



“Carlos Rafael Rodríguez Rodríguez”.



Título: *Sistema de Gestión de la Información para el Pago de Estipendio a Estudiantes Universitarios.*

Trabajo de Diploma para optar por el Título de Ingeniería en Informática.

Autor: Julio César Sardiñas Soca.

Tutores: Ing. Yeniersy Domínguez Díaz.

Ing. Kadir Héctor Ortiz.

Ing. Rafael Velásquez Fuster.

Facultad de Informática

Cienfuegos, junio del 2007

DECLARACIÓN DE AUTORÍA

Yo, Julio Cesar Sardiñas Soca declaro que soy el único autor de este trabajo de diploma y autorizo al Departamento de Finanzas, al Departamento de Informática y a las secretarías de cada facultad de la Universidad de Cienfuegos para que hagan el uso que estimen pertinente del mismo.

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

Firma del Autor
Julio Cesar Sardiñas Soca

Los abajo firmantes, certificamos que el presente trabajo ha sido realizado según acuerdo de la dirección de nuestro centro y el mismo cumple los requisitos que debe tener un trabajo de esta envergadura, referido a la temática señalada.

Firma del Tutor
Ing. Yeniersy Domínguez Díaz

Firma del Tutor
Ing. Hector Kadir Ortiz

Firma del Tutor
Ing Rafael Velásquez Fuster

Firma ICT

Firma ViceDecano

A mi Familia.

Agradezco:

- A mi mami y mi papi, todo su amor, sacrificio y preocupación constante.*
- A mi hermanita, su cariño y dedicación.*
- A mi cuquita por su amor.*
- A mi querida novia Lisbet por todo su amor, paciencia y entrega.*
- A toda mi familia, su preocupación y ayuda.*
- A Yeny y Kadir por su apoyo en todo momento*
- A Daimarelys por su tiempo y paciencia conmigo.*
- A Dailin y Danaisy su sincera amistad.*
- A todos los que de alguna forma han contribuido con su ayuda y apoyo, muchas gracias; este logro también es suyo.*

Resumen

El presente trabajo tiene como título: ***Sistema de Gestión de la Información para el Pago de Estipendio a Estudiantes Universitarios (SGIPEEU)***. Cuenta como objetivo general diseñar un sistema de Gestión de la información para el pago del Estipendio que permita perfeccionar la gestión de la información del pago estipendio estudiantil en la Universidad de Cienfuegos. El mismo se desarrolla en la Dirección de Economía en el Departamento de Finanzas de la Universidad de Cienfuegos (UCF).

La UCF ha ido insertándose en esta gran red de información, lo que constituye un reto en cuanto al manejo de la información por parte de cada una de los departamentos, facultades y áreas que la componen. Se ha venido trabajando, desde hace ya varios años, en la automatización de las principales tareas y funciones que en estos tienen lugar, de tal manera que se facilite su conocimiento. Para ello se han aplicado técnicas de investigación basadas en un estudio exploratorio, recopilación de datos a través de entrevistas informales y la formulación de hipótesis.

Se llevaron a cabo las etapas del proceso de desarrollo del software propuesto, según lo especificado en la metodología de Proceso Unificado de Desarrollo, obteniéndose como producto final el **Sistema de Gestión de Información del Pago de Estipendio a Estudiantes Universitarios(SGIPEEU)**. Como resultado se obtendrá un sistema que permita perfeccionar la gestión del pago de estipendio estudiantil basado en una interfaz Web con un ambiente amigable, legible y de fácil navegación ya que está destinado para el control y gestión de información.

El valor potencial de la investigación radica en sus implicaciones prácticas puesto que con la aplicación del sistema de gestión de información obtenido, se logra la agilización y perfeccionamiento de las labores de gestión del Departamento de Finanzas de la Universidad de Cienfuegos, y por tanto, una mayor eficiencia y calidad en sus servicios sustituyendo los procesos manuales ya existentes.

Introducción.	1
Capítulo 1. Fundamentación teórica.	6
1.1 Introducción.	6
1.2 Conceptos asociados al tema de investigación.	6
1.2.1 Estipendio a estudiantes universitarios.	6
1.2.2 Adeudo:	6
1.2.3 Reclamación:	6
1.2.4 Préstamo a estudiantes universitarios.	6
1.2.5 Ayudantía:	6
1.2.6 Nómina:	6
1.2.7 Gestión.	6
1.2.8 Gestión de la información.	7
1.3 Seguridad del sistema.	7
1.3.1 Formas de autenticación.	8
1.3.2 Kerberos.	8
1.3.3 Smart cards.	9
1.3.4 Autentificación de nombre de usuario.	9
1.3.5 Contraseñas.	9
1.4 Objeto de estudio.	9
1.4.1 Descripción general.	9
1.4.2 Ubicación.	10
1.4.3 Situación problemática.	10
1.4.4 Idea a defender.	10
1.4.5 Objetivo general de la investigación.	10
1.4.6 Objetivos específicos.	10
1.4.7 Análisis comparativo de la forma de llevar el proceso en cuestión con la propuesta.	12
1.5 Metodologías utilizadas.	12
1.5.1 Lenguaje de Modelación Unificado (UML).	12
1.5.2 Proceso Unificado de Desarrollo (RUP).	13
1.6 Paradigmas de programación.	14
1.6.1 Programación por capas.	14
1.7 Tecnologías a utilizar.	16
1.7.1 Interfaz de usuario (Capa de Presentación).	16
1.7.2 Lógica del negocio (Capa de la Lógica del Negocio).	17
1.7.2.1 Lenguajes de programación web.	17
1.7.2.2 Servicios Web.	18
1.7.2.3 SOAP.	19
1.7.2.4 XML.	20
1.7.2.5 Servidores de Aplicaciones Web.	22
1.7.3 Capa de acceso a datos.	23
1.8 Conclusiones.	24
Capítulo 2. Análisis y Diseño.	26
2.1 Introducción.	26
2.2 Descripción del modelo de negocio.	26
2.2.1 Descripción de los procesos del negocio.	27
2.3 Reglas del negocio a considerar.	28
2.4 Modelo de casos de uso del negocio.	29
2.4.1 Actores del negocio.	29
2.4.2 Trabajadores del negocio.	30

2.4.3 Casos de uso del negocio.	31
2.4.4 Diagramas de casos de uso del negocio.....	31
2.4.5 Descripción de los casos de uso del negocio.....	32
2.4.6 Diagramas de actividades del negocio.....	38
2.5 Modelo de objetos del negocio.	45
2.6 Descripción del sistema propuesto.	46
2.6.1 Concepción general del sistema.	46
2.6.2 Requerimientos funcionales.	46
2.6.3 Requerimientos no funcionales.	47
2.7 Modelo de casos de uso del sistema.	50
2.7.1 Actores del modelo de sistema.	50
2.7.2 Paquetes y sus relaciones.	50
2.7.3 Diagramas de casos de uso del sistema.....	51
2.7.4 Descripción de los casos de uso del sistema.....	53
2.8 Conclusiones.	65
Capítulo 3. Construcción de la solución propuesta.	66
3.1 Introducción.	66
3.2 Diseño de la base de datos.	66
3.2.1 Modelo lógico de datos.	66
3.2.2 Modelo físico de datos.	66
3.3 Diagrama de implementación.	66
3.4 Principios de Diseño del Sistema.	67
3.4.1 Estándares en la Interfaz de la Aplicación.	67
3.4.2 Tratamiento de Excepciones.....	68
3.4.3 Formatos de Reportes.....	68
3.4.4 Concepción General de la Ayuda.....	68
3.5 Modelo de Clases Web.....	69
3.6 Conclusiones.	70
Capítulo 4. Estudio de Factibilidad.	72
4.1 Introducción.	72
4.2 Planificación por puntos de función.	72
4.3 Determinación de los costos.....	77
4.4 Beneficios tangibles e intangibles.....	80
4.5 Análisis de costos y beneficios.	80
4.6 Conclusiones.	81
Conclusiones	82
Recomendaciones	83
Referencias Bibliográficas.	84
Bibliografía.....	87
Anexo A. Diagramas Clases Web.	89
Anexo B. Prototipos.	118
Anexo C. Diseño de la Base de Datos	131

Índice de tablas

Tabla 1. Comparación entre SGBD.....	23
Tabla 2. Descripción de los Actores del negocio.....	30
Tabla 3. Descripción de los trabajadores del negocio.....	30
Tabla 4. Descripción de los actores del sistema.....	50
Tabla 5. Diagrama de Clases Web. Paquete Módulo Administrativo.....	70
Tabla 6. Diagrama de Clases Web. Paquete Módulo Secretaria Docente.....	71
Tabla 7. Diagrama de Clases Web. Paquete Módulo Especialista en Finanzas.....	71
Tabla 8. Planificación: Entradas externas.....	73
Tabla 9. Planificación: Salidas externas.....	74
Tabla 10. Planificación: Peticiones.....	75
Tabla 11. Planificación: Ficheros internos.....	76
Tabla 12. Planificación: Punto de función.....	77
Tabla 13. Planificación: Miles de instrucciones fuentes.....	77
Tabla 14. Costos: Factores de escalas.....	78
Tabla 15. Costos totales.....	80

Índice de figuras

Figura 1. Diagramas de casos de uso del negocio.....	32
Figura 2. Diagramas de actividades del caso de uso Gestionar Préstamo Estudiantil.....	40
Figura 3. Diagrama de actividad Gestionar Estipendio Estudiantil.....	41
Figura 4. Diagrama de actividad Reclamar el pago de estipendio estudiantil no cobrado..	42
Figura 5. Diagrama de actividad Gestionar dinero para el pago del Estipendio estudiantil..	43
Figura 6. Diagrama de actividad Solicitar autorizo de cobro del estipendio estudiantil....	44
Figura 7. Diagramas de clases del modelo de objetos del negocio.....	45
Figura 8. Diagrama de Casos de Usos por Paquetes.....	51
Figura 9. Diagrama de Casos de uso del Sistema: Paquete Módulo Secretaria de Facultad.....	52
Figura 10. Diagrama de Casos de Uso del Sistema: Paquete Módulo Administrativo.....	52
Figura 11. Diagrama de Casos de uso del Sistema: Paquete Módulo Especialista en Finanzas.....	53
Fig. 12- Diagrama de Implementación.....	68

Introducción.

Las nuevas tecnologías van tomando auge e importancia a escala internacional y dentro de ella, la producción de software. Como es de esperar, Cuba no se ha quedado atrás, es por ello que una de las principales tareas del Gobierno Revolucionario Cubano es desarrollar esta industria, no solamente por los beneficios para el país en el desarrollo del sistema para su uso interno, sino también con el fin de insertarnos en el mercado a nivel mundial por su perspectiva económica.

Las actuales perspectivas de este desarrollo llevan a brindar la información oportuna y rápida, lo que le imprime valor y utilidad. Contar con aplicaciones desarrolladas en computadoras posibilita un mejor y más fácil acceso a la gestión de la información. Asimismo permite tomar la decisión más acertada en cada momento, clave del éxito que utiliza además de los conocimientos y las aptitudes del hombre, la tecnología informática actual.

En Cuba las diferentes entidades han encontrado en estos sistemas la forma de actualizarse al ritmo de los avances científico – técnicos, los que contribuyen al perfeccionamiento del trabajo.

El Ministerio de Educación Superior (MES) no ha estado ajeno a la introducción de la computación para uso y manejo de la información digitalizada. En este sentido, desde hace varios años, las diferentes dependencias e instituciones rinden la información solicitada en soporte digital y lo usan también en el control de la información interna.

Cada centro y departamento establece y organiza su información siguiendo las orientaciones generales del Ministerio, pero no hay una guía que oriente la gestión de la información para el pago del estipendio, por tanto no existe un criterio único al respecto; lo que motiva que cada uno lo organice e implemente a su manera.

La universidad de Cienfuegos “Carlos Rafael Rodríguez”, ha ido insertándose en esta gran red de información. Esto constituye un gran reto en cuanto a su manejo por parte de cada una de los departamentos, facultades y áreas que la componen. Se ha venido trabajando, desde hace ya varios años, en la automatización de las principales tareas y funciones que en

esto tienen lugar, de tal manera que se facilite el conocimiento. Esto posibilita que muchas de las labores que antes las personas realizaban manualmente, hayan sido sustituidas o apoyadas por aplicaciones.

En general los diferentes departamentos de la UCF han automatizado su información, pero el Departamento de Finanzas está presentando dificultades en la gestión de la información para el pago del estipendio estudiantil debido al trabajo manual que tienen que llevar a cabo los especialistas e integrantes para lograr la gestión y satisfacer de forma inmediata y eficiente el pago del estipendio estudiantil a los estudiantes universitarios cada mes.

Por otra parte, la necesidad de tener una base de datos para poder mantener la información actualizada con los datos de los estudiantes es una realidad imperante en el departamento. Para mantener la vigencia de estas necesidades, se necesita disponer de medios que no creen problemas derivados del tiempo, que permitan el desempeño pleno del departamento, y enriquezcan además el propio sistema de gestión de la información de pago de estudiantes en la UCF.

Muchos han sido los esfuerzos por perfeccionar esta labor aunque no se hayan materializado pues aún esos datos son controlados de forma manual. De ahí la necesidad de buscar una eficaz solución.

El presente trabajo trata de dar respuesta a esa necesidad, proponiendo una aplicación de gestión de la información que permita garantizar la calidad y rapidez en la obtención de dicha información, lo que se traduce en el mejoramiento del trabajo humano.

Se desarrolla bajo la concepción de un sistema que sustituya los procesos manuales existentes, aspectos que permiten una mayor consistencia y seguridad de la información almacenada, así como facilita el manejo y acceso a la misma de forma rápida.

La misión fundamental de la dirección de economía de la UCF consiste en brindar un servicio de asesoramiento a toda la gestión económica y financiera de la universidad, para garantizar la toma de decisiones de cada área a partir del registro y control eficiente de los hechos

económicos para ello tiene entre sus objetivos mostrar los resultados del departamento de finanzas, y además cumplir con el control de los parámetros que le evalúa el MES para lo cual debe garantizar:

1. Actualización de las operaciones bancarias garantizando las conciliaciones de las cuentas.
2. Agilidad en las respuesta y servicio a los clientes en cada puesto de trabajo, con énfasis en las operaciones de Caja, Emisión de Cheque, Nóminas y Divisa.
3. Cumplimiento en la entrega de la información contable y financiera en tiempo y con calidad.
4. Integración a la Facultad de Ciencias Económicas y Empresariales desarrollando una estrategia que incluya actividad docente, investigación y superación.
5. Integración de la Facultad de Informática en el desarrollo e implementación de Software para la gestión de la información no solo del Departamento de Finanzas sino de cualquier otro departamentos

Para lograr una solución acertada y consistente siempre resulta aconsejable llevar a cabo una investigación en las áreas del conocimiento relacionadas con el tema realizándose un objeto de estudio de la problemática existente para gestión de la información del estipendio a estudiantes universitarios por parte del Departamento de Finanzas en la UCF, estructurando un sistema teórico-conceptual que garantice la calidad del proceso de automatización.

Teniendo conocimiento de todo lo expuesto anteriormente se define como ***problema científico*** el siguiente:

La carencia de un mecanismo que gestione la información del pago del estipendio estudiantil universitario en el Departamento de Finanzas de la Universidad de Cienfuegos, función que se ve amenazada por el trabajo manual y tedioso en el procesamiento de la documentación para todo el proceso de confección de las nóminas y pago del estipendio estudiantil.

Se plantea como **Idea a defender:**

Si se diseña y aplica un sistema de gestión de la información del estipendio a estudiantes universitarios se perfecciona la gestión de la información para el pago del estipendio estudiantil.

Como **objetivo general** del presente trabajo se define:

Desarrollar un sistema informático para gestionar la información del Estipendio a estudiantes universitarios de la Universidad de Cienfuegos que permita la obtención inmediata y eficiente de los datos, así como que se realice en el mínimo tiempo y esfuerzo liberándolos de la mayoría de los trabajos monótonos y rutinarios.

De este objetivo general se desprenden los siguientes **objetivos específicos:**

1. Desarrollar una base de datos capaz de organizar y almacenar de manera eficiente la información que se manipula en el Departamento de Finanzas.
2. Desarrollar un módulo a la Especialista en Finanzas que le permita obtener por facultades toda la información de sus estudiantes de una forma mas rápida organizada y eficiente y que además, le permita la confección de las nóminas para el pago del estipendio estudiantil de forma automatiza por facultades, carreras y grupos de acuerdo con la información facilitada por cada secretaria de facultad al inicio de mes.
3. Desarrollar un módulo que permita a cada Secretaria de facultad gestionar la información desde las facultades a fin de reducir el tiempo y e incrementar la gestión de la información para el pago del estipendio, contribuyendo de esta manera a la calidad en sus servicios.
4. Confeccionar la ayuda del sistema a fin de facilitar su uso.
5. Determinar la factibilidad económica del *software* implementado.

Para cumplir estos objetivos se realizaron las siguientes **tareas científicas:**

1. Lograr representar todos los grupos de interés involucrados con el producto. Estos grupos de interés son:
 - Especialista en Finanzas

- personal que de una forma u otra intervienen en el proceso del pago de estipendio: secretarias de las facultades, cajero, administradores de las facultades.
 - autor y tutores de la investigación.
2. Estudiar cómo se gestiona la información del pago de estipendio de los estudiantes universitarios de la Universidad de Cienfuegos.
 3. Analizar todas las deficiencias que presenta el departamento de finanzas que dificultan el desempeño de la gestión de la información para el pago del estipendio estudiantil.
 4. Estructurar un sistema teórico-conceptual que garantice la calidad del proceso de automatización del pago del estipendio que permita a la especialista de Finanzas el perfeccionamiento de la gestión.
 5. Valorar y definir los procesos que pueden ser automatizados, y así obtener los requisitos.
 6. Plantear posibles soluciones que permitan el perfeccionamiento de la gestión de la información para el pago del estipendio estudiantil.
 7. Seleccionar las tendencias y tecnologías actuales posibles a emplear para dar solución al problema, teniendo en cuenta además las particularidades del área.
 8. Implantar el sistema en el Departamento de Finanzas de la Universidad de Cienfuegos y comprobar que da solución a los problemas planteados.

Los aportes prácticos de la investigación están dados en sus implicaciones prácticas puesto que con la aplicación del sistema de gestión de información obtenido, se logra la agilización y perfeccionamiento de las labores de gestión del Departamento de Finanzas de la Universidad de Cienfuegos, y por tanto, una mayor eficiencia y calidad en sus servicios sustituyendo los procesos manuales ya existentes.

Por lo anterior, se puede afirmar que esta investigación aportará datos útiles a las instituciones de educación superior si se hiciera extensivo el sistema y a todas las organizaciones vinculadas al quehacer educativo de nivel superior, en particular al departamento de Finanzas.

El presente documento está estructurado en cinco capítulos.

Capítulo 1: En este capítulo se aborda la fundamentación teórica del tema y los conceptos asociados al dominio del problema. Se explica en detalles el problema a resolver y se describe el campo de acción donde se desarrolla. Así como las tecnologías y metodologías utilizadas para su desarrollo teniendo en cuenta las tendencias actuales.

Capítulo 2: Este capítulo se centra fundamentalmente en analizar el modelo de negocio, así como la descripción de dicho proceso utilizando los artefactos de UML. También se describe la solución propuesta, utilizando los diagramas y modelos de casos de uso utilizados en el sistema a construir, con su correspondiente descripción, así como los requisitos funcionales y no funcionales, los casos de uso y el diagrama de casos de uso del sistema.

Capítulo 3: Se describe el diseño de la solución propuesta con sus correspondientes modelos y/o diagramas: modelo de clases, modelo de clases persistentes, modelo de implementación.

Capítulo 4: Se describe el estudio de factibilidad económica realizado para el sistema, en el que se determina si es factible o no el desarrollo del *software* propuesto.

Capítulo 1. Fundamentación teórica.

1.1 Introducción.

En los últimos años, la especialista del Departamento de Finanzas de la Universidad de Cienfuegos ha experimentado grandes esfuerzos humanos a la hora de actualizar la información referente al pago de estipendio de los estudiantes universitarios, empleando las técnicas que les sirven de soporte. Con el paso del tiempo es cada vez mayor el avance tecnológico, y no debe quedar atrás la posibilidad de crear nuevas formas de desarrollar esta tarea de una forma más rápida y confiable.

Por otra parte, la necesidad de mantener actualizada la Base de Datos de dichos estudiantes es una realidad imperante en el departamento. Sin embargo, para mantener la vigencia de estas necesidades, se necesita disponer de medios que no creen problemas derivados del tiempo, que permitan el desempeño pleno del departamento, y enriquezcan además el propio sistema de pago de estipendio.

1.2 Conceptos asociados al tema de investigación.

1.2.1 Estipendio a estudiantes universitarios.

“**Artículo- 309:** Comprende los gastos destinados a los pagos de estipendio a los estudiantes comprendidos en las disposiciones legales vigentes establecidas por el Ministerio de Educación Superior (MES)”. [1]

1.2.2 Adeudo: “acción y efecto de contraer deudo”. [2]

1.2.3 Reclamación: “acción y efecto de reclamar, oponerse o contradecir a una cosa como injusta, o mostrando no consentir”. [3]

1.2.4 Préstamo a estudiantes universitarios.

“**Artículo- 399:** Comprende los gastos destinados a los préstamos que se le otorguen a los estudiantes comprendidos en las disposiciones legales vigentes establecidas por el Ministerio de Educación Superior (MES)”. [1]

1.2.5 Ayudantía: “prestar cooperación, socorrer, hacer un esfuerzo, poner los medios para el logro de una actividad” .[4]

1.2.6 Nómina: “Es una lista o catálogo de cosas que en las oficinas han de percibir Haberes, estos haberes pueden ser las nóminas”. [5]

1.2.7 Gestión.

Gestión: es la acción y efecto de gestionar. Acción y efecto de administrar. La gestión es el gobierno de una empresa durante el período de actividad. Este gobierno comprende la adquisición y transformación de bienes y su transmisión o empleo para la consecución de los fines de la empresa, y el cumplimiento de esta función principal comprende otras secundarias en número variable según la clase de empresa de que se trate, pero que puedan reducirse de modo general a las siguientes: financiera, comercial, técnica, contable, de seguridad y administrativa. En sentido general y amplio equivale a toda diligencia realizada para la consecución de un fin. [6]

1.2.8 Gestión de la información.

La **gestión de la información** se puede definir como el conjunto de actividades realizadas con el fin de controlar, almacenar y, posteriormente, recuperar adecuadamente la información producida, recibida o retenida por cualquier organización en el desarrollo de sus actividades.[7]

1.3 Seguridad del sistema.

En el desarrollo de este documento, la seguridad se refiere a las políticas, acciones, y sistemas necesarios para proteger la integridad de la información de los sistemas de comunicación. Varios niveles de seguridad son necesarios para cumplir estos objetivos. Generalmente, los requisitos de seguridad de la información están agrupados en las siguientes categorías:

Integridad de la información: La garantía de que los datos no han sido alterados ni interceptados.

Confidencialidad: La garantía de que sólo las personas a las que van dirigidas los datos acceden a éstos.

Autenticación: La garantía de que el usuario o el grupo de trabajo que pide acceder a otro usuario, o grupo de trabajo, recurso o servicio es realmente ese usuario o grupo de trabajo. Además, la garantía de que la información descrita y asociada con el autor, o administrador, de un objeto digital no sea desconocida.

Verificación: Comprobar que los mecanismos de seguridad son sólidos, potentes y que están correctamente implementados.

Disponibilidad: Garantizar que los recursos estén disponibles cuando se necesiten.

Seguridad de la información: La confidencialidad e integridad de la información ha de basarse en un estricto control de los accesos. Esto es posible mediante privilegios del sistema y de objetos. Los privilegios pueden ser encapsulados en roles.

Diversos controles de acceso contribuyen a establecer el privilegio mínimo, es decir, que el usuario tenga únicamente aquel privilegio que necesita para hacer su trabajo. La integridad de los datos se garantiza a través de mecanismos de consistencia de datos: para ejecutar una transacción hay que confirmar una serie de datos.

Seguridad en los servidores: Las empresas operan en Internet y almacenan datos de sus clientes en una base de datos que reside detrás de un cortafuego (*firewall*) dentro de una Intranet. El reto consiste en poder acceder a la información situada en la intranet, protegida por el cortafuego, salvaguardando al mismo tiempo la confidencialidad e integridad de los datos.

Si la seguridad en la base de datos es un requisito previo y necesario para la seguridad en Internet, asegurar los datos frente a miradas acechantas mientras viajan a través de la red es también muy importante. Es posible cifrar mediante técnicas criptográficas la comunicación entre navegadores y servidores Web en Internet, o en una Intranet, utilizando el nivel SSL 3.0 (Secure Sockets Layer 3.0) y el sistema de encriptación SET (Secure Electronic Transaction).

1.3.1 Formas de autenticación.

Un sistema sin autenticación no es seguro. Como se había dicho anteriormente la autenticación no es más que el proceso de suministrarle al sistema la identidad del usuario, por lo que existen diversas formas de hacerlo. A continuación se enuncian algunas de ellas.

1.3.2 Kerberos.

Implementado por primera vez en Unix, el nombre proviene de la mitología, como una traducción del nombre con el que era denominado el perro de tres cabezas que cuidaba la puerta del reino de Hades. El usuario le brinda al sistema su cuenta y contraseña, los cuales son validados en la estación de trabajo. Luego dicha estación solicita al servidor Kerberos un identificador que es válido sólo para situación y tiempo determinados. Este identificador es conocido como “identificador de sesión de trabajo”. Es muy ventajoso ya que nunca se envía el nombre y contraseña por la red, sino que envía el identificador.

1.3.3 Smart cards.

Smart cards, Smart keys o lo que es considerado como **sistemas de desafío y respuesta** son un método similar al Kerberos pero las autenticaciones se realizan en la estación de trabajo.

La diferencia consiste en que este sistema no necesita de un servidor para crear el identificador que viaja por la red.

1.3.4 Autenticación de nombre de usuario.

Una forma un poco más segura es la verificación del nombre de usuario. Esta consiste en teclear el nombre del usuario y si está en la lista de usuarios válidos, entonces tendrá acceso al sistema.

1.3.5 Contraseñas.

Las contraseñas son un modelo muy utilizado de autenticación, generalmente vienen acompañados del nombre del usuario y se chequean en el servidor.

Este método tiene que ser especialmente utilizado ya que existen algunas normas específicas para la generación de contraseñas, que evitan el descubrimiento casual o dificultan el intencional.

Concluyendo este punto se puede decir que existen muchas y variadas formas de protección, cada una de ellas posee sus ventajas y desventajas, es entonces una responsabilidad del usuario escoger cual usará.

1.4 Objeto de estudio.

1.4.1 Descripción general.

La elaboración de este *software* lleva previamente una selección y un análisis de lo que se quiere obtener. Como resultado de ese análisis se obtienen las salidas por la vía más rápida posible de la información necesaria para el pago de estipendio, evitando de esta manera que

la especialista en Finanzas realice un gran esfuerzo manual. El sistema debe tener todo lo que necesitan los usuarios y una seguridad capaz de proteger los datos contenidos en la Base de Datos puesto que no toda persona debe tener la posibilidad de acceder, modificar y conocerlos.

1.4.2 Ubicación.

El proceso de investigación para el desarrollo del presente trabajo se enmarca en la Universidad de Cienfuegos “Carlos Rafael Rodríguez Rodríguez”, dentro de la misma en la Dirección de Economía, específicamente en el departamento de Finanzas.

1.4.3 Situación problemática.

El pago del estipendio a los estudiantes universitarios es uno de los procesos que se llevan a cabo en el Departamento de Finanzas función que se ve amenazada por el trabajo manual y tedioso en el procesamiento de la documentación para todo el proceso de confección de las nóminas de pago del estipendio estudiantil.

1.4.4 Idea a defender.

Si se diseña y aplica un sistema de gestión de la información del estipendio a estudiantes universitarios se perfecciona la gestión de la información para el pago del estipendio estudiantil.

1.4.5 Objetivo general de la investigación.

Automatización de un Sistema Informático para Gestionar la información del Estipendio a Estudiantes Universitarios de la Universidad de Cienfuegos que permita la obtención inmediata y eficiente de los datos, así como que se realice en el mínimo tiempo y esfuerzo liberándolos de la mayoría de los trabajos monótonos y rutinarios.

1.4.6 Objetivos específicos.

- Desarrollar una base de datos capaz de organizar y almacenar de manera eficiente la información que se manipula en el departamento de finanzas.
- Desarrollar un módulo a la Especialista en finanzas que le permita obtener por facultades toda la información de sus estudiantes de una forma mas rápida organizada y eficiente y que además, le permita la confección de las nóminas para el pago del estipendio estudiantil de forma automatiza por facultades, carreras y grupos de acuerdo con la información facilitada por cada secretaria de facultad al inicio de mes.

- Desarrollar un módulo que permita a cada Secretaria de facultad gestionar la información desde las facultades a fin de reducir el tiempo y e incrementar la gestión de la información para el pago del estipendio.
- Confeccionar la ayuda del sistema a fin de facilitar su uso.
- Se valida el funcionamiento del software.
- Determinar la factibilidad económica del *software* implementado.
- Implantar el sistema en el Departamento de finanzas de la Universidad de Cienfuegos y comprobar que da solución a los problemas planteados.

1.4.7 Análisis comparativo de la forma de llevar el proceso en cuestión con la propuesta.

Hasta los momentos actuales hay conocimiento de algunos sistemas informáticos que estén vinculados al tema de la investigación, estos son pocos flexibles y no hay ninguno que resuelva la problemática planteada.

Actualmente el Departamento de Finanzas cuenta con una especialista que atiende todo lo relacionado con la actividad del pago del estipendio a estudiantes universitarios, entre sus funciones se encuentra la de procesar y conformar las nóminas de pago, para lo cual se nutre de información enviada cada mes por las secretarías de la facultad. Esta información abarca cierto volumen de información de los estudiantes universitarios que debe ser procesada en un corto período de tiempo, por lo que la especialista tiene que ponerse en función del procesamiento de la misma y afectar el ritmo de sus actividades para evitar un incumplimiento en la tarea. Hoy el procesamiento de los datos se lleva a cabo por una entrada de datos mensual de los estudiantes con la correspondiente actualización en el monto a ser pagado, este método está sujeto a varias rectificaciones de la información debido a que la variabilidad recae en la especialista directamente lo que se convierte en un trabajo tedioso y agotador.

El Software nombrado “*Sistema de Gestión de la Información para el Pago de Estipendio a Estudiantes Universitarios*” brinda la posibilidad de actualizar toda la información necesaria en una bases de datos a través de un ambiente de entrada de datos, diseñado para un fácil uso de manipulación, ahorrando gran cantidad de tiempo, con mayor precisión, confiabilidad y calidad en los datos a ser procesados, así como las salidas automatizadas de todos los modelos necesarios aún más detallados y consultas asincrónicas de la información a través de las redes de información comunicación por parte de los directivos dentro o fuera de la universidad como dato de apoyo para la toma de decisiones, entre otras.

1.5 Metodologías utilizadas.

1.5.1 Lenguaje de Modelación Unificado (UML).

El Lenguaje de Modelado Unificado (UML - Unified Modeling Language) permite modelar, construir y documentar los elementos que forman un producto de software que responde a un enfoque orientado a objetos. Este lenguaje fue creado por un grupo de estudiosos de la Ingeniería de Software formado por: Ivar Jacobson, Grady Booch y James Rumbaugh en el año 1995. Desde entonces, se ha convertido en el estándar internacional para definir, organizar y visualizar los elementos que configuran la arquitectura de una aplicación

orientada a objetos. Con este lenguaje, se pretende unificar las experiencias acumuladas sobre técnicas de modelado e incorporar las mejores prácticas actuales en un acercamiento estándar.

UML no es un lenguaje de programación sino un lenguaje de propósito general para el modelado orientado a objetos y también puede considerarse como un lenguaje de modelado visual que permite una abstracción del sistema y sus componentes. [8]

Entre sus objetivos fundamentales se encuentran: [9]

1. Ser tan simple como sea posible, pero manteniendo la capacidad de modelar toda la gama de sistemas que se necesita construir.
2. Necesita ser lo suficientemente expresivo para manejar todos los conceptos que se originan en un sistema moderno, tales como la concurrencia y distribución, así como también los mecanismos de la ingeniería de software, como son el encapsulamiento y el uso de componentes.
3. Debe ser un lenguaje universal, como cualquier lenguaje de propósito general.
4. Imponer un estándar mundial.

1.5.2 Proceso Unificado de Desarrollo (RUP).

El Proceso Unificado de Desarrollo (RUP, por su denominación en inglés Rational Unified Process), fue creado por el mismo grupo de expertos que crearon UML, Ivar Jacobson, Grady Booch y James Rumbaugh en el año 1998. El objetivo que se perseguía con esta metodología era producir software de alta calidad, es decir, que cumpla con los requerimientos de los usuarios dentro de una planificación y presupuesto establecidos. Esta metodología concibió desde sus inicios el uso de UML como lenguaje de modelado.

Es un proceso dirigido por casos de uso, este avanza a través de una serie de flujos de trabajo, está centrado en la arquitectura y es iterativo e incremental. Además cubre el ciclo de vida de desarrollo de un proyecto y toma en cuenta las mejores prácticas a utilizar en el modelo de desarrollo de software. A continuación se muestran estas prácticas. [10]

- Desarrollo de software en forma iterativa.
- Manejo de requerimientos.
- Utiliza arquitectura basada en componentes.
- Modelación del software visualmente
- Verifica la calidad del software.
- Controla los cambios.

Para apoyar el trabajo con esta metodología ha sido desarrollada por la compañía norteamericana Rational Corporation la herramienta CASE (Computer Aided Assisted Automated Software Engineering) Rational Rose. Esta herramienta integra todos los elementos que propone la metodología para cubrir el ciclo de vida de un proyecto.

1.6 Paradigmas de programación.

1.6.1 Programación por capas.

La programación por capas es un estilo de programación en la que el objetivo primordial es la separación de la lógica de negocios de la lógica de diseño, un ejemplo básico de esto es separar la capa de datos de la capa de presentación al usuario. La ventaja principal de este estilo, es que el desarrollo se puede llevar a cabo en varios niveles y en caso de algún cambio, sólo se modifica el nivel requerido sin tener que revisar todo el código. Además permite distribuir el trabajo de creación de una aplicación por niveles, de este modo, cada grupo de trabajo está totalmente abstraído del resto de niveles, simplemente es necesario conocer la Interfaz de Programación de Aplicaciones (Application Programming Interface - API por sus siglas en Inglés) que existe entre los niveles.

En el diseño de sistemas informáticos actual se suele usar las arquitecturas multinivel o Programación por capas, teniendo en cuenta las ventajas que esta ofrece. En dichas arquitecturas a cada nivel se le confía una misión simple, lo que permite el diseño de arquitecturas escalables es decir que pueden ampliarse con facilidad en caso de que las necesidades aumenten.

El diseño más usado en la actualidad es el de tres capas [11].

1. Capa de presentación
2. Capa de lógica de negocio
3. Capa de datos

1.- Capa de presentación: Esta capa provee la interfaz de usuario (IU). Aquí es donde la aplicación presenta información a los usuarios y acepta entradas o visualiza respuestas. Idealmente, la IU no desarrolla ningún procesamiento de negocios o reglas de validación de negocios. Por el contrario, la IU debería relegar sobre la capa de negocios para manipular estos asuntos. Esto es importante, especialmente hoy en día, debido a que es muy común para una aplicación tener múltiples IU.

2.- Capa de negocio: es donde residen los programas que se ejecutan, recibiendo las peticiones del usuario y enviando las respuestas tras el proceso. Se denomina capa de negocio (e incluso de lógica del negocio - LN) pues 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 el almacenamiento o la recuperación de los mismos.

3.- Capa de datos: La capa de acceso a datos (AD) está formada por uno o más Sistemas Gestores de Bases de Datos (SGBD) que se encargan del almacenamiento y recuperación de los datos que necesita la capa de negocio. Todas estas capas pueden residir en un mismo equipo. Lo más usual es que haya una multitud de computadoras donde resida la capa de presentación, es decir los clientes de la arquitectura cliente/servidor, y una computadora central que ejerce la función de servidor en dicha arquitectura. Las capas de negocio y de datos pueden residir en un mismo servidor y si el crecimiento de las necesidades lo aconseja se pueden separar en dos o mas servidores. Así mismo, si el tamaño o la complejidad de la base de datos aumentara, se puede separar en varios servidores de datos, los cuales recibirán las peticiones del servidor donde reside la capa de negocio.

Si por el contrario fuese la complejidad en la capa de negocio lo que obligase a la separación, esta capa de negocio podría residir en una o más computadoras que realizarían solicitudes a una única base de datos. En sistemas muy complejos se llega a tener una serie de ordenadores sobre los cuales corre la capa de datos, y otra serie de ordenadores sobre los cuales corre la base de datos. En este caso se está haciendo referencia a Sistemas de Aplicaciones Distribuidas. En una arquitectura de tres niveles, los términos Capas o Niveles no significan lo mismo ni son similares. El término capa hace referencia a la forma como una solución es segmentada desde el punto de vista lógico: Presentación/ Lógica de Negocio/ Datos. El término nivel, corresponde a la forma como las capas lógicas, se encuentran distribuidas de forma física.

1.7 Tecnologías a utilizar.

1.7.1 Interfaz de usuario (Capa de Presentación).

Para desarrollar la interfaz de usuario de cualquier aplicación existen varias herramientas como Borland Delphi, Visual Fox Pro, Microsoft FrontPage, Macromedia Dreamweaver entre otras. [12]

Generalmente para el desarrollo de interfaz se utilizan herramientas web por las comodidades que ofrecen. Se realizó un análisis de las características fundamentales de Macromedia Dreamweaver 8 para justificar el porqué de su selección para la realización del diseño de la interfaz de este proyecto.

Macromedia Dreamweaver

Macromedia Dreamweaver es uno de los editores de desarrollo Web más utilizado a nivel profesional para la creación de sitios Web. Su amplio abanico de herramientas permite crear desde la más simple página Web personal hasta el sitio Web más completo y complejo para una gran empresa y utilizar casi todos los recursos de la Web. Este editor de HTML profesional para el diseño, codificación y desarrollo de páginas, sitios y aplicaciones Web; permite la edición visual, o sea, crear páginas rápidamente sin escribir código, así como también la codificación manual. Dreamweaver ayuda además a construir aplicaciones Web dinámicas apoyadas en bases de datos, es completamente personalizable. Se pueden crear objetos y comandos propios, modificar los accesos directos de teclado, e incluso escribir código *script* (guión) para extender las capacidades de las páginas web creadas con nuevos comportamientos.

Dreamweaver soporta varias tecnologías del servidor para la construcción de aplicaciones Web, tales como: Macromedia ColdFusion, Microsoft ASP (Active Server Page), Microsoft ASP.NET, Sun JavaServer Pages (JSP) y PHP (Profesional Home Page Tools).

AJAX

Es un acrónimo de Asynchronous Javascript And XML (Javascript Asincrónico y XML) no es una tecnología web, sino el uso coordinado de distintas tecnologías que en conjunto permiten una mayor rapidez y eficacia para las aplicaciones basadas en la Web. [13]

Hasta hace muy poco tiempo una de las principales diferencias entre las aplicaciones Web y las aplicaciones de escritorio era sin dudas su rapidez de ejecución, en el momento en que el

volumen de datos a gestionar era algo elevado, la ralentización era la normal. Una de las principales aportaciones de AJAX, es acelerar la velocidad de ejecución de estas aplicaciones web haciéndolas muy similares a las aplicaciones de escritorio. Para ello la clave está en modificar existencialmente el tipo de relaciones que se crean entre el navegador Web y el servidor.

1.7.2 Lógica del negocio (Capa de la Lógica del Negocio).

En la capa de lógica del negocio es necesario seleccionar un lenguaje de programación y un servidor de aplicaciones web. Teniendo en cuenta que existen varios lenguajes de programación web del lado del servidor (*Server Side Scripts*) como ASP, JSP, PHP entre otros, se analizaron las características fundamentales que permitieran seleccionar cual utilizar y en el caso de los servidores web se analizaron las características fundamentales de los dos servidores más difundidos en el mundo: Internet Information Server (IIS) y Apache.

1.7.2.1 Lenguajes de programación web.

ASP

Active Server Pages (ASP) es una tecnología de Microsoft que funciona del lado servidor permite generar páginas web dinámica, que ha sido comercializada como un anexo a Internet Information Services (IIS), está gratuito en el mercado, pero depende de la plataforma Windows. [16]

JSP

JSP es un acrónimo de Java Server Pages, (Páginas de Servidor Java). Es una tecnología orientada a crear páginas web con programación en Java. Con JSP pueden crearse aplicaciones web que se ejecuten en variados servidores web, de múltiples plataformas, ya que Java es un lenguaje multiplataforma. Las páginas JSP están compuestas de código HTML/XML mezclado con etiquetas especiales para programar scripts de servidor en sintaxis Java. [15]

PHP

PHP es un lenguaje de programación el cual se ejecuta en los servidores Web y que permite crear contenido dinámico en las páginas HTML, con un lenguaje propietario derivado del Perl. Al principio, PHP sólo estaba compuesto por algunas macros que facilitaban el trabajo a la hora de crear una página Web. Hacia mediados de 1995 se creó el analizador sintáctico y se llamó PHP/F1 Versión 2, y sólo reconocía el texto HTML y algunas directivas de MySQL. A partir de este momento, la contribución al código fue pública. El crecimiento de PHP desde entonces ha sido exponencial, y han surgido versiones nuevas como las actuales, PHP3, PHP4 y PHP5 orientado a objetos.

Dispone de múltiples herramientas que permiten acceder a bases de datos de forma sencilla, por lo que es ideal para crear aplicaciones para Internet.

Es multiplataforma, funciona tanto para Unix como para Windows de forma que el código que se haya creado para una de ellas no tiene porqué modificarse al pasar a la otra.

El lenguaje PHP es un lenguaje de programación de estilo clásico, con variables, sentencias condicionales, bucles, funciones, entre otras. La sintaxis que utiliza la toma de otros lenguajes muy extendidos como C y Perl. [16]

El lenguaje PHP, como todos los del tipo “guiones del lado del servidor”, envía al navegador código HTML, por lo que transparente al usuario final.

PHP se encuentra libre en el mercado y se puede acceder a él por medio de Internet.

Cuenta además con un repositorio muy completo de clases, PEAR (Repositorio de Aplicaciones y Extensiones de PHP), en el cual se puede encontrar desde clases para manejar ecuaciones matemáticas o para generar gráficos, hasta clases para generar hojas de cálculo en Excel, de una forma fácil y con funciones pocas veces vista con PHP. PEAR cuenta con 273 clases (hasta el momento de la revisión bibliográfica), está dividido en dos partes principales: PEAR y PCL, el primero lo constituyen paquetes escritos completamente en PHP, el segundo está compuesto por clases escritas en C o C++. [17]

1.7.2.2 Servicios Web.

Un servicio web es una entidad programable que proporciona alguna funcionalidad determinada, y es accesible a cualquier número de sistemas que usen las normas de Internet. Un servicio web puede ser usado internamente por una aplicación o ser publicado

hacia Internet. Estos servicios permiten la ejecución de sus funcionalidades sin importar la plataforma, sistema operativo, o lenguaje en el cual estén implementados. [18]

Los servicios web se pueden utilizar para intercambiar datos en redes de ordenadores como Internet. La interoperabilidad se consigue mediante la adopción de estándares abiertos. Las organizaciones OASIS y W3C son los comités responsables de la arquitectura y reglamentación de los servicios web. Para mejorar la interoperabilidad entre distintas implementaciones de servicios web se ha creado el organismo WS-I, encargado de desarrollar diversos perfiles para definir de manera más exhaustiva estos estándares. [18]

Los servicios web tienen una interfaz descrita en un formato que puede ser procesado por una máquina (específicamente WSDL) y otros sistemas interactúan con el servicio web utilizando mensajes SOAP.

Los servicios web brindan grandes ventajas dentro de las aplicaciones distribuidas como son: [18]

- Aportan interoperabilidad entre aplicaciones de software independientemente de sus propiedades o de las plataformas sobre las que se instalen.
- Los servicios web fomentan los estándares y protocolos basados en texto, que hacen más fácil acceder a su contenido y entender su funcionamiento.
- Al apoyarse en HTTP, los servicios Web pueden aprovecharse de los sistemas de seguridad firewall sin necesidad de cambiar las reglas de filtrado.
- Permiten que servicios y software de diferentes compañías ubicadas en diferentes lugares geográficos puedan ser combinados fácilmente para proveer servicios integrados.

Atendiendo a las ventajas, antes mencionadas, que brindan el uso de webservice se incluye su utilización en el desarrollo de este proyecto.

Los servicios web pueden ser utilizados a través de dos protocolos fundamentalmente: RPC (*Remote Procedure Call*) y SOAP (*Simple Object Access Protocol* – Protocolo de Acceso Simple a Objetos). El protocolo RPC tiene una dependencia de la plataforma y la tecnología que se utiliza para ejecutar los servicios web, en cambio SOAP es un protocolo que permite la independencia de plataforma y tecnología.

1.7.2.3 SOAP.

SOAP, siglas de *Simple Object Access Protocol*, es un protocolo estándar creado por Microsoft, IBM y otros, está actualmente bajo el auspicio de la W3C que define cómo dos

objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML (eXtensible Markup Language). SOAP es uno de los protocolos utilizados en los servicios Web. [19]

A diferencia de DCOM y CORBA, que son binarios, SOAP usa el código fuente en XML, que facilita la eliminación de errores, pero es menos efectivo. El intercambio de mensajes se realiza mediante tecnología de componentes. El término *Object* en el nombre significa que se adhiere al paradigma de la programación orientada a objetos.

SOAP es un marco extensible y descentralizado que permite trabajar sobre múltiples pilas de protocolos de redes informáticas. Los procedimientos de llamadas remotas pueden ser modelados en la forma de varios mensajes SOAP interactuando entre sí.

SOAP corre sobre cualquier protocolo de Internet, generalmente HTTP. SOAP tiene como base XML, con un diseño que cumple el patrón Cabecera-Desarrollo de diseño de software, como otros muchos diseños, verbigracia HTML. La cabecera (*header*) es opcional y contiene metadatos sobre enrutamiento (*routing*), seguridad o transacciones. El desarrollo (*body*) contiene la información principal, que se conoce como carga útil (*payload*). La carga útil se acoge a un esquema de XML propio.

1.7.2.4 XML.

El Lenguaje Extensible de Marcas, abreviado XML (eXtensible Markup Language), describe una clase de objetos de datos llamados documentos XML y describe parcialmente el comportamiento de los programas de computadora que los procesan.

XML es un "perfil de aplicación" o una forma restringida de SGML, el Lenguaje Estándar Generalizado de Marcación [ISO 8879]. Por construcción, los documentos XML son documentos SGML conformados. [20]

Los documentos XML están compuestos por unidades de almacenamiento llamadas entidades, que contienen tanto datos analizados como no analizados. Los datos analizados están compuestos de caracteres, algunos de los cuales, de la forma datos carácter, y otros de la forma etiquetas. Las etiquetas codifican una descripción de la estructura de almacenamiento del documento y su estructura lógica XML fue desarrollado por un Grupo de Trabajo (originalmente conocido como "*SGML Editorial Review Board*" – Comité Editorial de Revisión del SGML) formado bajo los auspicios del Consorcio World Wide Web (W3C), en 1996. Fue presidido por Jon Bosak de Sun Microsystems con la participación activa de un

Grupo Especial de Interés en XML (previamente conocido como Grupo de Trabajo SGML (*Standard Generalized Markup Language* – Lenguaje Estándar Generalizado de Marcado) también organizado en el W3C.

Los objetivos de diseño de XML son [20]:

- XML debe ser directamente utilizable sobre Internet.
- XML debe soportar una amplia variedad de aplicaciones.
- XML debe ser compatible con SGML.
- Debe ser fácil la escritura de programas que procesen documentos XML.
- El diseño de XML debe ser formal y conciso.
- Los documentos XML deben ser fácilmente creables.

XML es la solución a un problema de comunicación entre programas de computadoras. XML intenta ser un formato absolutamente genérico, con el que se puede describir cualquier tipo de archivo.

Características de XML [20]:

- Es una arquitectura más abierta y extensible. No se necesita versiones para que puedan funcionar en futuros navegadores. Los identificadores pueden crearse de manera simple y ser adaptados en el acto en internet/intranet por medio de un validador de documentos (parser).
- Mayor consistencia, homogeneidad y amplitud de los identificadores descriptivos del documento con XML (los RDF Resource Description Framework), en comparación a los atributos de la etiqueta <META> del HTML.
- Integración de los datos de las fuentes más dispares. Se podrá hacer el intercambio de documentos entre las aplicaciones tanto en el propio PC como en una red local o extensa.
- Datos compuestos de múltiples aplicaciones. La extensibilidad y flexibilidad de este lenguaje nos permitirá agrupar una variedad amplia de aplicaciones, desde páginas web hasta bases de datos.
- Gestión y manipulación de los datos desde el propio cliente web.
- Los motores de búsqueda devolverán respuestas más adecuadas y precisas, ya que la codificación del contenido web en XML consigue que la estructura de la información resulte más accesible.

- Se desarrollarán de manera extensible las búsquedas personalizables y subjetivas para robots y agentes inteligentes. También conllevará que los clientes web puedan ser más autónomos para desarrollar tareas que actualmente se ejecutan en el servidor.
- Se permitirá un comportamiento más estable y actualizable de las aplicaciones web, incluyendo enlaces bidireccionales y almacenados de forma externa.
- El concepto de "hipertexto" se desarrollará ampliamente (permitirá denominación independiente de la ubicación, enlaces bidireccionales, enlaces que pueden especificarse y gestionarse desde fuera del documento, hiperenlaces múltiples, enlaces agrupados, atributos para los enlaces, etc. Creado a través del Lenguaje de enlaces extensible (XLL).
- Exportabilidad a otros formatos de publicación (papel, web, cd-rom, etc.). El documento maestro de la edición electrónica podría ser un documento XML que se integraría en el formato deseado de manera directa.

1.7.2.5 Servidores de Aplicaciones Web.

Se realizó un estudio de dos de los servidores de aplicaciones web mas difundidos para determinar cuál será usado en el proyecto.

Apache

El **servidor Apache** es un servidor HTTP de código abierto para plataformas Uníx (BSD, GNU/Linux, etcétera), Windows y otras, que implementa el protocolo HTTP/1.1 (RFC 2616) y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA HTTPd 1.3, pero más tarde fue reescrito por completo. El servidor Apache se desarrolla dentro del proyecto http Server (httpd) de la Apache Software Foundation. Apache presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido, carece de una interfaz gráfica que ayude en su configuración.

En la actualidad, Apache es el servidor HTTP más usado, siendo el servidor http del 70% de los sitios web en el mundo y creciendo aún su cuota de mercado.

Internet Information Services

Internet Information Services (IIS) engloba una serie de herramientas administrativas que permite controlar sitios Web, FTP, SMTP (correo saliente) y Servicio de noticias (NNTP). Dispone también del soporte para crear páginas dinámicas (ASP), tecnología para el desarrollo de aplicaciones para Internet ampliamente extendida.

IIS es un servidor de distribución gratuita, pero tiene como condición que corre solamente sobre plataforma Windows. Al igual que Apache posee mensajes de error configurables, y a diferencia del mismo si posee una interfaz gráfica que ayuda en la configuración del sitio en su totalidad.

Después de analizar las características fundamentales de Apache e IIS se decide usar como servidor de aplicaciones web Apache teniendo en cuenta los siguientes aspectos:

- Se encuentra gratis en Internet,
- Posee una alta compatibilidad con PHP,
- Y cumple con uno de los objetivos específicos trazados al inicio de la investigación, que es lograr un sistema multiplataforma.

1.7.3 Capa de acceso a datos.

Sistemas Gestores de Base de Datos

Una Base de Datos (BD) es un conjunto de datos interrelacionados, almacenados con carácter más o menos permanente en la computadora, puede ser considerada una colección de datos variables en el tiempo. [21]

Un **Sistema Gestor de Base de Datos** (SGBD) 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. [21]

El objetivo fundamental de un SGBD consiste en suministrar al usuario las herramientas que le permitan manipular, en términos abstractos, los datos, o sea, de forma que no le sea necesario conocer el modo de almacenamiento de los datos en la computadora, ni el método de acceso empleado.

En el mercado existen un sinnúmero de gestores de base de datos, todos con sus características que lo hacen una opción a escoger, pero la elección, la mayoría de las veces, depende del cliente y no del desarrollador.

A continuación mostramos una tabla (Tabla 1) comparativa de algunos de los SGBD en cuanto a si son Multiplataforma o no, Gratis o no y en cuanto a la disponibilidad del código fuente.

Multiplataforma	Gratis	Código	Fuente
MSSQL	No	No	No
MSDE	No	Si	No
MySQL	Si	Si	Si
Postgres	Si	Si	Si
Firebird	Si	Si	Si
Interbase	Si	No	SI
SyBase	Si	No	No
INFORMIX	Si	No	No
Oracle	Si	No	No

Tabla 1. Comparación entre SGBD.

MySQL

MySQL es un sistema de administración de Base de Datos. Opera en una arquitectura cliente/servidor. Es el sistema gestor de bases de datos “*Open Source*” más popular, o sea que puede ser bajado de Internet y usarlo sin tener que pagar, además que cualquiera puede estudiar su código y adecuarlo a las necesidades que requiera. [22]

MySQL es muy rápido, fiable y fácil de usar, surge para manipular bases de datos muy grandes. Es un sistema multiplataforma de base de datos relacionales, lo que da velocidad y flexibilidad, cuenta con un sistema de privilegios contraseñas muy seguro que permite la autenticación básica para el acceso al servidor. [23]

El lenguaje *PHP* es altamente compatible con *MySQL*, por el amplio conjunto de comandos definidos para el tratamiento de este.

1.8 Conclusiones.

Después de un exhaustivo análisis de las tecnologías antes expuestas y debido a la gran variedad de plataformas en cuanto a Sistema Operativo (SO) y en cuanto a soporte para datos (SGBD) llegamos a las siguientes conclusiones preeliminares:

1. Se propone utilizar la programación orientada a objetos por los beneficios que está brinda, por lo que se considera RUP (Proceso Unificado de Desarrollo) la metodología más apropiada para el desarrollo del proyecto y UML como el lenguaje de modelación necesario en este caso.

2. Para la Interfaz de Usuario (IU) se utilizará Macromedia Dreamweaver, Adobe Photoshop.
3. Para la capa de la Lógica del Negocio (LN) se utilizará Apache como Servidor de Aplicaciones WEB, debido fundamentalmente a su gratuidad en el uso y la posibilidad multiplataforma del mismo. La elección del lenguaje de programación es evidente. Se seleccionará PHP por ser este uno de los lenguajes de código abierto (open source) más difundidos, ser independiente de plataforma y estar muy ligado al Servidor de Aplicación WEB seleccionado. Se utilizará PHP Expert Editor como herramienta de programación en este lenguaje. Para el manejo de los datos en esta capa se recomienda utilizar XML por ser independiente de plataforma y ampliamente utilizado por la mayoría de los sistemas modernos de comunicaciones para el intercambio de datos.
4. En la capa de acceso a datos se utilizará las bibliotecas PEAR escritas en PHP y como motor de base de datos MySQL, por ser multiplataforma y de código abierto.
5. Para la comunicación entre capas se utilizará XML como lenguaje de intercambio de información.

Capítulo 2. Análisis y Diseño.

2.1 Introducción.

Antes de comenzar a desarrollar un sistema es necesario comprender la organización bajo estudio y los procesos que en ella tienen lugar, a fin de lograr una mejor comprensión del problema a resolver y el común entendimiento entre clientes y desarrolladores.

Teniendo como guía la Metodología RUP, y utilizando uno de los artefactos que brinda dicha metodología se logra modelar y describir el Modelo del Negocio haciéndose evidente la consulta de las nuevas tendencias imperantes y del uso de una metodología que indique los pasos correctos a seguir. Además, se presenta una descripción detallada de las reglas del negocio que el objeto de automatización debe seguir para asegurar el cumplimiento de las restricciones que existen. También se hace una descripción de la solución propuesta para realizar un análisis de los requisitos funcionales y no funcionales del sistema que dará solución al problema planteado; quiénes interactuarán con él (actores del sistema) y las distintas funcionalidades que ofrecerá a cada uno de los actores.

2.2 Descripción del modelo de negocio.

Antes de comenzar a desarrollar un sistema es necesario comprender la organización bajo estudio y los procesos que en ella tienen lugar, a fin de lograr una mejor comprensión del problema a resolver y el común entendimiento entre clientes y desarrolladores; para lo cual se realiza la modelación del negocio [24].

El modelado del negocio es una técnica que permite comprender los procesos de negocio de la organización y se desarrolla en dos pasos:

- 1. Confección de un modelo de casos de uso del negocio que identifique los actores y casos de uso del negocio que utilicen los actores.
- 2. Desarrollo de un modelo de objetos del negocio compuesto por trabajadores y entidades del negocio que juntos realizan los casos de uso del negocio.

En este capítulo se describen los procesos de negocio del Departamento de Finanzas de la Universidad de Cienfuegos, a través de la descripción de los distintos artefactos utilizados para definir el modelo del negocio.

2.2.1 Descripción de los procesos del negocio.

El Departamento de finanzas de la Universidad de Cienfuegos tiene como uno de sus objetivos la gestión de la información del estipendio estudiantil para el correcto funcionamiento del pago a los estudiantes universitarios.

La gestión del estipendio estudiantil en la universidad se inicia a partir del envío por parte de cada secretaria de facultad de los listados de cada grupo de sus carreras. Estos listados se envían al inicio de curso y posteriormente cada mes se envía una actualización con las altas y bajas de estudiantes así como las respectivas actualizaciones por categorías de los estudiantes ya existentes, estas categorías se refieren a: alumnos ayudantes o préstamo estudiantil. El proceso continua cuando la especialista en finanzas recibe estos listados y de acuerdo a las categorías y cantidad de estudiantes conforma el monto total de dinero por facultades para la confección del cheque por su responsable. Luego se lo entrega al cajero para que el mismo lo cobre en el banco, posteriormente se distribuyen el dinero y las nóminas por cada facultad a los administradores de las mismas y se comienza el pago a cada estudiante. Si el estudiante no cobra este dinero se reintegra al banco y el estudiante lo puede reclamar el mes próximo, si lo reclama y no lo cobra este dinero va a una cuenta a parte para el fondo de la universidad.

Anteriormente se explicó como funciona en la actualidad la gestión de la información del estipendio estudiantil universitario; identificando entonces los siguientes procesos de negocio:

1. Gestionar Préstamo estudiantil.
2. Gestionar Estipendio estudiantil.
3. Gestionar dinero para el pago del estipendio estudiantil.
4. Solicitar autorizo de cobro del estipendio estudiantil.
5. Reclamar el pago de estipendio estudiantil no cobrado.

Gestionar Préstamo estudiantil persigue el objetivo de facilitar a cada estudiante la gestión de este proceso por parte de las secretarias y directivos de facultades. Este proceso se inicia cuando cada alumno solicita el modelo en su facultad para su llenado. Posteriormente se revisa y aprueba por los directivos que intervienen en el proceso.

Gestionar Estipendio estudiantil no es más que las operaciones que cada secretaria debe realizar en su facultad para facilitar a la especialista en finanzas la gestión de la información para el pago del estipendio estudiantil. También se debe tener en cuenta que es el proceso

que inicia todo el modelo de negocio al mandar los listados actualizados de los estudiantes que cobrarán el estipendio estudiantil cada mes.

Gestionar dinero para el pago del estipendio estudiantil. Consiste en darle continuidad al proceso de gestión que inició cada secretaria en su facultad y facilitarle al especialista en finanzas, responsable de cheque, cajero, administradores de facultades, funcionarios del banco toda la gestión del dinero para el pago del estipendio. Interviniendo en él varias acciones como son: la confección de cheques, la distribución de nóminas y su entrega a los administradores, el reintegro de las misma con el dinero no cobrado concluyendo con el cobro del estipendio por los estudiantes.

Solicitar autorizo de cobro del estipendio estudiantil consiste en la solicitud de un modelo oficial a la especialista en finanzas para designar a alguna persona que cobre el estipendio de otra. Este modelo debe llevar los datos y firma del solicitante y del autorizado y la firma del decano de su facultad.

Reclamar el pago de estipendio estudiantil no cobrado consiste en reclamar el estipendio del mes concluido que no fue cobrado por el estudiante. Este reclamo se hace mediante un modelo oficial y debe ser en un período de 10 días después que se haya finalizado el pago.

2.3 Reglas del negocio a considerar.

Después de identificar los procesos de negocio se definen las siguientes reglas del negocio: La secretaria de cada facultad debe enviar un listado oficial al inicio de curso. Este listado no debe incluir a los estudiantes:

extranjeros, planes especiales, Ley decreto 91, cadetes MININT y MINFAR. Estos deben ir en un listado aparte organizado por categorías.

La secretaria también debe enviar mensualmente a la especialista en finanzas de la universidad la actualización (alta y baja) de los listados de los estudiantes que cobraran ese mes.

La especialista en finanzas no puede confeccionar la nomina sin tener los listados que previamente la secretaria le envió.

Para la responsable de cheques poder confeccionar el cheque debe previamente haber recibido de la especialista en finanzas el monto total de dinero.

El cajero luego de confeccionado el cheque tiene 5 días hábiles para cobrar el mismo en el banco.

El cajero es el único que distribuye por facultades el dinero entregándoselo a cada administrador de la misma.

Cada administrador para pagar debe exigir el carné de identidad a cada estudiante y la firma antes de entregar el dinero al mismo.

Si el estudiante no cobra en un plazo de 5 días la nómina es devuelta al cajero y este reintegra el dinero al banco. A partir de esa fecha cada estudiante tiene 10 días hábiles para hacer la solicitud de reclamo de su estipendio a la especialista en finanzas mediante el modelo oficial.

Cada estudiante tiene derecho a solicitar un préstamo estudiantil en la secretaría de su facultad, para ello debe solicitar el Modelo de solicitud de préstamo en el cual llena los datos que le pide.

Luego de aprobado el préstamo estudiantil la especialista en finanzas lo incluye en la nómina a pagar del mes siguiente. Este contrato de préstamo debe renovarse todos los años. Cada estudiante en caso necesario puede solicitar un modelo de autorizo de pago en caso de no encontrarse en la fecha del mismo, para ello debe de llenar el modelo de autorizo previamente solicitado a la especialista en finanzas. Para que este modelo tenga validez debe de estar firmando por el solicitante, el que lo cobrará y el decano de su facultad.

2.4 Modelo de casos de uso del negocio.

El modelo de Casos de Uso del Negocio (CUN) describe los procesos de una empresa en términos de casos de uso y actores del negocio en correspondencia con los procesos del negocio y los clientes, respectivamente. El modelo de casos de uso presenta un sistema desde la perspectiva de su uso y esquematiza cómo proporciona valor a sus usuarios. Este modelo permite a los modeladores comprender mejor qué valor proporciona el negocio a sus actores. [25].

Este modelo es definido a través de tres elementos: el diagrama de casos de uso del negocio, la descripción de los casos de uso del negocio y el diagrama de actividades.

2.4.1 Actores del negocio.

Un **actor del negocio** es cualquier individuo, grupo, entidad, organización, máquina o sistema de información externos; con los que el negocio interactúa. Lo que se modela como actor es el rol que se juega cuando se interactúa con el negocio para beneficiarse de sus resultados [26].

Actor del negocio	Descripción
Estudiante	*Es el principal beneficiado en los procesos del negocio enmarcados en el cobro del estipendio estudiantil, para lo cual realiza diferentes procesos como: * Solicita a la especialista en finanzas el Modelo para el reclamo del estipendio estudiantil del mes no cobrado. * Solicita a la Secretaria de su facultad el modelo del préstamo estudiantil. *Solicita a la Especialista en Finanzas el modelo de autorizo de cobro.

Tabla 2. Descripción de los Actores del negocio.

2.4.2 Trabajadores del negocio.

Definición de trabajadores del negocio.

Un **trabajador del negocio** es una abstracción de una persona (o grupo de personas), una máquina o un sistema automatizado; que actúa en el negocio realizando una o varias actividades, interactuando con otros trabajadores del negocio y manipulando entidades del negocio. Representa un rol [27].

Trabajador del negocio	Descripción
Secretaria Docente	* Es la encargada de enviar el reporte mensual de altas y bajas de cada estudiante de su facultad * Es la encargada del llenado y verificación de la planilla de solicitud del préstamo estudiantil.
Especialista de Finanzas	* Solicita a la secretaria de cada facultad el listado mensual de los estudiantes que deben cobrar el estipendio. * Es la encargada de enviar el listado con el total de dinero de cada facultad para la extracción de fondos a la Especialista Principal de Finanzas. * Es la encargada de llenar el contrato del préstamo

	estudiantil.
Responsable de cheque	* Es el encargado de confeccionar el cheque con el monto total de dinero para el pago del estipendio estudiantil.
Cajero	* Es el encargado de cobrar el cheque para el pago del estipendio estudiantil en el banco. * Es el que entrega al administrador de cada facultad los listados para el pago del estipendio estudiantil. * Es el encargado de reintegrar el monto de dinero de estipendio no cobrado.
Funcionaria del Banco	*Es la encargada de interactuar con el cajero ya sea para el cobro del cheque o para el reintegro de dinero.
Administrador	* Es el encargado de pagar el estipendio a cada estudiante. * Es el encargado de reintegrar el dinero no cobrado al cajero.
Directivo	* Encargado de chequear y firmar el Préstamo Estudiantil.
Secretaria del Rector	*Es la encargada de entregar el modelo de préstamo estudiantil a la Especialista en Finanzas.

Tabla 3. Descripción de los trabajadores del negocio.

2.4.3 Casos de uso del negocio.

1. Gestionar Préstamo estudiantil.
2. Gestionar Estipendio estudiantil.
3. Gestionar dinero para el pago del estipendio estudiantil.
4. Solicitar autorizo de cobro del estipendio estudiantil.
5. Reclamar el pago de estipendio estudiantil no cobrado.

2.4.4 Diagramas de casos de uso del negocio.

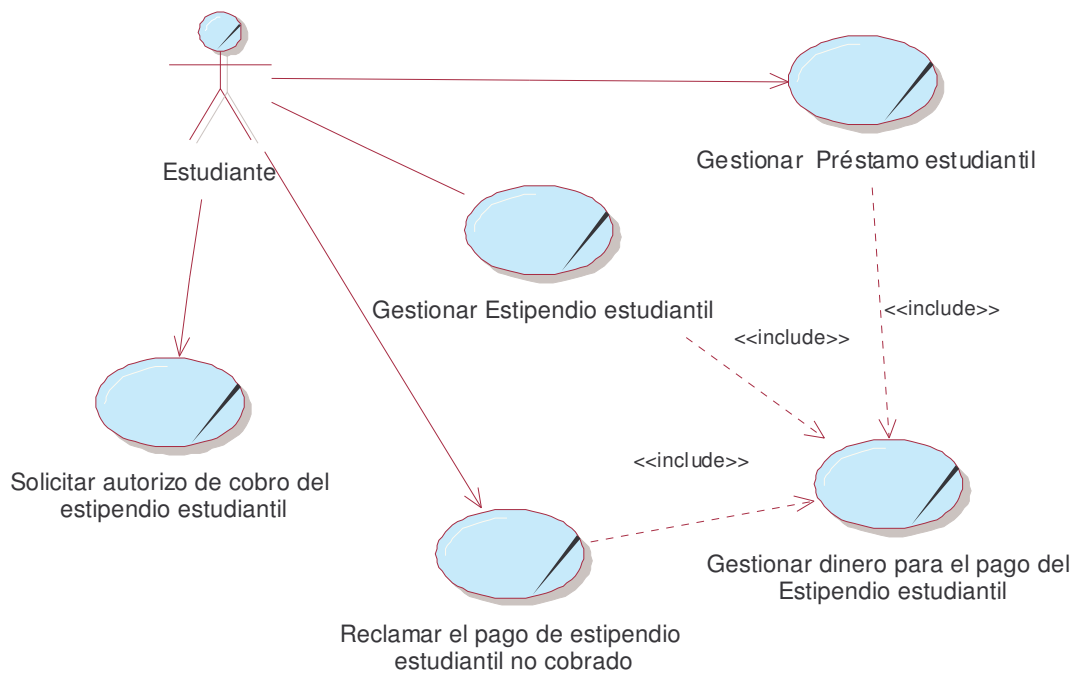


Figura 3. Diagramas de casos de uso del negocio.

2.4.5 Descripción de los casos de uso del negocio.

Caso de uso del negocio	Gestionar préstamo estudiantil.
Actores	Estudiante (Inicia)
Propósito	Obtener una ayuda económica.
Resumen: El caso de uso se inicia cuando el estudiante solicita el modelo del préstamo estudiantil. El estudiante realiza todas las funciones necesarias para el llenado del documento. El caso de uso termina cuando el estudiante obtiene el préstamo estudiantil.	
Acción del actor	Respuesta del proceso de negocio
1. El estudiante solicita el modelo de préstamo estudiantil a la secretaria de su facultad.	2. La secretaria recibe la solicitud. 3. La secretaria le entrega el modelo del préstamo estudiantil.
4. El estudiante introduce los	

datos solicitados y se lo entrega a la secretaria.	5. La secretaria recibe el modelo de préstamo estudiantil. 6. La secretaria le entrega el documento al decano. 7. El directivo recibe el documento. 8. El directivo llena los datos que le corresponde, y se lo entrega a la secretaria. 9. La secretaria recibe el modelo 10. La secretaria de la facultad le entrega el modelo de préstamo estudiantil a la Secretaria del rector. 11. La secretaria del rector recibe el documento. 12. Se lo entrega al directivo 13. El directivo recibe el modelo actualizado. 14. El directivo llena los datos que le corresponde, lo firma y se lo devuelve a su secretaria. 15. La secretaria del rector recibe el modelo actualizado 16. La secretaria del rector le envía a la especialista en finanzas el modelo. 15. La especialista en finanzas recibe el modelo. 16. Hacer referencia a caso de uso incluido (Gestionar dinero para el pago del estipendio estudiantil).
Prioridad	Proceso secundario del negocio.
Mejoras	El sistema brindará la posibilidad de imprimir el modelo de préstamo del estipendio estudiantil.
Caso de uso del negocio	Gestionar estipendio estudiantil.
Actores	Estudiante.

Propósito	Cobro del estipendio estudiantil.
Resumen El caso de uso se inicia cuando la secretaria manda los listados actualizados de su facultad con los estudiantes que cobrarán cada mes. Se realizan todos los procesos involucrados para el pago del estipendio. El caso de uso termina cuando el estudiante cobra su estipendio.	
Acción del actor	Respuesta del proceso de negocio
	1. La secretaria manda el listado general de su facultad al inicio de curso a la especialista en finanzas. 2. La especialista en finanzas recibe el listado. 3. Hacer referencia a caso de uso incluido (Gestionar dinero para el pago del estipendio estudiantil).
Prioridad	Es el principal proceso del negocio.
Flujos alternos	
Línea 1	* Si no es el inicio del curso manda un listado con las actualizaciones (altas, bajas) de los estudiantes de la facultad a cobrar.
Prioridad	Proceso primario del negocio.
Mejoras	El proceso se realizará automatizado permitiéndole a la secretaria actualizar los datos de los estudiantes quedando almacenados para su posterior consulta. Como resultado se evitará errores en la entrada y manipulación de los datos y se agilizará dicho proceso.

Caso de uso del negocio	Solicitar autorizo de cobro del estipendio estudiantil
--------------------------------	--

Actores	Estudiante (Inicia)
Propósito	Autorizar a otra persona a cobrar el estipendio estudiantil.
Resumen El caso de uso se inicia cuando el estudiante solicita el modelo de autorizo de cobro. El estudiante llena el modelo con los datos necesarios y las firmas correspondientes. El caso de uso termina cuando el autorizado cobra el estipendio estudiantil.	
Acción del actor	Respuesta del proceso de negocio
1. El estudiante solicita el modelo de autorizo de cobro de estipendio a la especialista en finanzas. 4. El estudiante llena el modelo con la información necesaria y se lo entrega al directivo. 7. El estudiante recibe el modelo firmado y aprobado.	2. La especialista en finanzas recibe la solicitud. 3. La especialista en finanzas le entrega el modelo para ser llenado. 5. El directivo recibe el modelo, 6. El directivo aprueba y firma el modelo y se lo entrega al estudiante.
Prioridad	Proceso secundario del negocio.
Mejoras	El sistema brindará la posibilidad de imprimir el modelo de solicitud de reclamo del estipendio estudiantil.
Caso de uso del negocio	Reclamar el pago de estipendio estudiantil no cobrado.

[illegible]

	(Gestionar dinero para el pago del estipendio estudiantil).
Prioridad	Proceso Secundario del negocio.
Flujos alternos	
Línea 2	* Si tiene 2 meses o menos sin reintegrar se le entrega el modelo sino no se le, finalizando así el caso de uso.
Línea 7	* El estudiante pierde el derecho de cobrar el estipendio.
Mejoras	El sistema brindará la posibilidad de imprimir el modelo de reclamo del estipendio estudiantil.

Caso de uso del negocio	Gestionar dinero para el pago del estipendio estudiantil.
Actores	Estudiante
Propósito	Procesos y etapas principales por las que pasa el pago del estipendio
Resumen El caso de uso se inicia cuando la especialista en finanzas confecciona la nómina de pago por facultades. Luego se siguen una series de pasos que hacen posible el pago del estipendio a cada estudiante. El caso de uso termina cuando el estudiante cobra el estipendio estudiantil o el dinero se reintegra al banco.	
Acción del actor	Respuesta del proceso de negocio
	1. La especialista en finanzas confecciona la nómina. 2 La especialista en finanzas envía a la Responsable de cheque el monto total del dinero. 3. La responsable de cheque confecciona el cheque con el monto total que le entregó la especialista en Finanzas. 4. El cajero recibe el cheque para luego cobrarlo.

10. El estudiante cobra su dinero.	5. El cajero cobra el cheque en el banco antes de cumplirse los 5 días hábiles de confeccionado el mismo. 6. La funcionaria del banco paga el cheque. 7. Cajero distribuye el dinero con sus nóminas por facultades y se lo entrega al administrador de cada facultad. 8. El administrador recibe las nóminas con el dinero a pagar. 9. El administrador paga el estipendio 11. El administrador envía la nómina al cajero con el dinero a reintegrar. 12. El cajero recibe el dinero y las nóminas de cada administrador. 13. El cajero contabiliza el dinero por facultades a reintegrar 14. El cajero reintegra el dinero no cobrado al banco. 15. La funcionaria del banco recibe el dinero y finaliza el caso de uso.
Prioridad	Proceso primario del negocio.
Flujos alternos	
Línea 5	* No puede cobrarlo pues carece de validez y concluye el caso de uso.
Línea 10	* Se reintegra el dinero al banco.
Mejoras	El sistema permitirá automatizar la confección de las nóminas.

2.4.6 Diagramas de actividades del negocio.

El diagrama de actividad es un grafo que contiene los estados en que puede hallarse la actividad a analizar. Cada estado de la actividad representa la ejecución de una sentencia de

un procedimiento, o el funcionamiento de una actividad en un flujo de trabajo. En resumen describe un proceso que explora el orden de las actividades que logran los objetivos del negocio. [28].

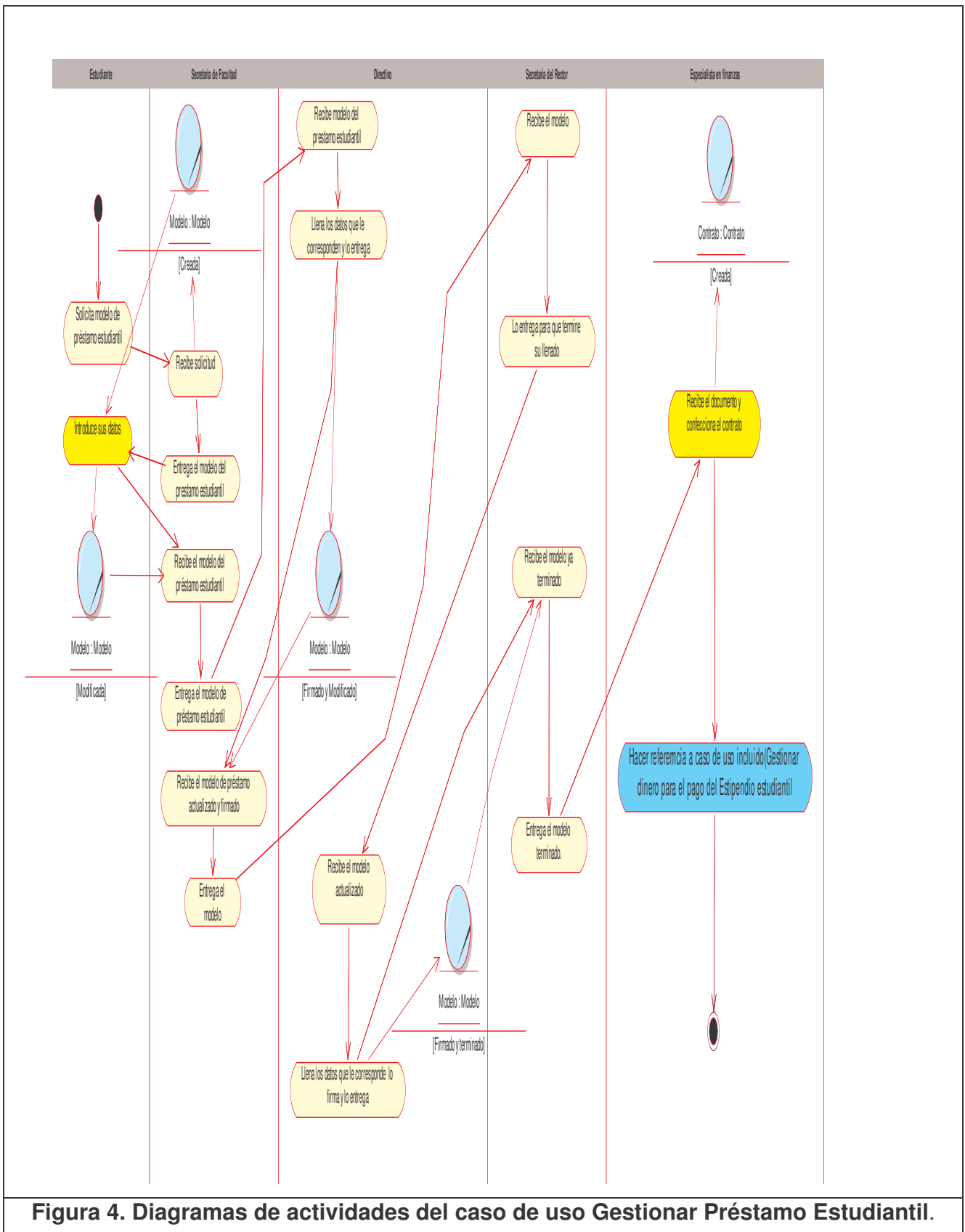


Figura 4. Diagramas de actividades del caso de uso Gestionar Préstamo Estudiantil.

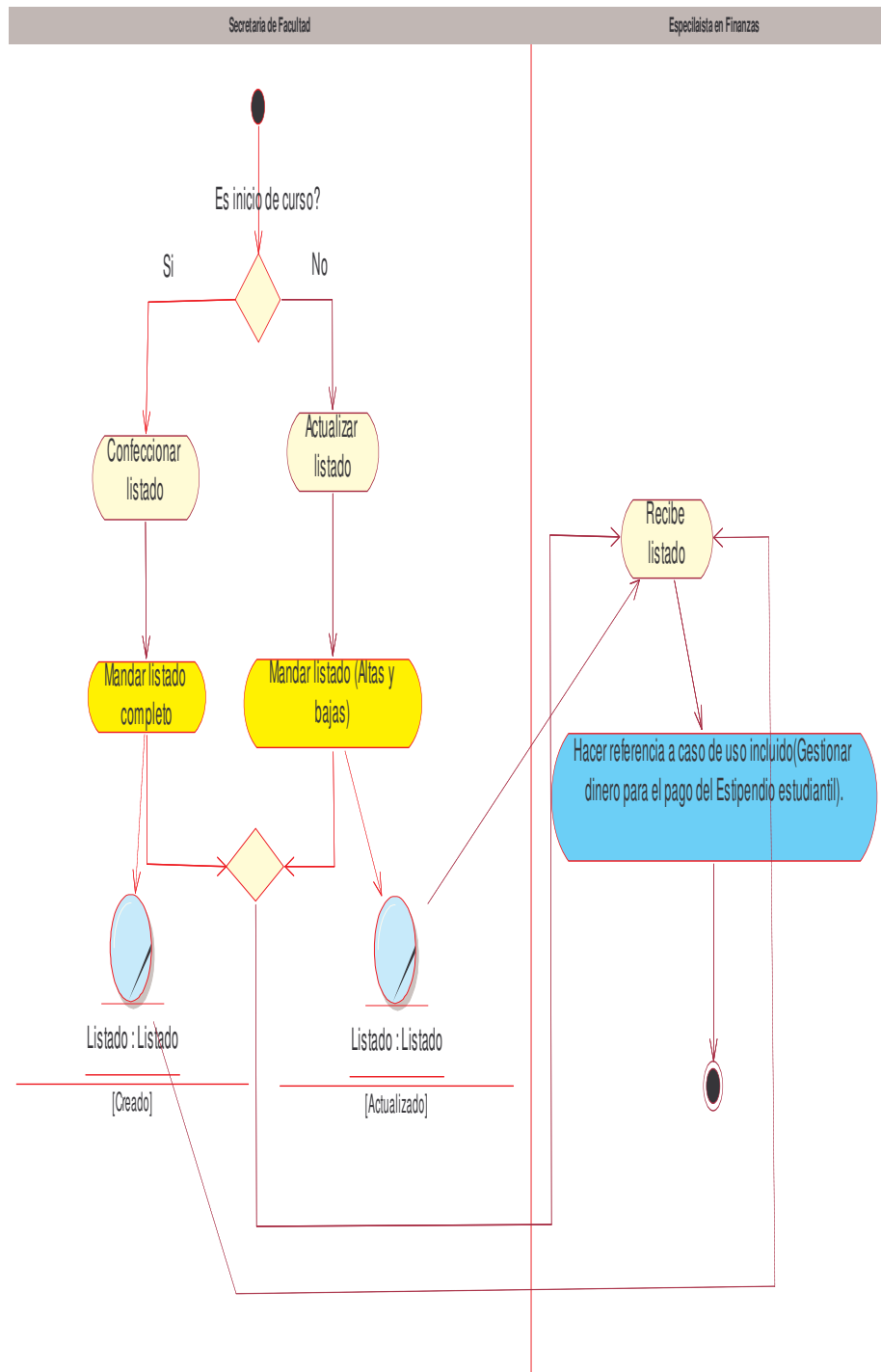


Figura 3. Diagrama de actividad Gestionar Estipendio Estudiantil.

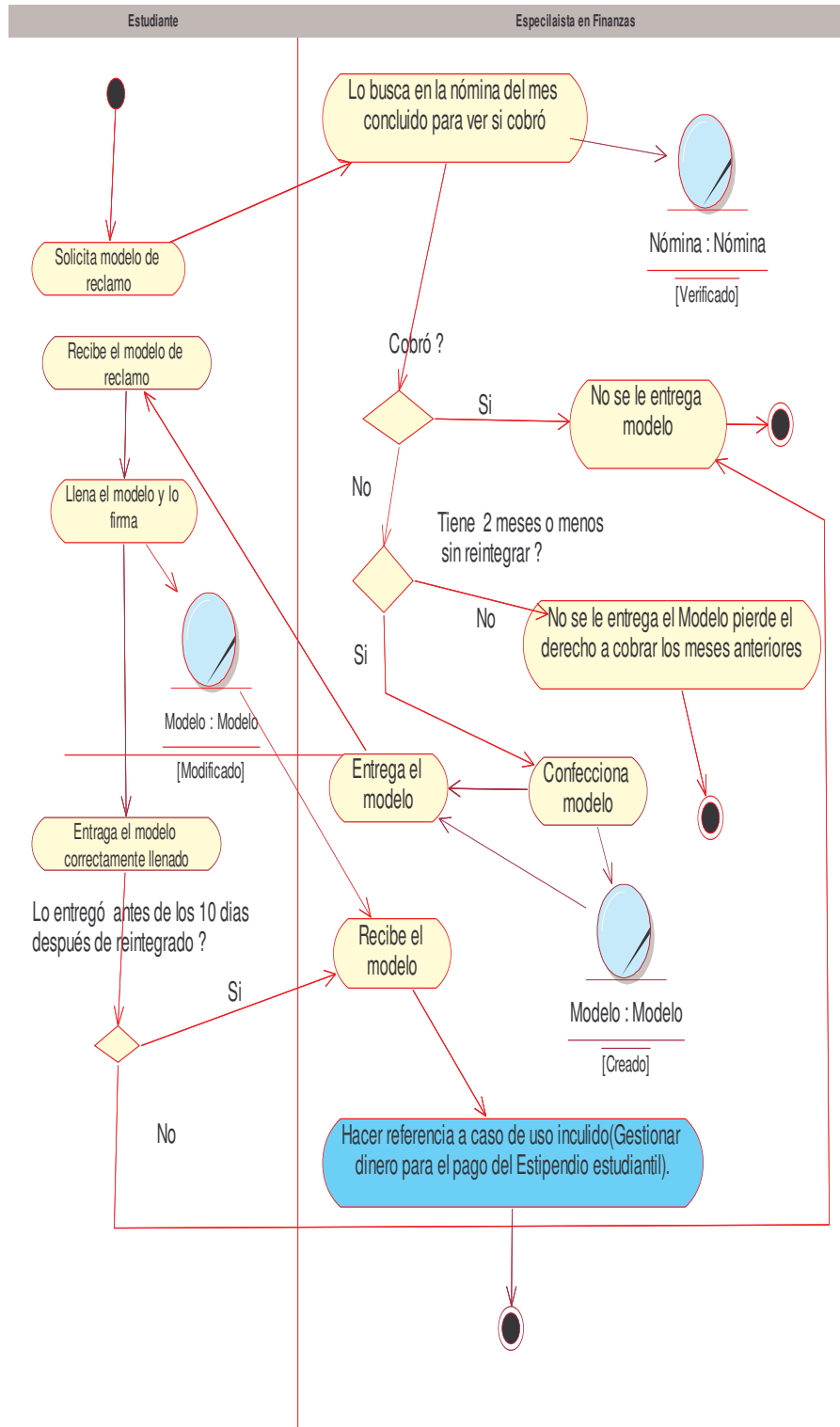


Figura 4. Diagrama de actividad Reclamar el pago de estipendio estudiantil no cobrado.

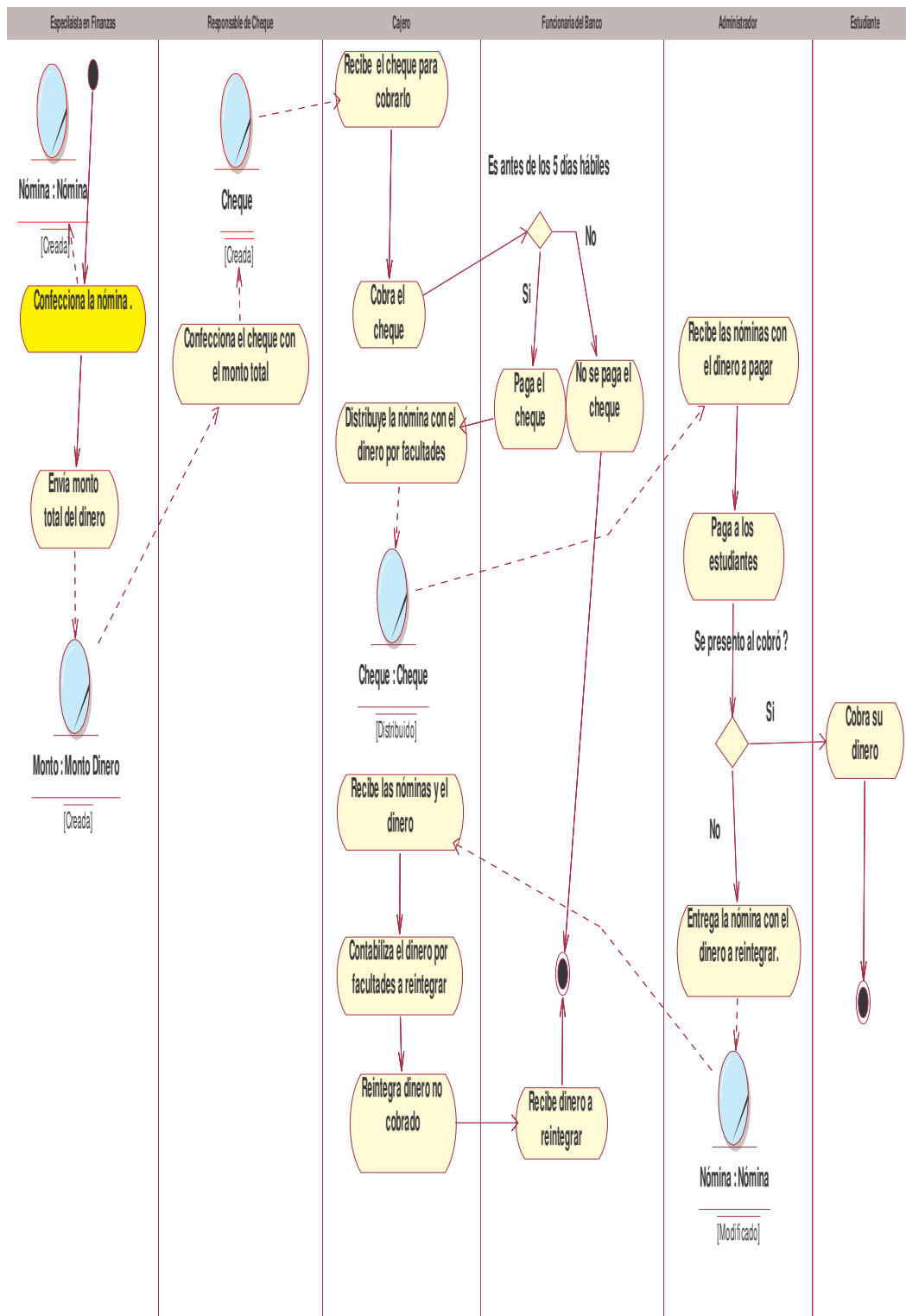
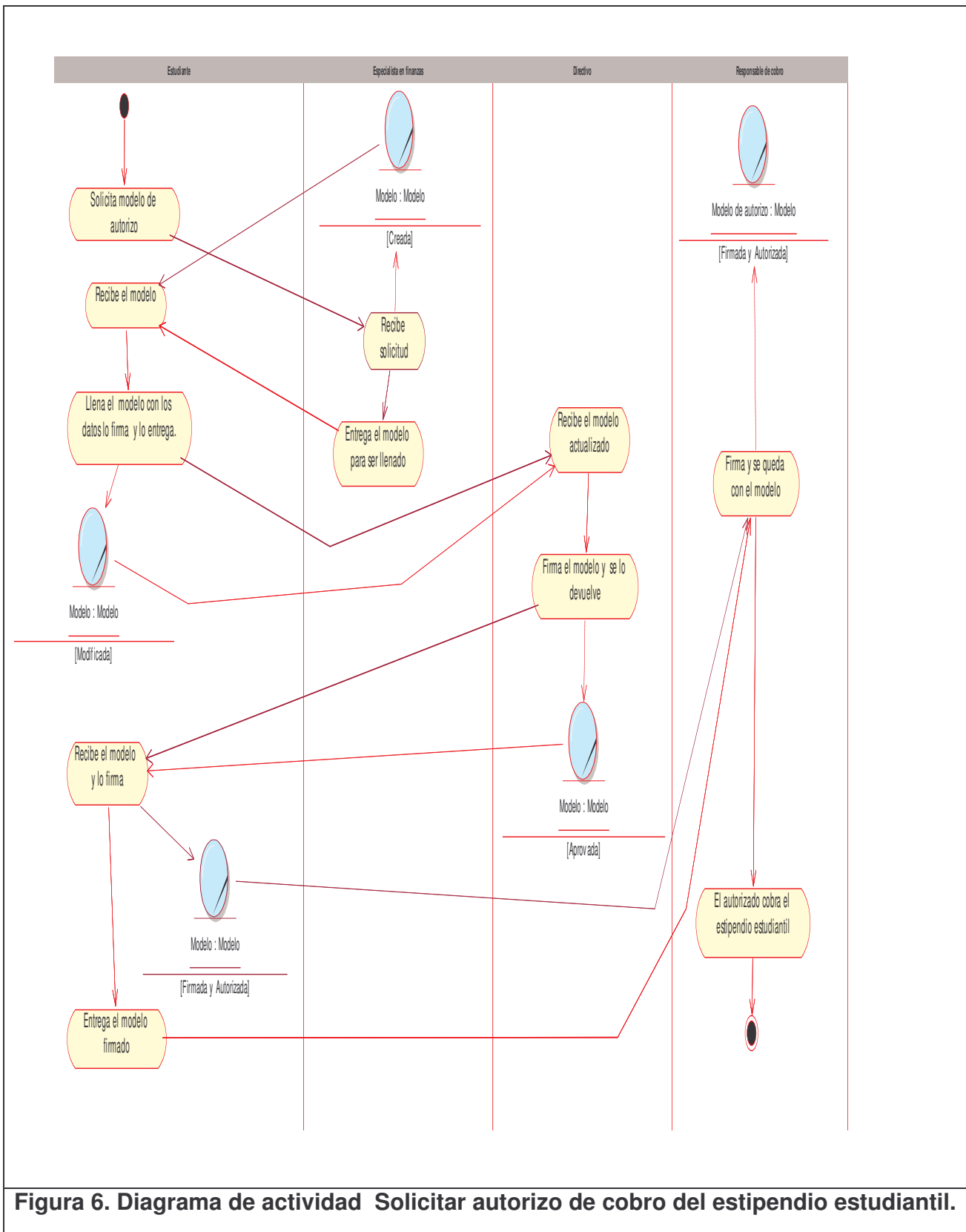


Figura 5. Diagrama de actividad Gestionar dinero para el pago del Estipendio estudiantil.



2.5 Modelo de objetos del negocio.

Un modelo de objetos del negocio es un modelo interno a un negocio. Describe como cada caso de uso del negocio es llevado a cabo por parte de un conjunto de trabajadores que utilizan un conjunto de entidades del negocio y unidades de trabajo. [29].

Una entidad del negocio representa algo, que los trabajadores toman, inspeccionan, manipulan, producen o utilizan en un caso de uso del negocio. El diagrama de clases del modelo de objeto, es un artefacto que se construye para describir el modelo de objetos del negocio. En la figura 7 se muestra el modelo de objetos del negocio estudiado.

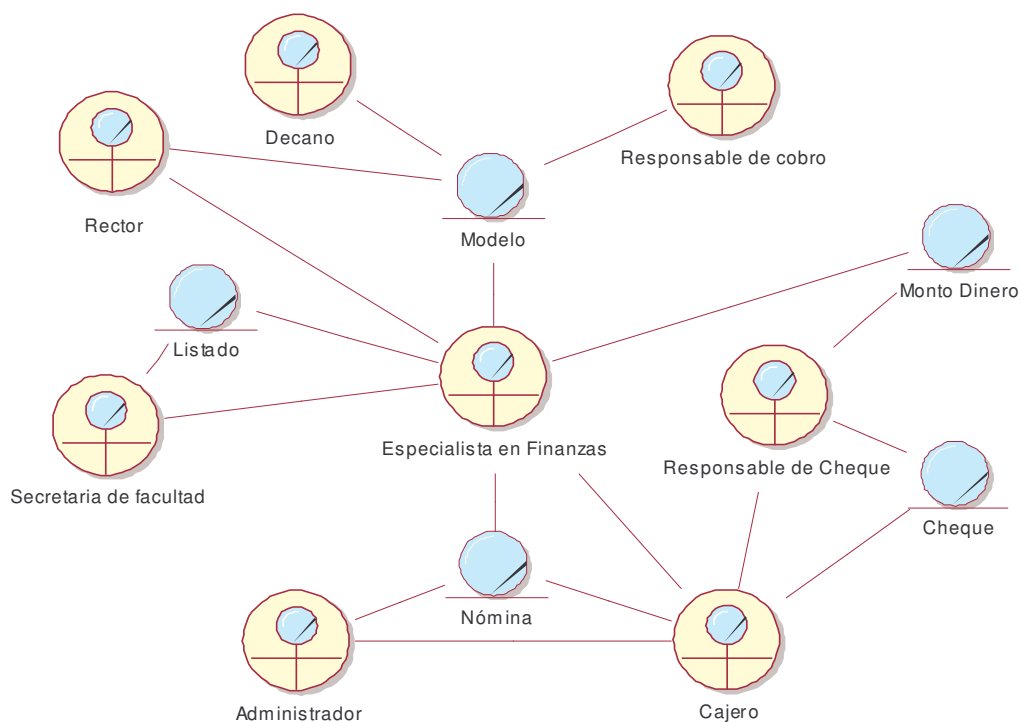


Figura 7. Diagramas de clases del modelo de objetos del negocio.

2.6 Descripción del sistema propuesto.

2.6.1 Concepción general del sistema.

El sistema propuesto dirigido a la gestión de la información relacionada al pago del estipendio estudiantil se implementará en tres módulos: un Módulo Administrativo, un Módulo para la Especialista en Finanzas y un Módulo para las Secretarías de cada Facultad.

Módulo Administrativo: dirigido al registro y acceso de los usuarios que operarán el sistema, así como el mantenimiento de la seguridad e integridad del mismo mediante la asignación de permisos a cada usuario al módulo que le corresponde. Brindará la posibilidad de cambiar la contraseña y habilitar e inhabilitar cada usuario.

Módulo de la Especialista en Finanzas: dirigido a la asignación de los tipos de pago (préstamos, reclamaciones y adeudos), a la creación de las nóminas para el pago del estipendio estudiantil y a la visualización del pago del estipendio a cada facultad, grupo y estudiante por los conceptos de (estipendio, orden 18, préstamos, ayudantías, reclamaciones y adeudos).

Módulo de la Secretaria de Facultad: dirigido al registro de cada estudiante, así como a la asignación, habilitación e inhabilitación de los tipos de pago (estipendio, orden 18 y ayudantía), así como a la visualización de reportes de pagos por conceptos mensualmente.

2.6.2 Requerimientos funcionales.

Los requerimientos funcionales permiten expresar una especificación más detallada de las responsabilidades del sistema que se propone. Ellos permiten determinar, de una manera clara, lo que debe hacer el mismo [30].

Los requerimientos funcionales del sistema propuestos son los siguientes:

1. Insertar un estudiante.
2. Modificar los datos de cada estudiante.
3. Buscar un estudiante.
4. Actualizar año y grupo a cada estudiante.
5. Dar alta, baja o graduar a cada estudiante.
6. Insertar un grupo académico.
7. Asignar un tipo de pago a cada estudiante.
8. Modificar un tipo de pago a cada estudiante.
9. Habilitar un tipo de pago a cada estudiante.
10. Inhabilitar un tipo de pago a cada estudiante.

11. Registrar reclamación(s) de cada estudiante.
12. Registrar estudiantes que no cobraron en el último pago efectuado.
13. Registrar adeudo(s) de cada estudiante.
14. Actualizar adeudo de cada estudiante.
15. Visualizar los tipos de pago de cada estudiante.
16. Visualizar las reclamaciones de cada grupo académico.
17. Visualizar las fechas reclamadas por estudiante.
18. Visualizar adeudo(s) de cada grupo académico.
19. Visualizar préstamos de cada grupo académico.
20. Visualizar dinero pagado a cada facultad por conceptos.
21. Total de dinero por reclamaciones.
22. Total de dinero por reclamaciones no cobradas.
23. Visualizar solicitud de fondos por conceptos..
24. Importe total de dinero por facultades.
25. Visualizar modelos de autorizo de cobro del estipendio.
26. Visualizar modelo de préstamo estudiantil.
27. Visualizar modelo de reclamación del estipendio.
28. Conformar nómina.
29. Insertar nómina.
30. Crear Usuario.
31. Iniciar sesión.
32. Cambiar Contraseña.
33. Habilitar Usuario.
34. Inhabilitar Usuario.
35. Asignar Permisos a Usuarios.

2.6.3 Requerimientos no funcionales.

Los requerimientos no funcionales especifican cualidades, propiedades del sistema; como restricciones del entorno o de la implementación, rendimiento, dependencias de la plataforma [31].

Los requerimientos no funcionales del sistema propuesto son los siguientes:

Apariencia o interfaz externa.

- El sistema debe tener una interfaz amigable, sencilla, agradable, legible y de fácil uso para el usuario. El contenido será mostrado de manera comprensible y fácil de leer.
- Deben evitarse imágenes y animaciones almacenadas en archivos de tamaño grande.
- Se debe informar al usuario dónde está y qué puede hacer desde allí, al proporcionar señales de navegación que conduzcan al usuario hasta el contenido que desea .
- Los mensajes de error deben ser reportados por la propia aplicación en la medida de las posibilidades y no por el Sistema Operativo.
- Los mensajes de las aplicaciones deben estar en español.
- La entrada de datos debe ser posible por varias vías, ya sea por el teclado o el mouse.

Requisitos de Usabilidad.

- El sistema podrá ser utilizado por cualquier persona, incluso con mínimos conocimientos en el manejo de la computadora y el ambiente Web en sentido general.
- El sistema será utilizado solo por personas registradas, estos serán la Especialista en Finanzas y Secretarias de cada facultad a las cuales se les asignan privilegios, es decir solo pueden trabajar con la información a la que tienen acceso.

Requisitos de Rendimientos.

- El sistema deberá ser rápido ante las solicitudes de los usuarios y en el procesamiento de la información. La eficiencia de la aplicación estará determinada en gran medida por el aprovechamiento de los recursos que se disponen en el modelo a tres capas, y la velocidad de las consultas a la base de datos. Se realizará la validación de los datos y la manipulación de eventos en el cliente y en el servidor aquellas que por cuestiones de seguridad, o de acceso a los datos lo requieran. Lográndose así un tiempo de respuesta más rápido, una mayor velocidad de procesamiento, y un mayor aprovechamiento de los recursos.

Requisitos de Soporte.

- Los servicios de instalación y mantenimiento del sistema será responsabilidad del administrador de la red del rectorado.

Requisitos de Seguridad.

- La información estará protegida contra accesos no autorizados utilizando mecanismos de autenticación y autorización que puedan garantizar el cumplimiento de esto: usuario y contraseña, de manera que cada uno tenga datos de acceso propios. Se usarán mecanismos de encriptación de los datos que por cuestiones de seguridad no deben viajar al servidor en texto claro, como es el caso de las contraseñas. Se guardará encriptada esta información en la base de datos utilizando para ello MD5 como algoritmo de encriptación.
- El sistema impondrá un estricto control de acceso que permitirá a cada usuario tener disponible solamente las opciones relacionadas con su actividad.
- El sistema no permitirá el acceso a informaciones a partir de puntos no autorizados.
- Las reglas de control de acceso deben ser aplicables a las bases de datos y a los sistemas que trabajan operativamente con los datos.
- No obstante los usuarios accederán de manera rápida y operativa al sistema sin que los requerimientos de seguridad se conviertan en un retardo para ellos.

Requisitos de Confiabilidad.

- El sistema debe ser tolerante ante los fallos; y las operaciones a realizar deben ser transaccionales.

Requisitos de Software.

- La aplicación debe poderse ejecutar en entornos *Windows*. Del lado del servidor se utilizará Apache como servidor web, del lado del cliente cualquiera de los exploradores existentes en el mercado.

Requisitos de Hardware.

- Se requiere de una máquina que funcione como servidor que los requerimientos específicos estarán en dependencia del sistema Gestor de Base de Datos a utilizar. Las computadoras clientes al menos deben cumplir los requisitos mínimos para poder ejecutar los navegadores de web.

Requisitos Políticos

- La aplicación debe cumplir con lineamientos, políticos y/o regulaciones del Ministerio de Educación Superior.

2.7 Modelo de casos de uso del sistema.

El modelo de casos de uso permite que los desarrolladores del software y los clientes lleguen a un acuerdo sobre los requisitos, es decir, sobre las condiciones y posibilidades que debe cumplir el sistema. Describe lo que hace el sistema para cada tipo de usuario [31].

2.7.1 Actores del modelo de sistema.

Un actor no es más que un conjunto de roles que los usuarios de Casos de Uso desempeñan cuando interaccionan con estos Casos de Uso. Los actores representan a terceros fuera del sistema que colaboran con el mismo. Una vez que hemos identificado los actores del sistema, tenemos identificado el entorno externo del sistema. [31].

Actor	Justificación
Secretaria de Facultad	* Es la encargada de gestionar toda la información de los estudiantes de la facultad.
Especialista en Finanzas	* Es el encargado de suministrar y registrar todos los datos necesarios para la confección de la nómina de pago de estipendio estudiantil.
Administrador	* Tiene control total sobre el sistema, es quien crea las cuentas de acceso al mismo y le asigna a cada usuario sus permisos en dependencia al rol a desarrollar.

Tabla 4. Descripción de los actores del sistema.

2.7.2 Paquetes y sus relaciones.

Dado el número de casos de uso y actores, se introducen paquetes en el modelo de casos de uso para tratar su tamaño, organizar los elementos en grupos y hacerlo más comprensibles.

En la figura 7 se muestra el diagrama de casos de uso por paquetes Paquete: Módulo Secretaria de Facultad, Paquete: Módulo Administrativo y Paquete: Módulo Especialista en Finanzas.

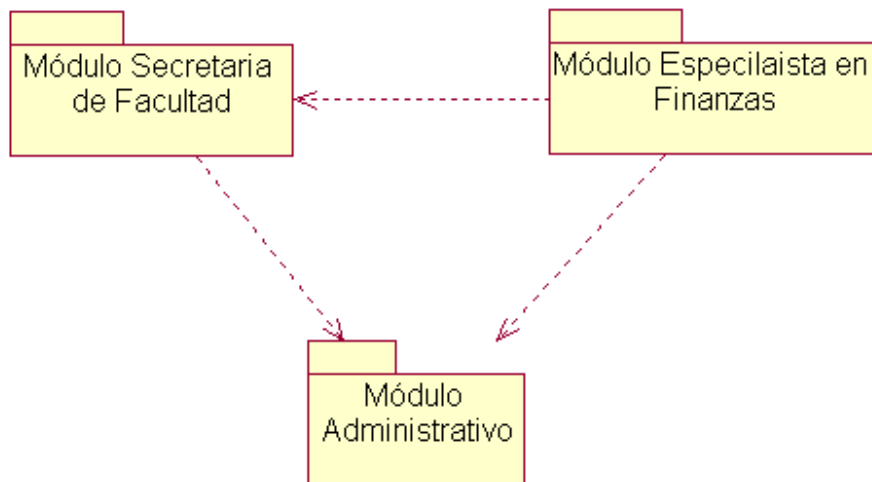


Figura 8. Diagrama de Casos de Usos por Paquetes.

2.7.3 Diagramas de casos de uso del sistema.

La forma en que interactúa cada actor del sistema con el sistema se representa con un Caso de Uso. Los Casos de Uso son “fragmentos” de funcionalidad que el sistema ofrece para aportar un resultado de valor para sus actores. De manera más precisa, un Caso de Uso especifica una secuencia de acciones que el sistema puede llevar a cabo interactuando con sus actores, incluyendo alternativas dentro de la secuencia [32].

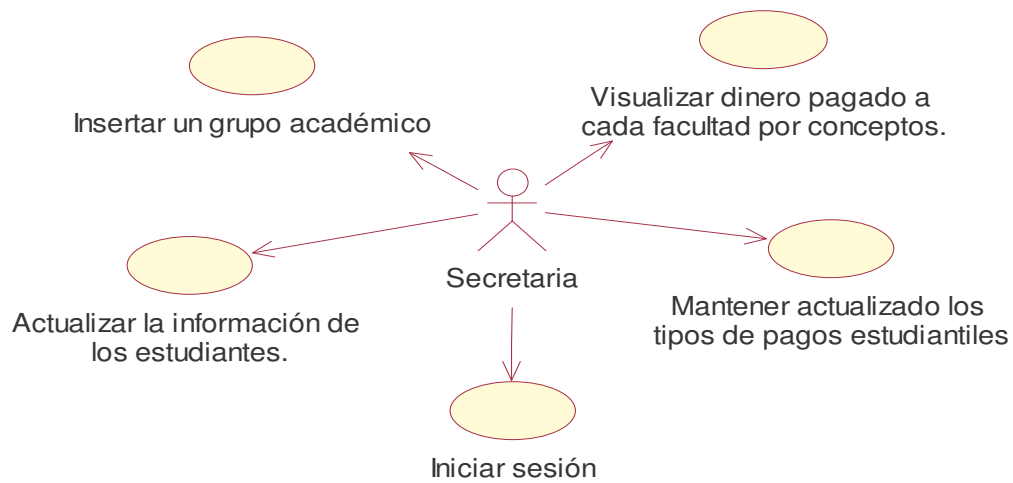


Figura 9. Diagrama de Casos de uso del Sistema: Paquete Módulo Secretaria de Facultad.

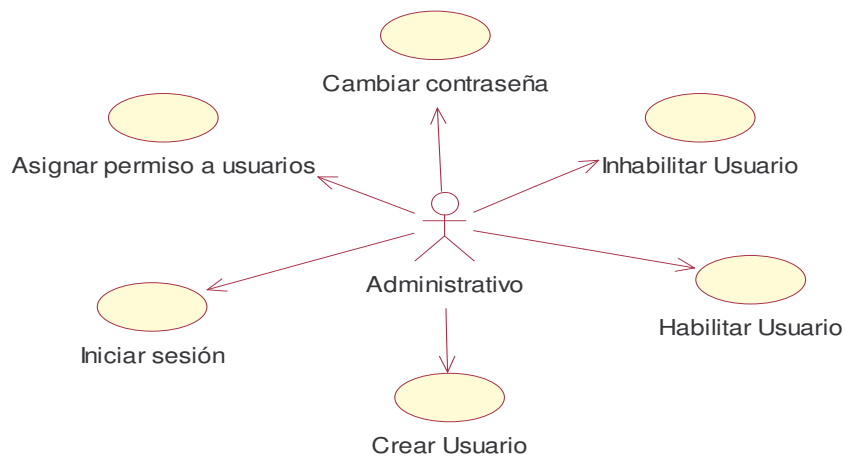


Figura 10. Diagrama de Casos de Uso del Sistema: Paquete Módulo Administrativo.

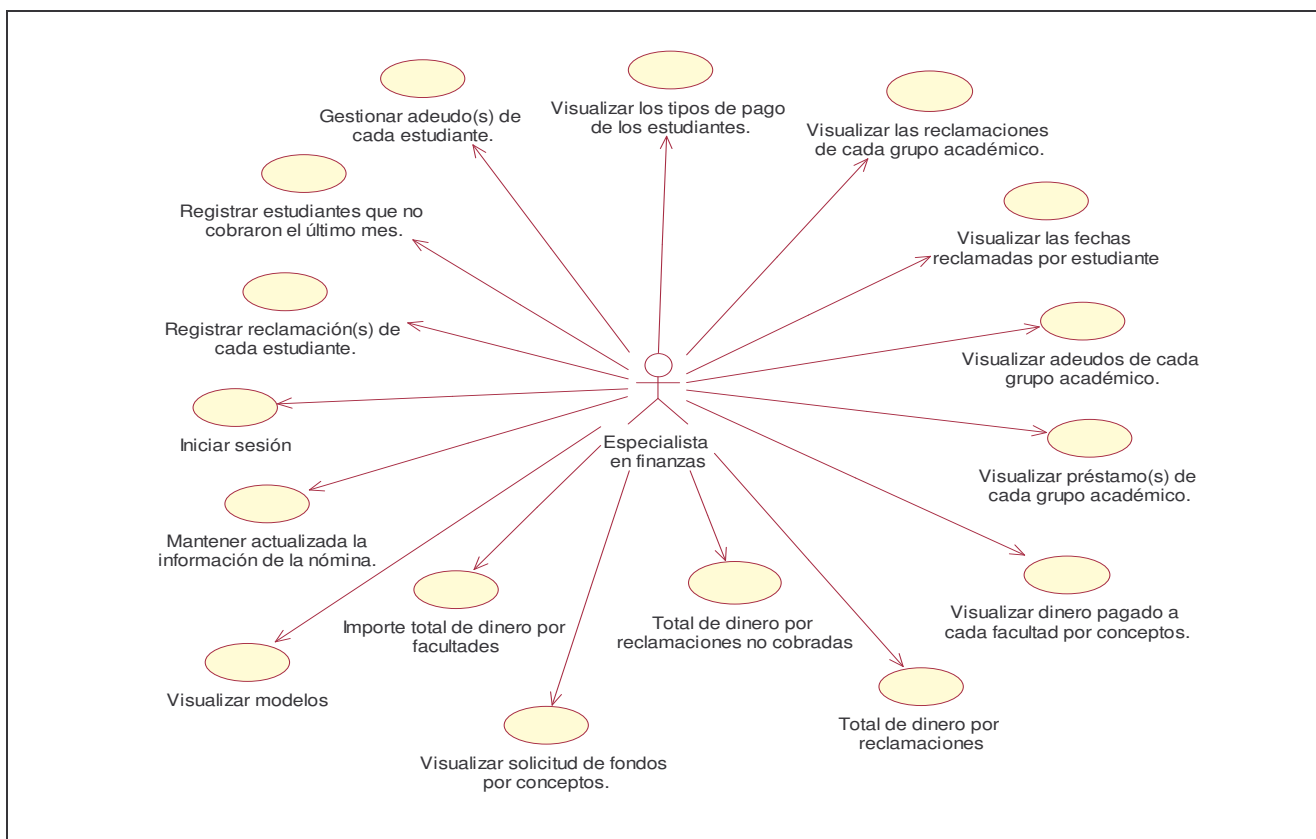


Figura 11. Diagrama de Casos de uso del Sistema: Paquete Módulo Especialista en Finanzas.

2.7.4 Descripción de los casos de uso del sistema.

CU-01	Actualizar la información de los estudiantes.
Actor: Secretaria de facultad (Inicia).	
Propósito: Mantener actualizada la información de cada estudiante para el correcto pago del estipendio estudiantil.	
Resumen: El caso de uso se inicia cuando la secretaria de cada facultad mediante el sistema necesite insertar o modificar a un estudiante de su facultad. Los datos a insertar o modificar son: el nombre, los dos apellidos, el carné, la nacionalidad, la vía de ingreso y el grupo, para su mayor facilidad se da la opción de buscar al estudiante por su nombre, carne y primer apellido. También se podrá dar alta, baja, graduar o cambiar de año y grupo académico a cada estudiante caso de uso termina cuando se obtiene la acción deseada.	
Precondiciones:	

Los estudiantes a modificar deben de estar en la base de datos. Para dar alta, baja o graduar un estudiante debe de tener un grupo académico asignado. Al buscar un estudiante el mismo debe de estar en la base de datos.
Referencias: R1, R2, R3,R4,R5.
Poscondiciones: Se actualiza la información de cada estudiante: Si acción insertar, se da alta al estudiante en la base de datos. Si acción modificar, se modifica los datos al estudiante. Si acción buscar se obtiene la información de cada estudiante. Si acción cambiar de grupo o año académico se cambiar de grupo a cada estudiante.
Prototipo: Ver Anexo B1, B2, B3, B4.

CU-02	Insertar un grupo académico.
Actor:	Secretaria de facultad (inicia).
Propósito:	Insertar un nuevo grupo académico.
Resumen:	El caso de uso se inicia cuando la secretaria de cada facultad mediante el sistema necesite insertar un grupo de su facultad. Para ello debe escoger el prefijo que le corresponde, la carrera y el año académico. El caso de uso termina cuando se obtiene la acción deseada.
Precondiciones:	Para insertar un grupo académico primero debe de estar en la base de datos el año al que ese grupo pertenece
Referencias:	R6.
Poscondiciones:	Si acción insertar el grupo académico queda insertado en la base de datos.
Prototipo:	Ver Anexo B5.

CU-03	Mantener actualizado los tipos de pagos estudiantiles.
Actores:	Secretaria de cada facultad o Especialista en Finanzas.
Propósito:	Mantener actualizado mensualmente los tipos de pago de cada estudiante.

Resumen: El caso de uso se inicia cuando se busca un estudiante determinado para habilitar, inhabilitar, actualizar un tipo de pago por los conceptos de: estipendio, orden 18 y ayudantía para el caso de la Secretaria de Facultad y para la Especialista en Finanzas solamente el préstamo estudiantil. El caso de uso termina cuando en el sistema queda actualizada toda la información.
Precondiciones: Para actualizar la información de cada estudiante deben existir los mismos en la base de datos.
Referencias: R7, R8, R9, R10, R3.
Poscondiciones: Información de cada estudiante debe quedar actualizada al inicio de cada mes.
Prototipo: Ver Anexo B6, B7, B8

CU-04	Registrar reclamación(s) de cada estudiante.
Actores:	Especialista en finanzas.
Propósito:	registrar las reclamaciones de cada estudiante en un mes determinado
Resumen:	El caso de uso se inicia cuando la Especialista en Finanzas registra en el sistema la reclamación de cada estudiante para ello debe de buscar primeramente al estudiante. Solo puede reclamar 3 veces al año y el mes último no cobrado. El caso de uso termina cuando en el sistema queda registrada la información.
Precondiciones:	Para registrar la reclamación de cada estudiante el sistema verificará que el mismo no cobró el mes que esta reclamando.
Referencias:	R11,R3.
Poscondiciones:	la reclamación de cada estudiante debe quedar actualizada en la base de datos para el pago del mes próximo.
Prototipo:	Ver Anexo B.9

CU-05	Registrar estudiantes que no cobraron el último mes.
Actores:	Especialista en finanzas.

Propósito: registrar los estudiantes que no cobraron en un pago efectuado.
Resumen: El caso de uso se inicia cuando la Especialista en finanzas busca en cada facultad, carrera, año y grupo académico cada estudiante para registrar en el sistema que no cobró en el una fecha determinada. El caso de uso termina cuando en el sistema queda registrada la información.
Precondiciones: El estudiante a registrar debe de estar en la base de datos.
Referencias: R12.
Poscondiciones: La actualización queda registrada en la base de datos
Prototipo: Ver Anexo B.10

CU-06	Gestionar adeudo(s) de cada estudiante.
Actores: Especialista en finanzas.	
Propósito: registrar adeudo(s) para cada estudiante en una fecha determinada.	
Resumen: El caso de uso se inicia cuando la Especialista en finanzas registra en el sistema a cada estudiante el adeudo que le corresponde, para ello debe buscar primero al estudiante que se le registrará el adeudo. El caso de uso termina cuando en el sistema queda almacenada la información.	
Precondiciones: Para registrar el adeudo de cada estudiante el estudiante debe de estar en la base de datos.	
Referencias: R13, R14,R3.	
Poscondiciones: Adeudo registrado para cada estudiante.	
Prototipo: Ver Anexo B.11, 12	

CU-07	Visualizar los tipos de pago de los estudiantes.
Actores: Especialista en finanzas.	
Propósito: Visualizar los tipos de pago de cada estudiant.	

Resumen: El caso de uso se inicia cuando la Especialista en finanzas busca un estudiante por nombre ,carné de identidad o primer apellido, además debe de introducir el tipo de pago que desea visualizar. El caso de uso termina cuando en el sistema le devuelve la información deseada.
Precondiciones: El estudiante debe tener al menos un tipo de pago asignado.
Referencias: R15, R3.
Poscondiciones:
Prototipo: Ver Anexo B.13

CU-08	Visualizar las reclamaciones de cada grupo académico.
Actores:	Especialista en finanzas.
Propósito:	Visualizar las reclamaciones de cada grupo académico.
Resumen:	El caso de uso se inicia cuando la Especialista en Finanzas busca para cada facultad, carrera, año y grupo académico las reclamaciones en un mes determinado de cada estudiante. El caso de uso termina cuando en el sistema le devuelve la información deseada.
Precondiciones:	Deben existir las facultades, carreras, años académicos y grupos en la base de datos.
Referencias:	R16.
Poscondiciones:	Estudiantes de cada grupo con reclamaciones.
Prototipo:	Ver Anexo B.14

CU-09	Visualizar las fechas reclamadas por estudiante.
Actores:	Especialista en finanzas (inicia).
Propósito:	Visualizar para cada estudiantes la fecha(s) que ha reclamado el estipendio durante el curso.
Resumen:	El caso de uso se inicia cuando la Especialista en finanzas desea visualizar las fechas reclamadas de cada estudiante, para ello debe de buscar al estudiante que desea visualizar las fechas reclamadas. El caso de uso termina cuando en el sistema le devuelve la

información deseada.
Precondiciones: El estudiante debe de estar en la base de datos
Referencias: R17, R3.
Poscondiciones: Fechas reclamadas por cada estudiante deseado.
Prototipo: Ver Anexo B.15

CU-10	Visualizar adeudo(s) de cada grupo académico.
Actores:	Especialista en finanzas.
Propósito:	Visualizar los adeudo(s) de cada grupo académico.
Resumen:	El caso de uso se inicia cuando la Especialista en Finanzas busca para cada facultad, carrera, año y grupo académico los estudiantes que tienen adeudo(s). El caso de uso termina cuando en el sistema le devuelve la información deseada.
Precondiciones:	Las facultades, carreras, años académicos y grupos así como los estudiantes deben de estar en la base de datos.
Referencias:	R18.
Poscondiciones:	Adeudo visualizado por cada grupo.
Prototipo:	Ver Anexo B.16

CU-11	Visualizar los préstamo(s) de cada grupo académico.
Actores:	Especialista en finanzas.
Propósito:	Visualizar las préstamos de cada grupo académico.
Resumen:	El caso de uso se inicia cuando la Especialista en finanzas busca para cada facultad, carrera, año y grupo académico los préstamos estudiantiles. El caso de uso termina cuando en el sistema le devuelve la información deseada.
Precondiciones:	Las facultades, carreras, años académicos y grupos así como los estudiantes deben de estar en la base de datos.
Referencias:	R19.
Poscondiciones:	Préstamos visualizados por grupos.
Prototipo:	Ver Anexo B.17

CU-12	Visualizar dinero pagado a cada facultad por conceptos.
Actores:	Especialista en finanzas.
Propósito:	Obtener total de dinero pagado a cada facultad por conceptos.
Resumen:	El caso de uso se inicia cuando la Especialista en Finanzas desea visualizar el total de dinero pagado en un mes determinado, para ello debe escoger cada facultad, el mes y los diferentes conceptos de: estipendio, orden 18, ayudantía, préstamos, extranjeros. El caso de uso termina cuando en el sistema le devuelve la información deseada.
Precondiciones:	La información obtenida debe de buscarse después de conformada la nómina.
Referencias:	R 20.
Poscondiciones:	Total del dinero visualizado por conceptos.
Prototipo:	Ver Anexo B.18

CU-13	Total de dinero por reclamaciones.
Actores:	Especialista en finanzas.
Propósito:	Obtener total de dinero de cada facultad por el concepto de reclamaciones.
Resumen:	El caso de uso se inicia cuando la Especialista en Finanzas necesita obtener de cada facultad, el total de dinero que la misma reclamó en un mes determinado. Para ello debe de especificar el mes y la facultad. El caso de uso termina cuando en el sistema le devuelve la información deseada.
Precondiciones:	La información obtenida debe de buscarse después de conformada la nómina.
Referencias:	R21.
Poscondiciones:	Total de dinero por reclamaciones visualizado.
Prototipo:	Ver Anexo B.19

CU-14	Total de dinero por reclamaciones no cobradas.
Actores: Especialista en finanzas.(inicia)	
Propósito: Obtener total de dinero de cada facultad por el concepto de reclamaciones no cobradas en un mes determinado.	
Resumen: El caso de uso se inicia cuando la Especialista en finanzas necesita obtener de cada facultad el total de dinero que la misma reclamó y no cobro en un mes determinado. Para ello debe de especificar el mes y la facultad. El caso de uso termina cuando en el sistema le devuelve la información deseada.	
Precondiciones: La información obtenida debe de buscarse después de conformada la nómina.	
Referencias: R22.	
Poscondiciones: Total de dinero por reclamaciones no cobradas visualizado.	
Prototipo: Ver Anexo B.20	

CU-15	Visualizar solicitud de fondos por conceptos.
Actores: Especialista en finanzas.	
Propósito: Solicitud de fondos por facultades por cada concepto.	
Resumen: El caso de uso se inicia cuando la Especialista en Finanzas solicita los fondos para el pago del estipendio estudiantil, estas solicitudes se hacen por cada concepto, mes y facultad. El caso de uso termina cuando en el sistema le devuelve la información deseada.	
Precondiciones: La información obtenida debe de buscarse después de conformada la nómina.	
Referencias: R23.	
Poscondiciones: Solicitud de fondos visualizada	
Prototipo: Ver Anexo B.21	

CU-16	Importe total de dinero por facultades.
Actores: Especialista en finanzas.	

Propósito: Obtener total de dinero pagado a cada facultad por conceptos en un mes dado
Resumen: El caso de uso se inicia cuando la Especialista en finanzas desea visualizar el total de dinero pagado en un mes determinado a cada facultad. El caso de uso termina cuando en el sistema le devuelve la información deseada.
Precondiciones: La información obtenida debe de buscarse después de conformada la nómina.
Referencias: R24.
Poscondiciones: Importe total visualizado.
Prototipo: Ver Anexo B.22

CU-17	Visualizar modelos.
Actores: Especialista en finanzas (inicia).	
Propósito: Que el sistema brinde la opción de visualizar los diferentes modelos de autorizo de cobro, para cada estudiante que lo solicite	
Resumen: El caso de uso se inicia cuando la especialista muestra al solicitante el modelo de autorizo de cobro, modelo de préstamo, reclamación.	
Precondiciones: El modelo debe estar confeccionado en el sistema	
Referencias: R25, R26, R27.	
Poscondiciones: Modelos visualizados.	
Prototipo: Ver Anexo B.23, 24, 25	

CU-18	Mantener actualizada la información de cada nómina.
Actores: Especialista en finanzas.	
Propósito: Confeccionar e insertar las nóminas de cada grupo para el pago del estipendio estudiantil.	
Resumen: El caso de uso se inicia cuando la Especialista en finanzas conforma e inserta las diferentes nóminas para el pago del estipendio estudiantil, La especialista busca para cada facultad, carrera , año y grupo académico y el sistema le devuelve los estudiantes de ese grupo	

con todos sus tipos de pagos, el total de dinero por cada tipo de pago y el total de dinero general para la nómina. La nómina de pago debe incluir los conceptos de estipendio, orden 18, ayudantía, retroactivo, adeudo y extranjeros para cada estudiante. El caso de uso termina cuando las nóminas para el pago del estipendio estudiantil quedan conformadas e insertadas en la base de datos.
Precondiciones: Cada facultad, carrera, año y grupo debe existir en la base de datos junto con los estudiantes y sus conceptos asignados
Referencias: R28, R29.
Poscondiciones: Nómina conformada y nómina insertada en la base de datos
Prototipo: Ver Anexo B.26

CU-19	Crear Usuario
Actores:	Administrador (inicia).
Propósito:	Añadir un nuevo usuario al sistema.
Resumen:	El caso de uso se inicia cuando el administrador busca al usuario por su carné de identidad para ver si el usuario está en el sistema. Si el usuario no está registrado en el sistema le introduce sus datos como son: nombre, apellidos, carné de identidad, sub-área a la que pertenece y le crea una su cuenta.
Precondiciones:	
Referencias:	R30
Poscondiciones:	Queda registrado el usuario y listo para asignarle los permisos.
Prototipo:	Ver Anexo B.27

CU-20	Iniciar Sesión
Actores:	Secretaria de facultad, Especialista en finanzas y administrador (inician).
Propósito:	Registrarse en el sistema.

Resumen:
El caso de uso se inicia cuando la secretaria de facultad o la especialista en finanzas o administrador desean acceder a información o servicios que se brindan a los usuarios que tienen determinados permisos, para esto debe introducir su cuenta y su contraseña. El sistema verifica en la base de datos si los datos introducidos son correctos y pasa a mostrar su perfil, culminando así el caso de uso.
Precondiciones: Para poder registrarse debe existir este usuario con esta contraseña.
Referencias: R31
Poscondiciones: El usuario puede acceder a toda la información a la que tiene permisos.
Prototipo: Ver Anexo B.28

CU-21	Cambiar Contraseña
Actores:	Administrador (inicia).
Propósito:	Permitir que la persona registrado cambie su contraseña
Resumen:	El caso de uso se inicia cuando un usuario registrado desea cambiar su contraseña, el sistema le da la posibilidad de poner su cuenta, la contraseña antigua y la nueva, si los datos son correctos la contraseña queda cambiada, de lo contrario se muestra un mensaje de error, finalizando así el caso de uso.
Precondiciones:	debe de existir el usuario en la base de datos.
Referencias:	32
Poscondiciones:	Queda actualizada la contraseña del usuario
Prototipo:	Ver Anexo B.29

CU-22	Habilitar Usuario
Actores:	Administrador (inicia).
Propósito:	Habilitar la cuenta de un usuario del sistema.
Resumen:	El caso de uso se inicia cuando el administrador desea activar la cuenta de un usuario para el uso del sistema.

Precondiciones: El usuario a habilitar debe estar previamente registrado como usuario del sistema.
Referencias: 33
Poscondiciones: Queda activada la cuenta del usuario.
Prototipo: Ver Anexo B.30

CU-23	Inhabilitar Usuario
Actores: Administrador (inicia).	
Propósito: Inhabilitar la cuenta de un usuario del sistema.	
Resumen: El caso de uso se inicia cuando el administrador desea desactivar la cuenta del usuario para el uso del sistema.	
Precondiciones: El usuario a inhabilitar debe estar previamente registrado como usuario del sistema.	
Referencias: 34.	
Poscondiciones: Queda desactivada la cuenta del usuario.	
Prototipo: Ver Anexo B.30	

CU-24	Asignar Permisos a Usuarios
Actores: Administrador (inicia).	
Propósito: Dar a los usuario permisos en los módulos y áreas correspondientes..	
Resumen: El caso de uso se inicia cuando el administrador desea asignar permisos a los usuarios del sistema, se muestran los módulos y los permisos posibles en cada módulo para seleccionarlos.	
Precondiciones: El usuario al que se le van a asignar los permisos debe estar previamente registrado como usuario del sistema.	
Referencias: 35.	
Poscondiciones: Quedan asignados los permisos del usuario.	
Prototipo: Ver Anexo B.30	

2.8 Conclusiones.

En este capítulo fueron descritos los procesos que tiene lugar en el Departamento de Finanzas de la Universidad de Cienfuegos, identificando a su vez los roles y objetos del negocio, así como su relación en esos procesos. Esta descripción fue realizada mediante el modelo del negocio, para lo cual se elaboraron los modelos de casos de uso y de actividad. Se logró una mejor comprensión del negocio que se trata, dando paso al modelado del sistema. Buscando establecer un común entendimiento entre usuario y el equipo de desarrollo, fueron definidos las principales funcionalidades del sistema y los requisitos adicionales que debe cumplir. Se identificaron, además, los tipos de usuarios que tendrá la aplicación y su comportamiento con cada uno de ellos, o sea, lo que debe hacer el sistema para cada uno de sus actores; obteniéndose el modelo de casos de uso del sistema.

Capítulo 3. Construcción de la solución propuesta.

3.1 Introducción.

En este capítulo se presenta el diseño de clases propuesto para desarrollar el sistema y los diagramas necesarios para la implementación de la solución propuesta.

3.2 Diseño de la base de datos.

Por la importancia de los datos manejados en los módulos implementados es necesario lograr un buen diseño de la información almacenada.

En este epígrafe se muestra el diseño de la base de datos del sistema propuesto a través de los diagramas de clases persistente y el esquema de la base de datos generados a partir de este, con el modelo de datos.

3.2.1 Modelo lógico de datos.

El diagrama de clases persistentes muestra todas las clases capaces de mantener su valor en el espacio y en el tiempo es mostrado en el anexo **C.1**

3.2.2 Modelo físico de datos.

El modelo de datos que muestra la estructura física de las tablas de la base de datos, obtenido a partir del diagrama de clases persistentes, es mostrado en el anexo **C.2**

3.3 Diagrama de implementación.

El modelo de implementación describe cómo los elementos del modelo de diseño se implementan en términos de componentes. Describe también cómo se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en el lenguaje o lenguajes de programación utilizados y cómo dependen los componentes unos de otros [31].

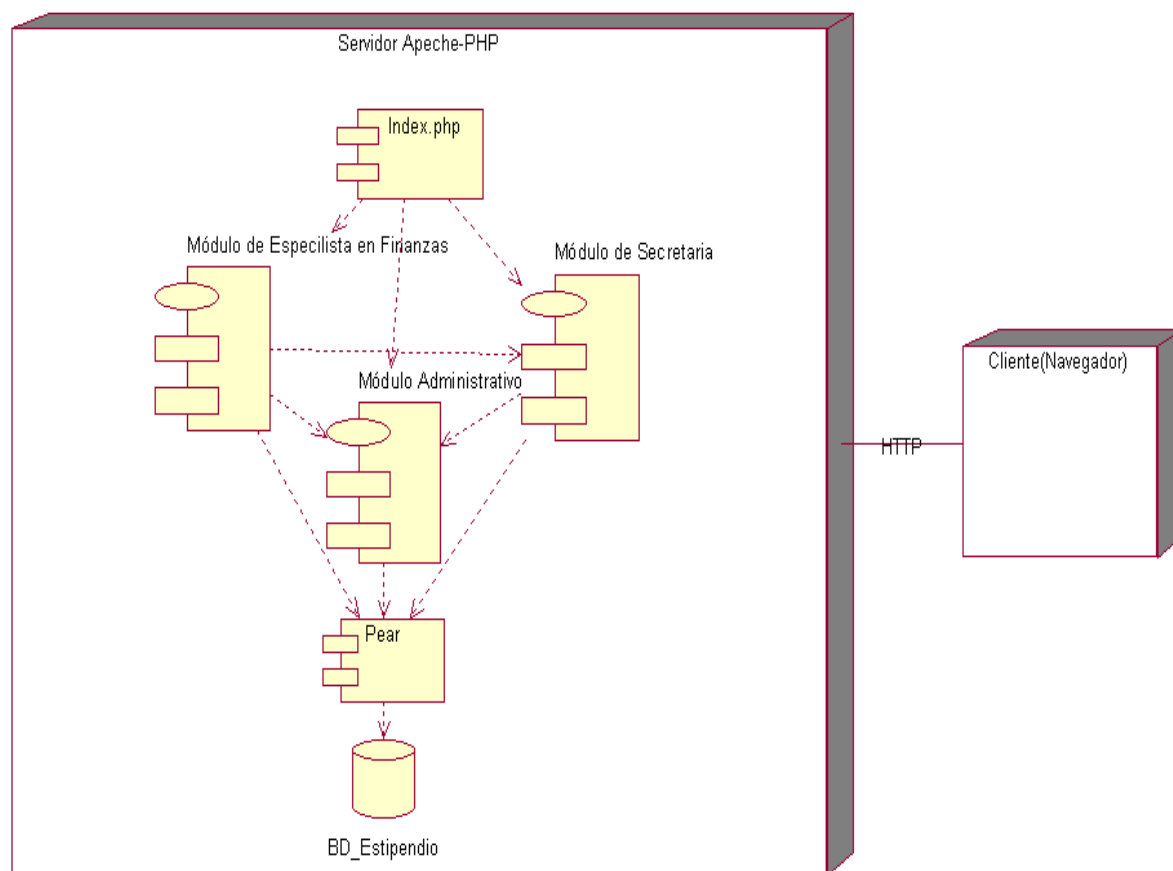


Fig. 12- Diagrama de Implementación.

3.4 Principios de Diseño del Sistema.

El **diseño de sistemas** se define como el proceso de aplicar ciertas técnicas y principios con el propósito de definir un dispositivo, un proceso o un sistema, con suficientes detalles como para permitir su interpretación y realización física.

3.4.1 Estándares en la Interfaz de la Aplicación.

La primera impresión del usuario cuando visita una aplicación web la brinda el diseño de la interfaz. Es por ello que, para lograr la apariencia adecuada y que el usuario se sienta confortable, se tienen en cuenta varios aspectos, sobre todo relacionados con tipografía, colores, gráficos, navegación, composición del sitio, etc, que a continuación se detallan. En el sistema, el diseño de la interfaz está basado en páginas Web, se utilizan las tonalidades de verdes por ser suaves y refrescantes en concordancia con los colores del logo de la

Capítulo 3: Construcción de la solución propuesta

universidad y facultad. El vocabulario manejado es lo menos técnico posible, acercándose al utilizado por los usuarios.

Se utilizan imágenes identificativas como vínculos para la navegación dentro del sitio web. La letra utilizada en todo el sistema es Times New Roman y Verdana (12, 16) lográndose un diseño estándar en todo el sitio. Los mensajes de error son pequeños y en Español. Se utilizan pequeños íconos para una mayor comprensión de las acciones, aunque se seleccionaron imágenes consecuentes con el significado que se quiere transmitir, cada imagen tiene asociado su ayuda rápida con el nombre de la acción que realiza.

El fondo de las páginas es de color blanco para mayor frescura de la vista. Todo esto se ha hecho con el objetivo de que el uso del sitio brinde comodidad y confort al usuario.

3.4.2 Tratamiento de Excepciones.

El diseño de la interfaz ha estado dirigido a evitar errores, teniendo en cuenta paralelamente la creación de interfaces útiles y amigables. Se ha buscado simplificar la validación de los datos garantizando una validación intrínseca de los mismos, procurando facilitar la corrección de errores lógicos tanto en la introducción de la información como en cualquier otro momento del tratamiento de la misma.

La técnica para el manejo de los errores en el sistema se concebirá de manera que cuando ocurra un error se genere una excepción; es decir, la ejecución normal se detenga y se transfiera el control a la zona de tratamiento de excepciones. Las excepciones internas se generan automáticamente por el sistema.

Los mensajes de error que emita el sistema ya sea de la base de datos o de la aplicación cliente se captarán y se traducirán a un lenguaje comprensible para el usuario.

Los formularios manejan los datos en memoria y sólo se actualiza en la base de datos cuando se indique salvarlos.

3.4.3 Formatos de Reportes.

Los reportes presentan un diseño similar a los reportes que se llevan en el área de finanzas del departamento de economía de forma manual pero a través de la web con el objetivo de que los cambios para quienes operan con los reportes sean mínimos.

3.4.4 Concepción General de la Ayuda.

Capítulo 3: Construcción de la solución propuesta

La ayuda constituye una parte imprescindible en todo sistema. En el menú principal aparece una opción *Ayuda* que explicara de forma detallada como funciona el sistema, tratando de aclarar los puntos que podría causar duda al usuario. Este menú aparece en todas las páginas para mayor comodidad del usuario.

Cada una de las opciones del sistema, así como las consideraciones que se asumen en la ejecución de ellas están propiamente documentadas para evitar cualquier tipo de confusión por parte del usuario. Cada aspecto de la ayuda ha sido diseñado con el objetivo de expresar explícitamente cómo y en qué orden debe operar el usuario. Al inicio de cada módulo se muestra una explicación del objetivo del módulo en sentido general.

3.5 Modelo de Clases Web.

Un diagrama de clases Web representa las colaboraciones que ocurren entre las páginas, donde cada página lógica puede ser representada como una clase. Al tratar de utilizar el diagrama de clases tradicional para modelar aplicaciones Web surgen varios problemas, por lo cual los especialistas del Rational plantearon la creación de una extensión al modelo de análisis y diseño que permitiera representar el nivel de abstracción adecuado y la relación con los restantes artefactos de UML.

Casos de Uso	Diagramas Clases Web
Iniciar sesión	Anexo A.1
Asignar Permisos a Usuarios	Anexo A.2
Cambiar Contraseña	Anexo A.3
Crear Usuario	Anexo A.4
Habilitar Usuario	Anexo A.5
Inhabilitar Usuario	Anexo A.6

Tabla 5. Diagrama de Clases Web. Paquete Módulo Administrativo.

Casos de Uso	Diagramas Clases Web
---------------------	-----------------------------

Capítulo 3: Construcción de la solución propuesta

Iniciar sesión	Anexo A.1
Mantener actualizado los tipos de pagos estudiantiles	Anexo A.13 y 14
Insertar un grupo académico	Anexo A.12
Actualizar la información de los estudiante	Anexo A.7, 8, 9, 10, 11

Tabla 6. Diagrama de Clases Web. Paquete Módulo Secretaria Docente.

Casos de Uso	Diagramas Clases Web
Registrar reclamación(s) de cada estudiante.	Anexo A.15
Registrar estudiantes que no cobraron el último mes.	Anexo A.17
Gestionar adeudo(s) de cada estudiante.	Anexo A.16 y 18
Visualizar los tipos de pago de los estudiantes.	Anexo A.19
Visualizar las reclamaciones de cada grupo académico.	Anexo A.20
Visualizar las fechas reclamadas por estudiante.	Anexo A.21
Visualizar adeudo(s) de cada grupo académico.	Anexo A.22
Visualizar los préstamo(s) de cada grupo académico.	Anexo A.24
Visualizar dinero pagado a cada facultad por conceptos.	Anexo A.23
Total de dinero por reclamaciones.	Anexo A.25
Total de dinero por reclamaciones no cobradas.	Anexo A.25
Visualizar solicitud de fondos por conceptos.	Anexo A.26
Importe total de dinero por facultades.	Anexo A.27
Visualizar modelos.	Anexo A.28
Mantener actualizada la información de cada nómina.	Anexo A.29

Tabla 7. Diagrama de Clases Web. Paquete Módulo Especialista en Finanzas.

3.6 Conclusiones.

En el presente capítulo se desarrollaron los diagramas de clases de la aplicación, el diseño de la base de datos y el modelo de implementación del sistema. Se describieron, además,

Capítulo 3: Construcción de la solución propuesta

los principios de diseño seguidos, específicamente, los temas de estándares de la interfaz, concepción del tratamiento de errores, y sistema de ayuda. Se muestran aquí los resultados de las etapas de diseño e implementación del sistema propuesto.

Capítulo 4. Estudio de Factibilidad.

4.1 Introducción.

En el presente capítulo se aborda el tema relativo al estudio de la factibilidad del producto, se ofrece una descripción de la planificación del proyecto, así como los costos asociados al mismo, los beneficios tangibles e intangibles que reportaría su elaboración y finalmente el análisis entre los costos y los beneficios para concluir si es o no factible el desarrollo del sistema.

Es necesario para la realización de un proyecto estimar el esfuerzo humano, el tiempo de desarrollo que se requiere para la ejecución del mismo y también su costo. Estas estimaciones pueden realizarse a través del método de puntos de función del modelo de COCOMO II.

4.2 Planificación por puntos de función.

Se utilizó para el cálculo de la estimación del esfuerzo, el tiempo de desarrollo y el costo del proyecto el método de puntos de características.

Para realizar el cálculo de los costos de desarrollo del sistema se deben obtener primero las instrucciones fuentes. Analizándose para esto las cantidades de entradas, salidas, peticiones, archivos lógicos e interfases externas preliminares que tiene el sistema. Para calcular la cantidad de instrucciones fuentes hay que tener en cuenta también que la conversión al *PHP*, *MySQL* y *Java Script* herramientas seleccionadas para implementar este software, es de 44, 40, 50 puntos respectivamente.

Después de este estudio se llegó a los siguientes resultados:

Nombre de la entrada externa	Cant de ficheros	Cant de Elem de datos	Clasificación(Bajo, Medio y Alto)
Iniciar sesión	1	2	Baja
Insertar datos a un estudiante	2	9	Media
Insertar Grupo	1	3	Baja
Insertar un nuevo usuario	3	10	Alta
Insertar un préstamo a un estudiante	1	5	Baja

Insertar reclamos a un estudiante	1	5	Baja
Insertar tipos de pagos a un estudiante	1	5	Baja
Insertar nóminas	1	12	Baja
Insertar estudiantes en cada nómina	4	17	Alta
Insertar adeudos de cada estudiante	1	6	Baja
Modificar estudiantes que no cobraron	1	3	Baja
Modificar adeudo de cada estudiante	1	4	Baja
Modificar los datos de un estudiante	2	9	Media
Modificar año y grupo a cada estudiante	1	2	Baja
Modificar tipos de pagos a un estudiante	1	5	Baja
Modificar permisos a un usuario	2	4	Baja
Modificar contraseña a un usuario	1	1	Baja
Modificar el estado de un estudiante	1	1	Baja
Eliminar permisos a un usuario	2	5	Media

Tabla 8. Planificación: Entradas externas

Nombre de la salida externa	Cant de ficheros	Cant de Elem de datos	Clasificación
Mostrar permisos de cada usuario	2	8	Media
Mostrar estado de cada	4	8	Alta

estudiante			
Mostrar préstamos por grupos	3	9	Media
Mostrar reclamos por grupos	3	8	Media
Mostrar adeudos por grupos	4	11	Alta
Mostrar reclamos por cada estudiante	3	6	Media
Mostrar total de dinero por conceptos	3	6	Media
Mostrar total de dinero por reclamaciones en una fecha dada	4	6	Alta
Mostrar total de dinero por reclamaciones no cobradas en una fecha dada	4	9	Alta
Mostrar listados mensuales de los estudiantes que cobrarán	4	13	Alta
Mostrar solicitud de fondos de cada facultad	4	7	Alta
Mostrar nóminas por grupos	6	16	Alta
Mostrar listados de los estudiantes por grupos	3	7	Media
Mostrar una facultad	1	1	Baja
Mostrar una carrera	2	1	Baja
Mostrar un año	1	1	Baja
Mostrar un grupo	3	1	Media

Tabla 9. Planificación: Salidas externas.

Nombre de la petición	Cantidad de ficheros	Cantidad de Elementos de datos	Clasificación(Bajo, Medio y Alto)
Obtener un estudiante	4	8	Alta

Obtener un operador	4	7	Alta
Obtener una fecha	2	1	Baja

Tabla 10. Planificación: Peticiones.

Nombre del fichero interno	Cantidad de records	Cantidad de Elementos de datos	Clasificación(Bajo, Medio y Alto)
Persona	1	5	Baja
Facultad	1	2	Baja
Carrera	1	3	Baja
Grupo	1	4	Baja
cod_nacionalidad	1	2	Baja
cod_viaingreso	1	2	Baja
Estudiante	1	5	Baja
Adeudos	1	7	Baja
Área	1	2	Baja
DetalleNomina	1	13	Baja
Módulos	1	3	Baja
Tipo_Pago	1	5	Baja
NominillaPago	1	7	Baja
Operador	1	4	Baja
operadorxmodulos	1	3	Baja
pagoxestudiante	1	5	Baja
Reclamaciones	1	7	Baja
Sesiones	1	5	Baja
sub_areas	1	3	Baja
Trabajador	1	2	Baja

Tabla 11. Planificación: Ficheros internos.

Elementos	Bajos	X Peso	Medios	X Peso	Altos	X Peso	Subtotal de puntos de función
Ficheros lógicos internos	20	7	0	10	0	15	140
Ficheros de interfaces externas	0	5	0	7	0	10	0
Entradas externas	18	3	3	4	2	6	78
Salidas externas	0	4	6	5	7	7	79
Peticiones	1	3	0	4	2	6	15
Total							312

Tabla 12. Planificación: Punto de función.

Características			Valor
Puntos de función desajustados			312
Lenguaje	PHP	MySQL	Java Script
Instrucciones fuentes por puntos de función	44	40	50
Por ciento de la aplicación en cuanto a requerimientos funcionales	50%	45%	5%
Instrucciones fuentes	6820	5580	780
Total de Instrucciones fuentes	13180		

Tabla 13. Planificación: Miles de instrucciones fuentes.

Miles de instrucciones fuentes (MF) = 13.180

4.3 Determinación de los costos.

Determinación de los valores de las variables de costos utilizadas en el cálculo de costos en la producción de software, como el cálculo del esfuerzo, el tiempo de desarrollo, la cantidad de hombres y el costo total del sistema.

Cálculo de:	Valor	Justificación
RCPX	1,00	BD moderada, no se requiere de amplia documentación. La aplicación Web tienen una moderada complejidad. (Nominal)
RUSE	1,00	Se implementa código reusable para el aprovechamiento de este en toda la aplicación. (Nominal)
PDIF	1,00	No tiene grandes restricciones en cuanto al tiempo de ejecución ya que el software podrá estar trabajando varias horas. EL Software no tiene limitación de memoria impuesta. La plataforma de aplicación tiene gran estabilidad. (Nominal)
PERS	0,83	Hay poco movimiento del personal. (Alto)
PREX	0,87	El equipo tiene buen dominio y posee conocimiento del lenguaje de programación. Con una experiencia de aproximadamente un años. (Alto)
FCIL	0,87	Se utilizan herramientas de programación como: Macromedia Dreamweaver 8, Rational Rose para la documentación, empleando como notación UML. (Alto)
SCED	1,00	La planificación se hace con moderada frecuencia. (Nominal)
PREC	3,72	El equipo de desarrollo posee una comprensión considerable de los objetivos del producto, no tiene experiencia en la realización de software de este

		tipo. (Nominal)
FLEX	3,04	El sistema cuenta con alguna flexibilidad en relación con las especificaciones de los requerimientos preestablecidos y a las especificaciones de interfaz externa. (Nominal)
TEAM	1,10	El equipo que va a desarrollar el software es altamente cooperativo.
RESL	4,24	Teniendo en cuenta la alta experiencia que existe en el país acerca de este tipo de estudios existen algunos factores de riesgo. (Nominal)
PMAT	6,24	Nivel I Alto porque se encuentra en su primera etapa un poco avanzada. (Bajo)

Tabla 14. Costos: Factores de escalas.

Multiplicador de esfuerzos

$$EM = \prod_{i=1} E_{mi} = RCPX * RUSE * PDIF * PERS * PREX * FCIL * SCED$$

$$EM = 1 * 1 * 1 * 0.83 * 0.87 * 0.87 * 1 = \mathbf{0.6282}$$

Factores de escala

$$SF = \sum SFi = PREC + FLEX + RESL + TEAM + PMAT$$

$$SF = 3.72 + 3.04 + 4.24 + 1.10 + 6.24 = \mathbf{18.34}$$

Valores de los coeficientes

$$A = 2,94; B = 0,91; C = 3,67; D = 0,24$$

$$E = B + 0,01 * SF$$

$$E = 0,91 + 0,01 * 18,34$$

$$E = \mathbf{1,0934}$$

$$F = D + 0,2 * (E - B)$$

$$F = 0,24 + 0,2 * (1,0934 - 0,91)$$

$$F = \mathbf{0,27668}$$

Esfuerzo

$$PM = A * (MF)^E * EM$$

$$PM = 2.94 * (13.180)^{1.0934} * 0.6282 = \mathbf{30.9713}$$

Cálculo del tiempo de desarrollo

$$TDEV = C * PM^F = 3.67 * (30.9713)^{0.27668} = \mathbf{9.4882 \text{ meses}}$$

Cálculo de la cantidad de hombres

$$CH = PM / TDEV$$

$$CH = 30.9713 / 9.4882 = \mathbf{3.2641 \approx 3 \text{ hombres}}$$

Costo

$$CHM = 3 * \text{Salario X Persona}$$

$$CHM = 3 * 225$$

$$CHM = \mathbf{\$ 675 / Mes}$$

$$\text{Costo} = CHM * PM$$

$$\text{Costo} = \$ 675 * 30.9713$$

$$\text{Costo} = \mathbf{\$ 20905.6275}$$

Los costos en los que se incurriría de desarrollarse el sistema serían:

Cálculo de:	Valor
Esfuerzo(PM)	30.9713
Tiempo de desarrollo	9 meses
Cantidad de hombres	3
Costo	\$ 20905.6275
Salario medio	\$225
RCPX	1,33
RUSE	1,00

PDIF	1,00
PREX	0,87
FCIL	0,87
SCED	1,00

Tabla 15. Costos totales.

4.4 Beneficios tangibles e intangibles.

Mantener un control más detallado y organizado de la información que se utiliza y procesa en el Departamento de Finanzas de forma digital; contribuye a mejorar la eficiencia en el trabajo de sus miembros, en la gestión de la información para el pago del estipendio estudiantil debido a que implica:

- Disminución del tiempo y esfuerzo que se invierte en esta tarea que se realiza, hasta ahora, de forma manual.
- Disminución de la acumulación de materiales impresos relacionados con el proceso de pago.
- Disminución de los gastos pues resulta menos costoso crear y procesar información digital que copias duras.
- Fácil y rápido acceso de la información actualizada.
- Fácil procesamiento de la información y obtención, dinámica, de reportes por cada facultad y en el Departamento de Finanzas en cualquier momento.

Por tanto los beneficios obtenidos con el desarrollo del software son fundamentalmente intangibles.

4.5 Análisis de costos y beneficios.

Desarrollar un producto informático cuesta. Justificar entonces su desarrollo depende de los beneficios que reportarían su implantación y utilización.

Una vez implantado el sistema éste contribuirá a aumentar la eficiencia del Departamento de Finanzas y de cada facultad de la UCF al disminuir el tiempo necesario a emplear en el registro, consulta y actualización de la compleja y diversa información; y generar reportes y nominas con mayor rapidez y certeza.

La utilización de este nuevo sistema para mostrar la información actualizada del Departamento de Finanzas de la UCF parte de la idea de concebir la información como un recurso estratégico para asistir a los integrantes del Departamento de Finanzas, estudiantes y secretarías de cada facultad. Este nuevo recurso del que dispondrá el Departamento de Finanzas y las Secretarías de cada facultad le permitirá tener actualizada toda la información para la gestión de la información para el pago del estipendio a estudiantes universitarios en la UCF. Por todo esto se concluye que la aplicación es factible.

4.6 Conclusiones.

Al desarrollo de todo producto informático va asociado un costo, el justificarlo depende de los beneficios tangibles e intangibles que produce.

La herramienta propuesta trae consigo una serie de beneficios sobre todo intangibles para la organización, pero no menos necesarios e importantes, porque va a contribuir a mejorar su funcionamiento, lo que indica que es factible implementar la herramienta propuesta. Una vez terminado el estudio de factibilidad del sistema, se estima un tiempo de **9** meses para su construcción por **3** hombres y su costo asciende a \$ **20905.6275**.

Conclusiones

Después de haber realizado un estudio de los flujos de trabajos de la oficina de Finanzas de la universidad de Cienfuegos se presenta una propuesta de sistema que basa su funcionamiento en los estándares de la W3C para la web. En esta propuesta:

- Se desarrolla una Base de Datos capaz de organizar y almacenar de manera eficiente la información que se manipula en el departamento de finanzas.
- Se desarrolla un módulo a la Especialista en Finanzas que le permita obtener por facultades toda la información de sus estudiantes de una forma mas rápida organizada y eficiente y que además, le permita la confección de las nóminas para el pago del estipendio estudiantil de forma automatiza por facultades, carreras y grupos de acuerdo con la información facilitada por cada secretaria de facultad al inicio de mes.
- Se desarrolla un módulo que permita a cada Secretaria de Facultad gestionar la información desde las facultades a fin de reducir el tiempo y e incrementar la gestión de la información para el pago del estipendio, contribuyendo de esta manera a la calidad en sus servicios.
- Se valida el funcionamiento del software.
- Se confecciona la ayuda del sistema a fin de facilitar su uso.
- Se determina la factibilidad económica del *software* implementado.

Recomendaciones

A pesar de que los objetivos trazados con la realización de este trabajo fueron cumplidos, los desarrolladores del mismo sugieren tomar esta propuesta solo como la primera versión del proyecto y continuar esta investigación.

Se recomienda entonces:

- Continuar con el estudio de los flujos de trabajos y de información de la Departamento en Finanzas para abarcar todos los aspectos referidos a esta, así como profundizar en los aspectos tratados con mayor detalle.
- Extender su uso a otras universidades para la validación de su funcionamiento.
- Mantener la concepción base utilizada de los servicios web para garantizar la interoperabilidad con otros módulos de otros sistemas, ya sean propios o desarrollados por terceros.
- Conservar la idea de que el sistema debe ser desarrollado sobre la base de la multiplataforma lo que permitirá adaptarse a las nuevas tecnologías de la informática moderna.
- Poner el sistema a prueba durante un período significativo para comprobar que sus funcionalidades se correspondan con las actividades que están gestionando.

Referencias Bibliográficas.

- [1] MES. *Metodología para elaboración del anteproyecto del presupuesto actividad Presupuestada 2004*. Ciudad Habana, 2004.
- [2] DEE. *Diccionario Enciclopédico Espasa*. Tomo 1, Novena edición, Madrid, 1984.
- [3] DEE. *Diccionario Enciclopédico Espasa*. Tomo 10, Novena edición, Madrid, 1984.
- [4] DEE. *Diccionario Enciclopédico Espasa*. Tomo 2, Novena edición, Madrid, 1984.
- [5] DEE. *Diccionario Enciclopédico Espasa*. Tomo 9, Novena edición, Madrid, 1984.
- [6] Aja Quiroga, Lourdes. *Gestión de información, gestión del conocimiento y gestión de la calidad en las organizaciones*
http://www.bvs.sld.cu/revistas/aci/vol10_5_02/aci04502.htm , Mayo del 2007.
- [7] [ENC 24] Enciclopedia Universal Ilustrada Europeo Americana, tomo XXV ,1924
- [8] Letelier Torres Patricio. *Desarrollo de Software Orientado a Objeto usando UML*, Departamento Sistemas Informáticos y Computación, Universidad Politécnica de Valencia, España. <http://www.creangel.com/uml/intro.php>, (15/01/2007)
- [9] Ferrá Grau Xavier. *Desarrollo orientado a objetos con UML*. Universidad Politécnica de Madrid, 2004.<http://www.clikear.com/manual/uml/introduccion.asp>, (20/10/2006).
- [10] Díaz Antón Maria Gabriela, Angélica Pérez Maria .*Propuesta de una metodología de desarrollo de software educativo bajo un enfoque de calidad sistemática* 2004.
<http://www.academiainteractiva.com>. (14/02/2007)
- [11] Teruel, Alejandro. *Introducción a la arquitectura de capas*.
<http://www ldc.usb.ve/~teruel/ci3715/clases/argCapas.html> (26/04/07)

- [12] Macromedia Dreamweaver 8 Getting Started. Ayuda.
- [13] Munz, Stefan. *Tecnologías Web ASP*.
<http://es.selfhtml.org/introduccion/tecnologias/asp.htm>. (3/05/07)
- [14] Idem. <http://es.selfhtml.org/introduccion/tecnologias/AJAX.htm> (3/05/07)
- [15] Idem. <http://es.selfhtml.org/introduccion/tecnologias/java.htm> (20/06/07)
- [16] Introducción a php <http://www.ciberteca.net/webmaster/php>. (20/01/2006)
- [17] Rodríguez Luis. *Conoce a Pear: El Repositorio de Aplicaciones y Extensiones de PHP*. http://programacion.com/blogs/60_php_extensiones/php.htm. (4/01/07)
- [18] Munz, Stefan. *XML y derivados de XML*.
<http://es.selfhtml.org/introduccion/tecnologias/webservices.htm> (10/01/07)
- [19] Idem. <http://es.selfhtml.org/introduccion/tecnologias/soap.htm> (10/01/07)
- [20] Idem. <http://es.selfhtml.org/introduccion/tecnologias/xml.htm> (10/01/07)
- [21] Matos, Rosa María. *Introducción al trabajo con Base de Datos*. Asignatura de Sistemas de Gestión de Base de Datos. Ciudad Habana, CUJAE, 2004
- [22] *Introducción a MySQL*, <http://www.lobocom.es/~claudio>, 2005 (21/05/2007)
- [23] Manual de Sql, <http://walter.freesevers.com> (21/05/2007)
- [24] Jacobson, I.; Booch, G. y Rumbaugh, J. *El Proceso Unificado de Desarrollo de software*. Addison-Wesley, EUA, 2000
- [25] Idem, pp. 115-118p.

[26] Idem,.116 p.

[27] Idem, 117p.

[26] Idem, 121p.

[28] Idem, 125p.

[29] Idem, 127p.

[30] Idem, 130p.

[31] Idem, 130p.

[32] Idem, 132p.

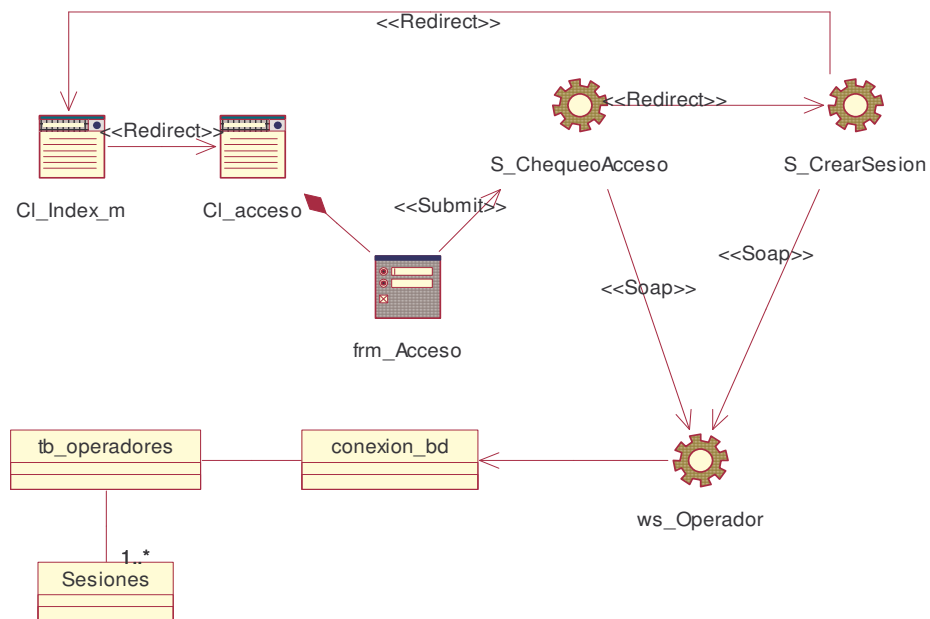
Bibliografía.

- [ARO 93] Aronow, D.B. *Information technology applications in quality assurance and quality improvement*, *The Joint Commision Journal on Quality Improvement*, (E.U) 19, 1993.
- [BEN 80] Bennatan, E.M. *Software Project Management: A Practitioner's Approach* / E.M.1980
- [BOE 81] Boehm, B. *Software Engineering Economics*. Prentice Hall, 1981.
- [BOO 96] Booch, Grady. *Análisis y Diseño Orientado a Objetos*. E. U: Ed. Addison-Wesley ,1996
- [DAT 97] Date, C.J. *An Introduction to Database Systems*. EU,1997
- [DIA 96] Díaz Ferrera, Carlos N. *Sistema de Almacenamiento de la Información*. Ciudad de la Habana: Editorial ENPES, 1982
- [FAI 96] Fairley, Richard. *Ingeniería del Software*. Mexico: Mc Graw-Hill, 1996.
- [GAR] García Hernández, L. *Bases de datos*. C. Habana. Editorial ,1989
- [JON 86] Jones, C. *Programming Productivity*. EE UU, 1986
- [KER 84] Kerr, J. *Inside RAD.*/ J. Kerr, R Junter __ EE UU,1994 ,200p.
- [MIC 98] Microsoft Co. *SQL Server Books Online*. EE. UU. Microsoft Press, 1998
- [PRE 98] Pressman, Roger. *Ingeniería de Software, un enfoque práctico*. E.U:, 1998
- [RUM 00] Rumbaugh, J. *El Lenguaje Unificado de Modelado. Manual de Referencia*. Madrid: 2000.

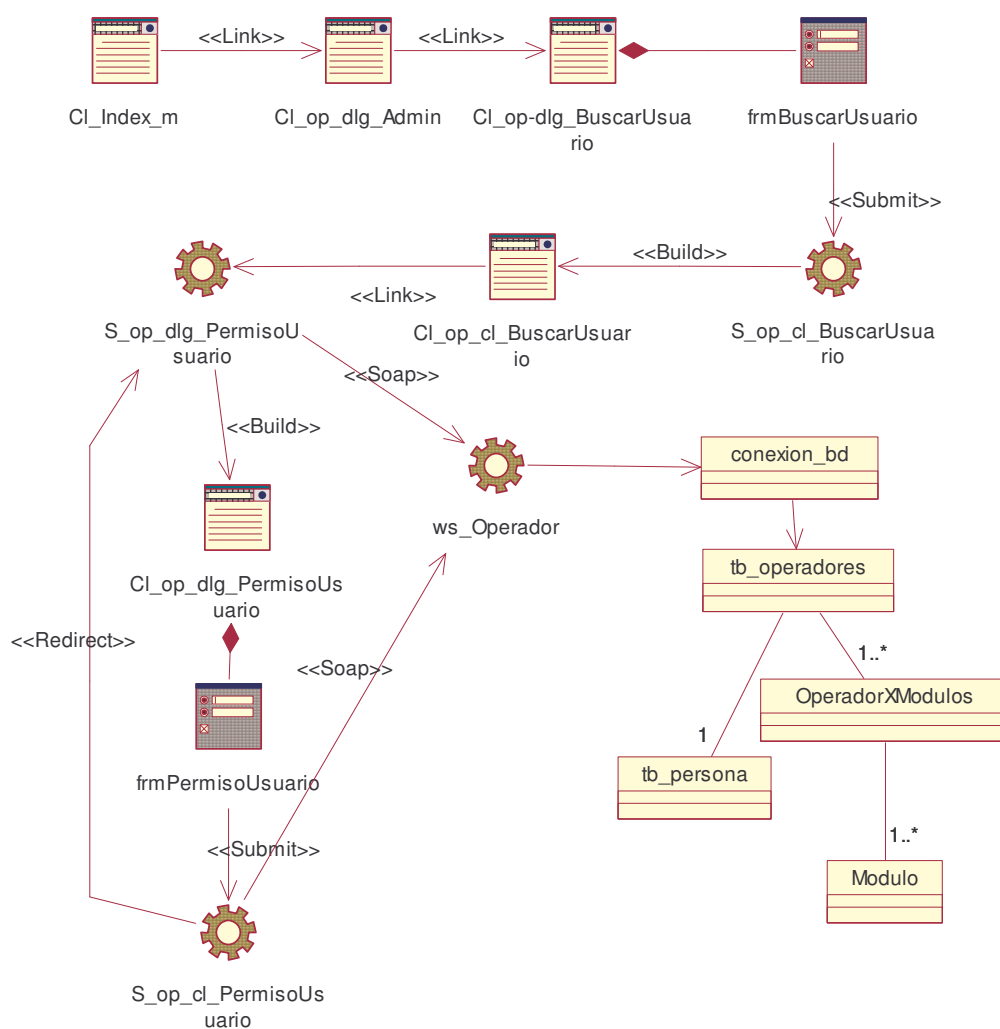
- [TAY 89]Taylor, E.S. *An Interim Report on Engineering Design*.E.U: Cambridge, Massachuett Institute of Technology, 1989.
- [TEO 86]Teory TJ. JP. *"A logical design methodology for relational database using the extended entity-relationship method"*.E.U,1986.
- [ULL 83]Ullman JD. *"Principles of Database Systems*.E U,1983.

Anexo A. Diagramas Clases Web.

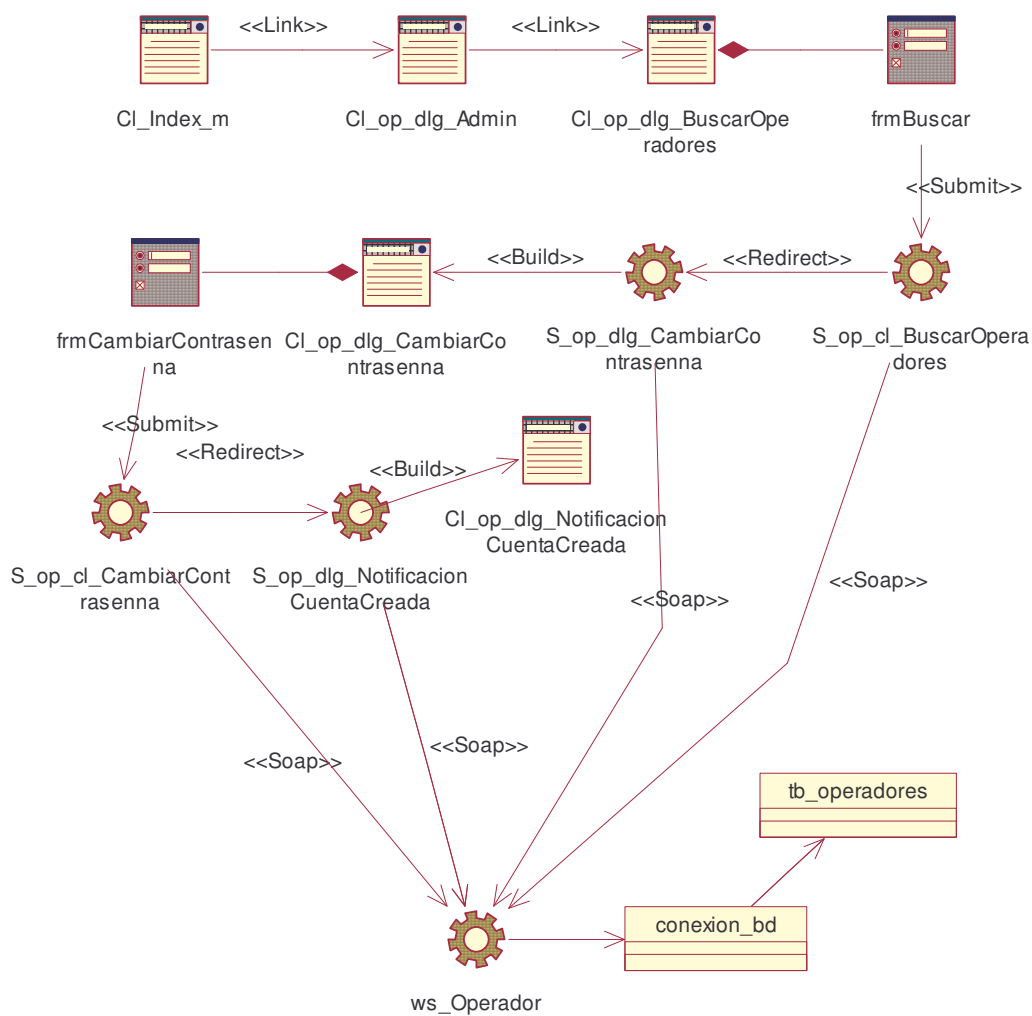
A.1-Iniciar Sesión



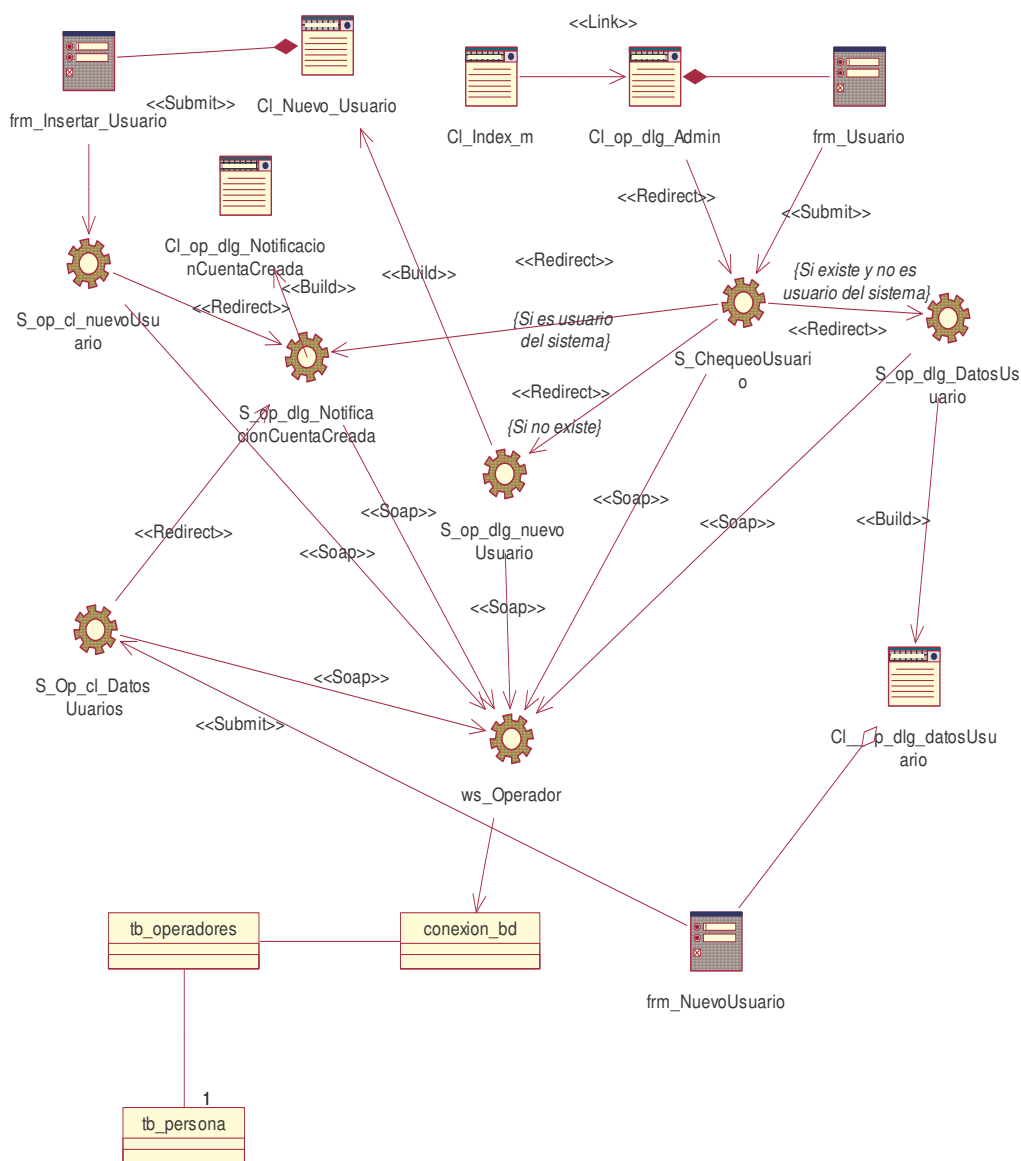
A.2- Asignar permiso a usuarios



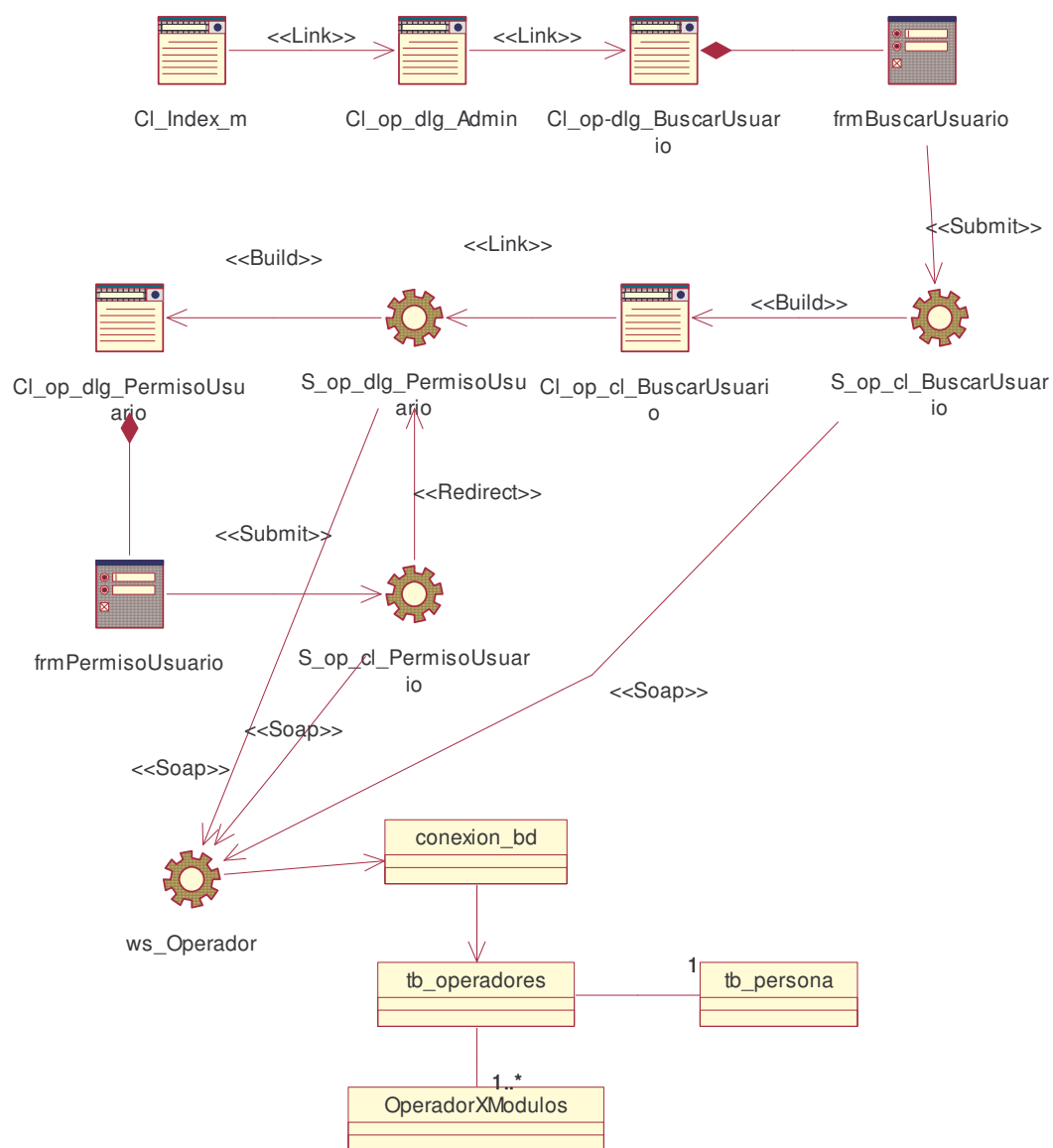
A.3-Cambiar Contraseña



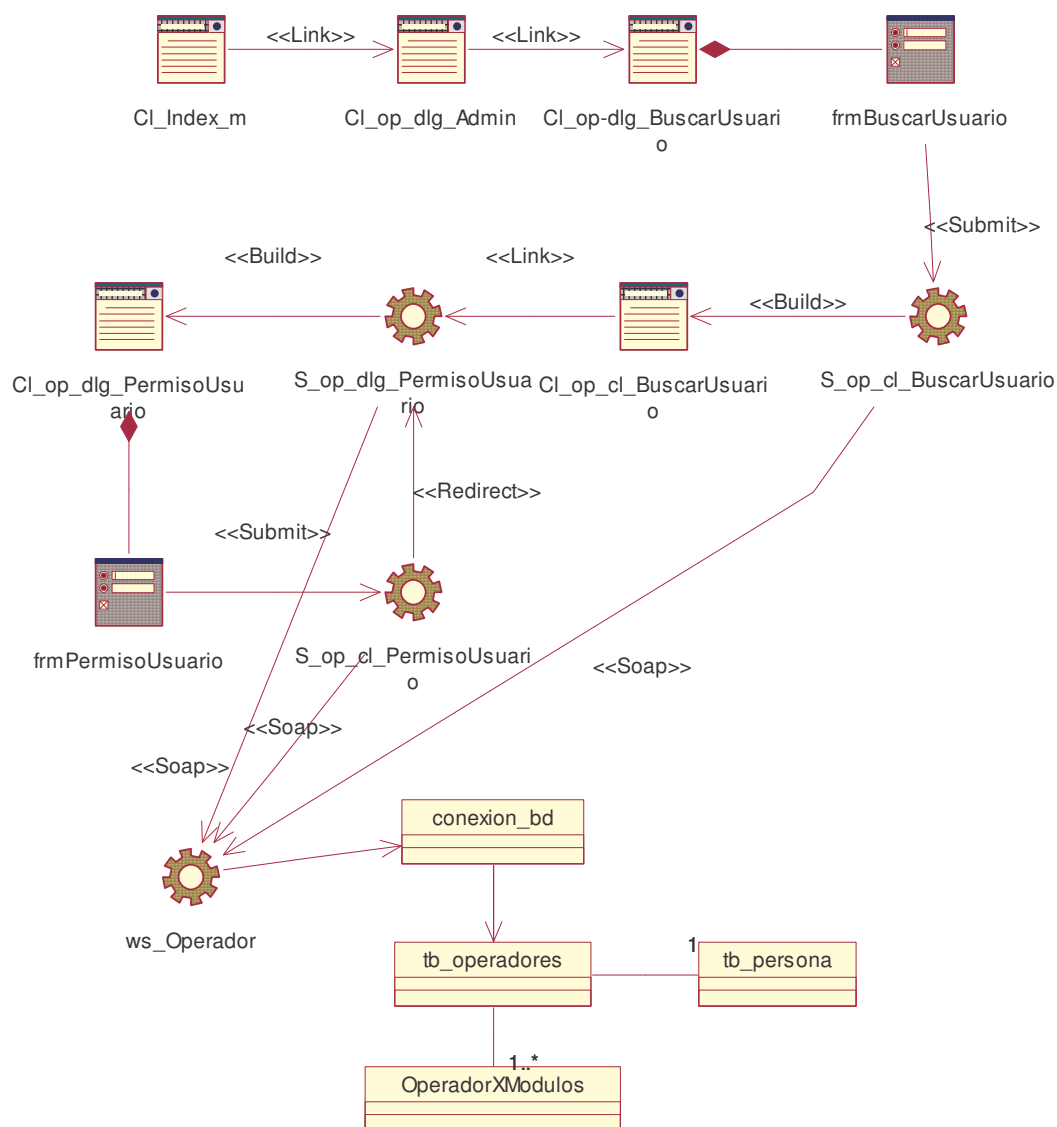
A.4-Crear Usuario



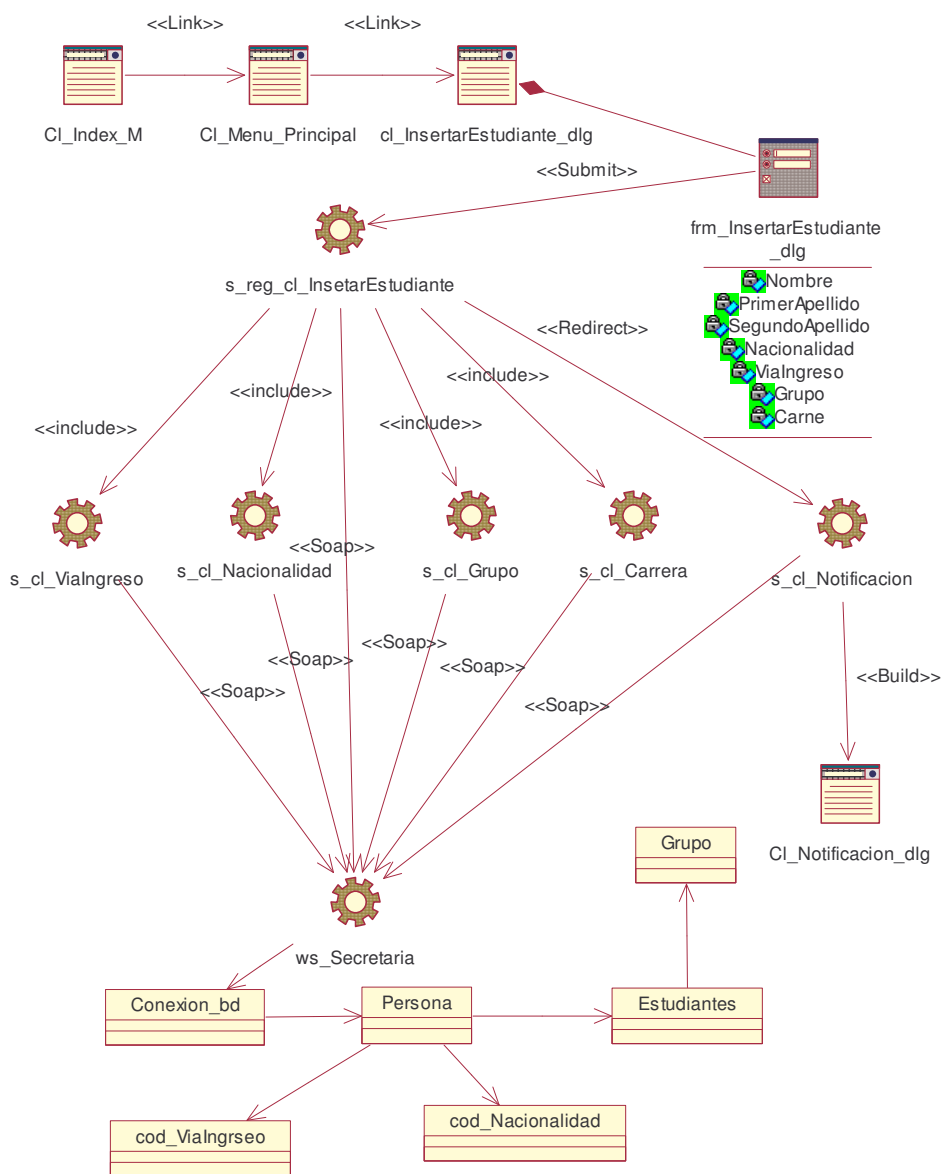
A.5-Habilitar Usuarios



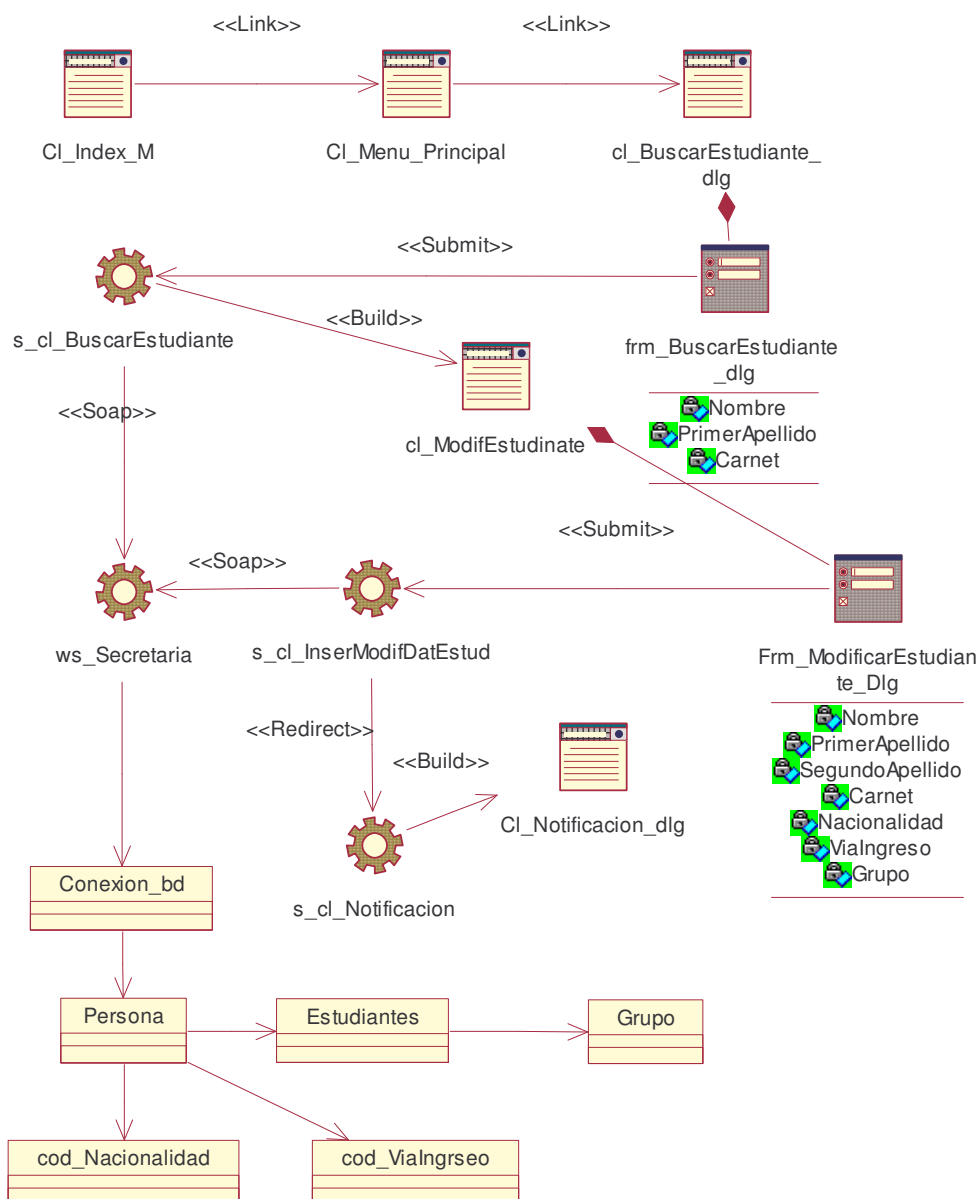
A.6- Inhabilitar Usuarios



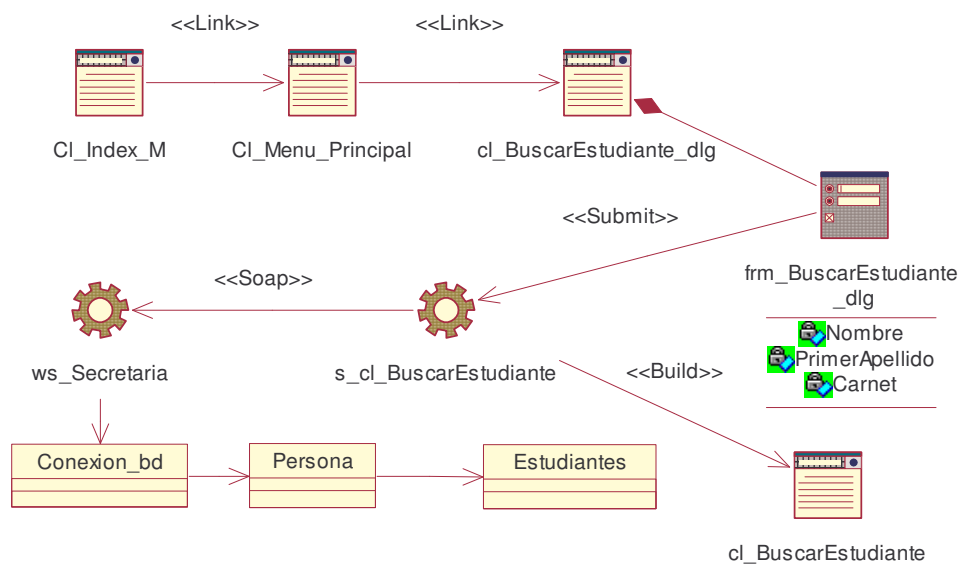
A.7- Insertar un estudiante



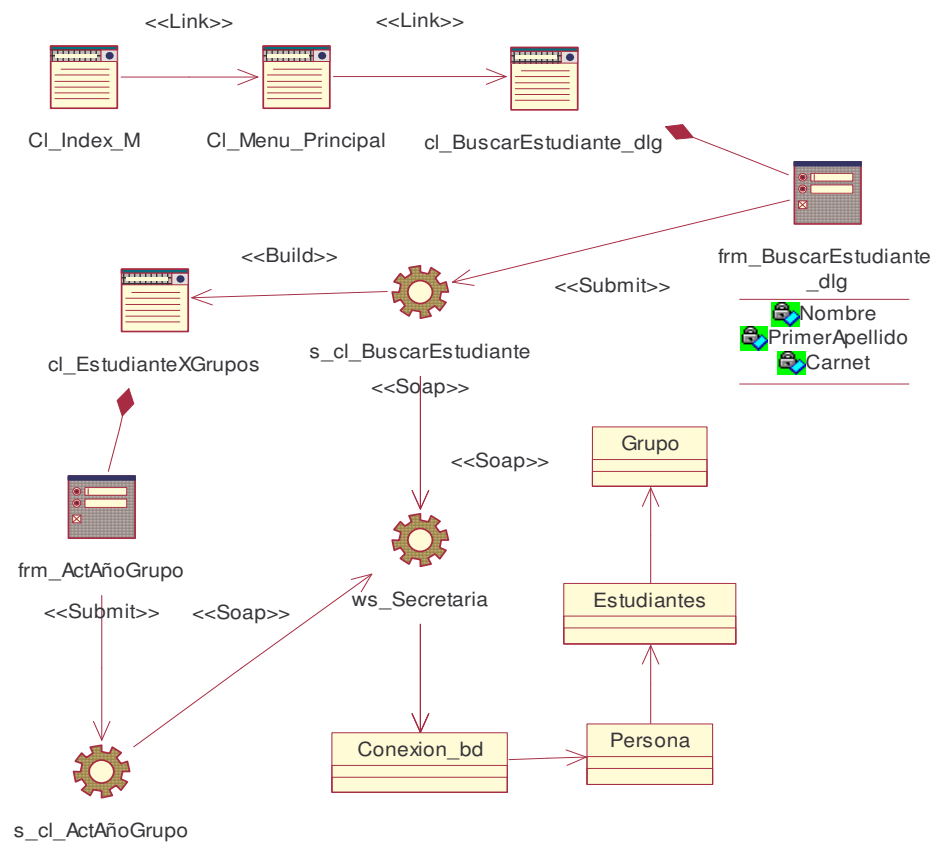
A.8- Modificar los datos de cada estudiante.



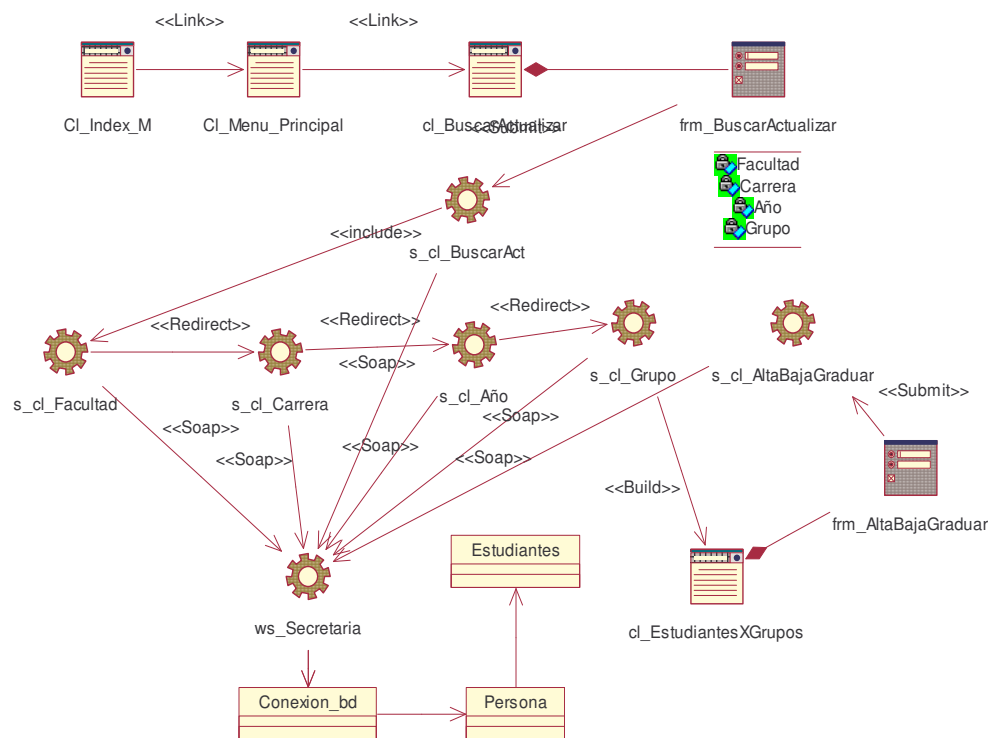
A.9 -Buscar un estudiante.



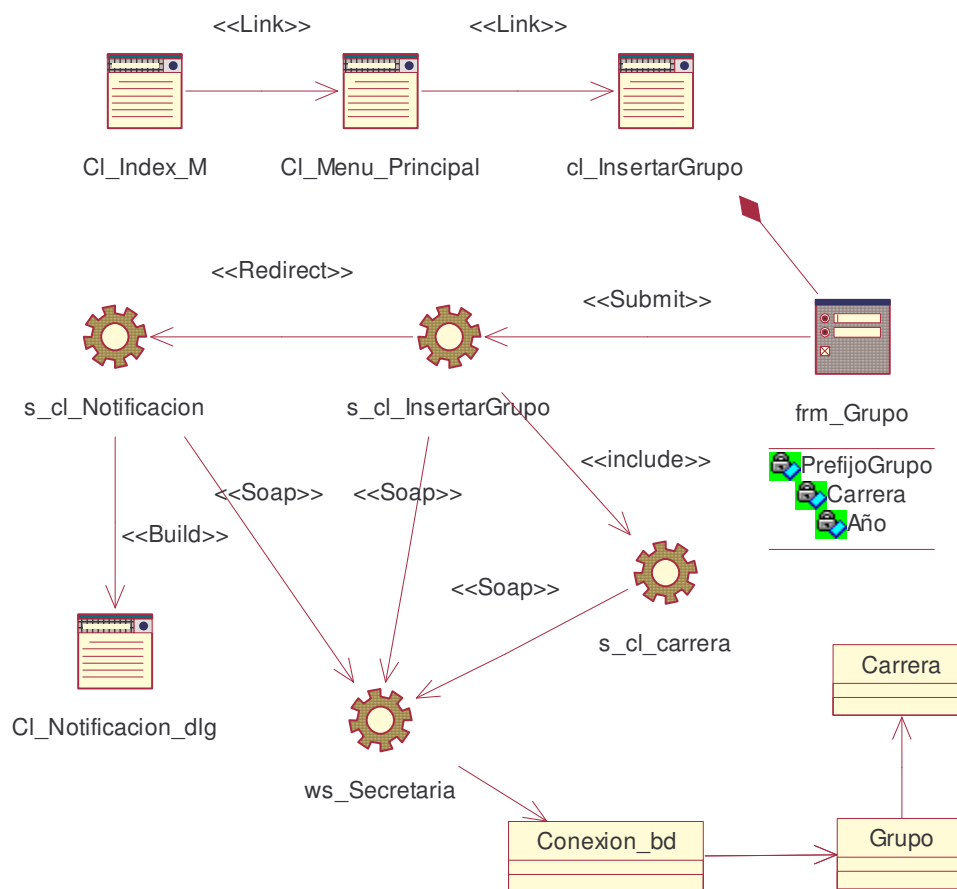
A.10- Actualizar año y grupo a cada estudiante.



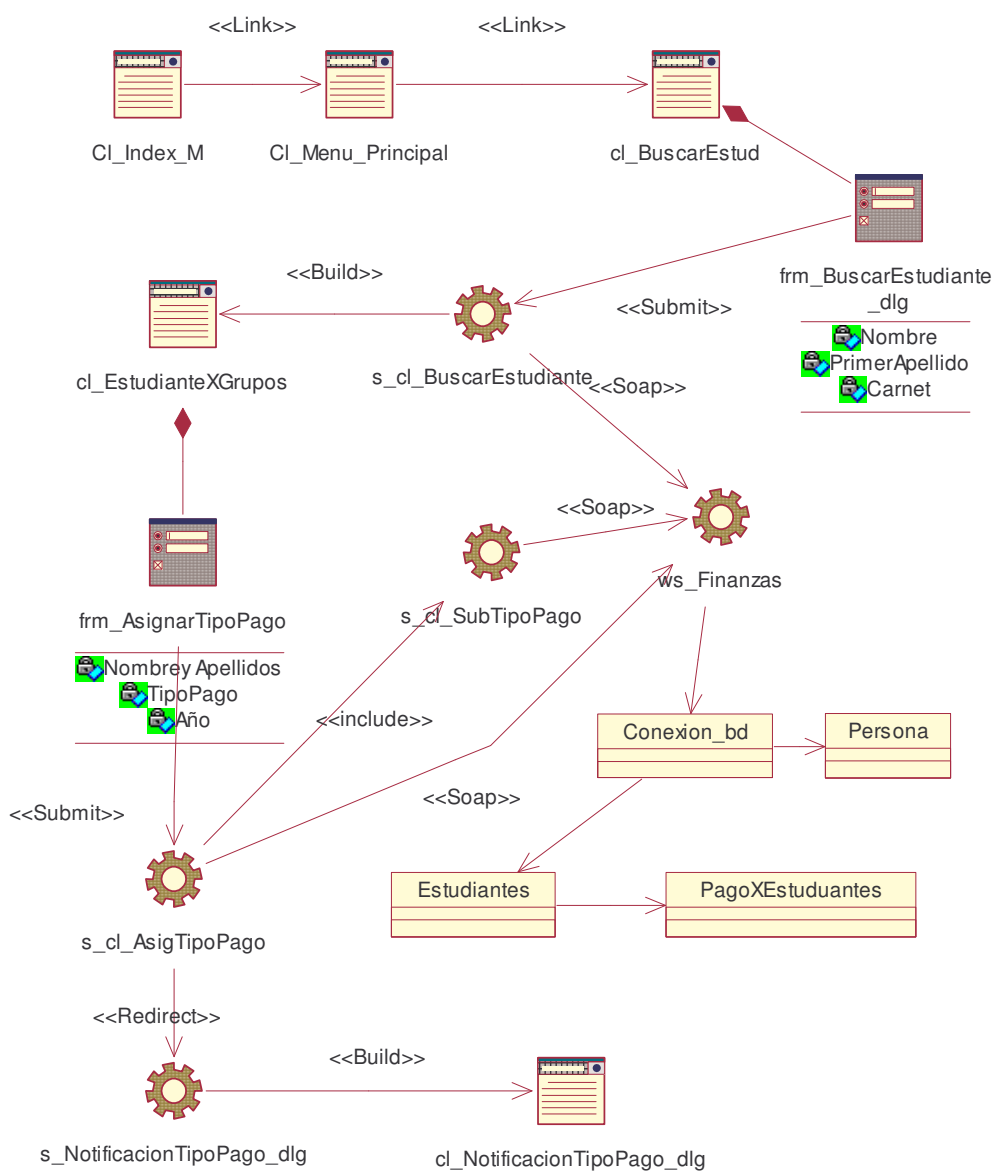
A11.-Dar alta, baja o graduar a cada estudiante.



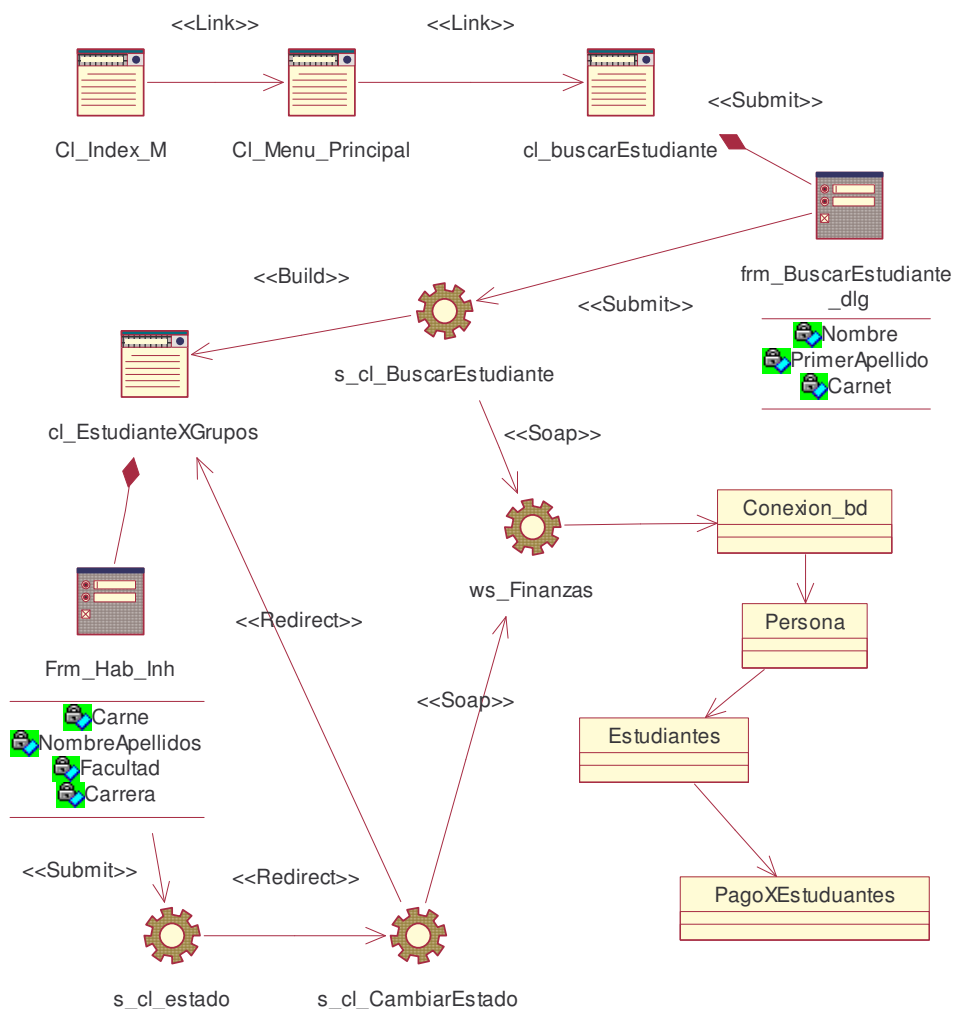
A.12- Insertar un grupo académico.



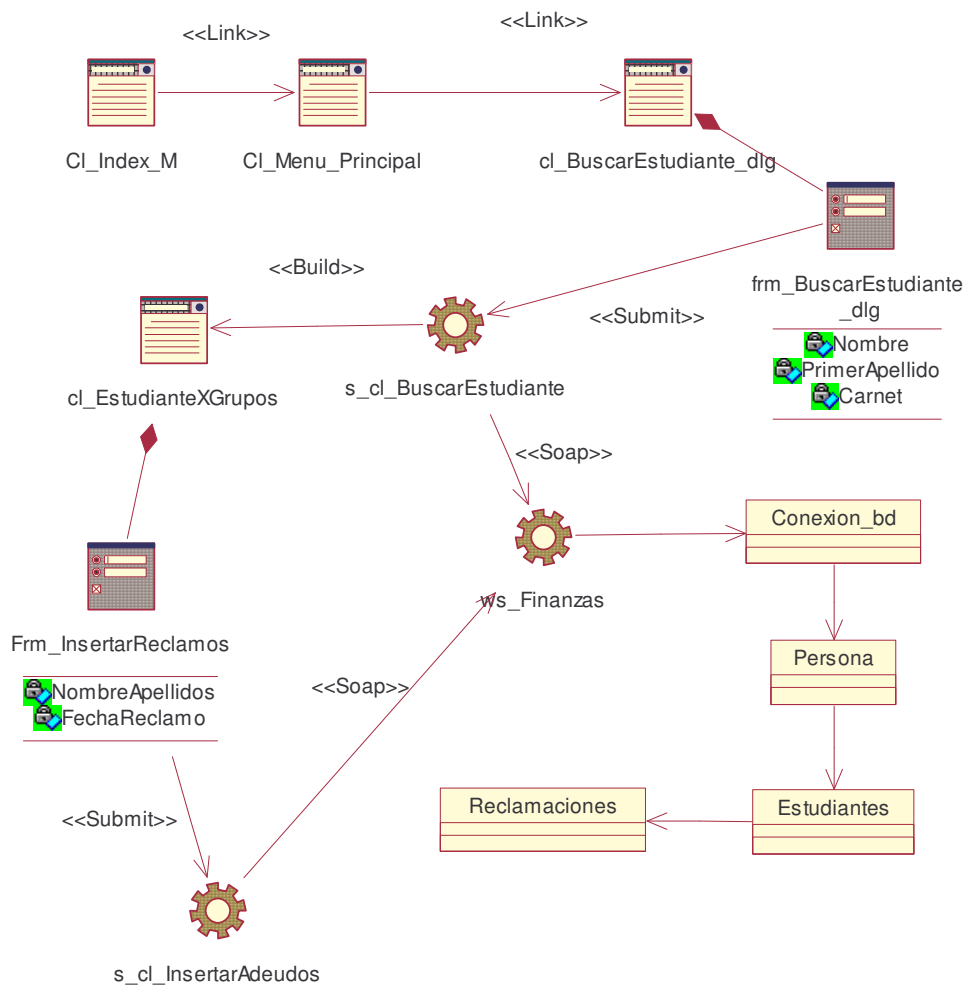
A.13 -Asignar un tipo de pago a cada estudiante.



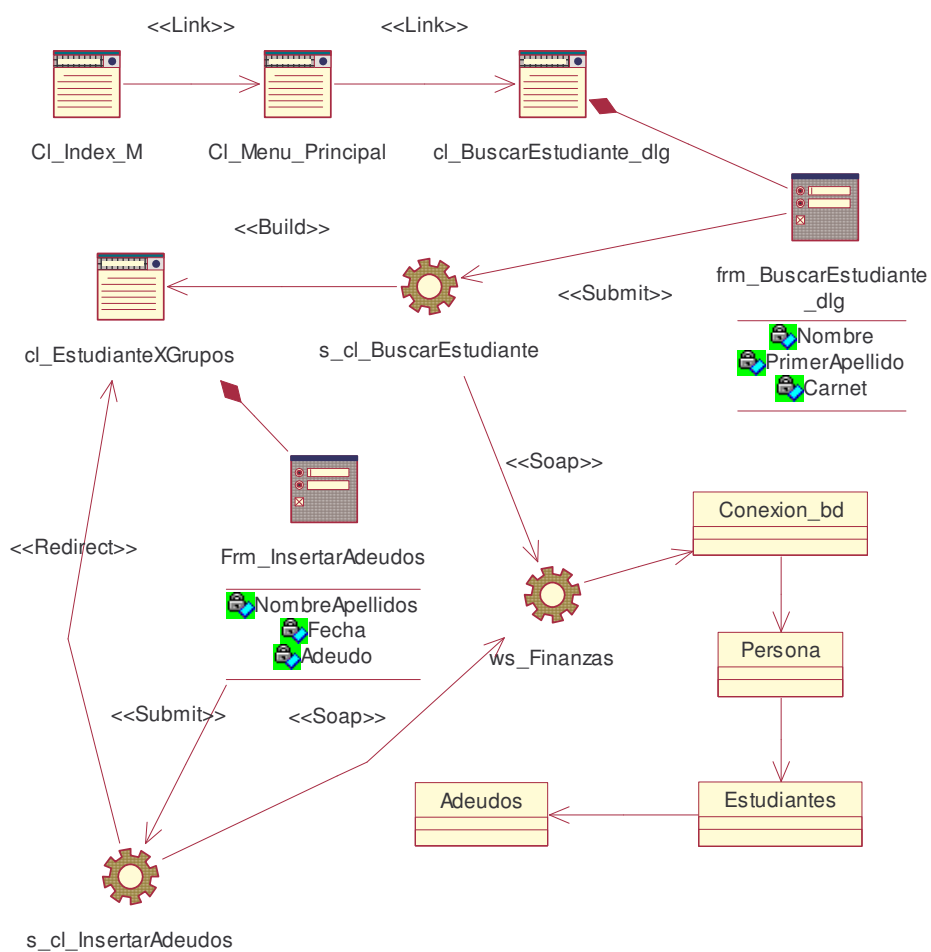
A.14- Habilitar e Inhabilitar un tipo de pago a cada estudiante.



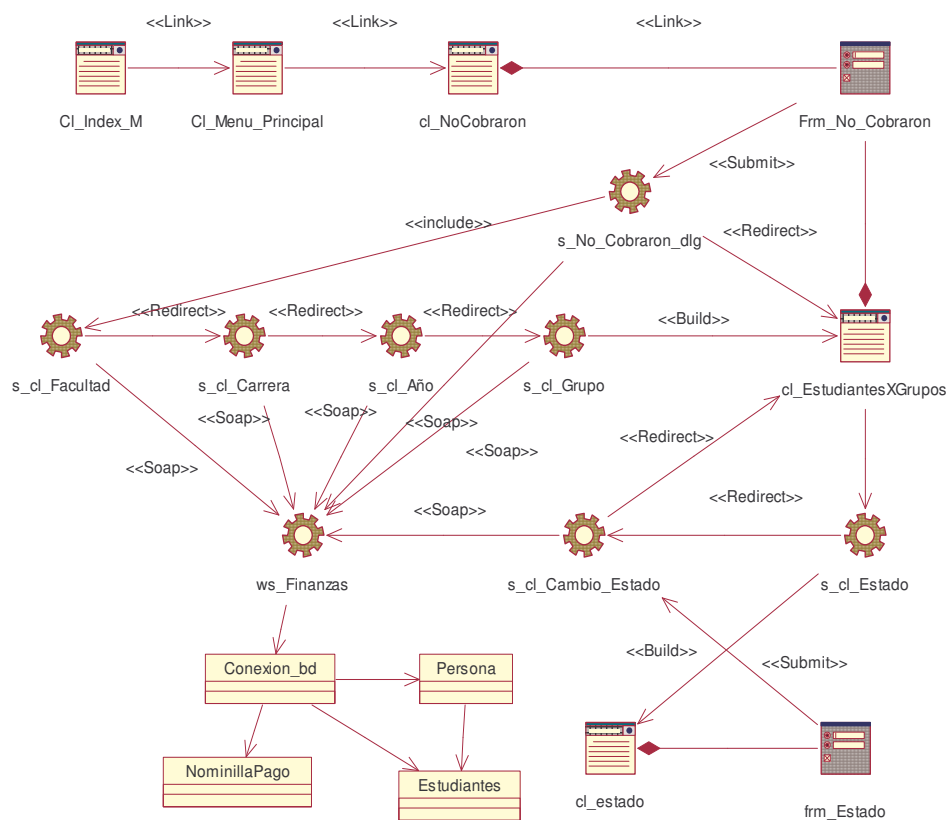
A.15 - Registrar reclamación(s) de cada estudiante.



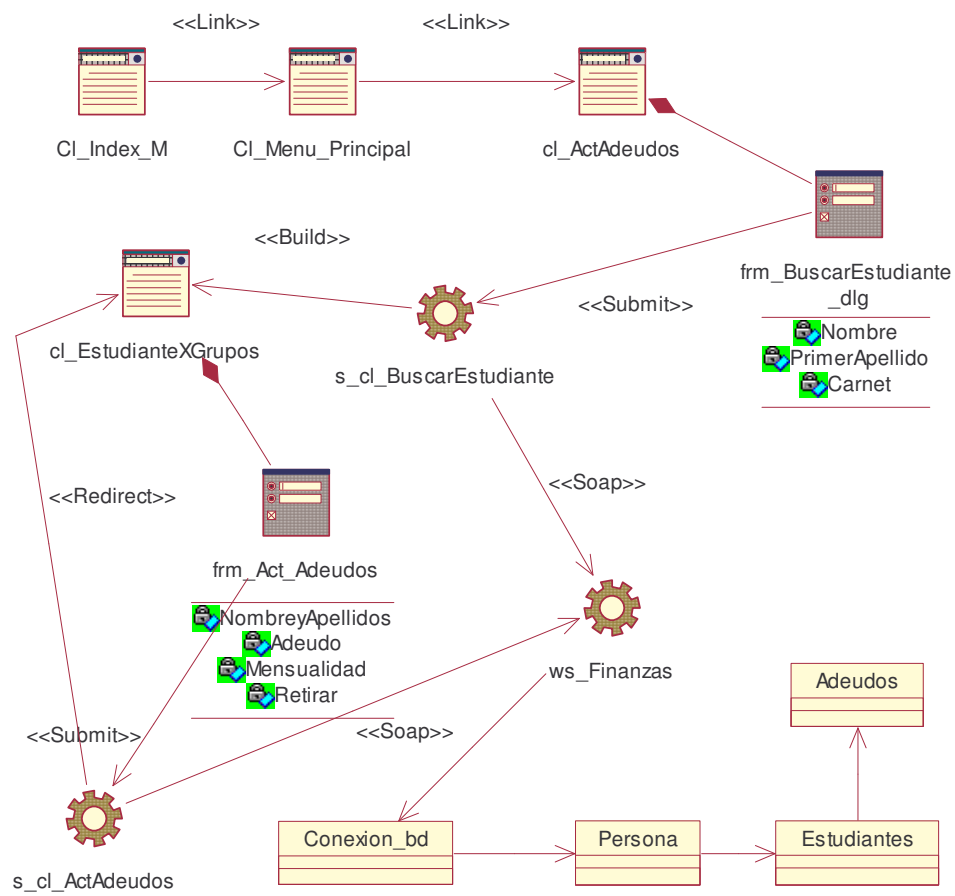
A.16- Registrar adeudo(s) de cada estudiante.



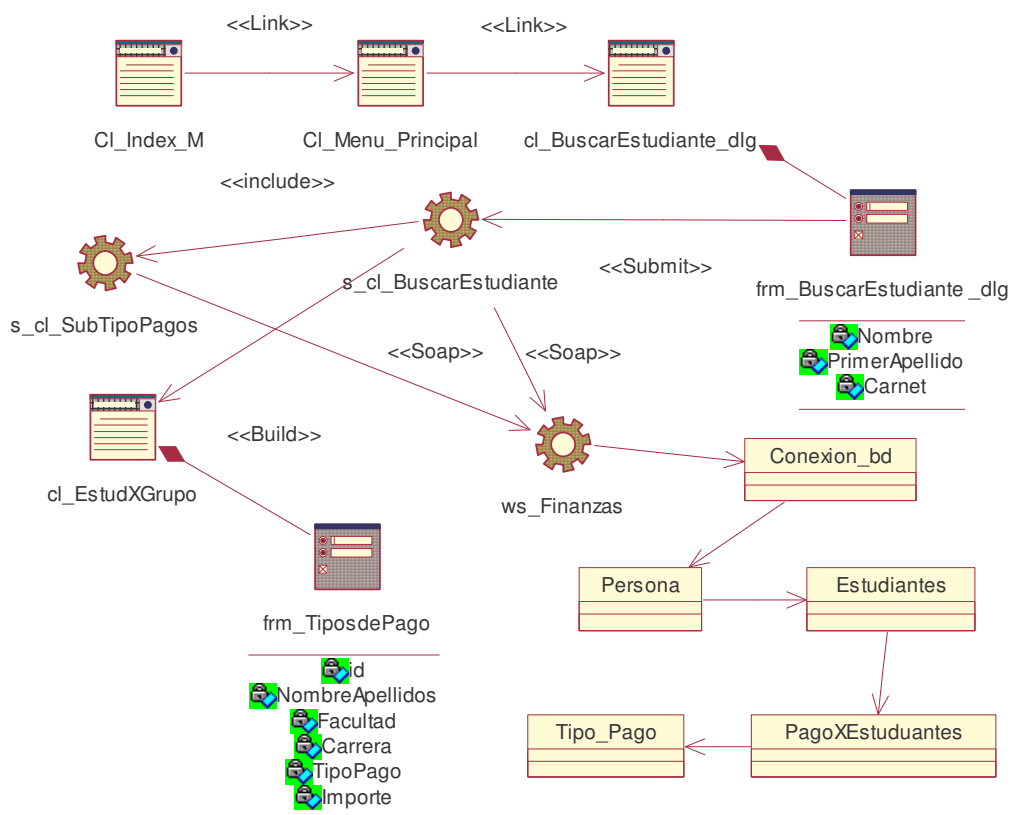
A.17- Registrar estudiantes que no cobraron en el último pago efectuado.



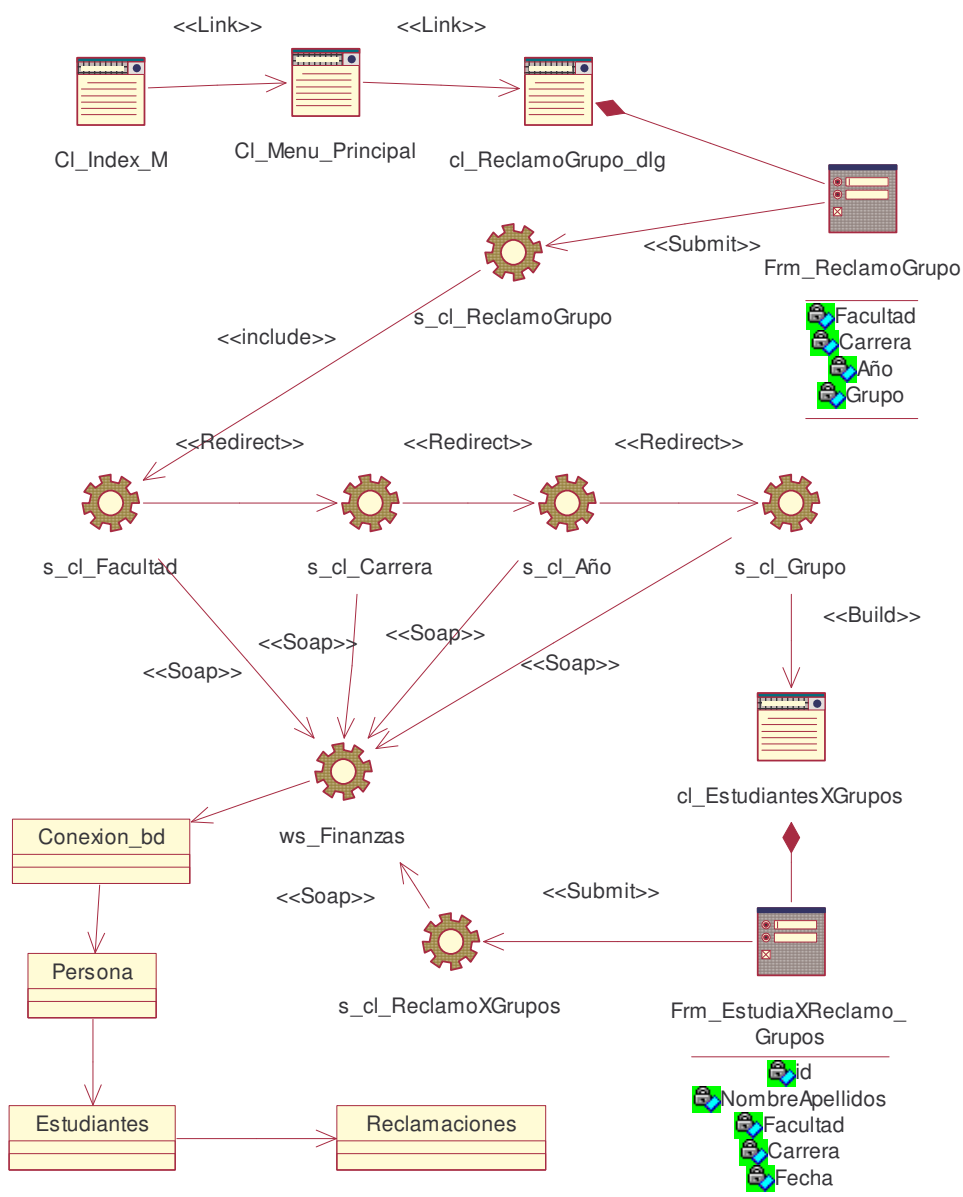
A.18- Actualizar adeudo de cada estudiante.



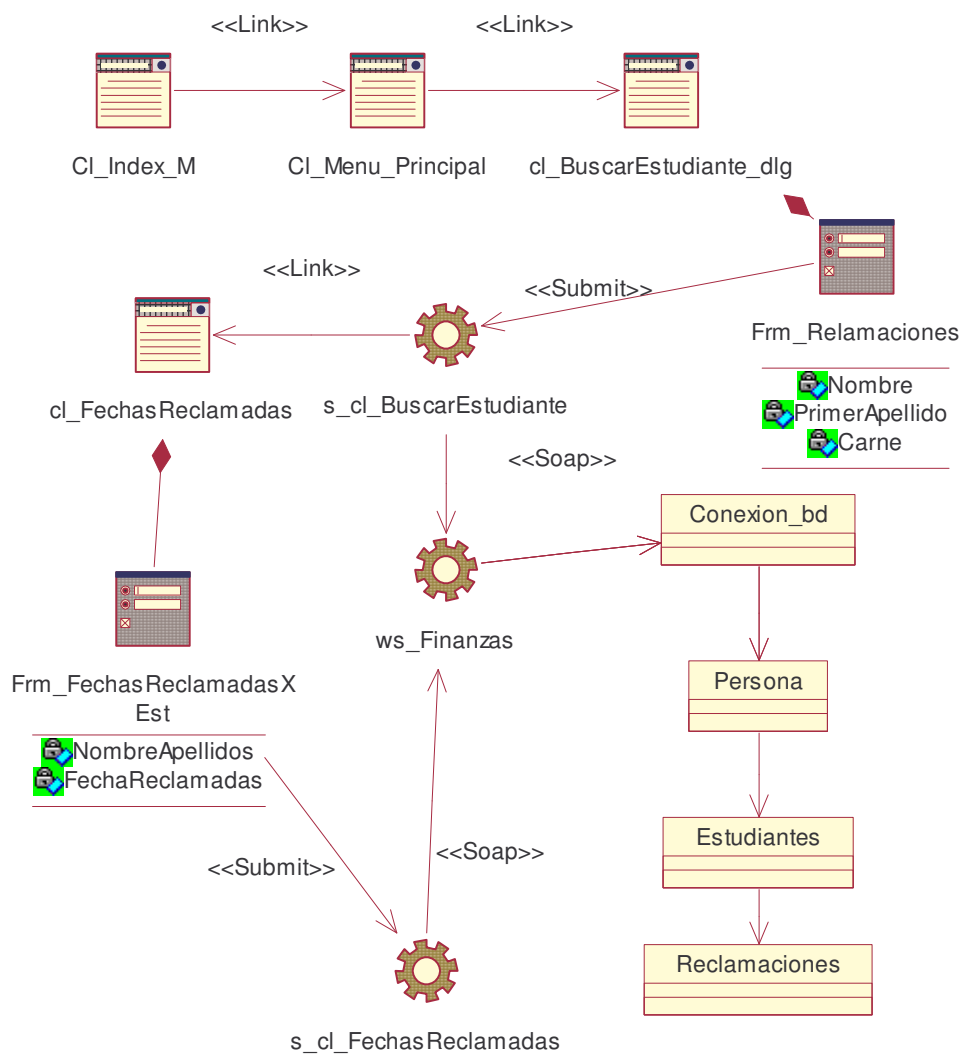
A.19- Visualizar los tipos de pago de cada estudiante.



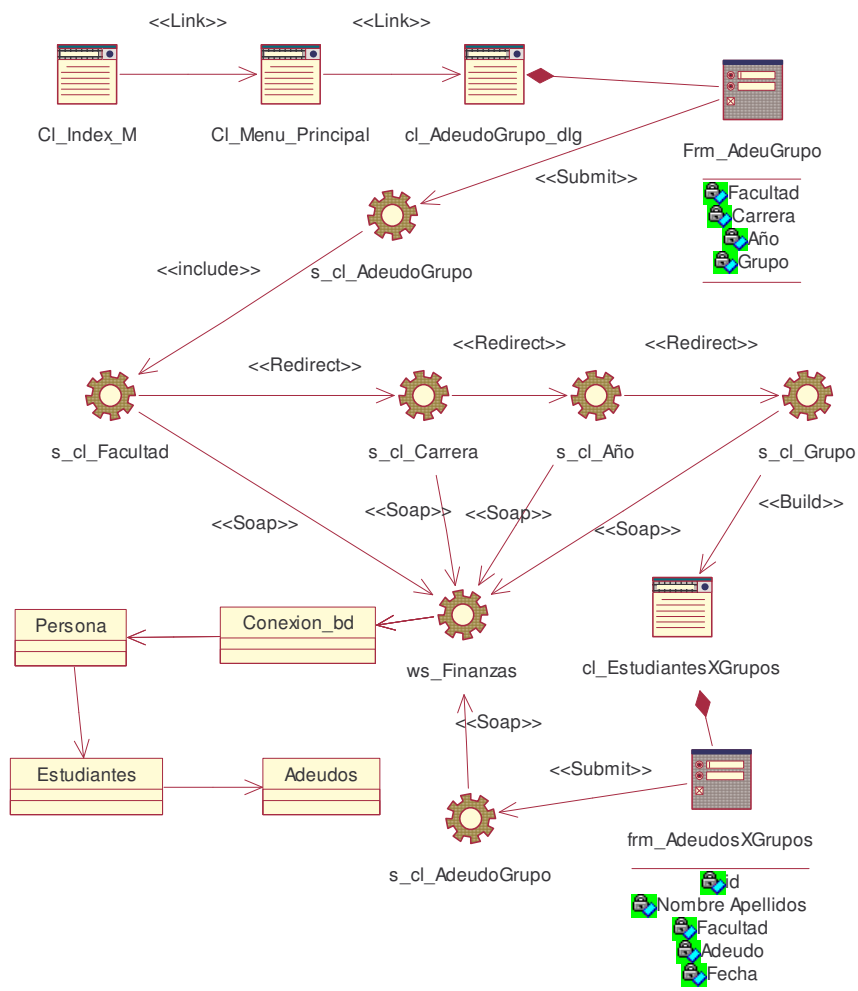
A.20- Visualizar las reclamaciones de cada grupo académico.



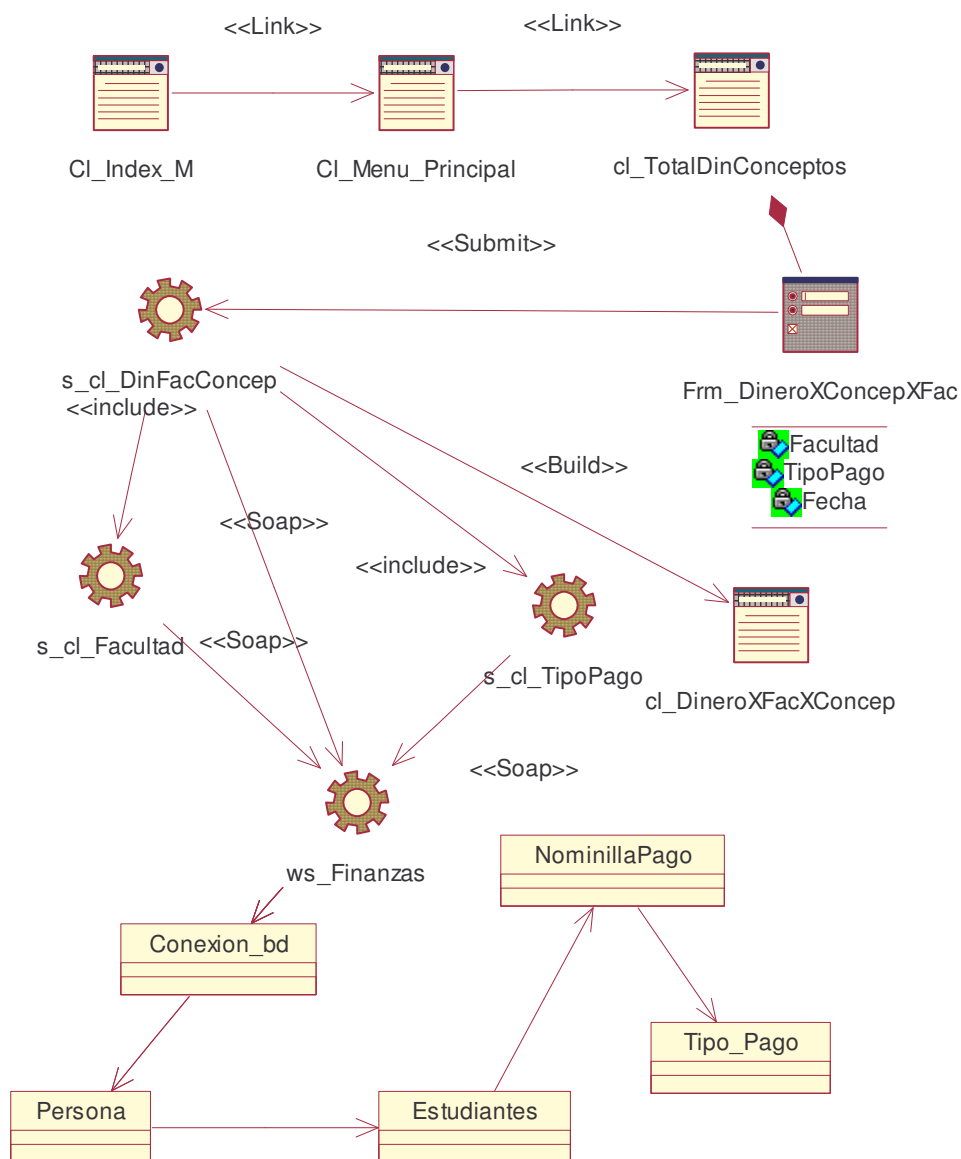
A.21- Visualizar las fechas reclamadas por estudiante.



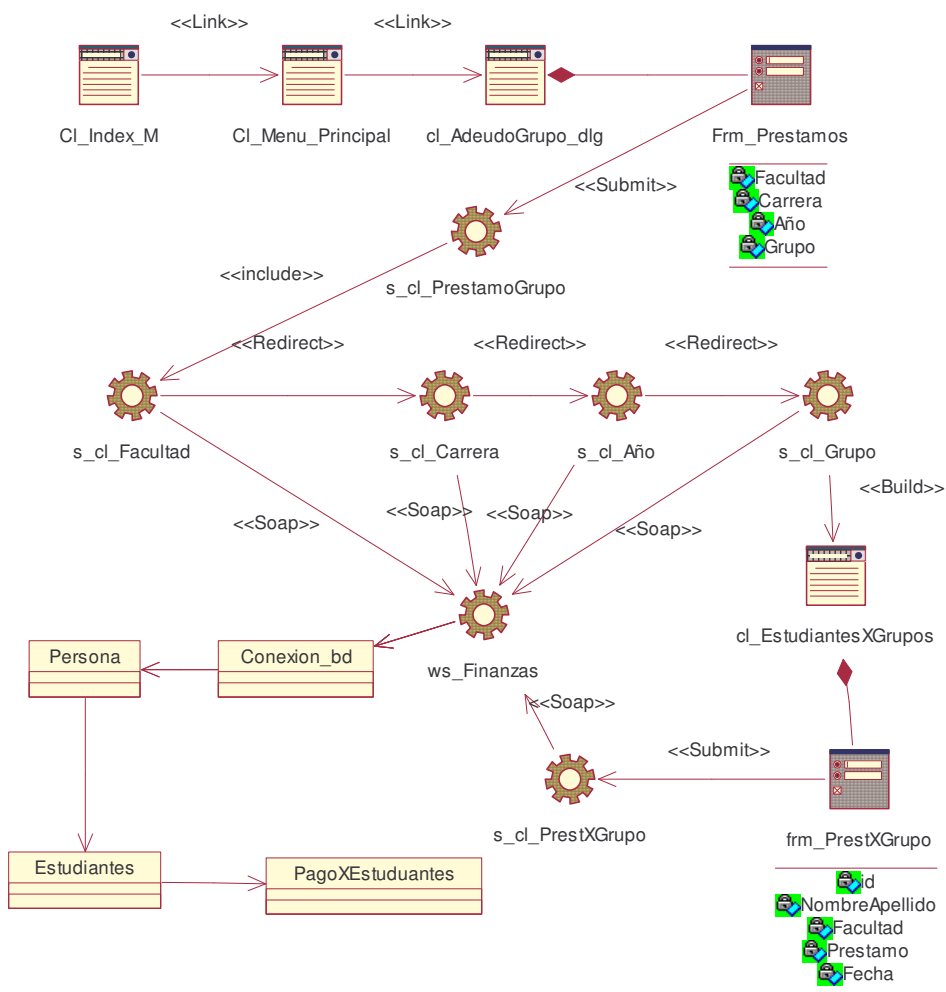
A.22- Visualizar adeudo(s) de cada grupo académico.



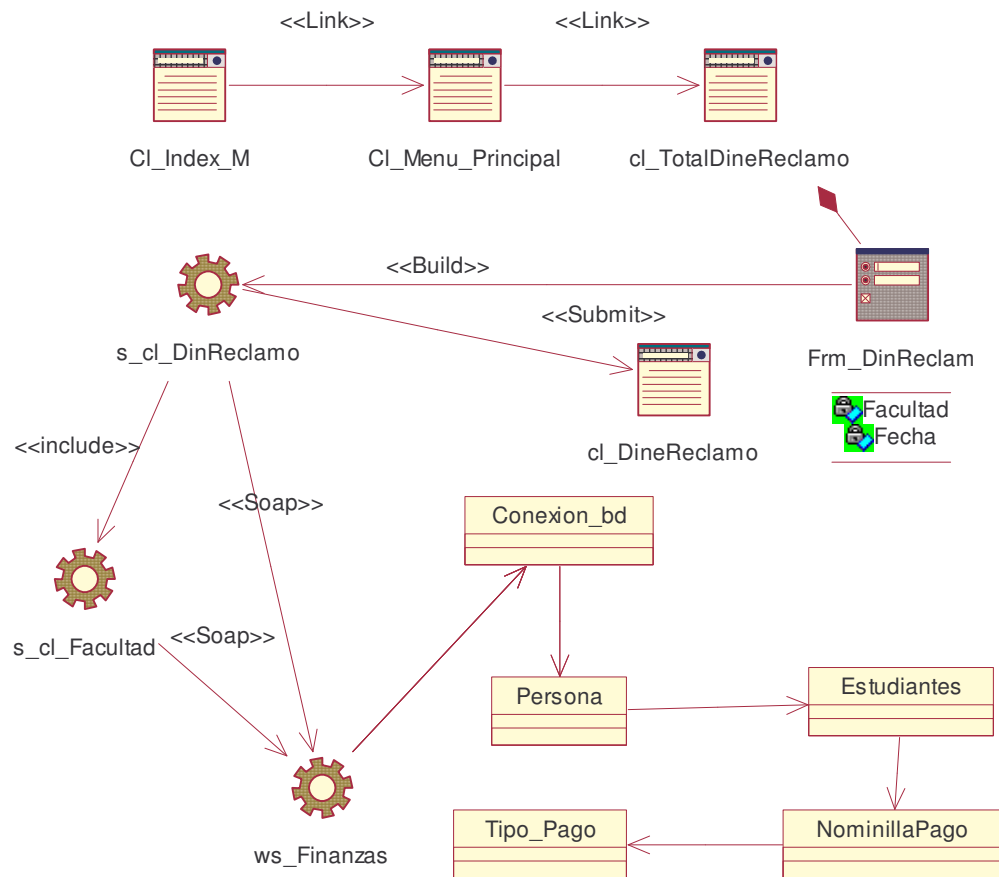
A.23- Visualizar dinero pagado a cada facultad por conceptos.



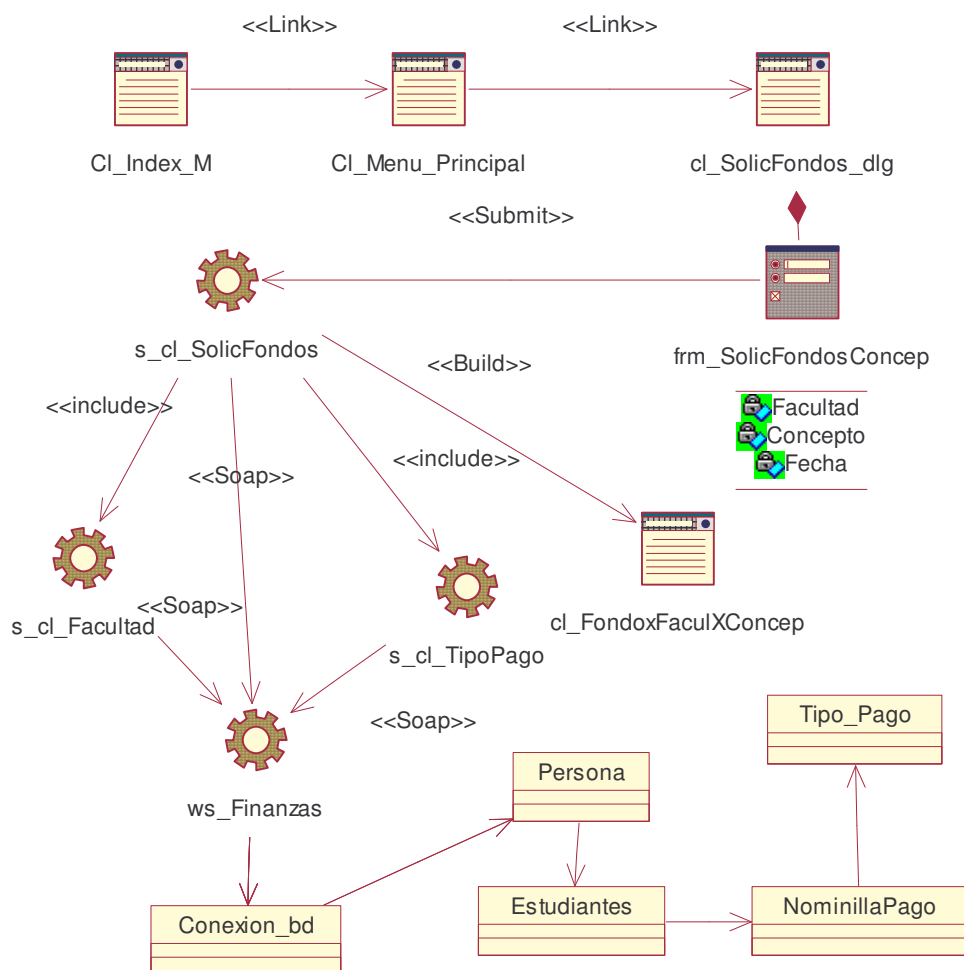
A.24- Visualizar préstamos de cada grupo académico.



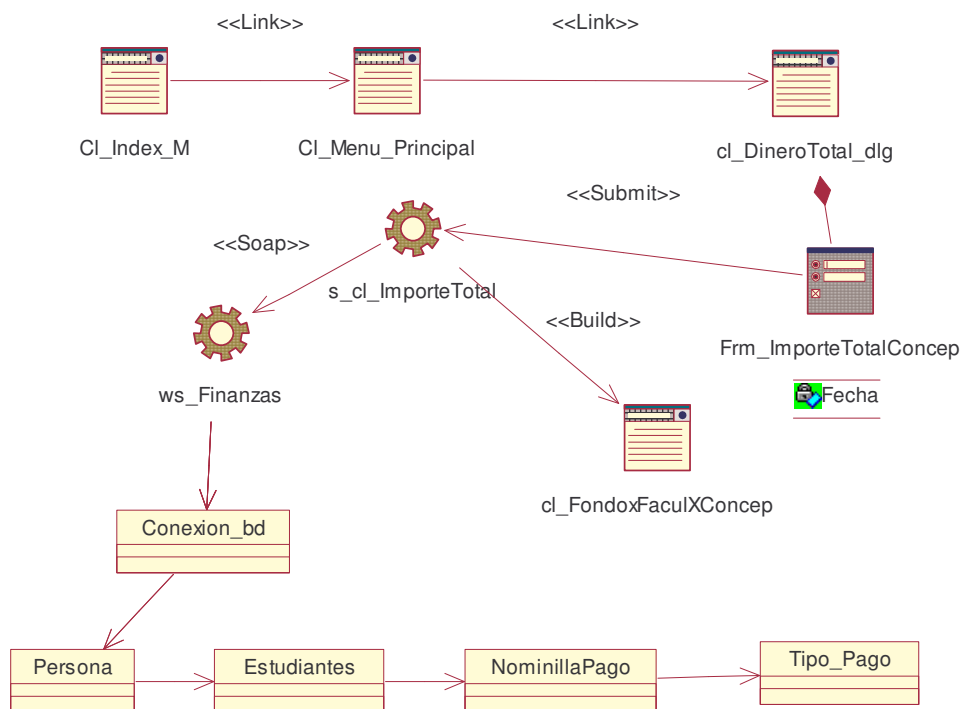
A.25- Total de dinero por reclamaciones.



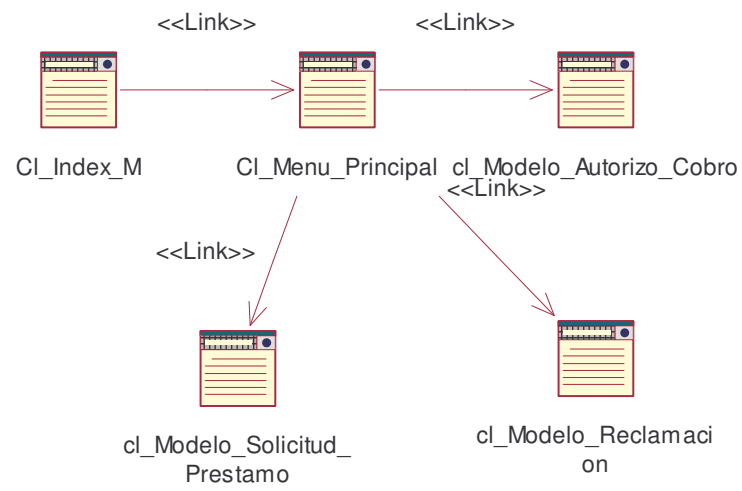
A.26 - Visualizar solicitud de fondos por conceptos.



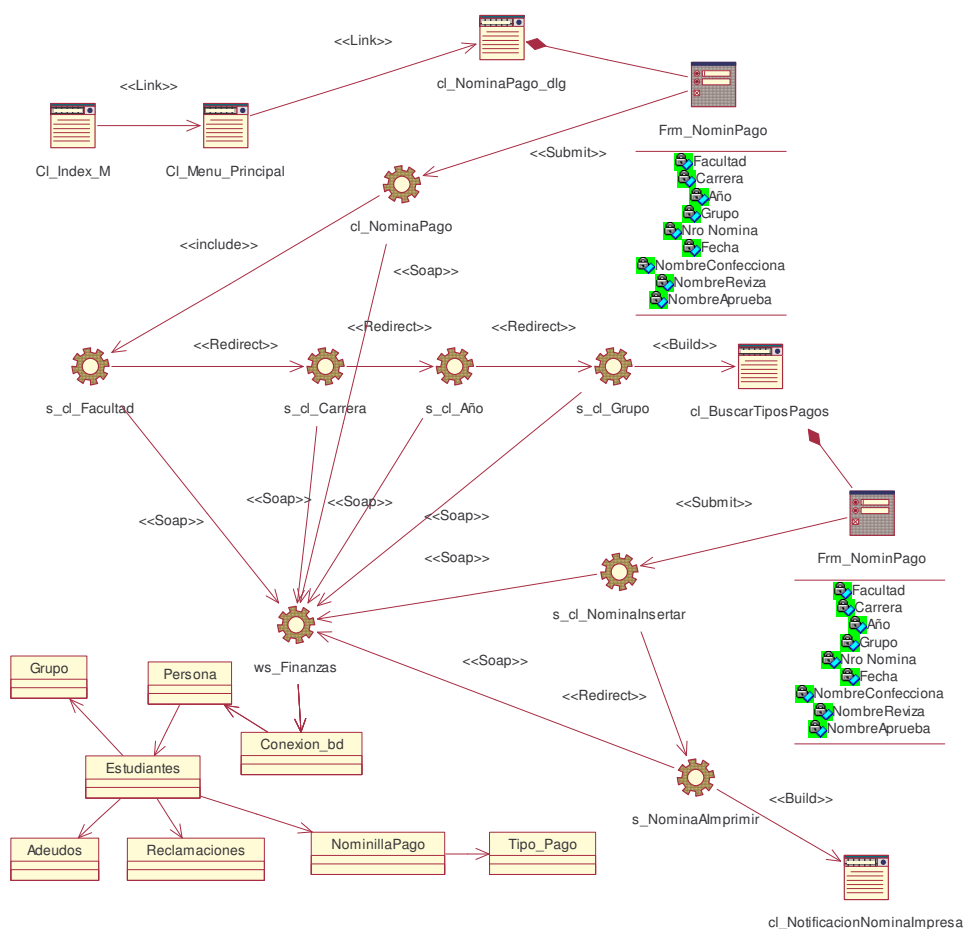
A.27- Importe total de dinero por facultades.



A.28- Visualizar Modelos



A.29- Mantener actualizada la información de cada nómina.



Anexo B. Prototipos.

Caso de uso actualizar la información de los estudiantes.

B.1- Insertar estudiante.

Secretaría de Facultad



Insertar estudiantes

Nombre		Nacionalidad	Seleccione v
Primer Apellido		Via Ingreso	Seleccione v
Segundo Apellido		Grupo	Seleccione un grupo v
Carnet de Identidad		(*) Campos Obligatorios	

Insertar

B.2- Modificar los datos de cada estudiante.

Secretaría de Facultad



Modificar datos a cada estudiante

Facultad : Informática

Nombre	Julio	Carnet de Identidad	79061212340
Primer Apellido	Sardinas	Nacionalidad	Cubano v
Segundo Apellido	Soca	Via Ingreso	Orden 18 v

Actualizar

B.3- Buscar un estudiante.

Buscar estudiante



Julio

Nombre

Buscar

Carnet	Nombre y Apellidos	Facultad	Carrera	Grupo
79061212340	Julio Sardinas Soca	Cultura Fisica	Cultura Fisica CRD	CFD51

B.4- Actualizar año y grupo a cada estudiante.

Secretaría de Facultad



Modificar año y grupo a un estudiante.

id	Nombre y Apellidos	Carrera	Año	Grupo
43	Julio Cesar Sardinas Soca	Cultura Fisica CRD	5 v	Nuevo Grupo v

Actualizar

Nuevo Grupo
CFD51

B.5- Dar alta, baja o graduar a cada estudiante.

☐	Nombre y Apellidos	Carrera	Año	Estado
☐	Kadir Hector Ortiz	Cultura Fisica CRD	5	✓ Alta
☐	Julio Cesar Sardinas Soca	Cultura Fisica CRD	5	✓ Alta
☐	Lisbet sardinas soca	Cultura Fisica CRD	5	✓ Alta
☐	Daylin Sosa Lopez	Cultura Fisica CRD	5	✓ Alta
☐	Lolo HHector OOrtiz	Cultura Fisica CRD	5	✗ Baja
☐	a a a	Cultura Fisica CRD	5	✗ Baja

✗
 ✓

B.5- Caso de uso insertar un grupo académico.

Secretaría de Facultad  Insertar grupos

Grupo	INF51
Carrera	Ingenieria Informatica
Año	5
Insertar	

Caso de uso mantener actualizado los tipos de pagos estudiantiles.

B.6- Asignar un tipo de pago a cada estudiante.

Secretaría de Facultad  Asignar un tipo de pago a un estudiante

Nombre y Apellidos	Tipo de Pago	Años	Fecha																																																
Kadir Hector Ortiz	Selecione Selecione Estipendio Orden-18 Ayudantia	Selecione	Mayo 2007 <table border="1"> <tr> <th>Sem</th> <th>Lun</th> <th>Mar</th> <th>Mie</th> <th>Jue</th> <th>Vie</th> <th>Sab</th> <th>Dom</th> </tr> <tr> <td>18</td> <td></td> <td>1</td> <td>2</td> <td>3</td> <td>4</td> <td>5</td> <td>6</td> </tr> <tr> <td>19</td> <td>7</td> <td>8</td> <td>9</td> <td>10</td> <td>11</td> <td>12</td> <td>13</td> </tr> <tr> <td>20</td> <td>14</td> <td>15</td> <td>16</td> <td>17</td> <td>18</td> <td>19</td> <td>20</td> </tr> <tr> <td>21</td> <td>21</td> <td>22</td> <td>23</td> <td>24</td> <td>25</td> <td>26</td> <td>27</td> </tr> <tr> <td>22</td> <td>28</td> <td>29</td> <td>30</td> <td>31</td> <td></td> <td></td> <td></td> </tr> </table>	Sem	Lun	Mar	Mie	Jue	Vie	Sab	Dom	18		1	2	3	4	5	6	19	7	8	9	10	11	12	13	20	14	15	16	17	18	19	20	21	21	22	23	24	25	26	27	22	28	29	30	31			
Sem	Lun	Mar	Mie	Jue	Vie	Sab	Dom																																												
18		1	2	3	4	5	6																																												
19	7	8	9	10	11	12	13																																												
20	14	15	16	17	18	19	20																																												
21	21	22	23	24	25	26	27																																												
22	28	29	30	31																																															

Hoy es Mar. 15. May 2007

B.7- Habilitar e Inhabilitar un tipo de pago a cada estudiante.

Secretaría de Facultad  Modificar un tipo de pago a un estudiante.

Buscar estudiante 

☐	Nombre y Apellidos	Carrera	Grupo	Descripción	Dinero	Estado
☐	Julio Cesar Sardinas Soca	Cultura Fisica CRD	CFD51	Estipendio	50	➕ Activo

B.8- Modificar un tipo de pago a cada estudiante

Secretaría de Facultad



Tipo de pago modificado

id	Nombre y Apellidos	Descripción	Año
43	Julio Sardinas Soca	Estipendio	2



Buscar otro estudiante

B.9- Caso de uso registrar reclamación(s) de cada estudiante.

Departamento de Finanzas



Buscar un estudiante para insertar su reclamo.

Buscar estudiante



Lisbet

Nombre



Buscar

Carnet	Nombre y Apellidos	Facultad	Carrera
2222222222	Lisbet sardinas soca	Cultura Fisica	Cultura Fisica CRD

B.10 –Caso de usos registrar estudiantes que no cobraron el último mes.

Departamento de Finanzas



Actualizar estudiantes que no cobraron

Facultad:	Humanidades	Carrera:	Estudios Socio Culturales
Año:	3	Grupo:	SOC31

Mayo

	Nro	Nombre y Apellidos	Mes	Estado
☐	64	Oslay Chamarro Hernandez	5	+ Cobro
☐	65	Yanedsis Fernandez Ramos	5	+ Cobro
☐	66	Vanessa Fernandez Bereau	5	+ Cobro
☐	67	Kenia Hernandez Gomez	5	- No cobro
☐	68	Maite Martinez Valladeres	5	+ Cobro
☐	69	Livania Perez Fernandez	5	+ Cobro
☐	70	Denise Sanchez Rodriguez	5	- No cobro
☐	71	Iyairys Toledo Fuentes	5	+ Cobro
☐	72	Maria Rodriguez Leyva	5	+ Cobro
☐	63	Jose Luis Bustillo Alejo	5	- No cobro



Inhabilitar



Habilitar

Caso de uso gestionar adeudo(s) de cada estudiante.

B.11- Registrar adeudo(s) de cada estudiante.

Departamento de Finanzas  Insertar adeudo de cada estudiante

Id	Nombre y Apellidos	Fecha	Adeudo(\$)
69	Livania Perez Fernandez	2007/05/21	20

Insertar

B.12- Actualizar adeudo de cada estudiante.

Departamento de Finanzas  Actualizar adeudo de cada estudiante

Id	Nombre y Apellidos	Adeudo	Mensualidad	Retirar
46	Yeniersy Dominguez Perez	60.00	130.00	30

Actualizar

B.13- Caso de uso visualizar los tipos de pago de los estudiantes.

Secretaria de Facultad  Modificar un tipo de pago a un estudiante.
 Buscar estudiante 

☐	Nombre y Apellidos	Carrera	Grupo	Descripcion	Dinero	Estado
☐	Julio Cesar Sardinas Soca	Cultura Fisica CRD	CFD51	Estipendio	50	 Activo

 Inhabilitar  Habilitar

B.14-Caso de uso visualizar las reclamaciones de cada grupo académico.

Departamento de Finanzas  Visualizar Reclamaciones

Facultad:	Humanidades	Carrera:	Estudios Socio Culturales
Año:	3	Grupo:	SOC31

Nro	Nombre y Apellidos	Importe(\$)	Fecha
63	Jose Luis Bustillo Alejo	80.00	2007-04-10

B.15-Caso de uso visualizar las fechas reclamadas por estudiante.

Departamento de Finanzas



Fechas reclamadas anualmente

Id	Nombre y Apellidos
43	Julio Cesar Sardinas Soca

Fechas Reclamadas

2007-03-15



Buscar otro estudiante

B.16-Caso de uso visualizar adeudo(s) de cada grupo académico.

Departamento de Finanzas



Ver Adeudos por Grupos

Facultad:	Cultura Fisica	Carrera:	Cultura Fisica CRD
Año:	5	Grupo:	CFD51

Nombre y Apellidos	Facultad	FechaInicial	Adeu Inic	Adeu Actual	Mensualidad
Yeniersy Dominguez Perez	Cultura Fisica	2007-03-23	60.00	60.00	130.00

B.17- Visualizar los préstamo(s) de cada grupo académico.

Departamento de Finanzas



Visualizar Préstamos

Facultad:	Humanidades	Carrera:	Estudios Socio Culturales
Año:	3	Grupo:	SOC31

	id	Nombre y Apellidos	Préstamo(\$)	Fecha	Estado
☐	68	Maite Martinez Valladeres	80.00	2007-05-17	Activo



Inhabilitar



Habilitar

B.18-Visualizar dinero pagado a cada facultad por conceptos.

Departamento de Finanzas



Total de dinero pagado por concepto a cada facultad

Facultad:	Humanidades	Concepto:	Estipendio
Mes:	Mayo	Aceptar	

Estipendio			
Facultad	Cant Estud	Importe	Total
Humanidades	9	75.00	675.00
Importe Total			\$675.00

B.19- Caso de uso total de dinero por reclamaciones.

Departamento de Finanzas



Total de dinero pagado por reclamaciones

Facultad:	Humanidades	Mes:	Abril
Aceptar			

Reclamaciones			
Facultad	Cant Estud	Importe	Total
Humanidades	1	80.00	80
Importe Total			80

B.20- Caso de uso total de dinero por reclamaciones no cobradas.

Departamento de Finanzas



Total de dinero por reclamaciones no cobradas

Facultad:	Humanidades	Mes:	Abril
Aceptar			

Reclamaciones no cobradas			
Facultad	Cant Estud	Importe	Total
Humanidades	2	80.00	160
Importe Total			160

B.21- Caso de uso visualizar solicitud de fondos por conceptos.

Departamento de Finanzas



Solicitud de fondos por conceptos

Facultad:	Humanidades	Concepto:	Estipendio
Mes:		Mayo	Aceptar

Estipendio				
Facultad	Anno	Cant Estud	Importe	Total
Humanidades	3	9	75.00	675.00
Importe Total				\$675.00

B.22- Caso de uso importe total de dinero por facultades.

Departamento de Finanzas 

Total del Dinero por Facultades

Mes:	Mayo	▼	Aceptar
------	------	---	---------

Total de dinero por Facultades		
Facultades	Cant Estud	Importe(\$)
Cultura Fisica	10	620.00.00
Humanidades	12	805.00.00
Importe Total(\$)		1425.00

Caso de usos Visualizar Modelos
B.23-Modelo de reclamación del estipendio

Universidad de Cienfuegos					
Modelo de Reclamación del Estipendio					
Carnet identidad:	79061212340 Buscar				
Nombre y Apellidos : Facultad: Especialidad: Año: Grupo:	Julio Cesar Sardinas Soca Cultura Fisica Cultura Fisica CRD 5 CFD51				
Firma:		Fecha		Fecha:	
Folio:		Firma Recibe :		Folio:	

B.24. Modelo de solicitud del préstamo estudiantil.

Ministerio de Educación Superior		Curso Diurno Solamente		Alta por solicitud de :		No	
Centro		Facultad		Especialidad			
I- Datos personales (Llenar a maquina de escribir o tinta , letra de molde) Orig. y copia							
Primer Apellido		Segundo Apellido		Nombre		Sex	Edad Est. Civil
Natural de: (Provincia y Municipio)		Inscripción SMG		Carnet Identidad		Nacionalidad	
- Dirección Particular							
Calle	No	Piso	Apto	Entrecalles:			
Reparto	Ciudad o Pueblo	Municipio	Provincia	País	Telef.		
Casa Propia	Casa de Huéspedes	Alquiler Mensual	Organizaciones	CDR	UJC		
De un familiar	Hotel				FMC	PCC	
II- Datos Docentes:							
Centro de Estudio precedente:		Provincia		Municipio		Indice Académico	
Becario anterior	Para la actual matricula	Curso	Año	Expedt e. No	Horario Lunes	De A Horario Miércoles	De A Horario Viernes
					Martes	Jueves	Sábado
Asignaturas pendientes		Del Año					

III- Datos del Núcleo Familiar

No	Nombre y Apellidos	Edad	Parentesco	Estado Civil	Centro de Estudio o Trabajo	Ingreso Mensual
----	--------------------	------	------------	--------------	-----------------------------	-----------------

1

2

Total.....

.....

Percápita (Resultado de dividir el ingreso total entre cantidad de componentes)

IV- Personas que dependen económicamente del solicitante fuera de su núcleo (Ídem Anterior)

1

2

Nombre y apellidos del Solicitante	Exp. Acum..	Año	Facultad	Especialidad
------------------------------------	-------------	-----	----------	--------------

V- Fundamentación. (Explique los motivos de su solicitud)

VI- Para solicitantes de préstamo estudiantil reintegrable solamente

¿Disponía de préstamo anteriormente?	Si N o	Centro donde lo recibía	Desde		Hasta	
			M	A	M	A

¿Subvencionado por la resolución 25867 del MINTRAB?	Si N o	Incorporado al movimiento de alumnos ayudantes	Si No	Préstamo mensual solicitado	Período de tiempo que solicita
---	--------------	--	----------	-----------------------------	--------------------------------

_____ de _____ de 20 _____
Firma del Solicitante

VII- Para uso de la facultad . Puede aprobar o negar (Proponer según tipo de solicitud)

Analizada la fundamentación
y los datos consignados se
decide...

Por un
Importe de

Con una
periodicidad

Desde

Hasta
D M A

Nombre

Cargo

Firma y Cuño

D M A

VIII-Observaciones (para la facultad)

(Para el Nivel Superior)

XI- Para uso del Nivel Superior (En las solicitudes que se requiera)

Analizada la fundamentación
y los datos consignados se
decide...

Por un
Importe de

Con una
periodicidad

Desde

Hasta
D M A

Nombre

Cargo

Firma y Cuño

D M A

Nombre y apellidos del Solicitante

Exp. Acum..

Año

Facultad

Especialidad

B.25- Modelo de contrato del préstamo estudiantil.

CONTRATO DE PRÉSTAMOS ESTUDIANTILES
ESTUDIANTES CURSOS REGULARES DIURNOS
CENTROS DE EDUCACION SUPERIOR.

Contrato.-----

I. DATOS PERSONALES

Nombre:

-----DIRECCION

PARTICULAR.

CALLE:----- Número-----entre-----y-----Rpto.o Finca--

-----Edificio-----Apto-----

CARNE DE IDENTIDAD: Serie y Numero:

Identidad permanente:-----

II.ACEPTACION DE LA DEUDA Y DE LAS CLAUSULAS DEL CONTRATO.

*Acepto la obligación de reintegrar al Banco Popular de ahorro o en su caso a la oficina de Cobros del Poder Popular, el importe total de las mensualidades que reciba por conceptos de prestamos estudiantil según la certificación que expida el Centro de Enseñanza Superior en base a las nómina firmadas por mí.
Dicho reintegro lo efectuare en no mas de ochenta y cuatro(84) mensualidades de no menos de veinte pesos(\$20.00) cada una.*

Asimismo acepto en todas sus partes las clausulas del presente contrato.

Fecha:-----Firma del deudor:

III:DATOS DEL CENTRO DE TRABAJO DEL DEUDOR.

Centro de pago:

organismo-empresa

Nombre:-----

Nombre.-----

Direccion-----

Direccion-----

IV.CERTIFICACION DEL CENTRO DE EDUCACION SUPERIOR.

-----en virtud de la facultad que me otorga la resolución conjunta MES-CEF-BPA-
del 6 de septiembre de 1990 en mi carácter del rector del CES-----
-----sito en-----y-----contador del
mismo.CERTIFICAMOS que la deuda contraida por el estudiante-----en el
periodo de -----a-----
---pesos\$() según consta en los registros contables de este centro docente.

Fecha:-----Firma del Rector:-----

Cuño del CES.

Firma del contador:-----

B.25 -Mantener actualizada la información de cada nómina.

B.26- - Caso de uso mantener actualizada la información de cada nómina.

Universidad de Cienfuegos	Nomina de Estipendio	Fecha		Nro Nom		Departamento de Finanzas
Facultad: Humanidades				Carrera: Estudios Socio Culturales		
Año: 3				Grupo: SOC31		

Nro	Nombre y Apellidos	Ext	Est	O-18	Pres	Ayud	Retro	Ade	Retirar	Importe	Firma
63	Jose Luis Bustillo Alejo	0.00	0.00	80.00	0.00	0.00	80.00	0.00	0.00	160.00	
64	Osley Chamorro Hernandez	0.00	75.00	0.00	0.00	0.00	0.00	20.00	20.00	55.00	
65	Yanedsis Fernandez Ramos	0.00	75.00	0.00	0.00	0.00	0.00	0.00	0.00	75.00	
66	Vanesa Fernandez Bereau	0.00	75.00	0.00	0.00	25.00	0.00	0.00	0.00	100.00	
67	Kenia Hernandez Gomez	0.00	75.00	0.00	0.00	25.00	0.00	0.00	0.00	100.00	
68	Maite Martinez Valladeres	0.00	75.00	0.00	80.00	0.00	0.00	0.00	0.00	155.00	
69	Livania Perez Fernandez	0.00	75.00	0.00	0.00	0.00	0.00	0.00	0.00	75.00	
70	Denise Sanchez Rodriguez	0.00	75.00	0.00	0.00	0.00	0.00	0.00	0.00	75.00	
71	Iyairys Toledo Fuentes	0.00	75.00	0.00	0.00	25.00	0.00	0.00	0.00	100.00	
72	Maria Rodriguez Leyva	0.00	75.00	0.00	0.00	0.00	0.00	0.00	0.00	75.00	
		0.00	675.00	80.00	80.00	75.00	80.00		20.00	970.00	

Confeccionado por: Margarita	Aprobado por: Damarys
Cargo: Especialista en Finanzas	Cargo: Directora Económica
Revizado por: Arelys	
Cargo: Especialista Principal en Finanzas	

Confeccionar

B.27- Caso de uso crear Usuario.

Carnet de Identidad: 64021612340 *

Nombre: Pepe *

Primer Apellido: Perez *

Segundo Apellido: Perez *

SubArea: Departamento Finanzas *

Cuenta: Pepe *

Clave: ***** *

Confirmar Clave: ***** *

Crear Cuenta

(*) Campos Obligatorios

B.28- Caso de uso iniciar Sesión.**Sistema de Gestión de Pago del Estipendio a Estudiantes Universitarios**

Introduzca su clave de acceso

Usuario		
Contraseña		
Aceptar		

B.29- Caso de uso cambiar contraseña.

Nombre: Daylin Sosa Lopez

Cuenta: DSLopez

Clave: *Confirmar Clave: ***B.30- Casos de usos: Habilitar usuario, inhabilitar usuario y asignar permisos a usuarios.**

Buscar Usuario

Buscar por



Nombre : Julio Sardinas Soca

Cuenta : JCesarSS

Activo: ☒

Asignar Permisos:

- ☒ Administrador
- ☐ Especialista en Finanzas
- ☐ Secretaria de Facultad

C.1 – Modelo lógico

