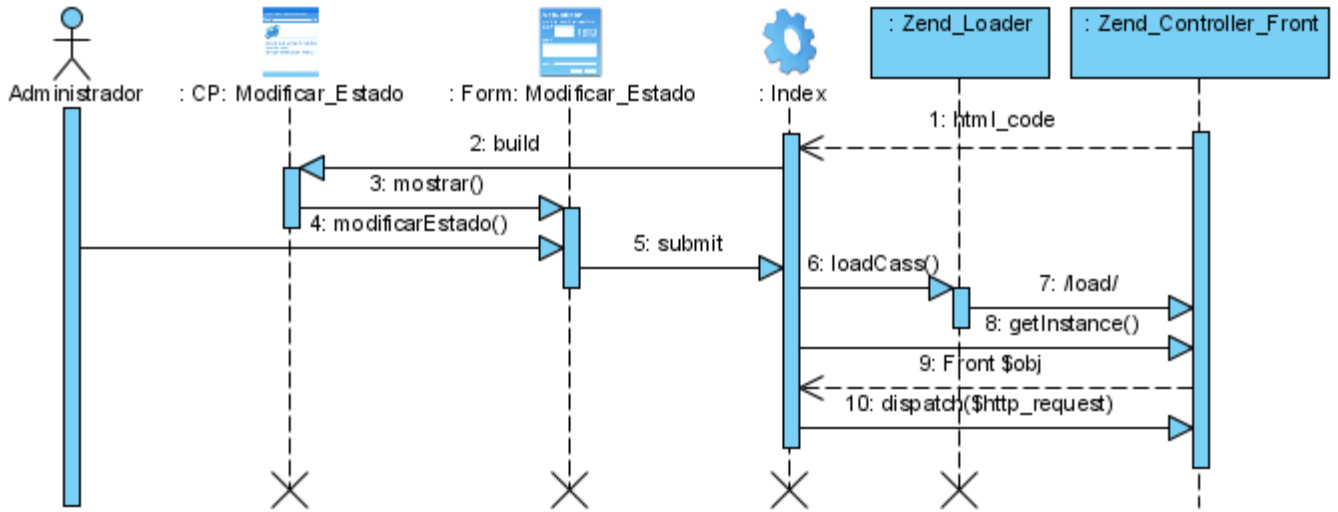


Figura 57: DS CUS _ Gestionar Estado: Sección Modificar Estado F

G

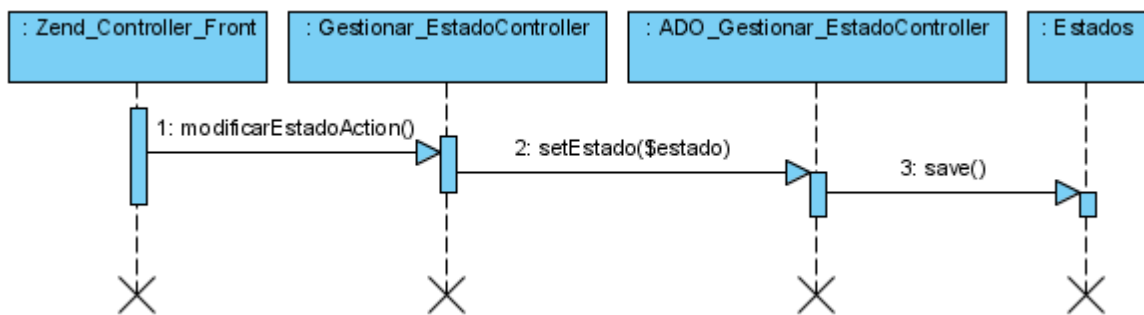


Figura 57: DS CUS _ Gestionar Estado: Sección Modificar Estado G

CUS: Gestionar Estado: Sección Eliminar Estado

A

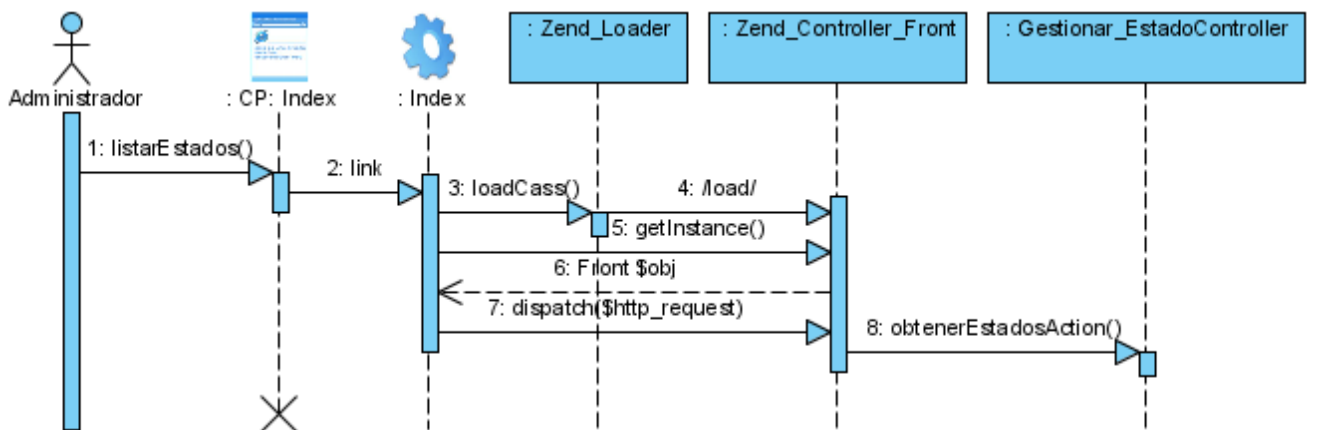


Figura 58: DC CUS _ Gestionar Estado: Sección Eliminar Estado A

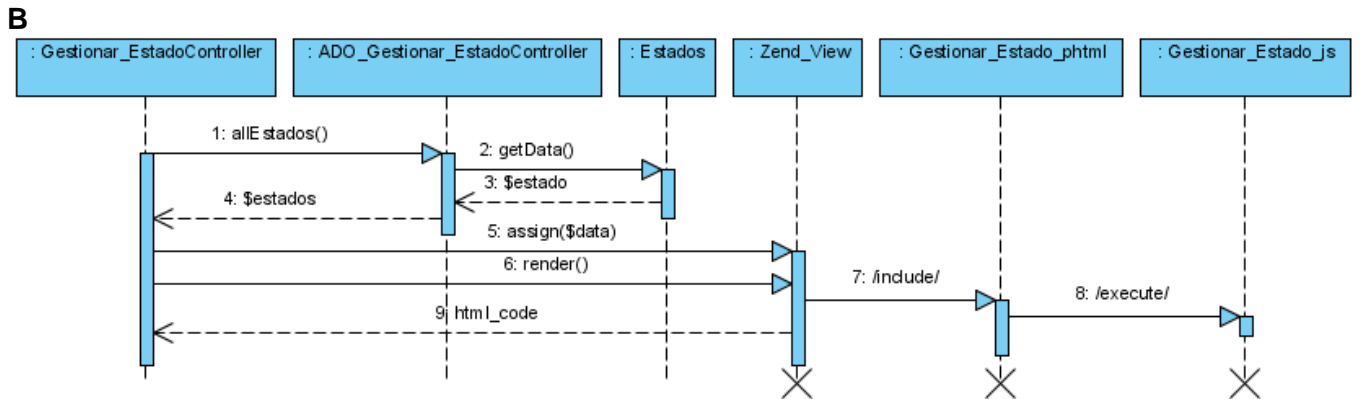


Figura 58: DS CUS _ Gestionar Estado: Sección Eliminar Estado B

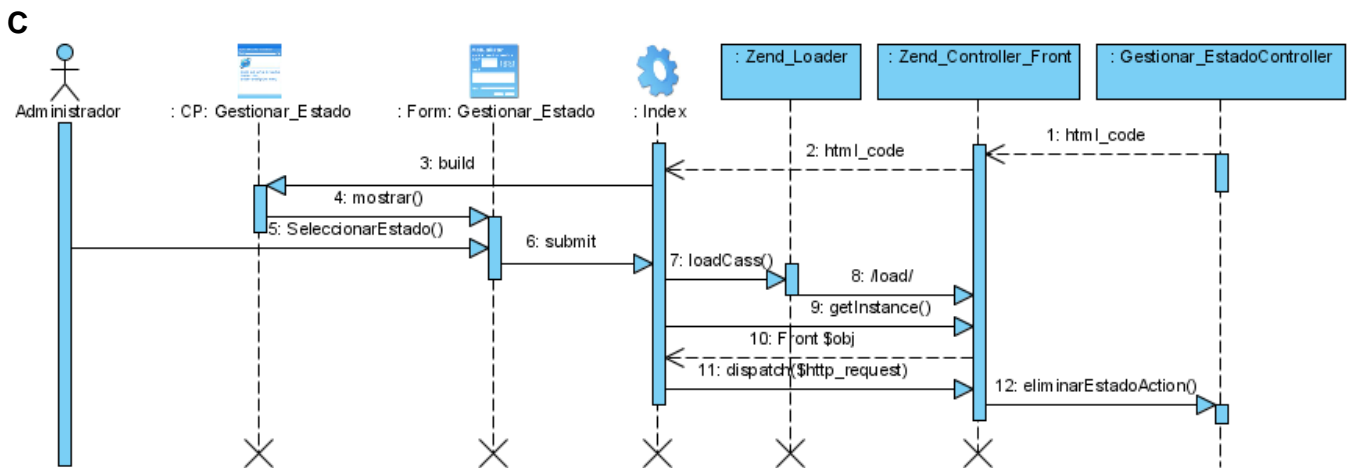


Figura 58: DS CUS _ Gestionar Estado: Sección Eliminar Estado C

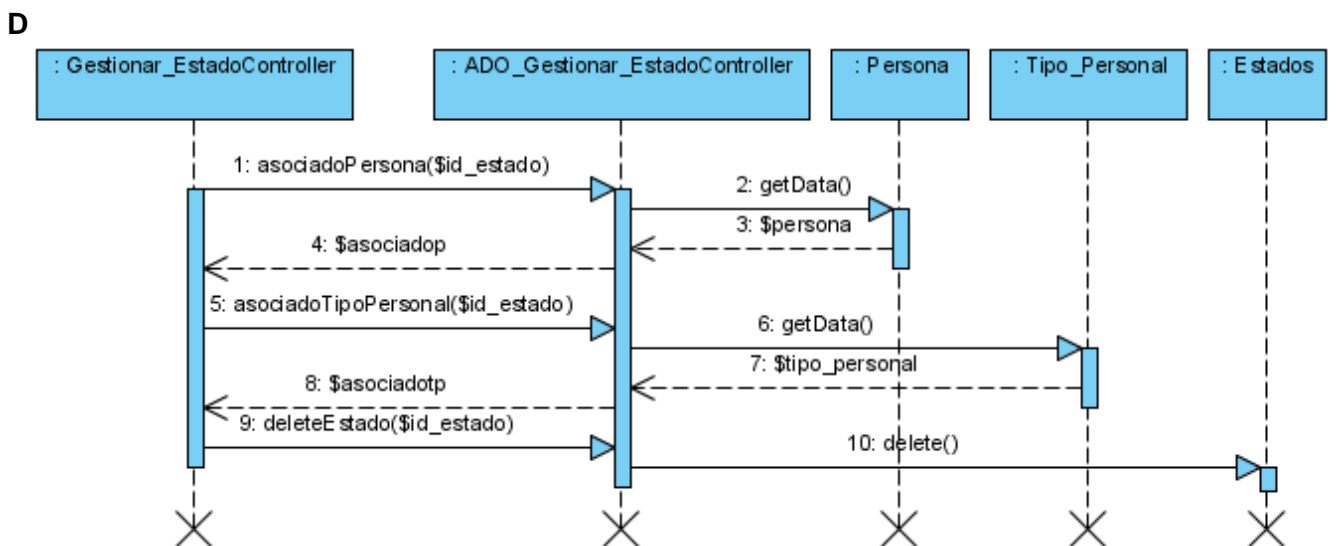


Figura 58: DS CUS _ Gestionar Estado: Sección Eliminar Estado D

CUS: Gestionar Datos Persona: Sección Registrar Datos Persona Manual

A

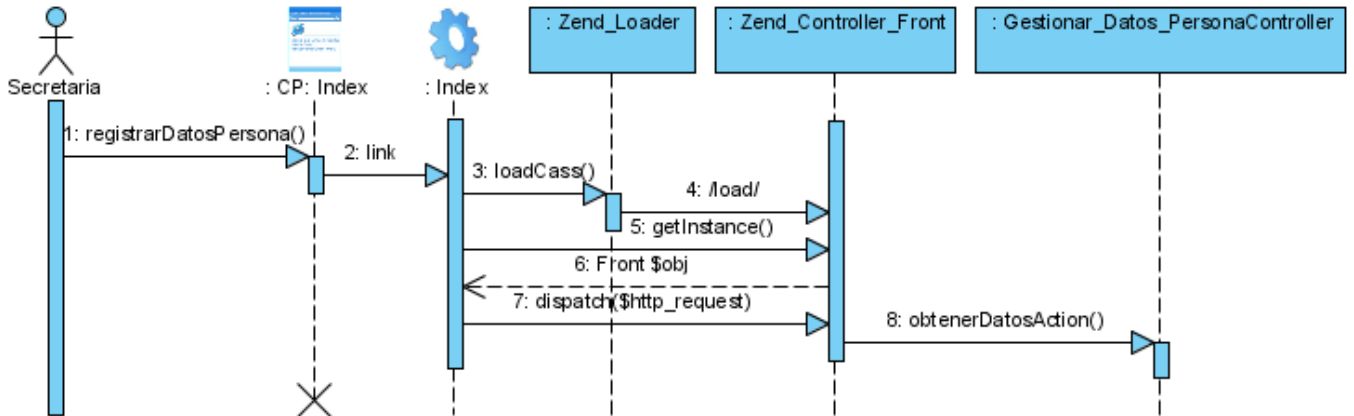


Figura 59: DS CUS _ Gestionar Datos Persona Sección Registrar Datos Persona Manual A

B

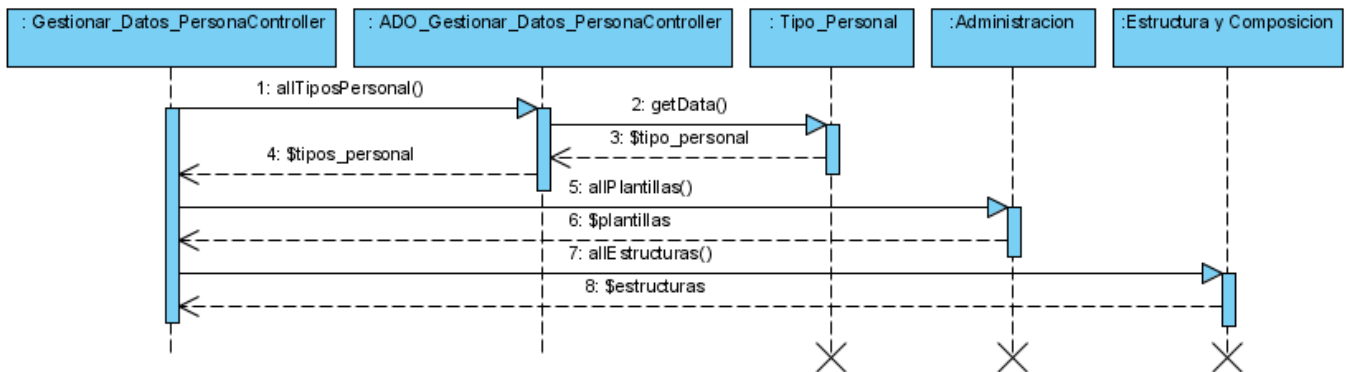


Figura 59: DS CUS _ Gestionar Datos Persona Sección Registrar Datos Persona Manual B

C

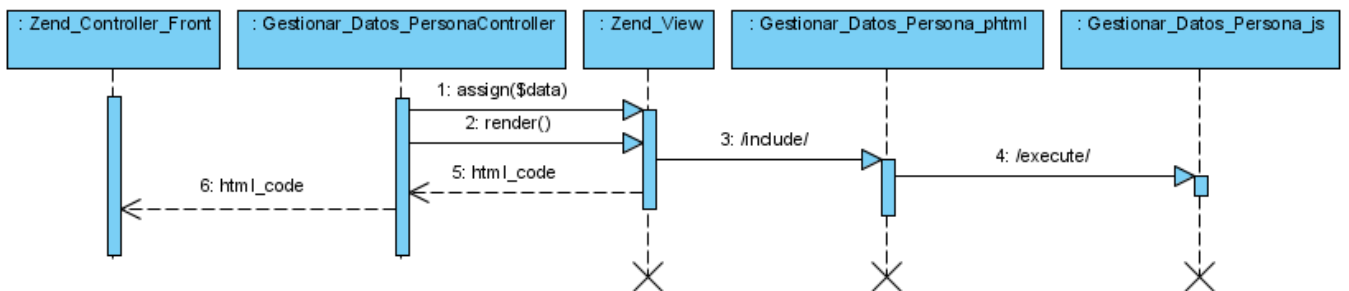


Figura 59: DS CUS _ Gestionar Datos Persona Sección Registrar Datos Persona Manual C

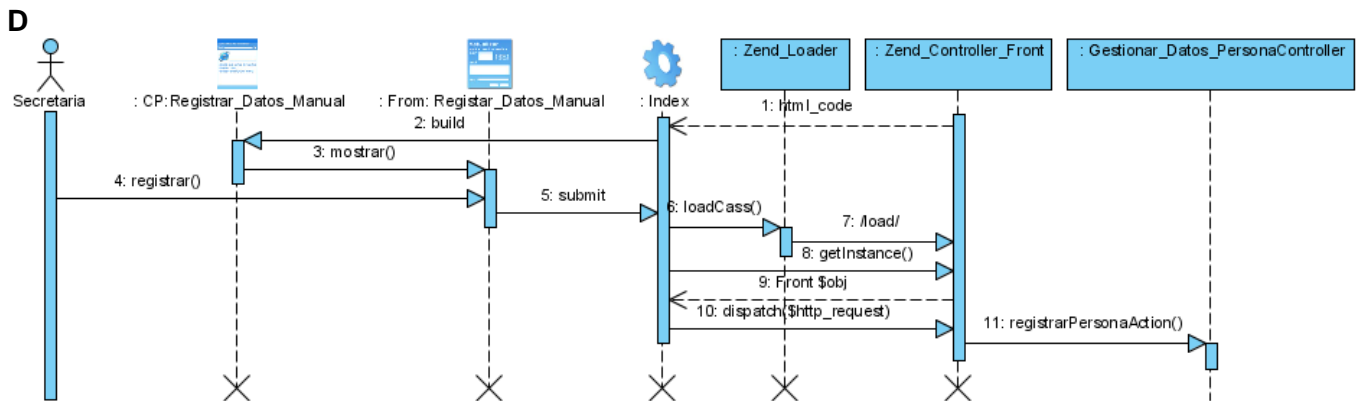


Figura 59: DC CUS _ Gestionar Datos Persona Sección Registrar Datos Persona Manual D

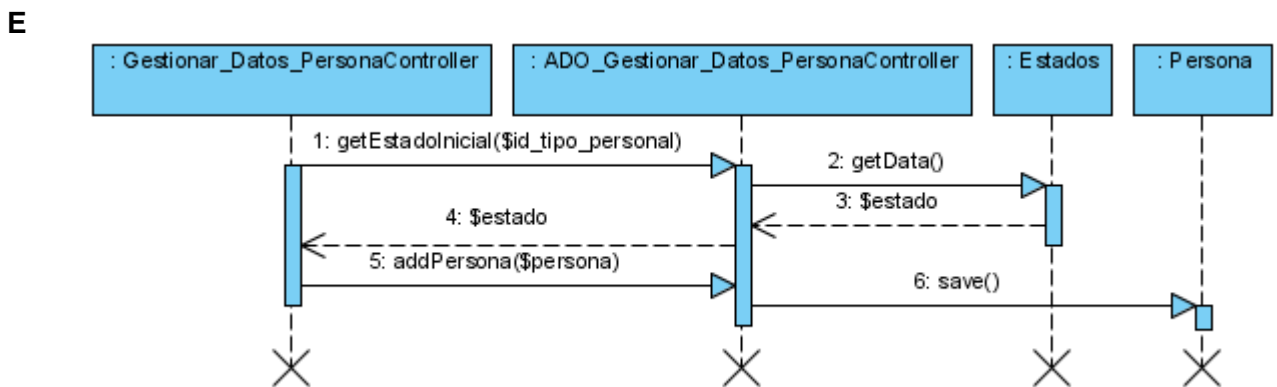


Figura 59: DS CUS _ Gestionar Datos Persona Sección Registrar Datos Persona Manual E

CUS: Gestionar Datos Persona: Sección Registrar Datos Persona Automático

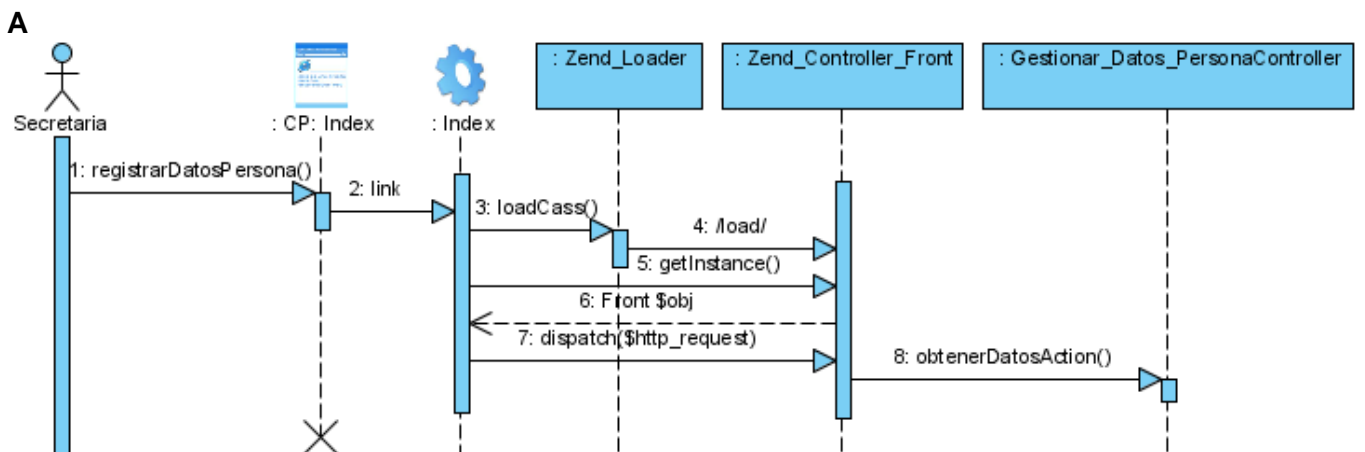


Figura 60: DS CUS _ Gestionar Datos Persona Sección Registrar Datos Persona Automático A

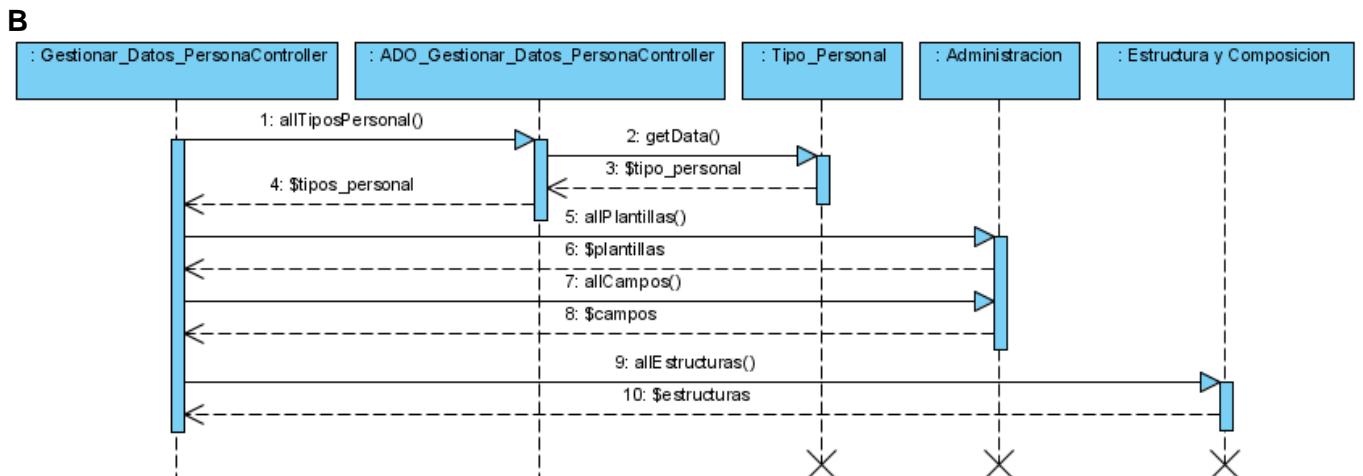


Figura 60: DS CUS_ Gestionar Datos Persona Sección Registrar Datos Persona Automático B

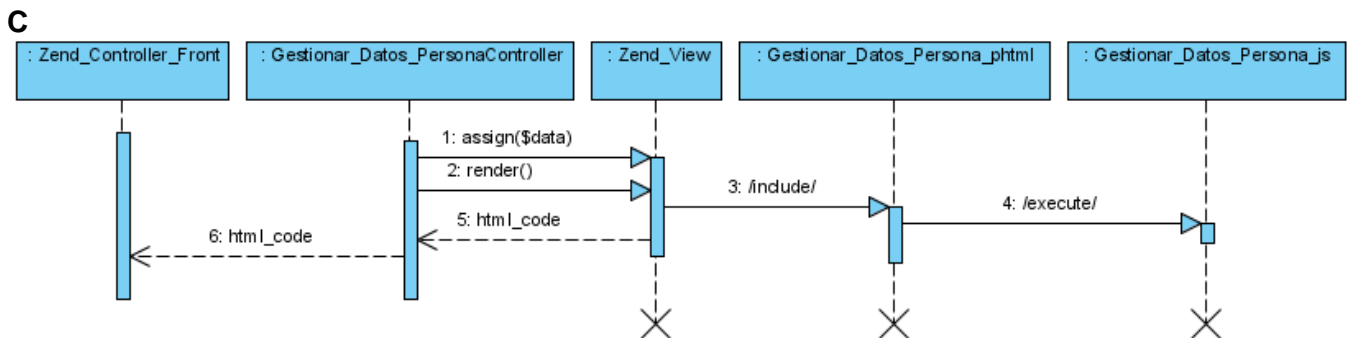


Figura 60: DS CUS_ Gestionar Datos Persona Sección Registrar Datos Persona Automático C

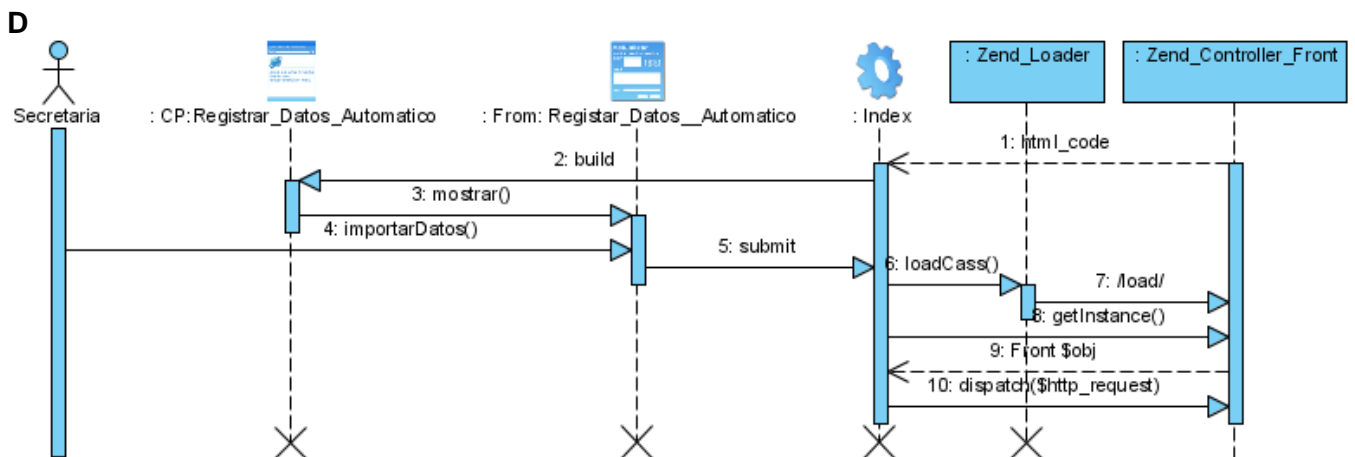


Figura 60: DS CUS_ Gestionar Datos Persona Sección Registrar Datos Persona Automático D

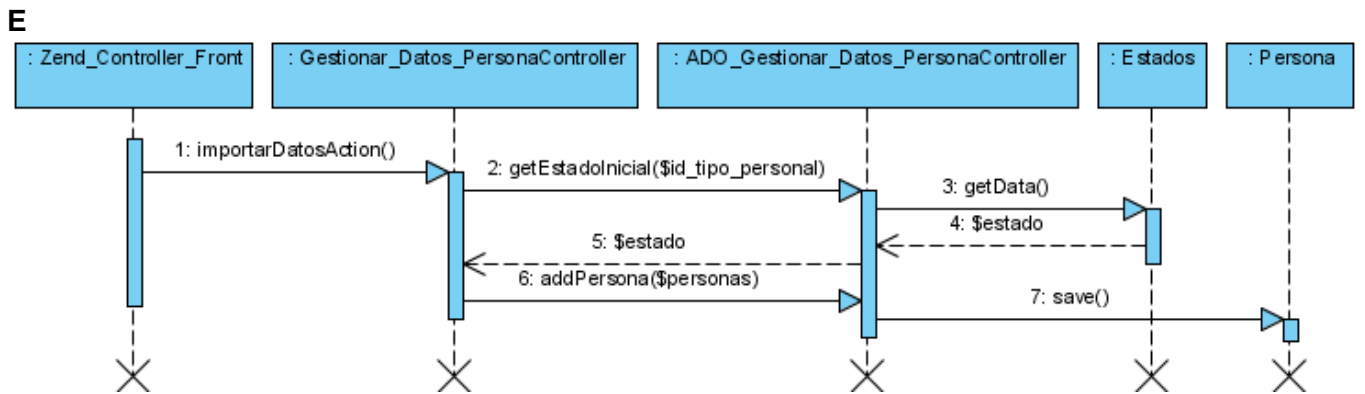


Figura 60: DS CUS Gestionar Datos Persona Sección Registrar Datos Persona Automático E

CUS: Gestionar Datos Persona: Sección Modificar Datos Persona

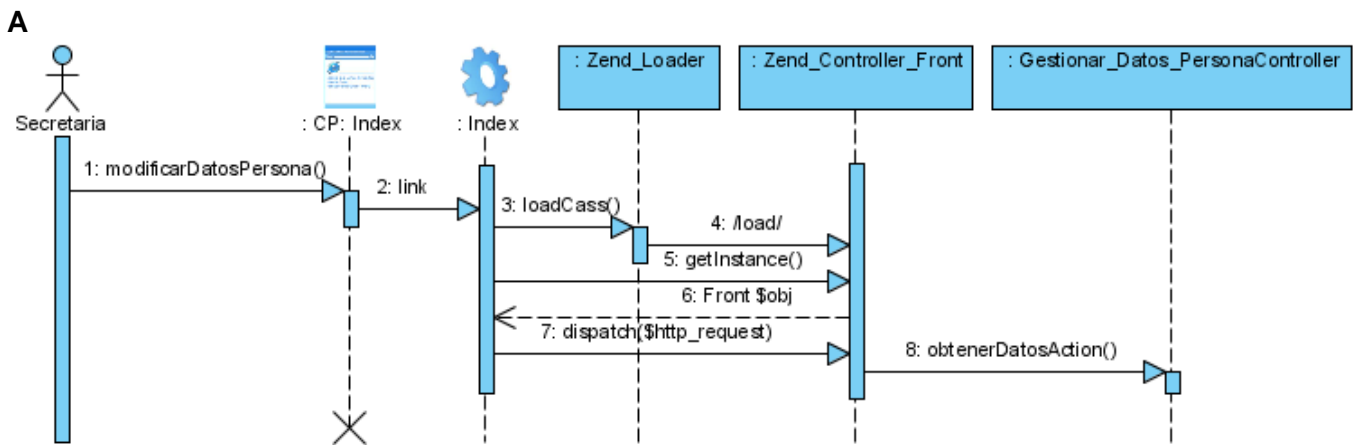


Figura 61: DC CUS_ Gestionar Datos Persona Sección Modificar Datos Persona A

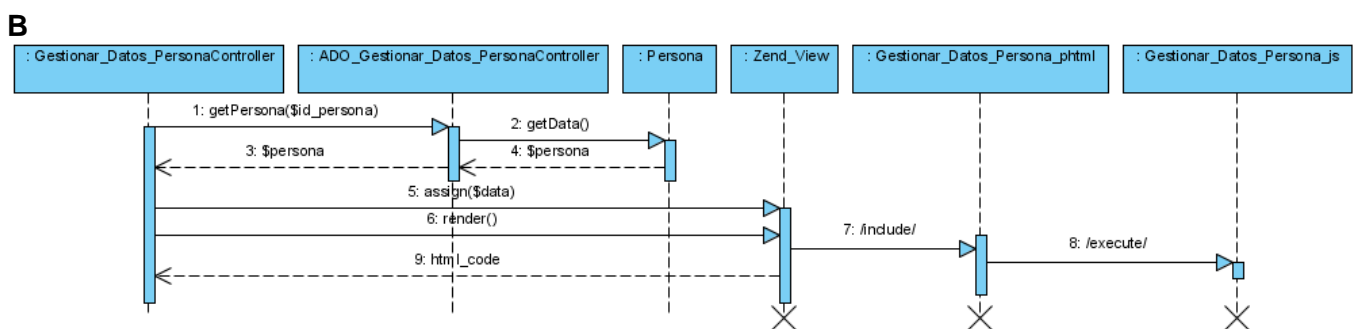


Figura 61: DC CUS_ Gestionar Datos Persona Sección Modificar Datos Persona B

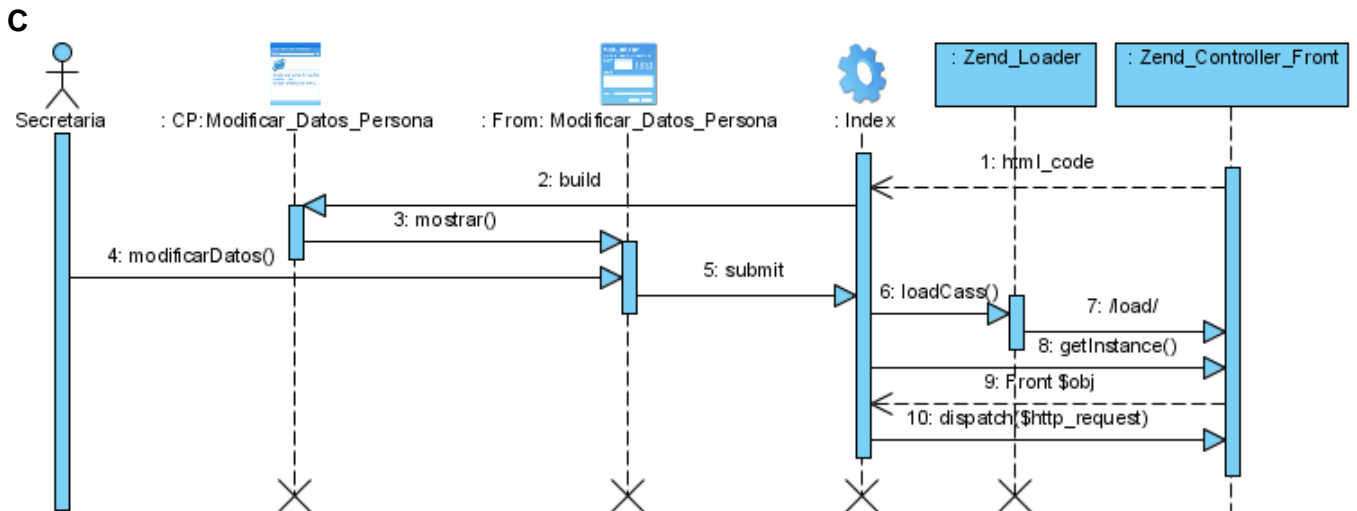


Figura 61: DC CUS_ Gestionar Datos Persona Sección Modificar Datos Persona C

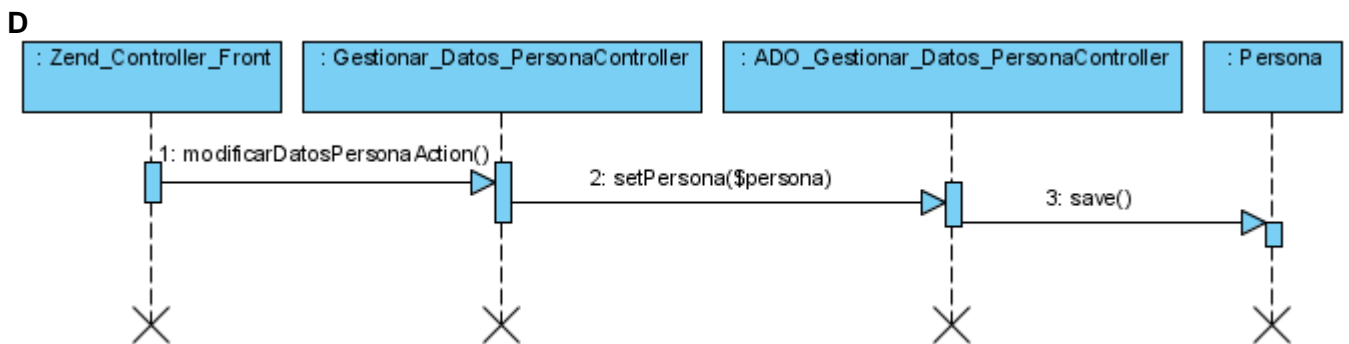


Figura 61: DC CUS_ Gestionar Datos Persona Sección Modificar Datos Persona D

CUS: Buscar Persona

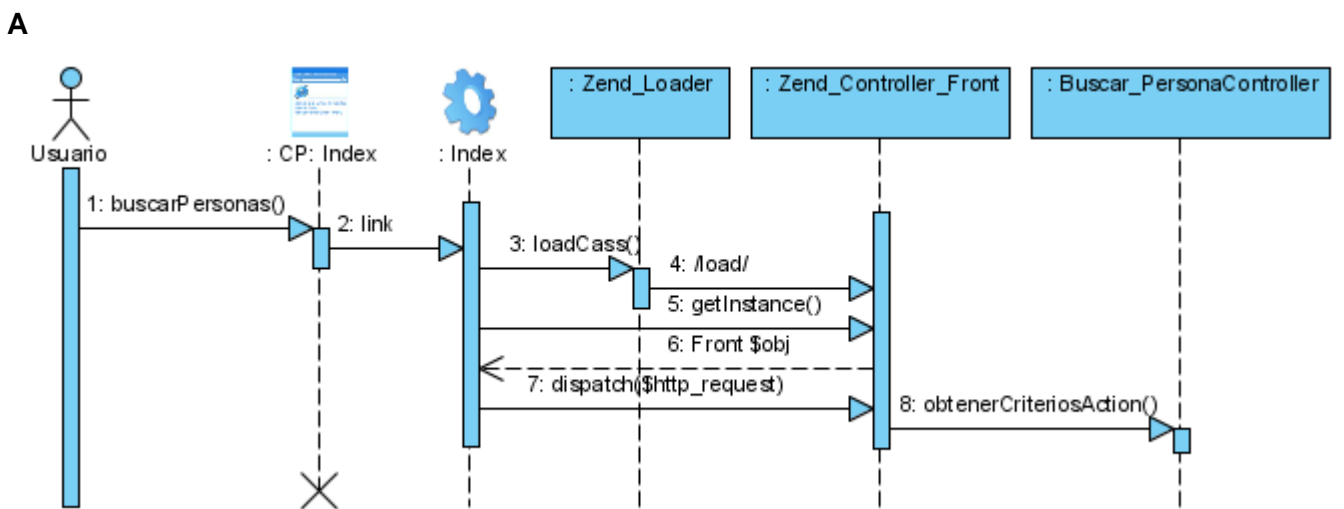


Figura 62: DS CUS_ Buscar Persona A

B

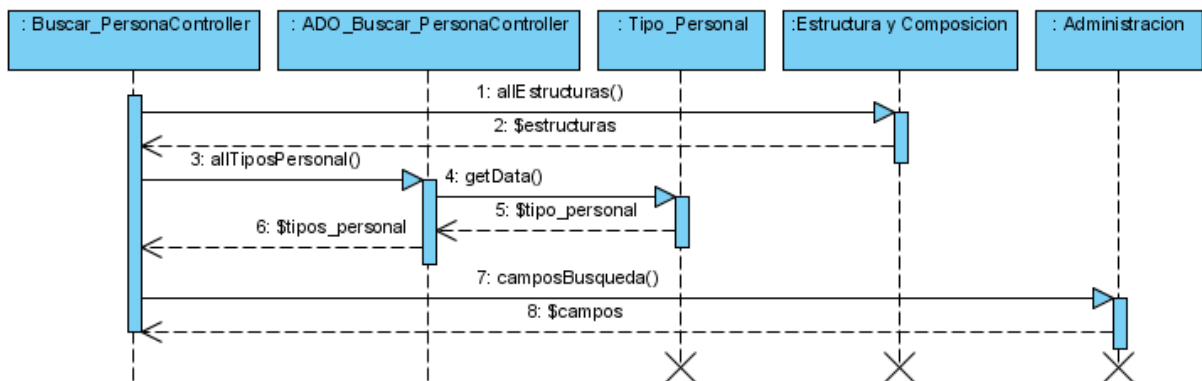


Figura 62: DS CUS_Buscar Persona B

C

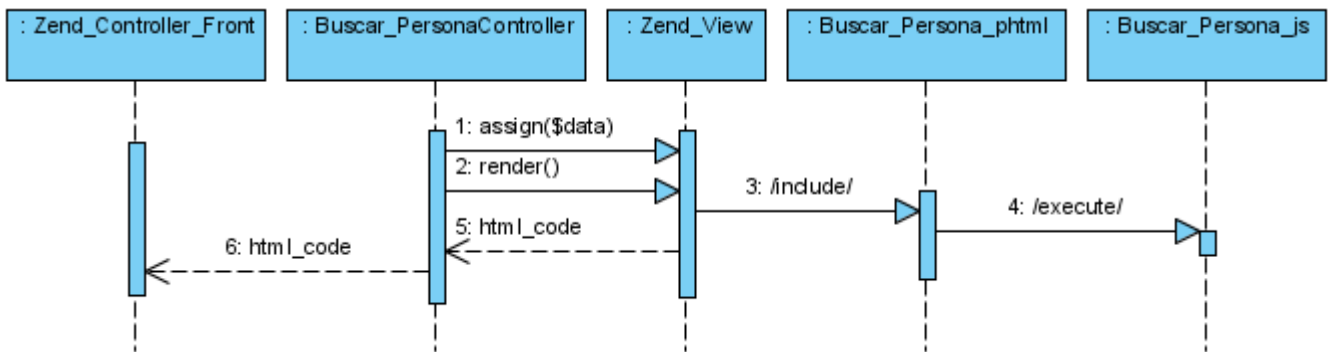


Figura 62: DS CUS_Buscar Persona C

D

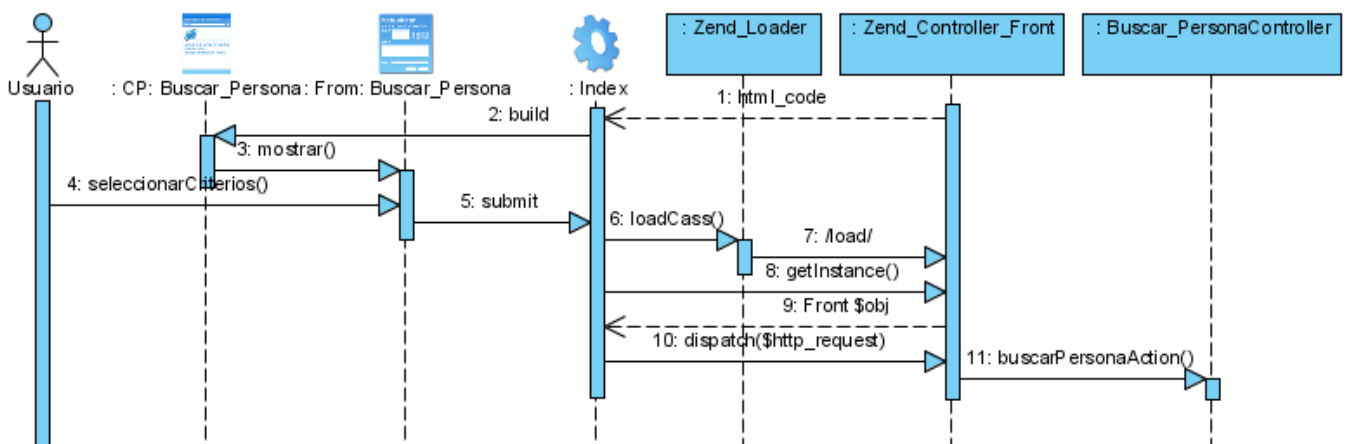


Figura 62: DS CUS_Buscar Persona D

E

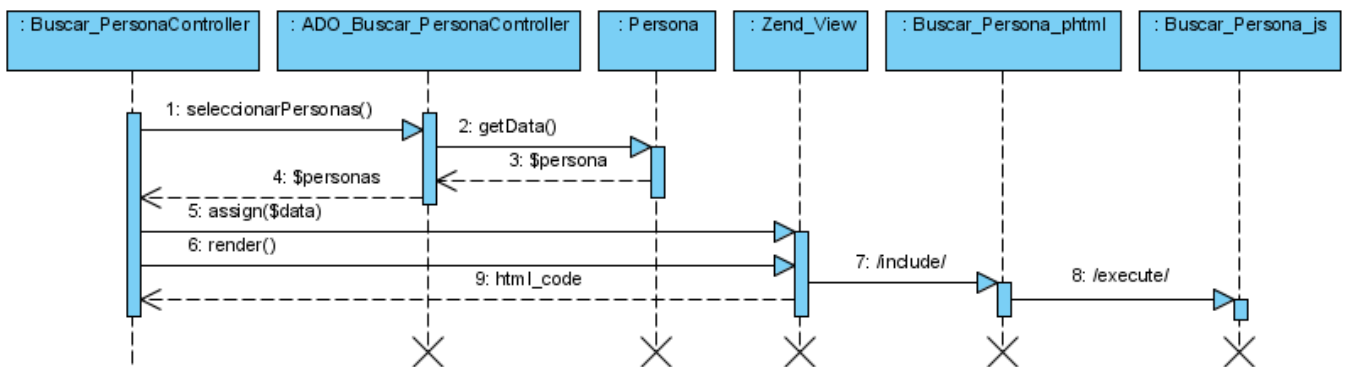


Figura 62: DS CUS_Buscar Persona E

F

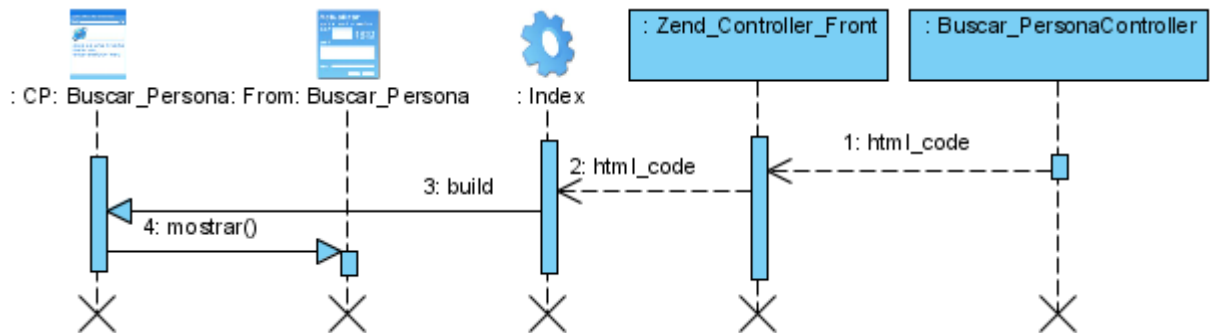


Figura 62: DS CUS_Buscar Persona F

CUS: Realizar Movimiento

A

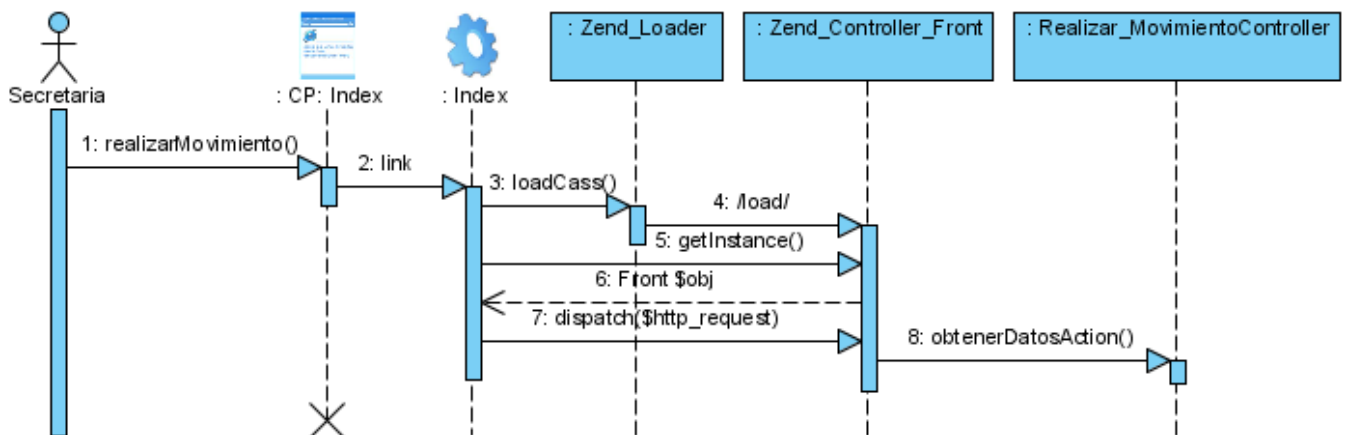


Figura 63: DS CUS_Realizar Movimiento A

B

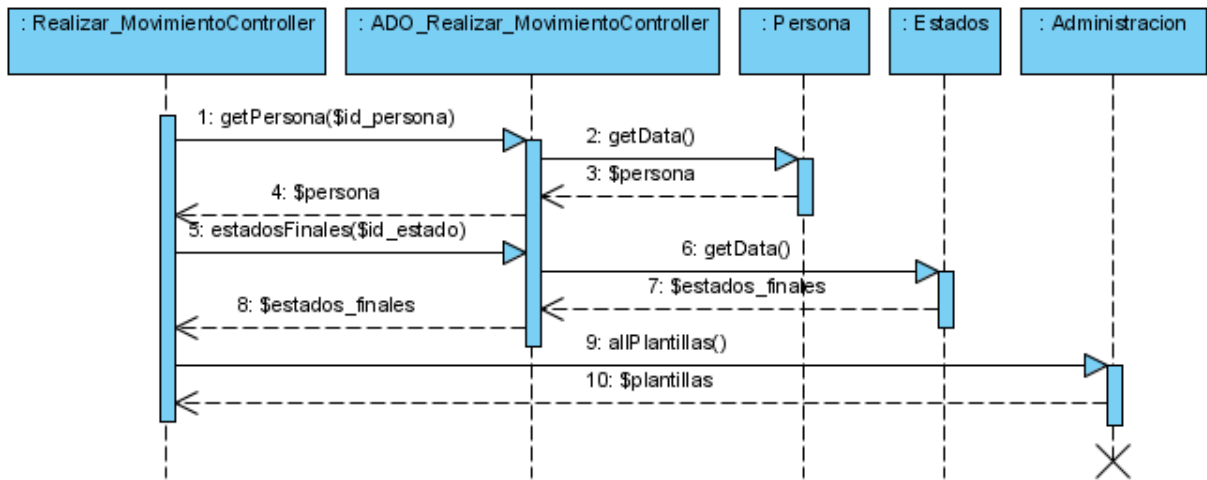


Figura 63: DS CUS _ Realizar Movimiento B

C

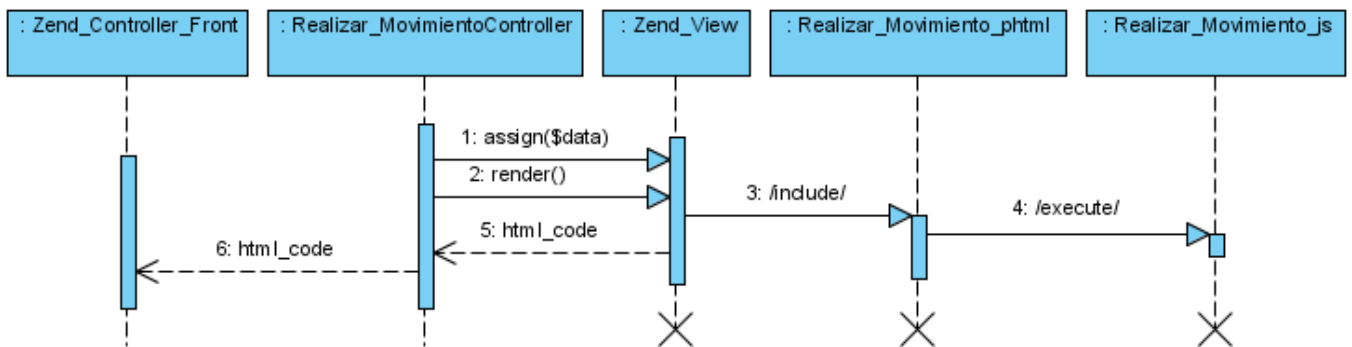


Figura 63: DS CUS _ Realizar Movimiento C

D

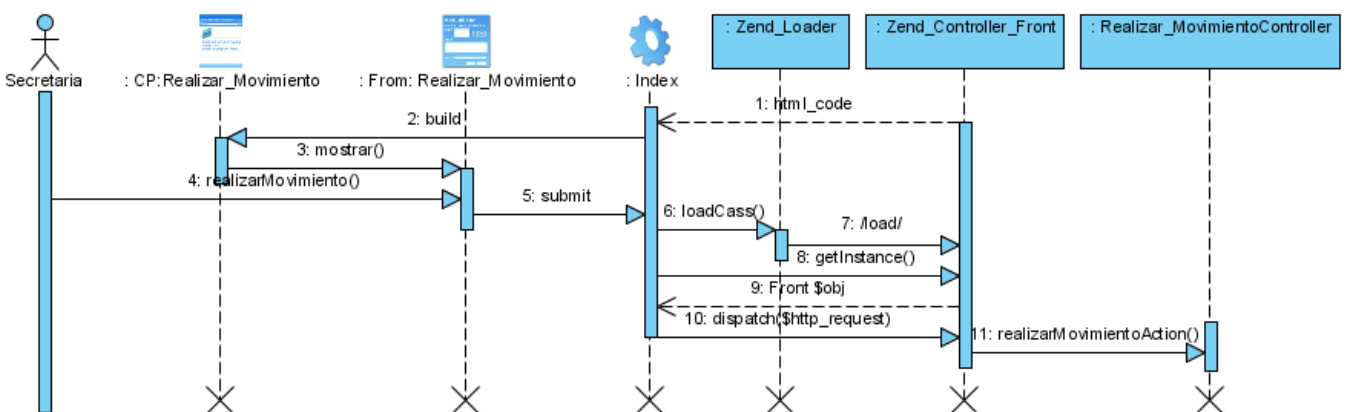


Figura 63: DS CUS _ Realizar Movimiento D

E

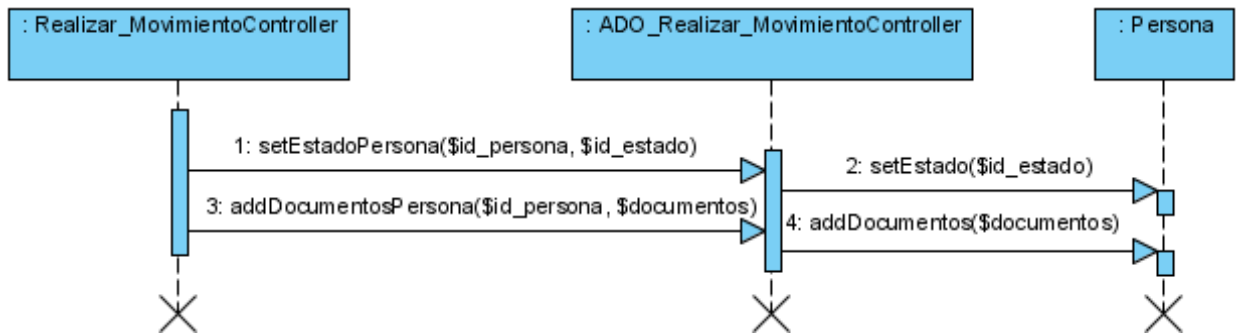


Figura 63: DS CUS _ Realizar Movimiento E

CUS: Ubicar Persona en Estructura

A

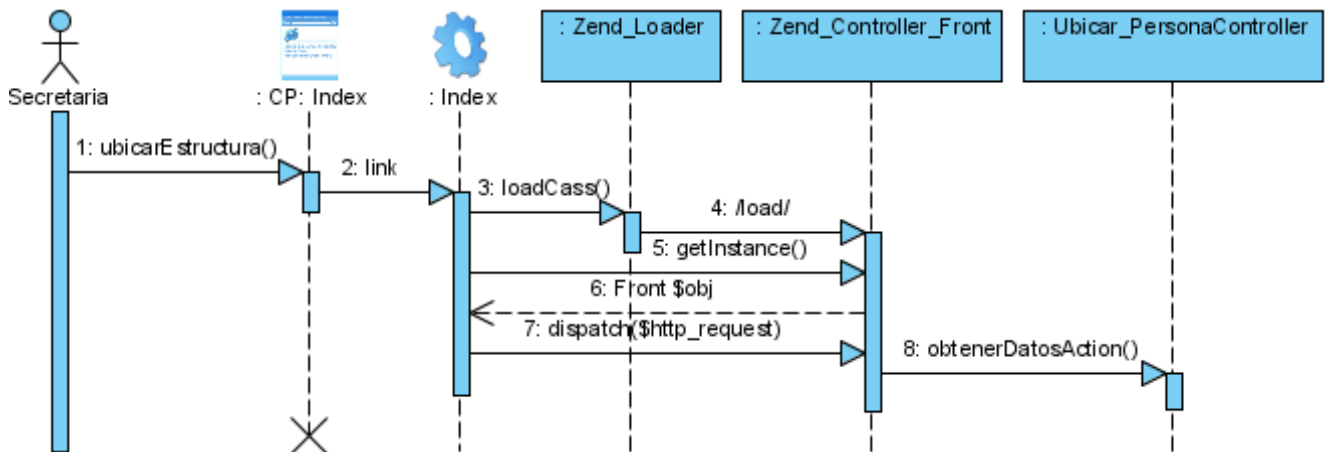


Figura 64: DS CUS _ Ubicar Persona en Estructura A

B

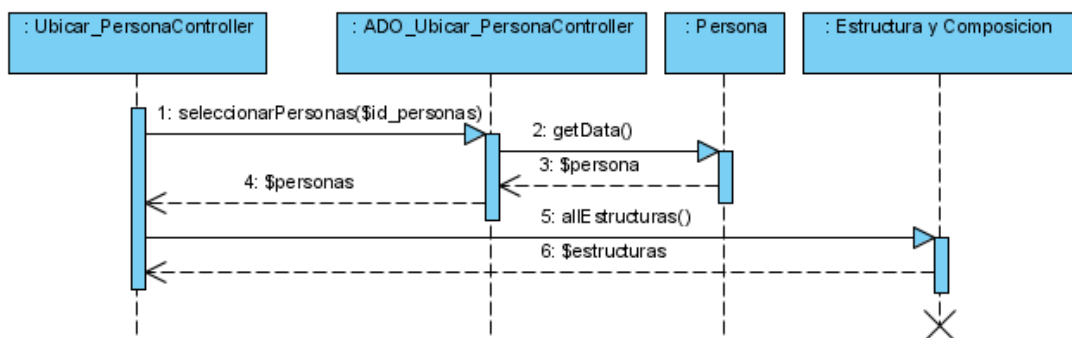


Figura 64: DS CUS _ Ubicar Persona en Estructura B

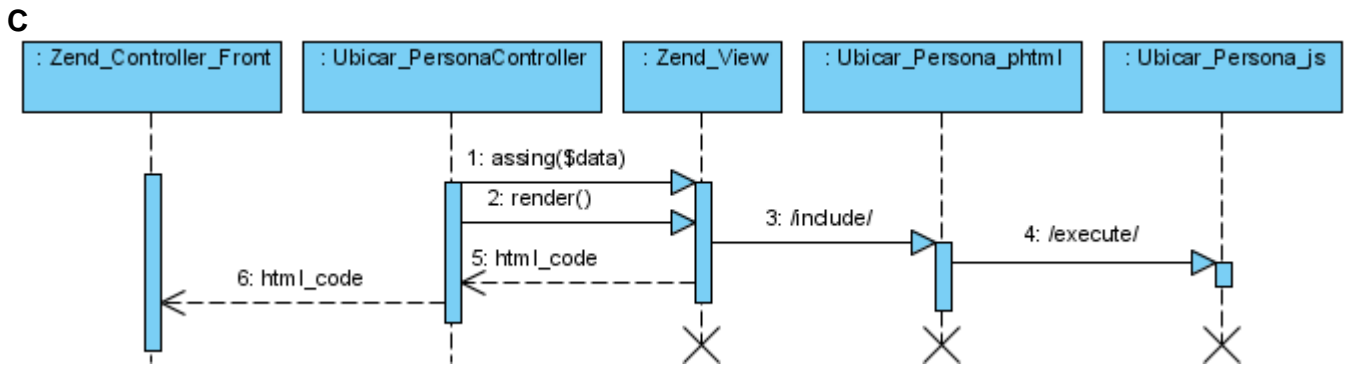


Figura 64: DS CUS_ Ubicar Persona en Estructura C

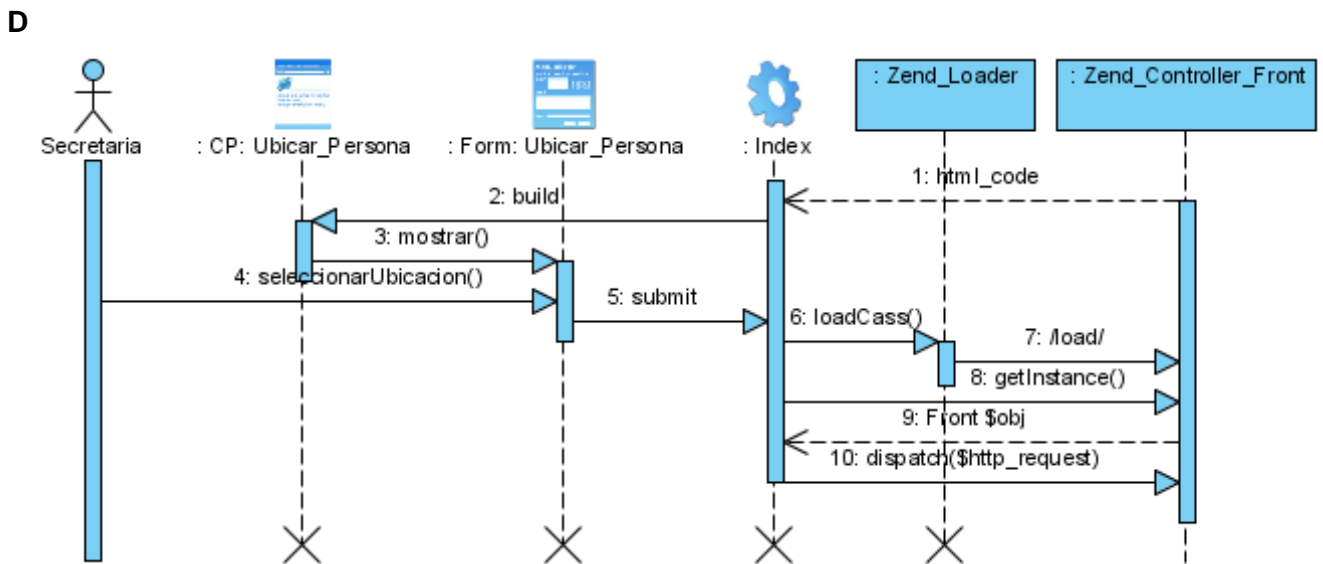


Figura 64: DS CUS_ Ubicar Persona en Estructura D

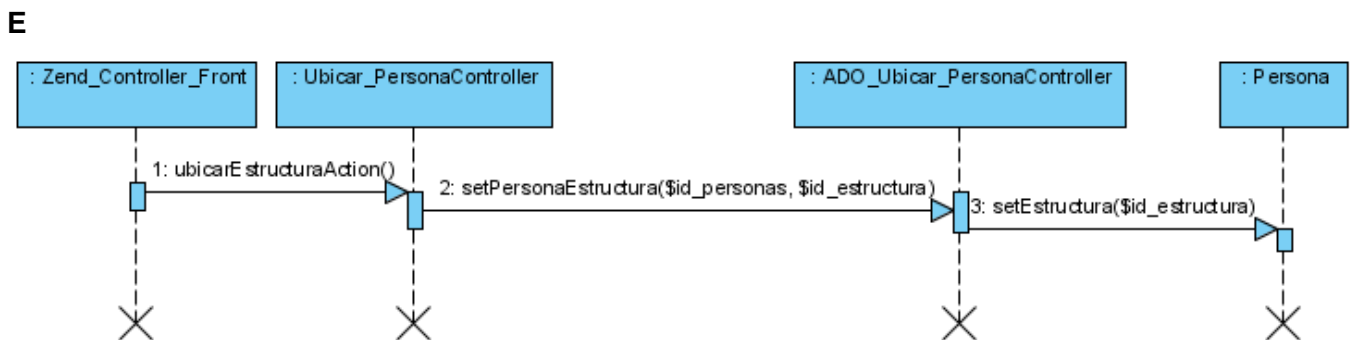


Figura 64: DS CUS_ Ubicar Persona en Estructura E

CUS: Asignar Asignaturas Profesor

A

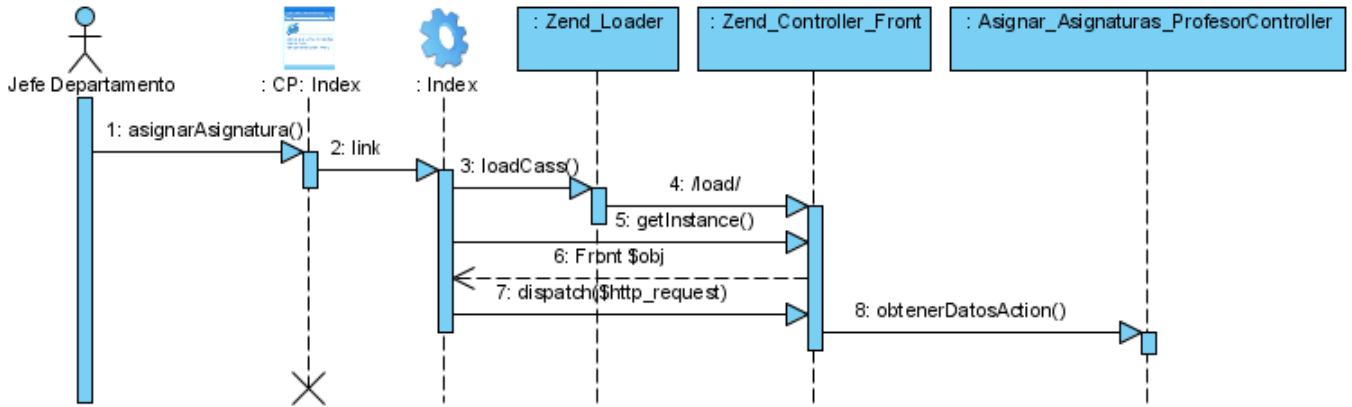


Figura 65: DS CUS _ Asignar Asignaturas Profesor A

B

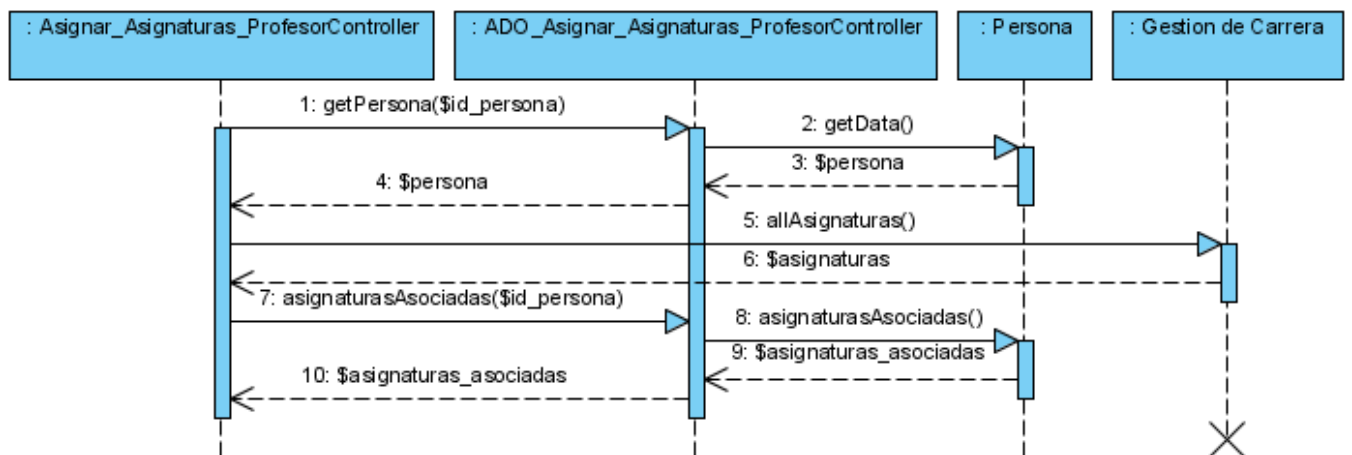


Figura 65: DS CUS _ Asignar Asignaturas Profesor B

C

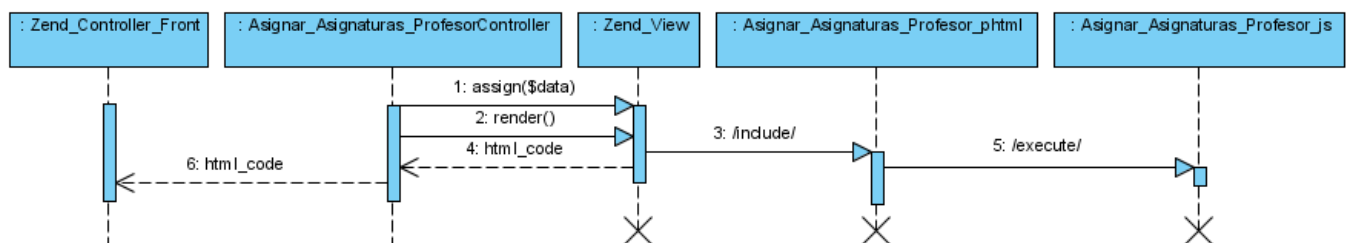


Figura 65: DS CUS _ Asignar Asignaturas Profesor C

D

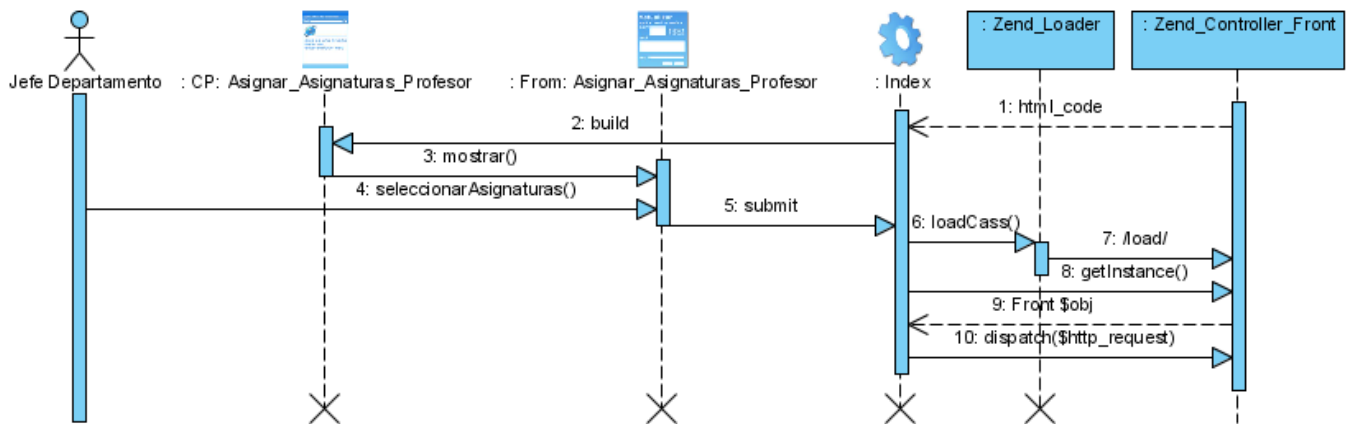


Figura 65: DS CUS_ Asignar Asignaturas Profesor D

E

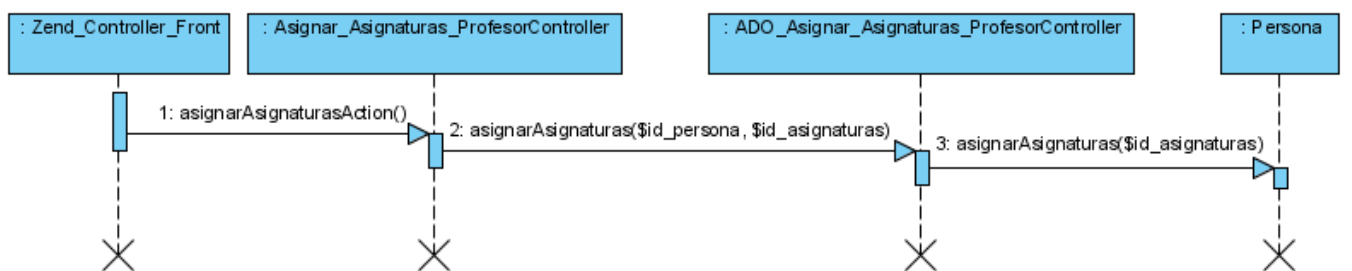


Figura: DS_ Asignar Asignaturas Profesor E

CUS: Asociar Plan de Estudio

A

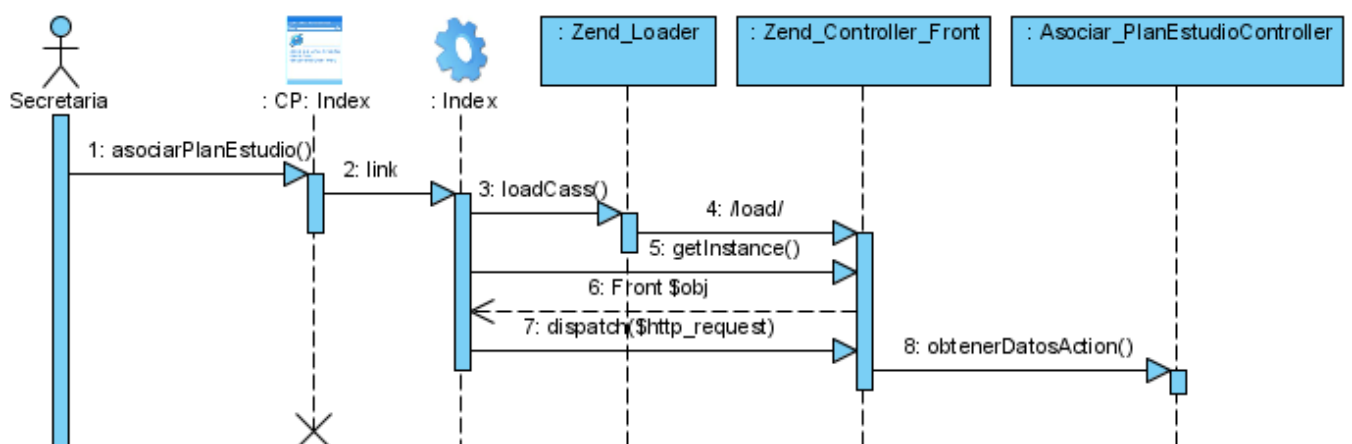


Figura 66: DS CUS_ Asociar Plan de Estudio A

B

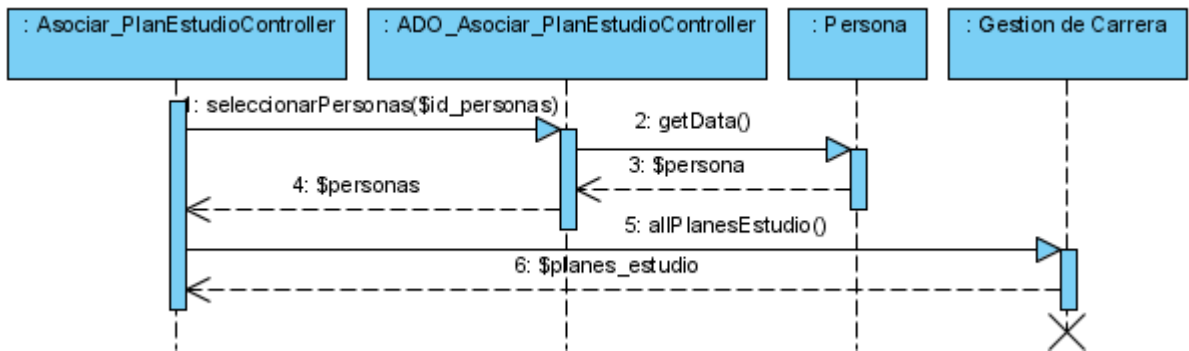


Figura 66: DS CUS _ Asociar Plan de Estudio B

C

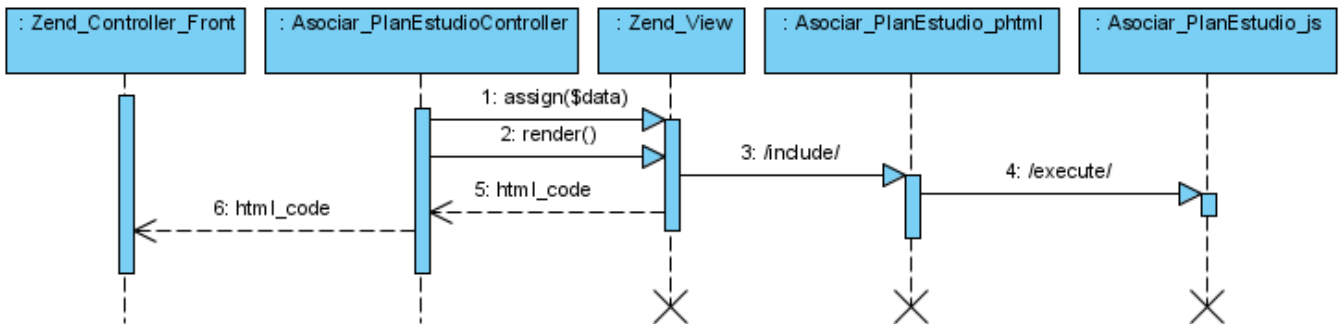


Figura 66: DS CUS _ Asociar Plan de Estudio C

D

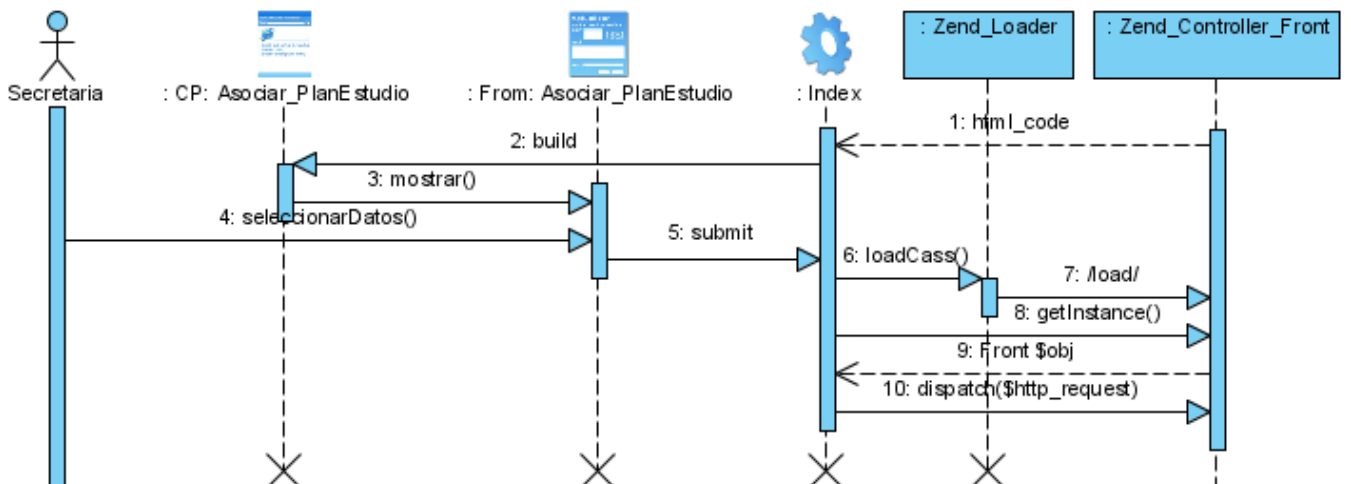


Figura 66: DS CUS _ Asociar Plan de Estudio D

E

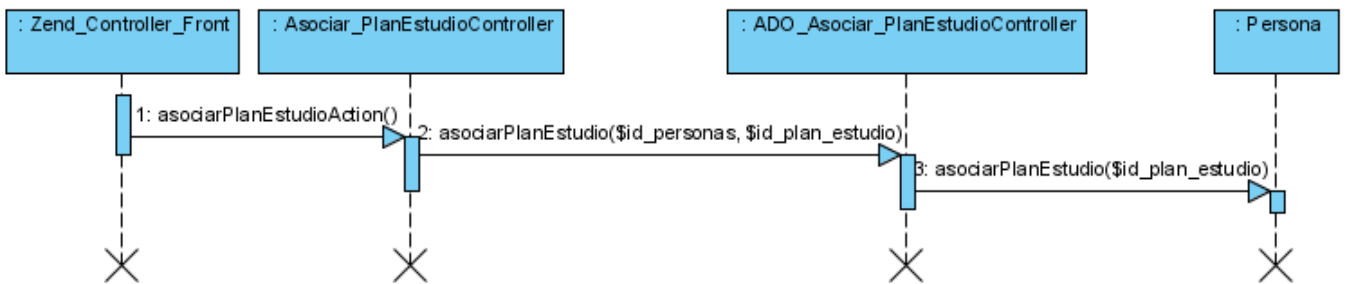


Figura 66: DS CUS _ Asociar Plan de Estudio E

CUS: Consultar Información

A

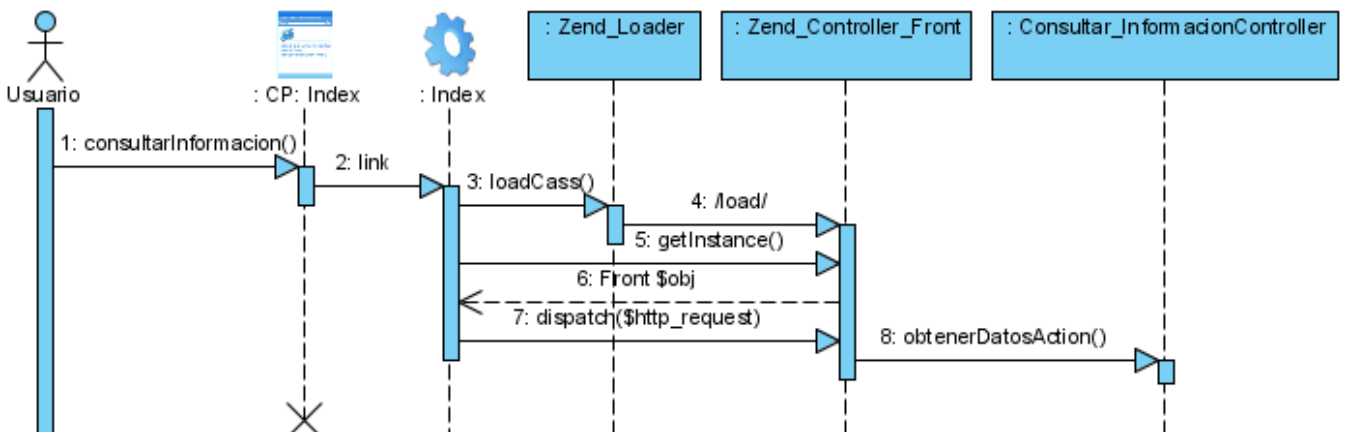


Figura 67: DS CUS _ Consultar Información A

B

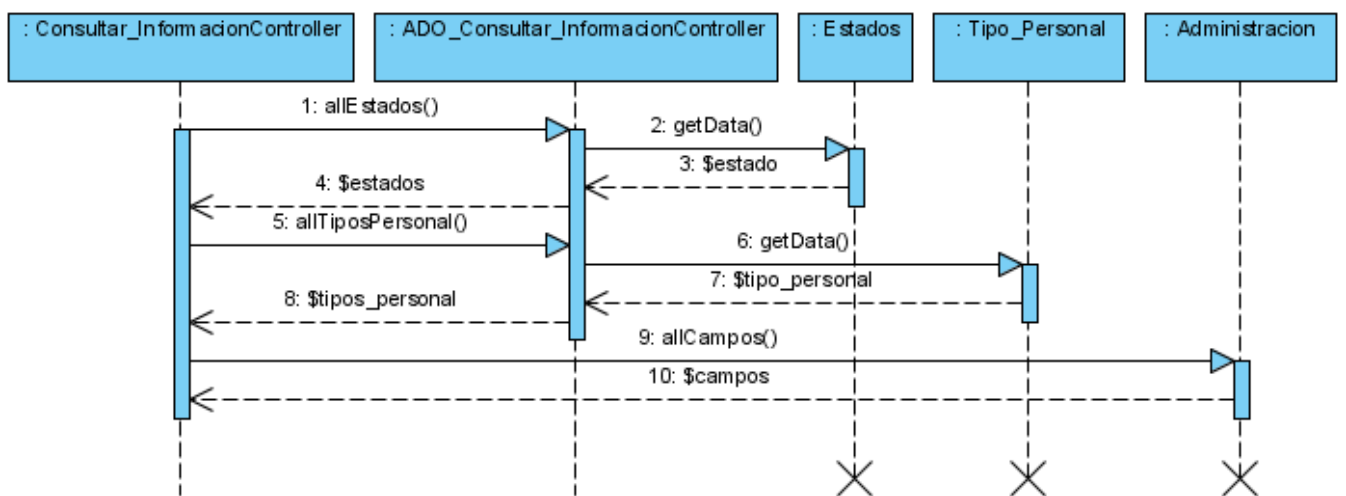


Figura 67: DS CUS _ Consultar Información B

C

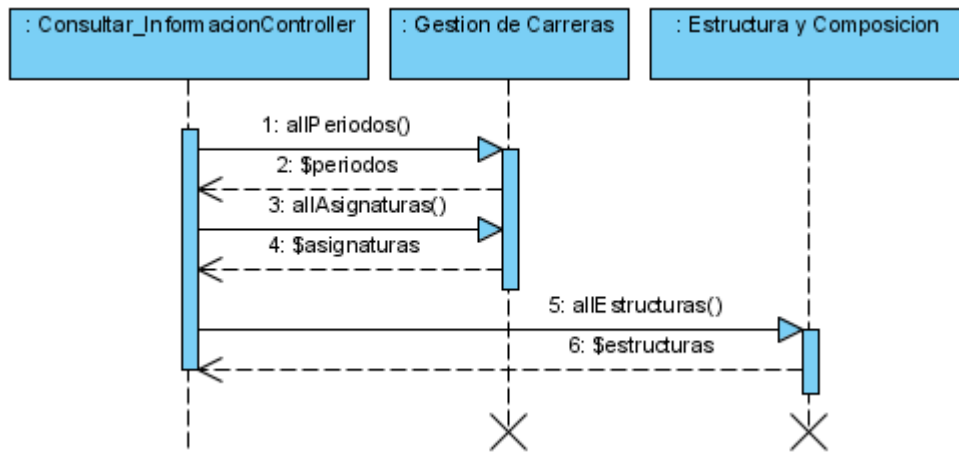


Figura 67: DS CUS_ Consultar Información C

D

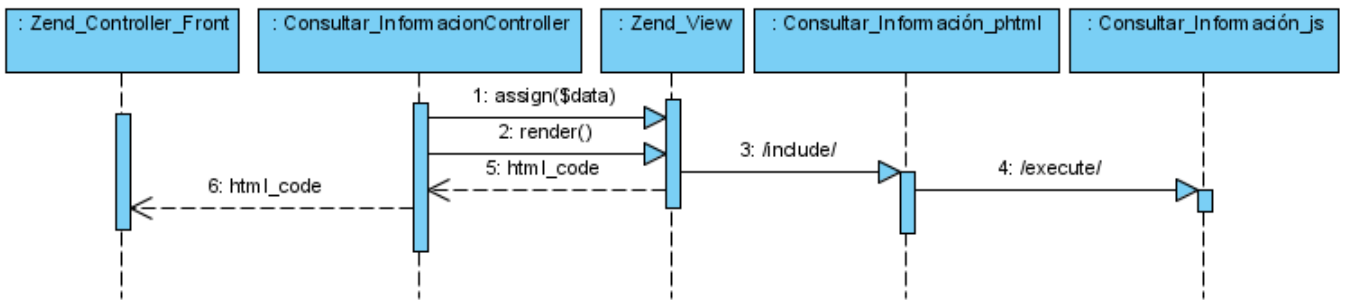


Figura 67: DS CUS_ Consultar Información D

E

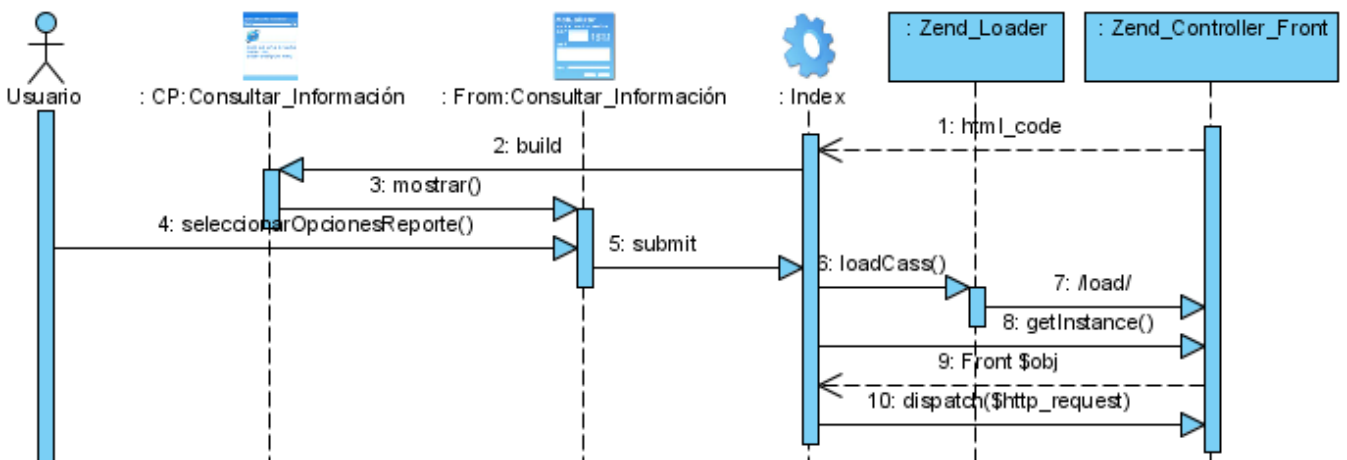


Figura 67: DS CUS_ Consultar Información E

F

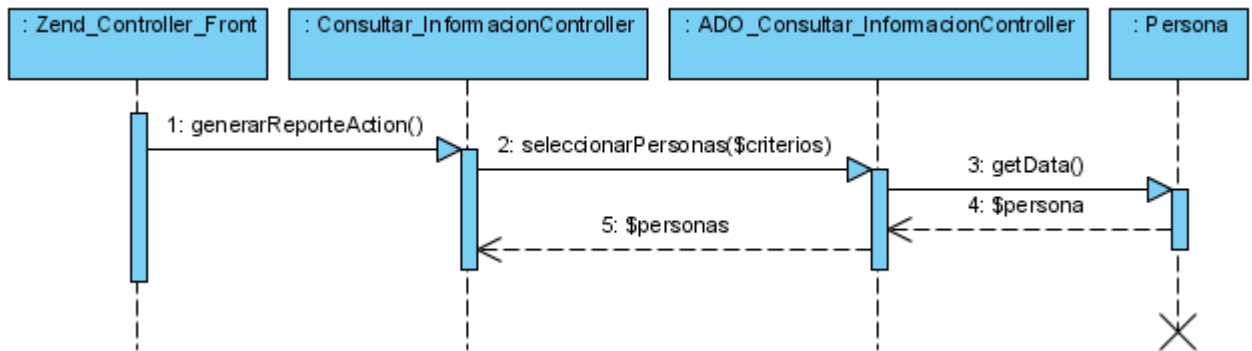


Figura 67: DS CUS_ Consultar Información F

G

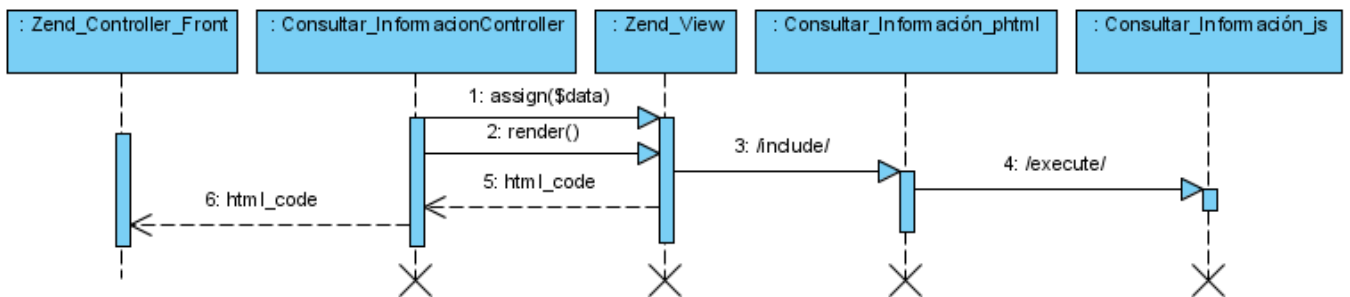


Figura 67: DS CUS_ Consultar Información G

H

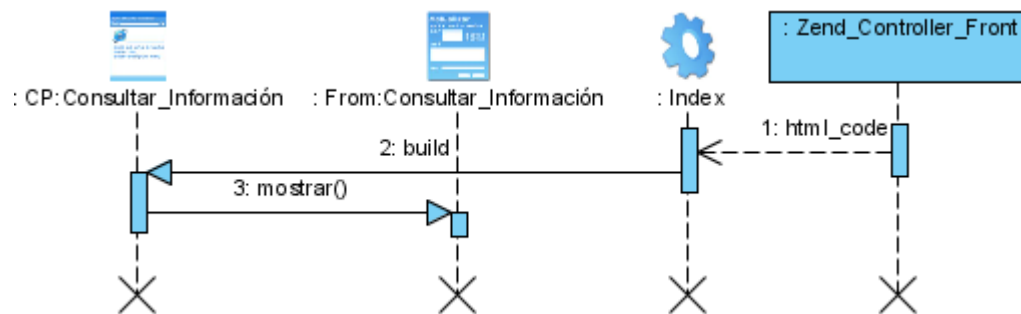


Figura 67: DS CUS_ Consultar Información H

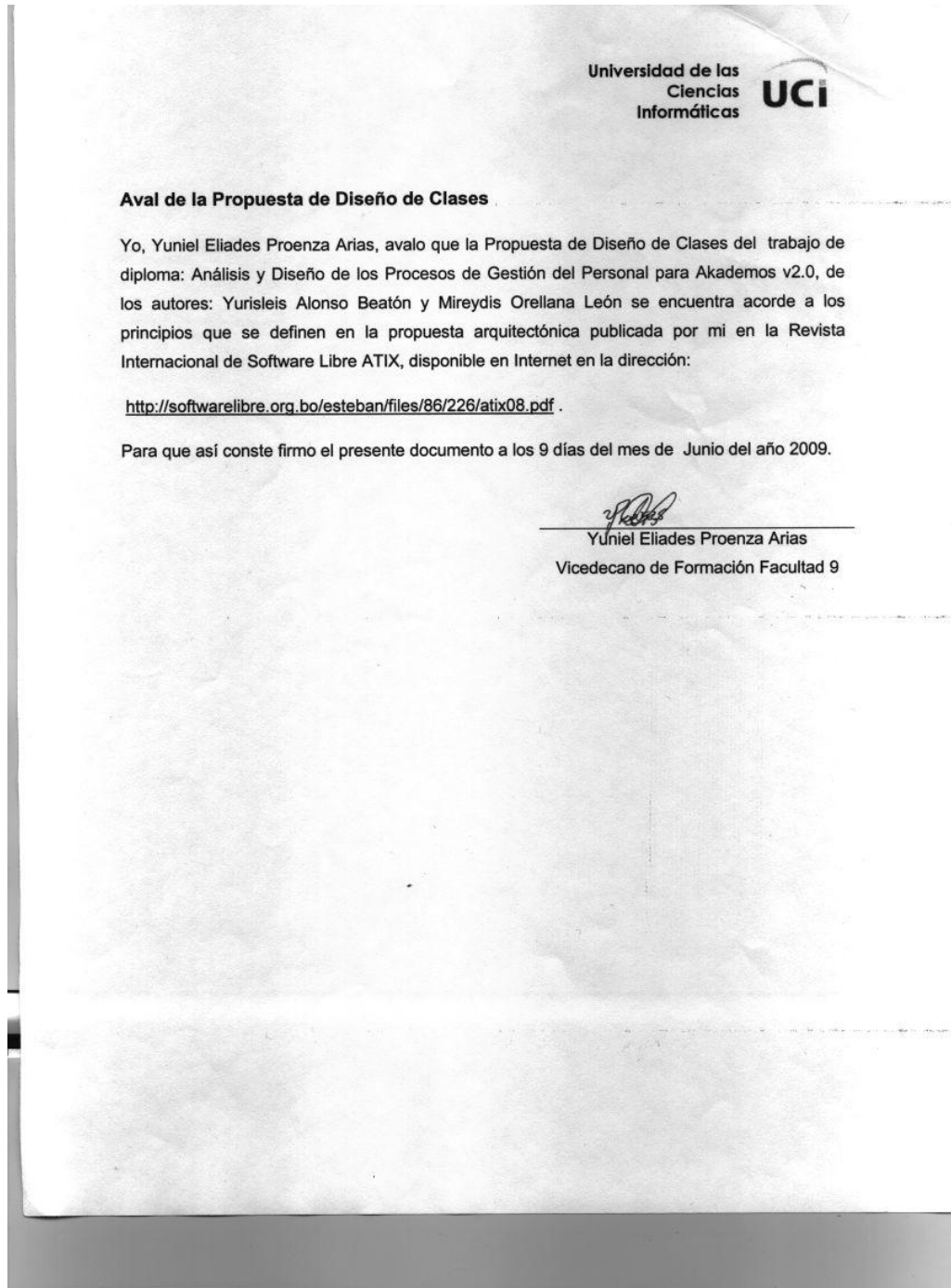
Anexo #5: Avals de la Propuesta de Diseño de Clases y Actas de Aceptación de Requerimientos.

Figura 68: Aval de la Propuesta de Diseño de Clases

Universidad de las
Ciencias Informáticas



Aval de la Propuesta de Diseño de Clases

Nosotros, Yulier Molina Ramírez y Dayron Cruz Ifigo, avalamos que la Propuesta de Diseño de Clases del trabajo de diploma: Análisis y Diseño del los Procesos de Gestión de Personal para Akademos v2.0, de los autores: Yurisleis Alonso Beatón y Mireydis Orellana León se encuentra acorde a los principios que se definen en la propuesta arquitectónica que se utiliza en el proyecto Akademos.

Para que así conste firmamos el presente documento a los 9 días del mes de Junio del año 2009.

Yulier Molina Ramírez

Arquitecto del Proyecto Akademos

Dayron Cruz Ifigo

Arquitecto Principal del Proyecto Akademos

Figura 69: Aval de la Propuesta de Diseño de Clases

ACTA DE ACEPTACIÓN

En la Ciudad de La Habana, a los 9 días del mes de junio del año 2009

De una parte, Facultad 9, representado en este acto por **Surama Borges Zayas**, quien a los fines y efectos derivados del presente documento se denominará como "**El Cliente**".

De otra parte, el **Grupo de Desarrollo del Proyecto Gestión Académica, Akademos**, representado en este acto por **Yanet Poza Bernal**, que a los fines y efectos derivados del presente documento se denominará como "**El Proyecto**".

Primero: Que en cumplimiento del Cronograma han sido efectuadas las actividades que se describen, **Las Partes DECLARAN**:

CONSIDERANDO: Que se han efectuado las actividades siguientes sobre el módulo **Gestión de Personal**:

1. **Especificación de los requerimientos del software**
2. **Modelado del sistema**

CONSIDERANDO: Que las actividades que se han ejecutado cumplen con los requerimientos de **El Cliente**.

CONSIDERANDO: Que **El Proyecto** ha mostrado la documentación que avala la ejecución de este acto a **El Cliente**.

POR TANTO: **Las Partes** acuerdan formalizar mediante la presente **Acta, Aceptadas** las actividades que han sido ejecutadas en esta fecha.

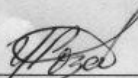
Y para que así conste, se extiende la presente **Acta** en dos ejemplares, rubricados por **Las Partes**.

Por El Cliente



Surama Borges Zayas
Secretaria Docente

Por El Proyecto



Yanet Poza Bernal
Líder del proyecto Akademos

Figura 70: Acta de Aceptación de Requerimientos

ACTA DE ACEPTACIÓN

En la Ciudad de La Habana, a los 9 días del mes de junio del año 2009.

De una parte, Facultad 10, representado en este acto por **Milagros E. Ramírez Muñoz**, quien a los fines y efectos derivados del presente documento se denominará como **"El Cliente"**.

De otra parte, el Grupo de Desarrollo del Proyecto Gestión Académica, Akademos, representado en este acto por **Yanet Poza Bernal**, que a los fines y efectos derivados del presente documento se denominará como **"El Proyecto"**.

Primero: Que en cumplimiento del Cronograma han sido efectuadas las actividades que se describen, **Las Partes DECLARAN:**

CONSIDERANDO: Que se han efectuado las actividades siguientes sobre el módulo **Gestión de Personal:**

1. Especificación de los requerimientos del software
2. Modelado del sistema

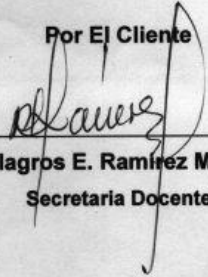
CONSIDERANDO: Que las actividades que se han ejecutado cumplen con los requerimientos de **El Cliente**.

CONSIDERANDO: Que **El Proyecto** ha mostrado la documentación que avala la ejecución de este acto a **El Cliente**.

POR TANTO: **Las Partes** acuerdan formalizar mediante la presente **Acta, Aceptadas** las actividades que han sido ejecutadas en esta fecha.


Y para que así conste, se extiende la presente **Acta** en dos ejemplares, rubricados por **Las Partes**.

Por El Cliente



Milagros E. Ramírez Muñoz
Secretaria Docente

Por El Proyecto



Yanet Poza Bernal
Líder del proyecto Akademos

Figura 71: Acta de Aceptación de Requerimientos

ACTA DE ACEPTACIÓN

En la Ciudad de La Habana, a los 9 días del mes de junio del año 2009

De una parte, Facultad 1, representado en este acto por **Mayelin Alvarez Marichal**, quien a los fines y efectos derivados del presente documento se denominará como "**El Cliente**".

De otra parte, el **Grupo de Desarrollo del Proyecto Gestión Académica, Akademos**, representado en este acto por **Yanet Poza Bernal**, que a los fines y efectos derivados del presente documento se denominará como "**El Proyecto**".

Primero: Que en cumplimiento del Cronograma han sido efectuadas las actividades que se describen, **Las Partes DECLARAN:**

CONSIDERANDO: Que se han efectuado las actividades siguientes sobre el módulo **Gestión de Personal:**

1. **Especificación de los requerimientos del software**
2. **Modelado del sistema**

CONSIDERANDO: Que las actividades que se han ejecutado cumplen con los requerimientos de **El Cliente**.

CONSIDERANDO: Que **El Proyecto** ha mostrado la documentación que avala la ejecución de este acto a **El Cliente**.

POR TANTO: **Las Partes** acuerdan formalizar mediante la presente **Acta, Aceptadas** las actividades que han sido ejecutadas en esta fecha.

Y para que así conste, se extiende la presente **Acta** en dos ejemplares, rubricados por **Las Partes**.

Por El Cliente



Mayelin Alvarez Marichal
Secretaria Docente

Por El Proyecto



Yanet Poza Bernal
Lider del proyecto Akademos

Figura 72: Acta de Aceptación de Requerimientos

GLOSARIO DE TÉRMINOS

A continuación, en orden alfabético, se muestra el significado de algunos términos usados en este documento cuyo uso no es común y que pueden dificultar la comprensión del mismo:

A

- **Ajax:** Acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas. Éstas se ejecutan en el cliente, es decir, en el navegador del usuario, y mantiene comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre la misma página sin necesidad de recargarla.

B

- **Bases de datos:** Es un conjunto integrado de datos junto con una serie de aplicaciones para su manejo, accesibles simultáneamente por diferentes usuarios y programas.
- **Bugs:** Es un error o un defecto en el software o hardware que hace que un programa funcione incorrectamente.

C

- **CU:** Caso de Uso.
- **CUN:** Casos de uso del negocio.
- **CUS:** Casos de uso del sistema.

D

- **DCA:** Diagrama de clases del análisis.
- **DCD:** Diagrama de clases del diseño.
- **DCUN:** Diagrama de casos de uso del Negocio.
- **DCUS:** Diagrama de casos de uso del sistema.

F

- **Framework:** representa una arquitectura de software que modela las relaciones generales de las entidades del dominio. Provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio.

H

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

- **Herramientas CASE:** Herramientas utilizadas para el desarrollo de proyectos de Ingeniería de Software.
- **HTML (HyperText Markup Language):** Lenguaje de Marcado de Hipertexto. Lenguaje en el que se escriben las páginas a las que se accede a través de navegadores WWW. Admite componentes hipertextuales y multimedia.
- **HTTP:** El protocolo de transferencia de hipertexto, usado en cada transacción de la Web (WWW). El hipertexto es el contenido de las páginas web y el protocolo de transferencia es el sistema mediante el cual se envían las peticiones de acceso a una página y la respuesta con el contenido.

I

- **Interfaz:** En informática, una interfaz es la parte del programa informático que permite el flujo de información entre varias aplicaciones o entre el propio programa y el usuario. En software también se habla de interfaz gráfica de usuario, que es un método para facilitar la interacción del usuario con el ordenador o la computadora a través de la utilización de un conjunto de imágenes y objetos pictóricos (iconos, ventanas, formularios, páginas web...).

L

- **Lenguaje:** En informática, cuando se habla de lenguaje se refiere generalmente al de programación, conjunto de instrucciones que las aplicaciones necesitan para que el ordenador ejecute determinadas operaciones. Hay lenguaje de alto y bajo nivel, de tercera y cuarta generación, lenguaje natural y lenguaje máquina.
- **Licencia GPL (GNU Public License):** La Fundación para el Software Libre (FSF - Free Software Foundation) está dedicada a eliminar las restricciones de uso, copia, modificación y distribución del software. Promueve el desarrollo y uso del software libre en todas las áreas de la computación. Específicamente, la Fundación pone a disposición de todo el mundo un completo e integrado sistema de software llamado GNU. La mayor parte de este sistema está ya siendo utilizado y distribuido.
- **Linux:** Linux es una implementación independiente con "espíritu" POSIX (especificación para sistemas operativos). Tiene extensiones System V y BSD, y ha sido escrito completamente a base de aportaciones. Linux no tiene código propietario. Linux está distribuido libremente bajo "GNU Public License". Actualmente solo trabaja en IBM PC (o compatibles) y con arquitecturas ISA e EISA, y requiere un procesador 386 o superior. El kernel de Linux está escrito por Linux Torvalds (Torvalds@kruuna.helsinki.fi), desde Finlandia y otros voluntarios de otras partes del mundo. La mayoría de los programas que ruedan bajo Linux son freeware, y muchos de ellos del

Proyecto GNU. Linux tiene todas las características que se pueden esperar de un moderno y flexible UNIX. Incluye multitarea real, memoria virtual, librerías compartidas, dirección y manejo propio de memoria y TCP/IP. Usa las características hardware de la familia de procesadores 386 para implementar las capacidades anteriores.

M

- **Microsoft:** Compañía que manufactura los sistemas de operación DOS y Windows.
- **MySQL:** Es un sistema de gestión de bases de datos relacional que cuentan con todas las características de un motor de BD comercial: transacciones atómicas, triggers, replicación, llaves foráneas entre otras. Su ingeniosa arquitectura lo hace extremadamente rápido y fácil de personalizar.

O

- **Open Source (Código Abierto):** Open Source es un término que describe partes de la licencia del movimiento por el software libre.
- **ORM- Object-Relational Mapper:** Es una interfaz que traduce la lógica de los objetos a la lógica relacional permitiendo acceder de forma efectiva a la base de datos desde un contexto orientado a objetos.

P

- **Perl:** Practical Extraction and Report Language. Es un lenguaje de programación desarrollado por Larry Wall inspirado en otras herramientas de UNIX como son: sed, grep, awk, c-shell.
- **PHP- Hypertext Pre-processor:** Lenguaje procesador de Scripts servidor. Libre y de código abierto.
- **Poscondición:** Restricción que ha de ser cierta al completarse una operación.
- **PostgreSQL:** es un Sistema de Gestión de Bases de Datos Objeto-Relacionales (ORDBMS) libre.
- **Precondición:** Restricción que ha de ser cierta cuando una operación es invocada.

R

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

S

- **Scripts:** Un conjunto de comandos escritos en un lenguaje interpretado.
- **Servidor:** Sistema que proporciona recursos (por ejemplo, servidores de ficheros, servidores de nombres). En Internet este término se utiliza muy a menudo para designar a aquellos sistemas que proporcionan información a los usuarios de la red.

- **SGBD:** Sistema de Gestores de Bases de Datos. Es el software que permite la utilización y/o la actualización de los datos almacenados en una (o varias) base(s) de datos por uno o varios usuarios desde diferentes puntos de vista y a la vez.
- **Software (componentes lógicos, programas, software):** Programas o elementos lógicos que hacen funcionar un ordenador o una red, o que se ejecutan en ellos, en contraposición con los componentes físicos del ordenador o la red.
- **SQL - Structured Query Language:** Es un lenguaje declarativo de acceso a bases de datos que permite especificar diversos tipos de operaciones sobre las mismas. Aúna características del álgebra y el cálculo relacional permitiendo lanzar consultas con el fin de recuperar información de interés de una base de datos.
- **Software libre:** Es la denominación del software que brinda libertad a los usuarios sobre un producto adquirido y por tanto, una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente.

U

- **UCI:** Universidad de las Ciencias Informáticas.
- **UML:** El Lenguaje Unificado de Modelado (Unified Modeling Language en inglés) prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan. Mientras que ha habido muchas notaciones y métodos usados para el diseño orientado a objetos, ahora los modeladores sólo tienen que aprender una única notación.
- **UNIX:** Sistema operativo portable, flexible, potente, con entorno programable, multiusuario y multitarea, muy difundido.

W

- **World Wide Web:** Telaraña mundial, Red mundial, WWW. Sistema global de la información basado en la tecnología del hipertexto. Soporta todo tipo de información (audio, video, imagen, texto) y se accede de manera fácil por los usuarios a través de los navegadores.

X

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