

Universidad de las Ciencias Informáticas

Facultad 4



Título: Análisis y diseño de un sistema automatizado para la entrega y control de los medios a estudiantes y profesores internos.

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

Autores: David Cabrera Gómez
Nimia I. Rodríguez Tabet

Tutor: Ing. Yunior A. González Nuzzo
Co-Tutora: Ing. Dinia Zayas Romero

Ciudad de la Habana
Junio 2008

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo al <nombre área> de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

David Cabrera Gómez

Nimia I. Rodríguez Tabet

Yunior A. González Nuzzo

AGRADECIMIENTOS

Nimia

A Maikel por haberme acogido en su casa aunque soy un desastre humano.

A mi tío Abigail y a Rebeca por todo el cariño brindado.

A mi piquete del pre: Anny, Yimy, Deno, la Yane, Alina, Dixo, Danne y a las mello por malcriarme, por haberme acompañado en el momento más malo, por darme el mejor cumple que he tenido y por todas las bachatas de la vocacional.

A mi little sis: viste mama, ya no me tienes que soportar más ni que coger lucha cuando no quiero comer algo y gracias por haber ido a mi casa.

A Yele por haberme soportado todo este tiempo y por haber estado conmigo en los momentos más difíciles en la escuela, ah, y lo de la casa también va contigo y al Yuni.

A mis padres Annia y Misael que aunque no están los quiero igual.

A mis buenas amistades de la escuela de todos estos años: Danay, Ilen, Lily, Chuchu, Lisney, Yoxcy, Carlos, Edwing, a la rata inmundada y presidiaria, Darwin, Ede y el Yoe, José Raúl y Jorge.

A danay: mama, ya no me vas a decir más M.

Al vieji, a Pipo y a Marlen por acogerme en su casa y tratarme como si fueran mi familia.

A María por creer en mí cuando yo ni creía que podía con algo, a Alexis y a Carmen por haberme tratado tan bien y aconsejarme.

A las chiquillas del 53105, Yu, Made, Yamisel, Anabel por las molestias causadas.

A mi tutor Yunior por toda la ayuda brindada con la tesis y por todas las veces que lo molesté.

A mi co-tutora Dinia que sin su ayuda nunca hubiese podido terminar la tesis.

A mi compañero de tesis David: viste papa, por fin nos graduamos.

A todos los que me han ayudado en todos estos años.

David

A Diniss, que sin ella no hubiese podido llegar hasta aquí, parte de lo que soy ahora, se lo debo, por su apoyo, y cariño ni sé por qué.

A Gretter, por dejarme ayudarla, como se pudo, y volver a rejuvenecerme con sus cosas, algo impresionante, por darme luz de nuevo, cuando ya no podía más, por ser única, y desmembrarme con valentía y originalidad.

A mi tutor, Yunior, por su apoyo interminable, algo que no tengo idea de porqué merecimos tener.

A la crema, un piquete único: el palomino (ingeniero ya), el pedal (tecla,-bestia), Maurice (El Pandion), el lanza (la mejor tumbadora y batá que voy a tener nunca), el lápiz (el gato), Herson, Maria Elena, Irina, Osly (que no está pero se siente igual) y Yicel, que la siento aquí mismo, a todos por enseñarme lo que puede hacer un piquete tan raro cuando se quiere.

Al piquete de Espirituanos, que ya no es tan Espirituano, ctorres, ymontesino, Bombino, Tito, El Willy, Jose (Mi sangre, siempre ahí cuando me hace falta el apoyo tan letal que nos damos), Johan y Maily, Eddy la bala y Yany, Maddiel y Carmona, El Frickie, Noel el no sé qué, Roberto Carlos, el primer pinareño industrialista que conozco, Jorgito, tarde pero uno de los buenos, Alexander, un casado de nacimiento, a todos no sé si gracias o la vida, pero mucho les debo.

A JC, por ser mi hermano tantos años aguantándome y comprendiéndome no sé como, y a mi mamá Mayté, por criarme y malcriarme tanto, principalmente este curso.

A mis tías Naryana, Haydéé, Cea y Nadia, por casi lo mismo que mi mamá, perdón por la preferencia, pero es mi madre.

A “El del traje gris” por acompañarme en todo, las flojeras, y las altas, aunque no hayamos logrado lo que siempre deseamos, lo que tú sabes, que por poco este curso, pero las mujeres son así, más cambiantes que la suerte. Al final siempre nos importó y nos importará un pito toda esa baba que dice Oliverio.

A la gente del doble, gente, son muchos no me obliguen a mencionarlos.

A la gente de pegada, El Coco, Edwin (El Tabicú), Orliandi, Jairol, el Juan Carlos de Jairol, Matos, Mas Flow, Rosendo, por dejarme adueñarme de su dominó tantas veces, a mi hijo Rigo.

A la gente vieja: Liso DJ (por apoyarme tanto, a pesar de no comunicar nada), Osley, que lo llevo siempre presente, El Bony, Lores, The Tiger, Yuricel, Yordanis, Angelino (por las pinturas, las canciones, y más) a todos, gracias por acompañarme tanto (Perdonen si se me queda alguno, es que son tantos).

A Lilita, “La Viajera”, mi sobrinita linda, que la quiero mucho.

A todo el piquete de trovadores, los mejores, Yuri, Jorgito (Medina) y Mairy lindita, Heriberto, que nunca viraste, Lorenzo, Carlos Felipe, Eliécer, que no lo veo hace siglos, pero ha sido mi mejor profesor no solo de programación, sino de la vida y muchas cosas más, de cómo tocar Sabina y más, Abelito el Beatle, Abelito la negrita con trenzas, a los que faltan, perdón, y a todos gracias, por compartir tantas canciones y poesía.

A Damián (Fuser), por ayudarme (principalmente con lo último) en todo, y sacarme de aprietos todo el tiempo y Arita lo mismo y más por ser tan linda conmigo.

A Dory por darme tanto sin conocerme aunque ella no lo sepa.

A La Pastela por momentos tan buenos y enseñarme mucho, pero mucho.

A Yanay, por su apoyo cuando estaba y cuando no.

A Ana Marys por su apoyo y su poesía inmortal.

A mi chuchundrita, por hacerme madurar un tilín con la genialidad de su persona y su figura de mujer “Ventosa”.

A Deigly, por todos los chats, y palabras justas y exactas.

A Yuri mi cañi, por tantos momentos tan lindos, y no permitirme mal llevar nuestra amistad, y al Dobo, gracias por quererla y cuidarla tanto.

A Leynita, por enseñarme tanto de humildad y de lo que es querer realmente, notas entre otras que llevo y llevaré siempre, por ser tan linda, por ser tan suave, por todo, te debo.

A la gente de San José, por ser tan especiales, Huguito, Im, Cesita, a todos, gracias, por todos los momentos, que no se acaban, me seguirán soportando.

A César Lage, por apoyarme desde siempre, desde años, eres un hermano, espero no se apague eso, por tus recortes de periódico en el pre, y aún más. El libro de Sabina, y su dedicatoria, y todo lo que pueda mencionar o agregar no te pagan.

A mi hijo, Irving, por salir mejor que yo, mucho mejor.

A mis primos, Juanky, por soportarme tanto durante toda la carrera y su paciencia, por mis visitas inescrupulosas e interesadas, y Aleco, que no se queda atrás.

A Juanca, Marcelo, Pedrito, por matarme el hambre, y enriquecerme con su amistad de lo mejor.

A Malcolm por enseñarme algo de música, no mucho, pero algo.

A todo el piquete de músicos locos, Yuri, José, mi tocayo, el mejor baterista de la UCI, Ernestico, otro grande de la percusión, Kenny, El Migue, el mejor tocando a Ringo, Toscano, mi maestro, Michel un cantante malísimo, pero decidido, al Tosco, un hermano.

Al Yoga, un hermano de siempre, al Yosho, un suertudo inigualable, y un hermano igual.

A Yaniris, por escuchar toda mi baba cíclica y obstinada.

A toda la gente buena de Aduana, Cobo, que no se me queda, y los demás.

A Yunier Garcés, un grande que espero salga adelante como se debe, porque puede y yo quiero, me enseñaron mucho.

A mis tías por su preocupación constante y su participación, casi directa en mi trabajo.

A la Yabó (Arellys), que sin su aporte no hubiese obtenido ningún avance en la tesis.

A mi mamá Migdalia por educarme tan bien.

DEDICATORIA

Nimia

A la memoria de mi abuela Rosalba que fue la persona que más me quiso y malcrió desde chiquita hasta mis 17 años.

A mis padres y a mi abuelo por haberme dado la dedicación, el cariño y el apoyo brindado todos estos años.

A mi papá Pancho por el cariño, por ayudarme y por ser tan comprensivo y a Mily.

A mis hermanos Neiza, Neiser y Meily que los quiero muchísimo.

A mi novio Daimar por los buenos momentos y por haberme dado una razón para querer ser responsable.

A mi abuelas Nica y Monga por estar siempre pendientes de mí y por el cariño todos estos años.

A mis tías Ángela y Milene por el cariño que me han dado.

A mis primos Richi, Úbelo y Sarita por ser más que mis primos hermanos mis hermanos.

A mi primito Viti, por darme tanta alegría y por ser mi niño lindo.

A la Guajira por ser como una madre para mí.

A mis hermanas de crianza Adisneysi y Eleinne, que aunque no nos veamos siempre nos seguimos queriendo igual y por ser fieles al pacto que hicimos y a Ana por aguantar todas mis locuras y quererme mucho.

David

A la memoria de mi primo Amaury, quien ha sido mi guía por mucho en silencio.

A mis padres por ser tanto, por ser todo, por traerme a este momento, por ser lo que no merezco, geniales.

A mi abuela Nere, por ser tan linda y tan fuerte por todos.

A la memoria de mis abuelos y abuela Ana.

A mi chuchitica linda, por darme todo lo que necesito para mantenerme vivo.

A Rosell por mostrarme muchas cosas, incluida ella.

A Neldys, Rosy, Neidy, Dai, mis ninfas de la buena suerte.

A mis tíos siempre tan atentos, pendientes.

A abuelito Rodolfo, cuya fortaleza eterna me hace temblar, y admirarlo cada día más.

A Juan Carlos (Mi tío), que aun me tiene presente desde lejos.

Al Micho, mi hermano arrancado de mi lado, que me apoyó mucho en los 5 años, y no puede estar.

A Abuelita Marina y Patato, por su cariño interminable.

RESUMEN

La gestión eficiente de la entrega y el control estricto de los materiales son cada vez más necesarios en la Universidad de las Ciencias Informáticas, debido a que se encuentran becados en su residencia una gran cantidad de personal, que incluye a estudiantes y profesores. El presente trabajo de diploma se refiere al desarrollo del análisis y diseño de un sistema automatizado para la entrega y control de los medios que se les entregan a estudiantes y profesores internos. En la actualidad estos medios son gestionados manualmente, siendo propensos: la aparición de desvío de recursos, el papeleo innecesario, la demora en la entrega de informaciones y recursos de importancia para decidir cuestiones importantes, el no conocimiento de determinadas situaciones, entre otros problemas, que se derivan de la carencia de un sistema automatizado que mantenga la fidelidad de los datos.

Esta propuesta de sistema, abarca la entrega de artículos de aseo, de avituallamiento, de módulos docentes, de libros de texto, así como los préstamos externos que se efectúan en la biblioteca a los internos de la Universidad. Se incluye también una propuesta para hacer menos compleja la tramitación de las bajas, pues actualmente es un proceso bastante extenso y que requiere de muchos subprocesos que lo entorpece. La generación de reportes que muestren el estado de alguno de estos procesos es otro de los requisitos que se tuvieron en cuenta a la hora de desarrollar este diseño. La implementación de este sistema significará un cambio notable en las áreas afectadas.

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	5
1.1 Introducción.....	5
1.2 Control de inventarios en el mundo.	5
1.3 Control de inventarios en Cuba.....	14
1.4 Control de inventarios en la UCI.	16
1.5 Control de avituallamiento en la UCI.....	17
1.6 Control de aseo personal en la UCI.	17
1.7 Tecnologías a utilizar	18
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA	38
2.1 Introducción.....	38
2.2 Objeto de estudio	38
2.3 Modelo del Negocio	40
2.4 Especificación de los requisitos del software.....	46
2.5 Definición de los casos de uso.....	56
CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA	76
3.1 Introducción.....	76
3.2 Análisis.....	76
3.3 Diseño	82
3.4 Arquitectura.....	97
3.5 Diseño de la BD	98
3.6 Definiciones del diseño que se apliquen.....	103
3.7 Tratamiento de errores.....	104
3.8 Seguridad.....	104
3.9 Interfaz	105
3.10 Concepción de la ayuda.....	105
3.11 Diagrama de Despliegue.....	106
3.12 Conclusiones.....	107
CONCLUSIONES	108
RECOMENDACIONES	109
BIBLIOGRAFÍA	110
GLOSARIO DE TÉRMINOS	112

INTRODUCCIÓN

Actualmente, la Universidad de las Ciencias Informáticas (UCI), en su sede central, cuenta con aproximadamente diez mil estudiantes y más de ochocientos profesores internos distribuidos en diez facultades docentes y más de cien edificios de residencia.

Para lograr que los estudiantes tengan acceso la mayor cantidad de materiales que le sirvan para su preparación como futuros profesionales de la informática, la dirección de la Universidad se ha preocupado por adquirir una gran variedad de libros de texto que se han puesto a disposición tanto de estudiantes, como de profesores, pues el claustro también debe estar bien preparado. La entrega de estos libros de texto se realiza en locales o pequeños almacenes habilitados para esto; pero existe una carencia de un sistema automatizado que se encargue de archivar los datos de los principales procesos que se llevan a cabo diariamente en dichos almacenes, principalmente la entrega y la recogida de libros de texto a los estudiantes o profesores que necesitan de ellos, o que permita obtener reportes. La realidad es que los inventarios se realizan utilizando herramientas de Ofimática, específicamente Microsoft Excel. Las consecuencias de esta situación se reflejan en el papeleo innecesario, puesto que por cada estudiante o profesor hay archivada una tarjeta donde se registran los libros que ha solicitado, entre otros datos de interés.

Otro de los medios que pone la Universidad a disposición de los estudiantes y los profesores, son los módulos de materiales docentes que les son entregados cada semestre. Este módulo contiene principalmente, materiales de oficina: libretas, bolígrafos, gomas de borrar, etc., y a la hora de gestionar la entrega de éstos, se encuentran los mismos problemas que se habían visto anteriormente con la entrega de los libros. Esta problemática es común para una gran parte de los procesos que tienen lugar en la Universidad; lo mismo ocurre con la gestión de préstamos externos de libros en la biblioteca, con la gestión de los módulos de avituallamiento, con la gestión del aseo personal, con la asignación y control de medios básicos de los apartamentos, en fin, que el flujo de información que se maneja tiende a tener errores, redundancias, demoras en las entregas; factores que facilitan la corrupción y el desvío de algunos de los recursos que se manejan.

Uno de los procesos más complejos que se realizan en la UCI es la tramitación de la baja: el interno, ya sea estudiante o profesor, debe recorrer varias oficinas de la Universidad para que se le compruebe que no debe ningún medio (entiéndase libros, artículos de avituallamiento, medios básicos), y en caso de que lo deba, debe ingresar una cantidad de dinero en efectivo o reponer el medio, en dependencia de lo que se haya acordado en el Acta de Responsabilidad Material. Este proceso resulta muy dificultoso, puesto que estas oficinas se encuentran muy alejadas entre sí. Y después de finalizado, el interno debe presentar el Modelo de Tramitación de Bajas, con todas las firmas y los cuños, para que se haga efectiva la baja correspondiente.

La informatización de la sociedad es una de las metas que se ha planteado la Universidad desde sus inicios, es por ello que una correcta solución a esta problemática sería el uso de una herramienta informática que se encargue de la gestión y control de los procesos antes mencionados. Pero para lograr esto, hay que realizar primeramente un diseño acertado para este software. Surge así el **problema**: ¿Cómo obtener el diseño de una aplicación para la gestión de los medios que se entregan a profesores internos y estudiantes en la UCI?

Para dar respuesta a la pregunta formulada, se desarrollará este trabajo de diploma que se plantea como **objeto de estudio**, los procesos de gestión y control de la información de los medios que se entregan al personal de diferentes centros; y como **campo de acción** los procesos de gestión de la información de los medios que se entregan a los estudiantes y profesores internos en la UCI.

Basándose en la **hipótesis** de que si se contara con el diseño de una herramienta los procesos del inventario o medios básicos, del aseo personal, del avituallamiento así como del módulo de libros de texto y los materiales docentes que se le entregan a los estudiantes y profesores internos, los préstamos externos de la biblioteca a internos, así como el proceso de baja, se logrará un modelo que permitirá implementar un sistema informático que logre optimizar la gestión de la información referente a los mismos y eliminar el papeleo innecesario; se plantea el **objetivo general** de este trabajo de diploma: diseñar un sistema para la gestión y control de los recursos que se le entregan a estudiantes y profesores internos.

Como **objetivos específicos** se obtuvieron los siguientes:

- Diseñar un módulo para la gestión de los inventarios o medios básicos que se entregan a los estudiantes y profesores internos.
- Diseñar un módulo para la gestión de avituallamiento que se entrega a los estudiantes y profesores internos.
- Diseñar un módulo para la gestión del aseo personal que se entrega a los estudiantes y profesores internos.
- Diseñar un módulo para la gestión de baja de estudiantes y profesores internos.
- Diseñar un módulo para la gestión de libros y materiales docentes.
- Diseñar un módulo para la gestión de los préstamos externos de la biblioteca.

Para dar cumplimiento a estos objetivos se han trazado las siguientes **tareas**:

- Investigar cómo se realiza el proceso de entrega de los medios básicos de un apartamento a estudiantes y profesores internos.
- Investigar cómo se realiza el proceso de entrega de avituallamiento a los estudiantes y profesores internos.
- Investigar cómo se realiza el proceso de entrega del aseo personal a estudiantes y profesores internos.
- Investigar cómo se realiza el proceso de entrega de materiales docentes a estudiantes y profesores internos.
- Investigar cómo se realiza el proceso de entrega del modulo de libros a estudiantes y profesores internos.
- Investigar cómo se realizan los procesos de gestión de estos medios dentro del centro.
- Investigar cómo se realiza el proceso de dar baja a un interno dentro del centro.
- Investigar cómo se realiza el proceso de los préstamos externos de la biblioteca a estudiantes y profesores internos.
- Realizar un análisis de las tecnologías existentes para el desarrollo de la aplicación.
- Realizar un análisis sobre las herramientas existentes para la implementación de la aplicación.
- Realizar al análisis y diseño de cada uno de estos módulos.

Estructura de la tesis

La tesis está estructurada por tres capítulos y estos a su vez por epígrafes. A continuación se hace una breve reseña sobre el contenido que se trata en cada uno de ellos.

El capítulo 1 expone el panorama de cómo se realizan el control de inventarios, de avituallamiento y aseo personal en el mundo, en Cuba y en dentro de la Universidad, haciéndose una exposición de que son los inventarios, sus principales funciones, las clasificaciones de los mismos y una breve explicación de algunos software de inventario en el mundo y en el país. Se tratan además conceptos básicos como el de ingeniería de software, proceso de desarrollo de software, la metodología RUP y se hace un resumen de las herramientas y tecnologías a utilizar para el desarrollo del trabajo.

El capítulo 2 expone las características del sistema y se tratan en él además, el objeto de estudio, el modelado del negocio, las especificaciones de los requerimientos de software y la definición de los casos de uso del sistema, incluyendo una breve reseña de los mismos así como de la justificación de los actores que intervienen en ellos.

En el capítulo 3 se trata el análisis y diseño del sistema, sus principales artefactos, así como una breve explicación de los diagramas y las clases que intervienen en ellos. Entre los principales artefactos a los que se hace alusión podemos encontrar el modelo del análisis, el modelo de diseño, la descripción de la arquitectura, el diseño de la base de datos y el modelo de despliegue, de manera general, se hace alusión además a otros temas de interés como las definiciones del diseño aplicadas, el tratamiento de errores, la seguridad, interfaz así como la concepción de la ayuda.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

En este capítulo se abordarán aspectos relacionados con el proceso de gestión de los distintos medios que se le otorgan a los estudiantes y profesores internos en la UCI, entre los que se encuentran el control de inventarios de los medios básicos de los apartamentos, control del aseo personal y el control de avituallamiento, aplicándolos al personal relacionado con esto en la Universidad, a partir de la información obtenida en las investigaciones y los conceptos fundamentales que de dicha información se derivan. Se tratarán además, las tecnologías y herramientas que se han de utilizar para desarrollar el diseño de la aplicación.

1.2 Control de inventarios en el mundo.

1.2.1 ¿Qué son los inventarios?

Los inventarios forman una parte muy importante para la contabilidad de mercancías, porque la venta del inventario es el corazón del negocio. El inventario es, por lo general, el activo mayor en sus balances generales, y los gastos por inventarios, llamados costo de mercancías vendidas, son usualmente el gasto mayor en el estado de resultados.

1.2.2 ¿Qué funciones cumple el Inventario?

En cualquier organización, los inventarios añaden una flexibilidad de operación que de otra manera no existiría. En fabricación, los inventarios de producto en proceso son una necesidad absoluta, a menos que cada parte individual se lleve de máquina a máquina y que éstas se preparen para producir una sola parte.

Funciones:

- Eliminación de irregularidades en la oferta
- Compra o producción en lotes o tandas
- Permitir a la organización manejar materiales perecederos
- Almacenamiento de mano de obra

1.2.3 Tipos De Inventario

- Inventario Físico
- Inventario de Anticipación
- Inventario de Lote o de Tamaño de Lote
- Inventario Estacionales
- Inventario Intermitente
- Inventario Permanente
- Inventario Cíclico

1.2.4 Software de inventarios.

En el mundo existen productos de software que sirven para gestionar el inventario, tanto personal como de medianas y grandes empresas. Muchos de ellos son propietarios por lo que no son de fácil adquisición para cualquier persona o empresa que necesite llevar el control de sus medios. Existe también un gran número de softwares que están disponibles en Internet para su adquisición gratis, lo que sería un elemento provechoso para aquellos que no tienen recursos, pero la utilización de los mismos, conlleva a que a los 30 días, sea necesaria la renovación del mismo y se pierda así la información antes gestionada. Una desventaja que presentan algunos softwares es que no son multiplataforma, es decir, que solo están implementados para correr sobre una plataforma, lo que hace que en muchos casos no sea posible la utilización de algunos de estos. Dentro de la gran gama de softwares de inventario se encuentran podemos mencionar: *SoftTrader, Silver Inventory System, MyHome Inventory System, ICash, SIMMS Software, Start EZ Inventory, Full Trust, Swellog Inventory, atACC, Golden Inventory System, VMSmart, BS/1 Enterprise, R6 Home Inventory, Atrex, Elohai Free Accounting, Asset Manage, Inventory Manager, MP* y muchos otros.

1.2.4.1 MP

El MP (Software de Mantenimiento Preventivo), es un software que informa sobre los trabajos de mantenimiento de una empresa. El mismo, posee una interfaz amigable, práctica y de fácil manejo que incluye 3 programas: Mantenimiento Preventivo, Inventario de Refacciones y Control de Herramientas.

Inventario de Refacciones.

El Inventario de Refacciones es un programa profesional diseñado especialmente para responder a las necesidades de muchas empresas que requieren una solución específica y práctica para el manejo óptimo de su almacén de refacciones y materiales de consumo interno, ofreciendo un entorno realmente amigable para el almacenero y personal de compras.

Para operar el Inventario de Refacciones el programa cuenta con poderosas herramientas de búsqueda en línea que permiten trabajar cómodamente sin necesidad de memorizar números de parte o complicados códigos para ejecutar las funciones del programa.

Un avanzado sistema de filtros y ordenamiento permiten desplegar en pantalla y visualizar la información tal y como la requiere el usuario. Para agilizar aún más su operación, el programa permite dar de alta productos, proveedores y catálogos auxiliares en línea, sin necesidad de abandonar o salir de un proceso. El Inventario de Refacciones permite mantener organizada y actualizada toda la información del almacén de materiales y refacciones, como existencias, costos, localización o incluso fotografías de sus productos. Permite además controlar eficientemente diferentes tipos de movimientos de entradas y salidas, así como evaluar el almacén por diferentes métodos de costeo. Permite también un registro detallado de sus proveedores, relacionar diferentes proveedores para un producto y la opción de manejar multi-almacenes. Incluye un poderoso módulo de compras que permite calcular el abastecimiento y organizar las compras por proveedor antes de emitir las órdenes de compra. El programa permite además llevar control sobre el estado de las órdenes de compra, informando sobre vencimientos en los plazos de entrega, arribos y entregas parciales.

El inventario de refacciones permite además:

- Asociación de imágenes o fotografías
- Catálogo detallado de proveedores con teléfono, extensión y correo electrónico de hasta 3 ejecutivos
- Capacidad de manejar multi-almacenes. Permite hacer traspasos entre almacenes y consultar existencias de un producto en los diferentes almacenes.
- Permite registrar diferentes tipos de movimientos de entrada y salida, foliando consecutivamente los documentos en forma corporativa o independiente.
- Detalles de entradas y salidas de cada producto
- Manejo de divisas
- Cálculo del abastecimiento
- Módulo para organizar las compras por proveedor
- Historial de movimientos
- Diferentes niveles de acceso para cada usuario
- Amplia variedad de reportes
- Respaldo automático de información
- Importación de información de otros sistema

Aunque este software proporciona una gran gama de opciones y facilidades, no permite o no implementa las funcionalidades necesarias para el proceso de gestión de la información del inventario de cada uno de los apartamentos de la universidad, ni permite hacer los reportes y recuperaciones de información por áreas: a nivel de universidad, de residencia, de facultad, de edificio y de apartamento, como se necesita en la universidad.

El programa de inventario de MP esta diseñado para llevar a cabo el control de una empresa que se dedique a la fabricación de equipos, no está diseñado para el control de medios básicos de un apartamento, por lo tanto, una de las características que tiene, como la de diferentes niveles de acceso para los usuarios, no se modela de manera que pueda servir en la aplicación para el control de estos medios, pues en MP esta funcionalidad se hace a nivel de rangos dentro de la empresa, y solo puede tener acceso a ella el personal con cargos dentro de la misma, tratando de aplicar esto al problema

presente, solo tendrían acceso a la información los trabajadores involucrados en el proceso de control de los medios básicos, lo que excluye a otros usuarios como los profesores y estudiantes que también necesitan realizar sus búsquedas dentro de la aplicación, para por ejemplo saber, en caso de la pérdida de uno de estos medios, a quién tiene que dirigirse y cuánto le costará la reposición del mismo.

1.2.4.2 ICash

Este es un software especialmente diseñado para llevar el inventario personal, que registra sus medios, cuentas, ingresos, así como transacciones bancarias. El mismo es de fácil uso, puesto que permite manejar la información aún sin tener conocimientos de contabilidad. Este proporciona todo un conjunto de herramientas para saber hacia dónde se mueven sus medios y su dinero. Es un software más bien enfocado a la gestión que a la contabilidad. Es un programa ideal para aplicar en clubes, asociaciones, trabajadores autónomos, pequeñas empresas o para gestionar la contabilidad doméstica. Presenta una interfaz amigable y de fácil utilización. Tiene soporte para divisas y editor de tipos de cambio. Permite además la generación de reportes detallados de un período determinado, así como la búsqueda por múltiples criterios, y está hecho para Macintosh y Windows, pero a pesar de brindar estas posibilidades, por ejemplo, ICash, podría no ser aplicable puesto a que el mismo no está implementado para correr sobre Linux, no es multiplataforma, y actualmente en la Universidad y en el país se está haciendo la migración hacia software libre, por lo tanto, esta es una característica que podría dificultar su aplicación. Así mismo, no posibilita la opción de diferentes niveles de acceso a la información a los usuarios, y la universidad necesita de una aplicación que lo permita para que cada uno de los involucrados en el proceso de control de los medios, tenga solamente acceso a la información concerniente a su nivel, evitando así el uso indebido de la misma por parte de estos. ICash, como está diseñado para la gestión de los medios domésticos así como de pequeñas empresas, no se podrían realizar con él las recuperaciones de información necesarias para la universidad, como por ejemplo, los listados de un determinado medio a nivel de universidad, a nivel de residencia o de un edificio, pues aplicándolo al problema presente, vendría siendo aplicable para gestionar solo la información de un apartamento.

1.2.4.3 VMSmart

VMSmart es un software diseñado para las pequeñas empresas que se adapta a las necesidades operacionales de las mismas con una fuerte orientación hacia la gestión de almacenes y de inventario. Incluye control de inventario, gestión de almacén, distribución, gestión de activos, y gestión de módulos de contacto.

Está definido y diseñado como multi-todo, permite gestionar múltiples clientes y almacenes y proporciona todas las herramientas necesarias para el acceso óptimo, grupo y tipo de filtro así como solo los datos relevantes para una determinada cuenta. Permite la agregación de un carrito de compras y la automatización del inventario personal y el almacén. Permite además la presentación de informes detallados, módulos de órdenes de venta y factura, y la integración de una gama de funciones para la gestión de la información. VMSmart está implementado únicamente para plataformas Windows, no está implementado para Linux o Macintosh, lo que le traería problemas para su utilización a múltiples usuarios debido a la fuerte tendencia que hay de la utilización de software libre, y en particular, en el país se está migrando hacia Linux al igual que en la universidad, por lo tanto, restringiría bastante el área de aplicación del mismo. Por otra parte, este software no está diseñado para el control de grandes volúmenes de información, puesto a que ha sido creado para las pequeñas empresas, y aplicando el mismo dentro de la universidad, para el control del inventario, no solo no podríamos hacer todas las recuperaciones de información pertinentes en las distintas áreas, sino que no habría una correcta protección de la información y todo el mundo tendría acceso a la misma, y en la universidad se necesita de una aplicación que además de ser capaz de generar los reportes desde cualquier tipo de área, pueda gestionar además, el acceso de los distintos usuarios a la información, con el objetivo de que no vaya a existir un uso indebido de la misma.

1.2.4.4 atACC

atACC es útil para cualquier pequeña o mediana empresa. Es de fácil utilización e instalación y proporciona dos elementos: Contabilidad e Inventario. Brinda las funciones de inversión de cálculo, la transacción en moneda extranjera. Ofrece soporte para múltiples divisas y dispone de una función para obtener los tipos de cambio en Internet. Tiene opciones para la inicialización de cuentas, productos,

servicios, lotes y el presupuesto y un menú detallado de inventario de las transacciones. Se pueden generar gran cantidad de informes con representación grafica de datos y además presenta funciones especializadas para inventario. A pesar de esto, este software no proporciona o no es capaz de generar primeramente los reportes o recuperaciones de información que son necesarias para la universidad, que son desde los medios básicos presentes en un apartamento hasta los existentes a nivel de universidad, los medios entregados a un estudiante determinado, ni gestionar la boleta de baja y orientar tanto al estudiante como al profesor de qué hacer con sus medios o en caso de alguna pérdida a qué lugar debe dirigirse para solucionar su problema. Además no presenta la opción de la validación de distintos niveles de información para cada usuario, y esto podría a conllevar a un uso indebido de la misma.

1.2.4.5 BS/1 Enterprise

BS/1 Enterprise es un sistema multi-moneda que incluye libro mayor, cuentas por pagar, cuentas por cobrar, inventario, órdenes de compra, órdenes de venta y cotizaciones, análisis de venta. Crea facturas automáticamente a partir del ingreso de los datos. Muestra informes de inventario, niveles de compra y de pedidos y ventas. El hecho de que es multi-moneda facilita la compra y venta de productos a nivel internacional, y realiza el tipo de cambio a moneda nacional así como los ingresos y gastos una vez hecha una transacción, y muestra informes detallados de los inventarios. Este software está diseñado sólo para versiones de Windows, por lo que hace que pueda ser dificultosa su utilización dada la situación de la migración hacia las plataformas de software libre en el país y en la universidad como Linux. Además, este software no proporciona o no es capaz de generar primeramente los reportes o recuperaciones de información que son necesarias para la universidad, que son desde los medios básicos presentes en un apartamento hasta los existentes a nivel de universidad, los medios entregados a un estudiante determinado, ni gestionar la boleta de baja y orientar tanto al estudiante como al profesor de que hacer con sus medios o en caso de alguna pérdida a que lugar debe dirigirse para solucionar su problema y no presenta la opción de la validación de distintos niveles de información para cada usuario.

1.2.4.6 Golden Inventory System

Este es un sistema completo de inventario que realiza las funciones de compra, venta y pago. Crea listas de productos, de proveedores, órdenes de compra, las órdenes de venta, las órdenes de pago, las facturas. Realiza órdenes de transferencia entre lugares, vendedores y clientes y saldos y diversos tipos

de informes para el seguimiento del negocio. Jerarquiza grupos de productos, mejora las funciones de importación y exportación de información al sistema, es multi-moneda y brinda fecha de caducidad para los últimos productos. A pesar de esto, este software está diseñado sólo para versiones de Windows, por lo que hace que pueda ser dificultosa su utilización dada la situación de la migración hacia las plataformas de software libre en el país y en la universidad. Además, este software no proporciona o no es capaz de generar primeramente los reportes o recuperaciones de información que son necesarias para la universidad, que son desde los medios básicos presentes en un apartamento hasta los existentes a nivel de universidad, los medios entregados a un estudiante determinado, ni gestionar la boleta de baja y orientar tanto al estudiante como al profesor de que hacer con sus medios o en caso de alguna pérdida a que lugar debe dirigirse para solucionar su problema y no presenta la opción de la validación de distintos niveles de información para cada usuario.

1.2.4.7 MyHome Inventory System

Software ideal para organizaciones pequeñas o medianas, no para grandes organizaciones. Es un sistema de dirección de inventario completo que puede realizar el control de existencias para un depósito. Constituye una respuesta a cualquier problema de dirección de inventario y es de fácil uso e instalación. Genera facturas, reportes de importación y exportación. Genera más de 200 tipos de informes diferentes. El sistema muestra las etiquetas de Producto junto con sus códigos de barras, clientela y equilibrios de vendedores, órdenes de transferencia entre las posiciones diferentes. Algunos aspectos únicos incluyen cosas como la capacidad de crear transacciones de planificación y hacer una réplica de facturas almacenadas. Es muy poderoso y envasado por aspectos aún esto se jacta de simplicidad y facilidad del uso. El sistema también ha incorporado la seguridad de nivel; éste rechaza a cualquier persona externa aprovechar sus bases de datos. A pesar de esto, este software está diseñado sólo para versiones de Windows, por lo que hace que pueda ser dificultosa su utilización dada la situación de la migración hacia las plataformas de software libre en el país y en la universidad. Además, este software no proporciona o no es capaz de generar primeramente los reportes o recuperaciones de información que son necesarias para la universidad, que son desde los medios básicos presentes en un apartamento hasta los existentes a nivel de universidad, los medios entregados a un estudiante determinado, ni gestionar la boleta de baja y orientar tanto al estudiante como al profesor de que hacer con sus medios o en caso de alguna pérdida a

que lugar debe dirigirse para solucionar su problema y no presenta la opción de la validación de distintos niveles de información para cada usuario, y esto podría conllevar a un uso indebido de la misma.

1.2.4.8 Silver Inventory System

Este es un software de bajo costo para la gestión de ventas, compras y pagos. Crea de facturas, órdenes de venta, pedidos de compra, recepción de las listas, recibos de pago, las etiquetas de los productos con códigos de barras y cualquier tipo de los informes de la empresa. Permite el control de los saldos y proveedores. Es de una interfaz intuitiva, puesto a que puede realizar las operaciones sin necesidad de leer la ayuda, utiliza 20 caracteres alfanuméricos para los códigos de los productos, calcula los precios de venta y lista de precios y costes. Usa las descripciones de los productos por separado para las facturas y órdenes de compras. Calcula el tiempo real para equilibrio entre el cliente y el proveedor y actualiza el inventario en el momento en que las facturas o pedidos están entrando. Son configurables el aspecto de las facturas, las órdenes de pago y las recepciones y puede imprimir cualquiera de los informes relacionados con la información manejada. A pesar de esto, este software está diseñado sólo para versiones de Windows, por lo que hace que pueda ser dificultosa su utilización dada la situación de la migración hacia las plataformas de software libre en el país y en la universidad. Además, este software no proporciona o no es capaz de generar primeramente los reportes o recuperaciones de información que son necesarias para la universidad, que son desde los medios básicos presentes en un apartamento hasta los existentes a nivel de universidad, los medios entregados a un estudiante determinado, ni gestionar la boleta de baja y orientar tanto al estudiante como al profesor de que hacer con sus medios o en caso de alguna perdida a que lugar debe dirigirse para solucionar su problema y no presenta la opción de la validación de distintos niveles de información para cada usuario.

1.2.4.9 SIMMS

Este software ofrece control de los datos relativos a la recepción de mercancías, el movimiento de mercancías a la recepción de las mercancías, el movimiento de mercancías dentro o entre las ubicaciones, la venta, disposición de bienes, y la valoración precisa de las mercancías y la condición de permanecer en el inventario en un momento dado en el tiempo. SIMMS de control de inventarios de software ha sido diseñado para acomodar las necesidades de las grandes y pequeñas empresas. SIMMS hace más que simplemente automatizar su proceso de control de inventario, también ofrece la gestión de

los materiales sólidos. A pesar de esto, este software está diseñado solo para versiones de Windows, por lo que hace que pueda ser dificultosa su utilización dada la situación de la migración hacia las plataformas de software libre en el país y en la universidad. Además, este software no proporciona o no es capaz de generar primeramente los reportes o recuperaciones de información que son necesarias para la universidad, que son desde los medios básicos presentes en un apartamento hasta los existentes a nivel de universidad, los medios entregados a un estudiante determinado, ni gestionar la boleta de baja y orientar tanto al estudiante como al profesor de que hacer con sus medios o en caso de alguna pérdida a que lugar debe dirigirse para solucionar su problema y no presenta la opción de la validación de distintos niveles de información para cada usuario.

1.2.4.10 SoftTrader

SoftTrader es una completa gestión de los inventarios y cuentas de servicios. Permite tomar el control total sobre el inventario y la facturación. Analiza las facturas y hace el seguimiento de las ventas. Realiza el balance general. Permite almacenar varios precios para un solo producto, útiles para el comercio al por menor. A pesar de esto, este software está diseñado sólo para versiones de Windows, por lo que hace que pueda ser dificultosa su utilización dada la situación de la migración hacia las plataformas de software libre en el país y en la universidad. Además, este software no proporciona o no es capaz de generar primeramente los reportes o recuperaciones de información que son necesarias para la universidad, que son desde los medios básicos presentes en un apartamento hasta los existentes a nivel de universidad, los medios entregados a un estudiante determinado, ni gestionar la boleta de baja y orientar tanto al estudiante como al profesor de que hacer con sus medios o en caso de alguna pérdida a que lugar debe dirigirse para solucionar su problema y no presenta la opción de la validación de distintos niveles de información para cada usuario.

1.3 Control de inventarios en Cuba.

El control de inventarios en Cuba registra todos los documentos primarios que tradicionalmente generan las entradas y salidas en los almacenes; además tiene incorporada una gran cantidad de opciones que permiten ejecutar los conteos físicos y el tratamiento de los diferentes tipos de inventarios.

1.3.1 Software de Inventarios en Cuba

1.3.1.1 VERSAT SARASOLA

El Versat Sarasola constituye el primer software certificado en Cuba para la planificación económica, el control y el análisis de gestión. El mismo, está diseñado para ser empleado en cualquier tipo de entidad empresarial o presupuestada. Permite llevar a cabo el control y registro contable e individual de todos los hechos económicos que se originan en las estructuras internas de las entidades, así como exponer el estado financiero y toda la información económica y contable, incluyéndose el inventario de una entidad. Se estructura en un grupo de subsistemas en los cuales se procesan y contabilizan los documentos primarios, donde se anotan los movimientos, los recursos materiales, laborales y financieros que se utilizan en una entidad. Logra establecer un proceso de interacción usuario-sistema y posee una gran rapidez y agilidad a partir de la configuración del proceso de contabilización de los documentos primarios y de las propias posibilidades de trabajo contenidas en cada subsistema. A pesar de esto, y de la orientación en el país de la utilización del mismo por cada entidad, no consta es un software que se utiliza con fines de llevar los medios a nivel de empresa o entidad, es decir, que con la aplicación del mismo no se podría llevar el inventario de un apartamento en específico, o de un determinado edificio, por citar algún ejemplo. Tampoco consta de la opción de distintos niveles de acceso a la información para los usuarios, por lo que con la aplicación del mismo, podría ejercerse un uso indebido de la misma. Como es un software diseñado para propósitos generales, no se podrían hacer las recuperaciones de información necesitadas por la universidad, es decir, si se deseara conocer los medios básicos pertenecientes a un área determinada, no se podría hacer, además de que no gestionaría la boleta de baja de un estudiante o profesor, en la cual se hace referencia a la entrega de los medios que ha puesto la universidad en sus manos, siendo esto una de las principales funcionalidades necesarias.

1.3.1.2 AT06

Este software consta de muchas funcionalidades y brinda un gran número de facilidades para llevar a cabo el proceso de inventario entre las que podemos destacar que permite la captura y/o actualización, permite revisar y realizar los movimientos del día, el inventario físico, la carga inicial del inventario, inicializar datos históricos, realizar el registro de usuarios, registro de datos procesados, el registro de entradas y salidas, la conciliación por almacenes, la tarjeta de submayor de inventario, las existencias de productos, los valores de los inventarios, los comprobantes contables, el resumen de movimientos,

documentos emitidos por contrapartidas, documentos por cuenta de inventario, productos con lento movimiento, registro de facturas emitidas, movimientos de productos por contrapartidas, visualizar los documentos procesados, permite hacer codificaciones de productos, de existencias por almacén, de unidades contables, de familias contables, de almacenes, custodios y entidades, cuentas contables, subelementos de gastos y orígenes y destinos, permite exportar comprobantes, verificar y corregir información así como recalculan el submayor y hacer ajustes en él, hacer saneamiento de ficheros, unificación de códigos, variaciones de precios, captura de nuevos precios, listado por valor de inventario, listado por costo de inventario, revalorizar costo de inventario, reconstruir ficheros índices, borrar registros marcados, importar datos de versiones anteriores de este programa, y posee además, herramientas como la calculadora. Para el caso de este software, el registro de usuarios, se implementa de una manera parecida a la que se necesita en la universidad, pues una vez que se crean los usuarios y el administrador le da los permisos, sólo tiene acceso a la información del nivel que le corresponde, pero no se puede por ejemplo, utilizar esto para una gran cantidad de usuarios como lo requiere la universidad, pues se requiere de que todos los usuarios tanto estudiantes como profesores que sean internos puedan acceder a la aplicación y ver los medios básicos de su apartamento, además, esta aplicación está diseñada para tratar información a nivel de empresa, no al nivel tan específico como un apartamento.

1.4 Control de inventarios en la UCI.

El control de los inventarios se realiza en el departamento de economía de la universidad, el cual divide las distintas áreas de la beca en residencias. Existen 3 residencias estudiantiles, la cuales están formadas por un grupo de facultades, en la residencia numero 1 se encuentran la beca de las facultades 1,2 y 3; en la residencia número 2 se encuentran las facultades 4, 5, 6, y 7 y por último en la residencia número 3 se encuentran las facultades 8, 9,10. Existe además una residencia de profesores. En el departamento de economía de cada una de las residencias antes mencionadas, se lleva a cabo el control de inventario, a nivel de facultades; y dentro de cada facultad, se controlan los medios que existen en cada edificio perteneciente a la misma. En el control que se lleva a cabo en cada edificio, las instructoras controlan los medios básicos que tiene cada apartamento a través del inventario o Acta de Entrega de Recursos y Activos de los Dormitorios de la Residencia Universitaria para los Estudiantes. Entre estos medios se encuentran: literas, bastidores, mesa de estudio, mesa de comedor, mesa de computadora, sillas, taquillas, gaveteros, butacas, sofás, mesa de centro, espejo de tocador, espejo de baño, libreros, soporte

de TV, televisor, CPU, teclado de computadora, monitor, back up, estabilizador de línea, Mouse, dispensador de agua, lámparas recargables, mando de TV, antena de TV, teléfono, ventilador de pie, ventilador de pared, ventilador de mesa, minibar, refrigerador, cable de red, pomos de agua, jarras, vasos, colchones, almohadas, cortina de baño, encendedores de 20w, encendedores de 40 w, pilas de agua, tapa de tanque de baño, cubo grande, cubo chico, escoba, trapeador, haragán, cestos de papeles, recogedor, cortijero de baño, extensiones eléctricas, lámparas de 40 w, lámparas de 20 w, tubos de luz de 40 w, tubos de luz de 20 w, interruptores, tomacorrientes, consola de aire acondicionado, aire acondicionado, cable coaxial, tapas de tomacorrientes, platos y llaves.

1.5 Control de avituallamiento en la UCI.

El control de avituallamiento se realiza en el departamento de economía de la universidad, el cual para un mejor control del mismo, divide las distintas áreas de la beca en residencias. Existen 3 residencias estudiantiles, la cuales están formadas por un grupo de facultades, en la residencia número 1 se encuentran la beca de las facultades 1,2 y 3; en la residencia número 2 se encuentran las facultades 4, 5, 6, y 7 y por último en la residencia número 3 se encuentran las facultades 8, 9,10. Existe además una residencia de profesores. El control de avituallamiento se lleva a cabo en el área de lavandería de cada una de las residencias antes mencionadas mediante la Tarjeta de Control de Entrega y Recepción de Avituallamiento. Los medios que se entregan son sábanas, toalla, colcha, percheros, almohada y funda.

1.6 Control de aseo personal en la UCI.

El control del aseo personal se realiza en el departamento de economía de la universidad, el cual para un mejor control del mismo, divide las distintas áreas de la beca en residencias.

Existen 3 residencias estudiantiles, la cuales están formadas por un grupo de facultades, en la residencia número 1 se encuentran la beca de las facultades 1,2 y 3; en la residencia número 2 se encuentran las facultades 4, 5, 6, y 7 y por último en la residencia número 3 se encuentran las facultades 8, 9,10. Existe además una residencia de profesores. En el departamento de economía de cada una de las residencias antes mencionadas se lleva a cabo el control del aseo, específicamente el aseo que se entrega a cada uno de los edificios pertenecientes a esta residencia, y dentro del edificio, la instructora tiene el control del aseo que se le da a cada uno de los estudiantes mediante la Tarjeta de Control de Entrega del Módulo de

Aseo. Los productos que se entregan son: jabón de baño, jabón de lavar, pasta dental, desodorante, máquina de afeitar, papel higiénico, almohadilla sanitaria, shampoo, detergente y condones.

1.7 Tecnologías a utilizar

1.7.1 Arquitectura Cliente/Servidor.

El esquema cliente-servidor "es un modelo de computación en el que el procesamiento requerido para ejecutar una aplicación o conjunto de aplicaciones relacionadas se divide entre dos o más procesos que cooperan entre sí. Usualmente la mayoría del trabajo pesado se hace en el proceso llamado servidor y el (los) proceso(s) cliente(s) sólo se ocupa de la interacción con el usuario (aunque esto puede variar). Los principales componentes del esquema cliente-servidor son entonces los Clientes, los Servidores y la infraestructura de comunicaciones.

Los Clientes interactúan con el usuario, usualmente en forma gráfica. Frecuentemente se comunican con procesos auxiliares que se encargan de establecer conexión con el servidor, enviar el pedido, recibir la respuesta, manejar las fallas y realizar actividades de sincronización y de seguridad.

Los Servidores proporcionan un servicio al cliente y devuelven los resultados. En algunos casos existen procesos auxiliares que se encargan de recibir las solicitudes del cliente, verificar la protección, activar un proceso servidor para satisfacer el pedido, recibir su respuesta y enviarla al cliente. Además deben manejar los interbloques, la recuperación ante fallas, y otros aspectos afines. Por las razones anteriores la plataforma computacional asociada con los servidores es más poderosa que la de los clientes. Por esta razón se utilizan PCs poderosos, estaciones de trabajo, mini computadores o sistemas grandes. Además deben manejar servicios como administración de la red, mensajes, control y administración de la entrada al sistema ("login"), auditoría y recuperación y contabilidad. Usualmente en los servidores existe algún tipo de servicio de bases de datos.

Para que los clientes y los servidores puedan comunicarse se requiere una infraestructura de comunicaciones, la cual proporciona los mecanismos básicos de direccionamiento y transporte. La mayoría de los sistemas Cliente/Servidor actuales se basan en redes locales y por lo tanto utilizan protocolos no orientados a conexión, lo cual implica que las aplicaciones deben hacer las verificaciones. La red debe tener características adecuadas de desempeño, confiabilidad, transparencia y administración.

Ventajas:

Uno de los aspectos que más ha promovido el uso de sistemas Cliente/Servidor es la existencia de plataformas de hardware cada vez más baratas. Esta constituye a su vez una de las más palpables ventajas de este esquema, la posibilidad de utilizar máquinas considerablemente más baratas que las requeridas por una solución centralizada, basada en sistemas grandes. Además de lo anterior, se pueden utilizar componentes, tanto de hardware como de software, de varios fabricantes, lo cual contribuye considerablemente a la reducción de costos y favorece la flexibilidad en la implantación y actualización de soluciones.

Además de lo anterior, el esquema Cliente/Servidor facilita la integración entre sistemas diferentes y compartir información, permitiendo, por ejemplo que las máquinas ya existentes puedan ser utilizadas pero utilizando interfaces más amigables al usuario. De esta manera podemos, por ejemplo, integrar PCs con sistemas medianos y grandes, sin que todas las máquinas tengan que utilizar el mismo sistema operacional.

Al favorecer el uso de interfaces gráficas interactivas, los sistemas construidos con este esquema tienen una interacción más intuitiva con el usuario. Si se utilizan interfaces gráficas para interactuar con el usuario, el esquema Cliente/Servidor presenta la ventaja, con respecto a uno centralizado, de que no es siempre necesario transmitir información gráfica por la red pues ésta puede residir en el cliente, lo cual permite aprovechar mejor el ancho de banda de la red.

Una ventaja adicional del uso del esquema Cliente / Servidor es que es más rápido el mantenimiento y el desarrollo de aplicaciones pues se pueden emplear las herramientas existentes (por ejemplo los servidores de SQL o las herramientas de más bajo nivel como los sockets o el RPC). La estructura inherentemente modular facilita además la integración de nuevas tecnologías y el crecimiento de la infraestructura computacional, favoreciendo así la escalabilidad de las soluciones. El esquema Cliente/Servidor contribuye además a proporcionar a los diferentes departamentos de una empresa soluciones locales, pero permitiendo además la integración de la información relevante a nivel global.

1.7.2 Lenguajes de programación.

Desde los inicios de Internet, fueron surgiendo diferentes demandas por los usuarios y se dieron soluciones mediante lenguajes estáticos. A medida que pasó el tiempo, las tecnologías fueron desarrollándose y surgieron nuevos problemas a dar solución. Esto dio lugar a desarrollar lenguajes de programación para la Web dinámicos, que permitieran interactuar con los usuarios y utilizaran Sistemas de Bases de Datos Actualmente existen diferentes lenguajes de programación para desarrollar en la Web, estos han ido surgiendo debido a las tendencias y necesidades de las plataformas. Cada uno de ellos posee diferentes características que lo hacen útiles para desarrollar determinadas aplicaciones. Estos se clasifican de acuerdo a la arquitectura Cliente/Servidor en:

- ✚ Lenguajes del lado del servidor: son aquellos reconocidos, ejecutados e interpretados por el propio servidor y que se envían al cliente en un formato comprensible para él entre los que se encuentran PHP, ASP, JSP, Perl, Python y Ruby.
- ✚ Lenguajes del lado del cliente: son aquellos que pueden ser directamente comprendidos por el navegador como el VBScript, el Javascript y Ajax. Estos dos últimos son incluidos en el código HTML.

Ambos tienen ventajas y desventajas puesto que un lenguaje del lado del cliente es totalmente independiente del servidor, lo cual permite que la página pueda ser albergada en cualquier sitio mientras que un lenguaje del lado servidor es independiente del cliente por lo que es mucho menos rígido respecto al cambio de un navegador a otro o respecto a las versiones del mismo; por otra parte, los *scripts* son almacenados en el servidor quien los ejecuta y traduce a HTML por lo que permanecen ocultos para el cliente.

A continuación se presentan las características de algunos de los lenguajes de programación para la Web de los cuales se escogerá el más idóneo según las necesidades actuales de la universidad y teniendo en cuenta sus ventajas y desventajas.

Del lado del cliente

HTML

Es un lenguaje estático para el desarrollo de sitios Web (acrónimo en inglés de HyperText Markup Language, en español Lenguaje de Marcas Hipertextuales). Desarrollado por el World Wide Web Consortium (W3C). Los archivos pueden tener las extensiones (htm, html).

Ventajas:

- Sencillo que permite describir hipertexto.
- Texto presentado de forma estructurada y agradable.
- No necesita de grandes conocimientos cuando se cuenta con un editor de páginas Web o WYSIWYG.
- Archivos pequeños.
- Despliegue rápido.
- Lenguaje de fácil aprendizaje.
- Lo admiten todos los exploradores.

Desventajas:

- Lenguaje estático.
- La interpretación de cada navegador puede ser diferente.
- Guarda muchas etiquetas que pueden convertirse en “basura” y dificultan la corrección.
- El diseño es más lento.
- Las etiquetas son muy limitadas.

Javascript

Javascript fue creado por Brendan Eich en la empresa Netscape Communications, es un lenguaje que puede ser utilizado por profesionales y para quienes se inician en el desarrollo y diseño de sitios Web.

No requiere de compilación ya que el lenguaje funciona del lado del cliente, los navegadores son los encargados de interpretar estos códigos. Gracias a su compatibilidad con la mayoría de los navegadores modernos, es el lenguaje de programación del lado del cliente más utilizado. Es similar a Java, aunque no es un lenguaje orientado a objetos, el mismo no dispone de herencias, más bien se podría decir que es un lenguaje interpretado, basado en prototipos. Entre las acciones típicas que se pueden realizar en Javascript se tienen dos vertientes. Por un lado los efectos especiales sobre páginas Web, para crear contenidos dinámicos y elementos de la página que tengan movimiento, cambien de color o cualquier otro dinamismo. Por el otro, Javascript permite ejecutar instrucciones como respuesta a las acciones del usuario, con lo que se puede crear páginas interactivas con programas como calculadoras, agendas, o tablas de cálculo. Javascript es un lenguaje con muchas posibilidades, permite la programación de pequeños scripts, pero también de programas más grandes, orientados a objetos, con funciones, estructuras de datos complejas, etc. Además, Javascript pone a disposición del programador todos los elementos que forman la página Web, para que éste pueda acceder a ellos y modificarlos dinámicamente. Este lenguaje posee varias características, entre ellas se pueden mencionar que es un lenguaje basado en acciones que posee menos restricciones. Además, es un lenguaje que utiliza Windows y sistemas X- Windows, gran parte de la programación en este lenguaje está centrada en describir objetos, escribir funciones que respondan a movimientos del *Mouse*, aperturas, utilización de teclas, cargas de páginas, entre otros. Nació con la necesidad de permitir a los autores de sitios Web crear páginas que permitan intercambiar con los usuarios, ya que se necesitaba crear webs de mayor complejidad. El HTML solo permitía crear páginas estáticas donde se podían mostrar textos con estilos, pero se necesitaba interactuar con los usuarios. Javascript es el siguiente paso, después del HTML, que puede dar un programador de la Web que decida mejorar sus páginas y la potencia de sus proyectos.

Ventajas:

- Lenguaje de *scripting* seguro y fiable.
- Los scripts tienen capacidades limitadas, por razones de seguridad.
- El código Javascript se ejecuta en el cliente.

Desventajas:

- Código visible por cualquier usuario.

- El código debe descargarse completamente.
- Puede poner en riesgo la seguridad del sitio, con el actual problema llamado XSS (significa en inglés Cross Site Scripting renombrado a XSS por su similitud con las hojas de estilo CSS).

En una aplicación Web se deben usar ambos lenguajes del lado del cliente ya que el lenguaje HTML es un lenguaje estático y no permite hacer validaciones del lado del cliente lo cual es muy importante en una aplicación Web donde se necesita en todo momento interactuar con el usuario, por lo cual es necesario usar Javascript, el cual genera páginas HTML y a su vez las enriquece dándole más funcionalidad y dinamismo, puesto que con Javascript se pueden validar los campos, entre otras muchas funcionalidades que hacen más eficiente a la aplicación.

Ajax

AJAX.- AJAX, acrónimo de Asynchronous JavaScript And XML (en inglés JavaScript y XML asíncronos).

AJAX es una técnica que se usa para el desarrollo de webs, sirve para crear aplicaciones más interactivas y no necesita recargar completamente la Web ya que utiliza:

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

No es considerado un lenguaje de programación, puesto que reúne diversos lenguajes para poder obtener una nueva tecnología que sirve para el desarrollo de aplicaciones Web.

Algunas ventajas de ajax son:

- No se necesita recargar las páginas en el envío de información.

- Permite la reconfiguración de la página, mostrando los datos instantáneamente.

Navegadores que permite Ajax:

- Internet Explorer para Windows versión 5.0 y superiores, y los navegadores basados en él.
- Navegadores basados en Gecko como Mozilla, Mozilla Firefox, SeaMonkey, Camino, Flock, Epiphany, Galeon y Netscape versión 7.1 y superiores.
- Navegadores con el API KHTML versión 3.2 y superiores implementado, incluyendo Konqueror versión 3.2 y superiores, Apple Safari versión 1.2 y superiores, y el Web Browser for S60 de Nokia tercera generación y posteriores.
- Opera versión 8.0 y superiores, incluyendo Opera Mobile Browser versión 8.0 y superiores.


Del lado del servidor

PHP

PHP es un acrónimo recursivo que significa “Hypertext Pre-processor”, (inicialmente se llamó Personal Home Page). Es un lenguaje de script interpretado en el lado del servidor utilizado para la generación de páginas web dinámicas, embebidas en páginas HTML y ejecutadas en el servidor. PHP no necesita ser compilado para ejecutarse. Para su funcionamiento necesita tener instalado Apache o IIS con las librerías de PHP. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas. Los archivos cuentan con la extensión (php).

En cuanto a la seguridad es un poderoso lenguaje e intérprete, ya sea incluido como parte de un servidor web en forma de módulo o ejecutado como un binario CGI separado. Es capaz de acceder a archivos, ejecutar comandos y abrir conexiones de red en el servidor. PHP está diseñado específicamente para ser un lenguaje más seguro para escribir programas CGI que Perl o C, y con la selección correcta de opciones de configuración en tiempos de compilación y ejecución, siguiendo algunas prácticas correctas de programación.

Entre sus características fundamentales se encuentran:

-  Gratuito. Al tratarse de software libre, puede descargarse y utilizarse en cualquier aplicación, personal o profesional, de manera completamente libre.

- ✚ Gran popularidad. Existe una gran comunidad de desarrolladores y programadores que continuamente implementan mejoras en su código.
- ✚ Enorme eficiencia. Con escaso mantenimiento y un servidor gratuito (en este caso, Apache), puede soportar sin problema millones de visitas diarias.
- ✚ Sencilla integración con múltiples bases de datos. Algo esencial para una página Web verdaderamente dinámica, es una correcta integración con la base de datos. Aunque MySQL es la base de datos que mejor trabaja con PHP puede conectarse también a PostgreSQL, Oracle, dbm, filePro, interbasem o cualquier otra base de datos compatible con ODBC (Open Database Connectivity Standard).
- ✚ Versatilidad. PHP puede usarse con la mayoría de los sistemas operativos, ya sea basados en UNIX (Linux, Solares, FreeBSD), como con Windows, el sistema operativo de Microsoft.
- ✚ Gran número de funciones predefinidas. A diferencia de otros lenguajes de programación, PHP fue diseñado especialmente para el desarrollo de páginas Web dinámicas. Por ello, está dotado de un gran número de funciones que simplificarán enormemente tareas habituales como descargar documentos, enviar correos, trabajar con cookies y sesiones, etc. (Desarrollo web con PHP y MySQL).

Ventajas:

- Muy fácil de aprender.
- Se caracteriza por ser un lenguaje muy rápido.
- Soporta en cierta medida la orientación a objetos, clases y herencia.
- Es un lenguaje multiplataforma: Linux, Windows, entre otros.
- Capacidad de conexión con la mayoría de los manejadores de base de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, entre otras.
- Capacidad de expandir su potencial utilizando módulos.
- Posee documentación en su página oficial la cual incluye descripción y ejemplos de cada una de sus funciones.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.

- Incluye gran cantidad de funciones.
- No requiere definición de tipos de variables ni manejo detallado del bajo nivel.

Desventajas:

- Se necesita instalar un servidor web.
- Todo el trabajo lo realiza el servidor y no delega al cliente. Por tanto puede ser más ineficiente a medida que las solicitudes aumenten de número.
- La legibilidad del código puede verse afectada al mezclar sentencias HTML y PHP.
- La programación orientada a objetos es aún muy deficiente para aplicaciones grandes.
- Dificulta la modularización.
- Dificulta la organización por capas de la aplicación.

ASP

Es una tecnología del lado del servidor desarrollada por Microsoft para el desarrollo de sitio web dinámicos. ASP significa en inglés (Active Server Pages). Necesita tener instalado Internet Information Server (IIS). ASP no necesita ser compilado para ejecutarse. Existen varios lenguajes que se pueden utilizar para crear páginas ASP. El más utilizado es VBScript, nativo de Microsoft. El código ASP puede ser insertado junto con el código HTML. Los archivos cuentan con la extensión (asp).

Ventajas:

- Usa Visual Basic Script, siendo fácil para los usuarios.
- Comunicación óptima con SQL Server.
- Soporta el lenguaje JScript (Javascript de Microsoft).

Desventajas:

- Código desorganizado.
- Se necesita escribir mucho código para realizar funciones sencillas.
- Tecnología propietaria.
- Hospedaje de sitios web costosos.

JSP

Es un lenguaje para crear sitios web dinámicos, acrónimo de Java Server Page y está orientado a desarrollar páginas web en Java. Es multiplataforma, fue creado para ejecutarse del lado del servidor y desarrollado por Sun Microsystems. Comparte ventajas similares a las de ASP.NET, desarrollado para la creación de aplicaciones web potentes. Posee un motor de páginas basado en los *servlets* de Java. Para su funcionamiento se necesita tener instalado un servidor Tomcat.

Características:

- Código separado de la lógica del programa.
- Las páginas son compiladas en la primera petición.
- Permite separar la parte dinámica de la estática en las páginas web.
- Los archivos se encuentran con la extensión (jsp).
 - El código JSP puede ser incrustado en código HTML.

Elementos de JSP

Los elementos que pueden ser insertados en las páginas JSP son los siguientes:

- Código: se puede incrustar código "Java".
- Directivas: permite controlar parámetros del servlet.
- Acciones: permite alterar el flujo normal de ejecución de una página.

Ventajas:

- Ejecución rápida del servlets.
- Crear páginas del lado del servidor.
- Multiplataforma.
- Código bien estructurado.
- Integridad con los módulos de Java.

- La parte dinámica está escrita en Java.
- Permite la utilización de servlets.

Desventajas:

- Complejidad de aprendizaje.

Python

Es un lenguaje de programación creado en el año 1990 por Guido van Rossum y es el sucesor del lenguaje de programación ABC. Python es comparado habitualmente con Perl. Los usuarios lo consideran como un lenguaje más limpio para programar. Permite la creación de todo tipo de programas incluyendo los sitios web. Su código no necesita ser compilado. Es un lenguaje de programación multiparadigma, lo cual fuerza a que los programadores adopten un estilo de programación particular:

- Programación orientada a objetos.
- Programación estructurada.
- Programación funcional.
- Programación orientada a aspectos.

Ventajas:

- Lenguaje de propósito general.
- Gran cantidad de funciones y librerías.
- Sencillo y rápido de programar.
- Multiplataforma.
- Licencia de código abierto (Open source).
- Orientado a Objetos.
- Portable.

Desventajas:

- Lentitud por ser un lenguaje interpretado.

Se escoge como lenguaje a utilizar del lado del servidor PHP puesto que la facultad necesita un sistema que sea seguro y confiable pues se va a manejar información sensible de los profesores y PHP posee estas ventajas, además de ello su eficiencia permitirá que el trabajo sea más rápido y eficaz y su versatilidad permite un marco más amplio de posibilidades en cuanto al uso de Sistemas Gestores de Base de Datos. Otra ventaja que lo convierte en el elegido es que el mismo es un software de código abierto, lo cual es una característica fundamental para la universidad, puesto a que se encuentra en un proceso de migración hacia el software libre y las aplicaciones que se desarrollen en la misma deben ir encaminadas cada día más a esta tecnología. También cuenta con la ventaja de ser gratuito, lo cual disminuye el gasto de recursos a la universidad en la compra de licencias.

1.8 SGBD

MySQL (*Desarrollo web con PHP y MySQL*)

MySQL es, sin duda, la base de datos más popular y utilizada a la hora de desarrollar páginas Web dinámicas y sitios de comercio electrónico. Este se suele trabajar en combinación con PHP. Fue creada por la empresa sueca MySQL AB pero hace poco fue comprada por la Sun Microsystems por lo que una de sus ventajas más significativas que era su gratuidad actualmente es una desventaja.

Ventajas:

- ✓ La velocidad a la hora de realizar las operaciones, lo que le hace uno de los gestores que ofrecen mayor rendimiento.
- ✓ Su bajo consumo lo hacen apto para ser ejecutado en una máquina con escasos recursos sin ningún problema.
- ✓ Las utilidades de administración de este gestor son envidiables para muchos de los gestores comerciales existentes, debido a su gran facilidad de configuración e instalación.
- ✓ Tiene una probabilidad muy reducida de corromper los datos, incluso en los casos en los que los errores no se produzcan en el propio gestor, sino en el sistema en el que está.

Desventajas:

- ✓ Carece de soporte para transacciones, *rollback* y subconsultas.
- ✓ El hecho de que no maneje la integridad referencial, hace de este gestor una solución pobre para muchos campos de aplicación, sobre todo para aquellos programadores que provienen de otros gestores que sí poseen esta característica.
- ✓ No es viable para su uso con grandes bases de datos, a las que se acceda continuamente, ya que no implementa una buena escalabilidad.[8]
- ✓ No es gratuito.

Microsoft SQL Server

Microsoft SQL Server es un sistema de gestión de bases de datos relacionales (SGBD) basada en el lenguaje SQL, capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea.

Entre sus ventajas están:

- ✓ Soporte de transacciones.
- ✓ Gran estabilidad.
- ✓ Gran seguridad.
- ✓ Escalabilidad.
- ✓ Soporta procedimientos almacenados.
- ✓ Incluye también un potente entorno gráfico de administración, que permite el uso de comandos Lenguaje de Definición de Datos (DDL) y Lenguaje de Manipulación de Datos (DML) gráficamente.
- ✓ Permite trabajar en modo cliente-servidor donde la información y datos se alojan en el servidor y las terminales o clientes de la red sólo acceden a la información.
- ✓ Además permite administrar información de otros servidores de datos.

Desventajas:

- ✓ Tiene soporte solamente en el sistema operativo Windows.
- ✓ Es un software de licencia propietaria.

Oracle.

Es un manejador de base de datos relacional que hace uso de los recursos del sistema informático en todas las arquitecturas de hardware, para garantizar su aprovechamiento al máximo en ambientes cargados de información.

Ventajas:

- ✓ Posee igual interacción en todas las plataformas (Windows, Unix, Macintosh y Mainframes). Más del 80% de los códigos internos de Oracle son iguales a los establecidos en todas las plataformas de Sistemas Operativos.
- ✓ Oracle soporta bases de datos de todos los tamaños, desde severas cantidades de bytes y gigabytes en tamaño.
- ✓ Soporte de transacciones.
- ✓ Es mucho más fácil la realización de backups (copias de seguridad).
- ✓ Es mucho más eficaz y eficiente.
- ✓ Tiene una amplia gama de herramientas para operar con la Base de Datos tanto como usuario y como Administrador.

Desventajas:

- ✓ Elevado precio comercial.
- ✓ La seguridad de la plataforma, y las políticas de suministro de parches de seguridad, modificadas a comienzos de 2005 y que incrementan el nivel de exposición de los usuarios.
- ✓ Exige una gran cantidad de recursos de la máquina donde se encuentre instalado.

PostgreSQL.

PostgreSQL es el servidor de bases de datos de código abierto más potente que existe y es por tanto la alternativa a MySQL cuando se necesitan características avanzadas como transacciones, procedimientos almacenados, triggers, vistas, etcétera.

Es el servidor de bases de datos más utilizado por los programadores de *serv/lets* de Java y, en general, por todos aquellos que realizan aplicaciones cliente-servidor complejas o críticas en el mundo Linux/Unix.

Ventajas:

- ✓ Posee una gran escalabilidad. Es capaz de ajustarse al número de Unidades Centrales de Procesamiento (CPUs) y a la cantidad de memoria que posee el sistema de forma óptima, haciéndole capaz de soportar una mayor cantidad de peticiones simultáneas de manera correcta.
- ✓ Implementa el uso de *rollback*, subconsultas y transacciones, haciendo su funcionamiento mucho más eficaz.
- ✓ Tiene la capacidad de comprobar la integridad referencial, así como también la de almacenar procedimientos en la propia base de datos.
- ✓ Permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos.

Desventajas:

- ✓ Tiene un límite de 8K por fila, aunque se puede aumentar a 32K, con una disminución considerable del rendimiento.
- ✓ Es lento.

Evaluación de los SGBD.

Criterios	Sistemas Gestores de Bases de Datos			
	Oracle	SQL Server	MySQL	PostgreSQL
Plataforma	Windows/Linux	Windows	Windows/Linux	Windows/Linux

Velocidad	+	+	+	-
Volumen Datos	+	+	+	+
Integridad	+	+	-	+
Potencia	+	+	+	+
Coste	-	-	-	+
Requerimientos de hardware	-	+	+	+
Código abierto	-	-	+	+

+ Positivo - Negativo.

Dadas las características anteriores y la tabla de comparación, el SGBD que se propone para desarrollar el sistema es PostgreSQL, ya que SQL Server no es multiplataforma y además hay que pagar un importe por su licencia, Oracle por su parte, tiene un coste más elevado aún que el de SQL Server, además de que el Oracle necesita una máquina con elevados recursos (memoria RAM, velocidad de microprocesador, capacidad de disco duro), en cuanto al MySQL se puede decir que cuando se almacena gran cantidad de datos estos pueden perderse o dañarse, o sea que falla en la integridad de los datos, por otro parte al ser comprado por la Sun Microsystems deja de ser gratuito, por lo que su coste sería un inconveniente al igual que el de los SGBD antes mencionados, por todo lo explicado se decidió escoger PostgreSQL porque a pesar de ser el más lento, tiene dos grandes ventajas y es que es de código abierto y es gratuito, las cuales son características fundamentales en el momento de hacer la elección puesto que además de que la UCI se encuentra en un proceso de migración hacia el software libre, al ser gratuito le ahorra recursos a la universidad y al país.

1.9 Herramientas a utilizar

Herramienta Case

Las herramientas de modelado permiten hacer diferentes tipos de diagramas durante todos los ciclos de desarrollo del software, una de las más usadas hoy en día en el mundo es el Rational Rose que es una herramienta creada por "Rose" y que tiene muchas ventajas, porque permite la ingeniería directa e

inversa del producto, además de que facilita la comunicación entre los desarrolladores al trabajar con un mismo lenguaje de modelado, posibilita la creación de diagramas de casos de uso, de objetos, así como la interacción entre los diagramas de secuencia y colaboración. Además el Rational Rose organiza sus diagramas en vistas: la vista de casos de uso, la vista lógica, la vista de componentes y la vista de despliegue. EL uso de estas vistas facilita la organización del trabajo para una mejor comprensión del mismo.

Otra de las herramientas muy populares hoy en día es el Visual Paradigm, es una herramienta muy potente que permite visualizar y diseñar elementos de un producto de software, desarrollar un producto de calidad rápidamente, además al igual que el Rational Rose permite desarrollar la ingeniería directa e inversa. Su diseño está centrado en casos de uso y presenta un entorno de trabajo para UML 2.0.

Esta última es la herramienta que esta predeterminada para el trabajo con este sistema junto con el lenguaje de modelado UML, por todas sus comodidades de trabajo y comprensión de la misma, además del producto con calidad que resulta y la facilidad de su utilización para hacer los diagramas que se necesiten y su documentación. También porque trabajar con software libre es uno de los objetivos de nuestro país para la producción de software cubano y la utilización de esta herramienta lo permite debido a su capacidad de ser multiplataforma.

Zend Studio

Se trata de un programa de la casa Zend, impulsores de la tecnología de servidor PHP, orientada a desarrollar aplicaciones Web, como no, en lenguaje PHP.

El programa, además de servir de editor de texto para páginas PHP, proporciona una serie de ayudas que pasan desde la creación y gestión de proyectos hasta la depuración de código.

El programa entero está escrito en Java, lo que a veces supone que no funcione tan rápido como otras aplicaciones de uso diario. Sin embargo, esto ha permitido a Zend lanzar con relativa facilidad y rapidez versiones del producto para Windows, Linux y MacOS, aunque el desarrollo de las versiones de este último sistema se retrase un poco más.

Zend Studio consta de dos partes en las que se dividen las funcionalidades de parte del cliente y las del servidor. Las dos partes se instalan por separado, la del cliente contiene el interfaz de edición y la ayuda.

Permite además hacer depuraciones simples de scripts, aunque para disfrutar de toda la potencia de la herramienta de depuración habrá que disponer de la parte del servidor, que instala Apache y el módulo PHP o, en caso de que estén instalados, los configura para trabajar juntos en depuración.

Lo más destacable es que contiene una ayuda contextual con todas las librerías de funciones del lenguaje que asiste en todo momento ofreciendo nombres de las funciones y parámetros que deben recibir. Aunque esta ayuda contextual no sólo se queda en las funciones definidas en el lenguaje, sino que también reporta ayudas con las funciones que se vayan creando, incluso en páginas que haya incluidas con la función include().

Otras ayudas que ofrece a la hora de escribir son las típicas en editores avanzados, como permitir editar varios archivos, y moverse fácilmente entre ellos, marcar a qué elementos corresponden los inicios y cierres de las etiquetas, paréntesis o llaves, moverse al principio o al final de una función, identificación automática del código, etc.

1.10 Ingeniería de Software

La ingeniería de software, es una tecnología multicapa en la que, según Pressman, se pueden identificar: los métodos (indican como construir técnicamente el software), el proceso (es el fundamento de la Ingeniería de Software, es la unión que mantiene juntas las capas de la tecnología) y las herramientas (proceso automático o semiautomático para el proceso y los métodos).



1.11 Lenguaje Unificado de Modelado (UML)

Es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software. Está compuesto por diversos elementos gráficos que se combinan para conformar diagramas.

Es importante recalcar que UML no es una guía para realizar el análisis y diseño orientado a objetos, es decir, no es un proceso. UML es un lenguaje que permite la modelación de sistemas con tecnología orientada a objetos. Su objetivo es visualizar, especificar, construir y documentar los artefactos que se crean durante el proceso de desarrollo.

1.12 Proceso Unificado de Desarrollo (RUP)

RUP es una metodología, que constituye una forma disciplinada de asignar tareas y responsabilidades. RUP define además 4 fases, la fase de inicio o conceptualización, de elaboración, de construcción y la fase de transición, así como 9 flujos de trabajo entre los que se destacan, el flujo de trabajo de modelado del negocio, el de requerimientos, el de análisis y diseño, el de implementación y el flujo de trabajo de prueba. RUP se caracteriza por ser dirigido por casos de uso, centrado en la arquitectura e iterativo e incremental.

- **Dirigido por casos de uso:** Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. A partir de aquí los casos de uso guían el proceso de desarrollo ya que los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso (cómo se llevan a cabo).
- **Centrado en la arquitectura:** La arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente. RUP se desarrolla mediante iteraciones, comenzando por los CU relevantes desde el punto de vista de la arquitectura.
- **Iterativo e Incremental:** RUP propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros. Las iteraciones hacen referencia a pasos en los flujos de trabajo, y los incrementos, al crecimiento del producto.

1.13 Proceso de desarrollo de software

Un proceso define QUIÉN (Trabajadores), QUÉ (Artefactos), CÓMO (Actividades) y CUANDO (Flujo de Actividades). Un Proceso de Desarrollo de Software es la definición del conjunto de actividades que guían los esfuerzos de las personas implicadas en el proyecto, a modo de plantilla que explica los pasos necesarios para terminar el proyecto.

1.14 Conclusiones

En este capítulo se trató el control de los medios que se le otorgan a los estudiantes y profesores internos que es el objeto de estudio, el que se divide en módulos como el Control de Inventario, Control de Avituallamiento y Control de Aseo Personal dentro de la universidad, así como las tendencias actuales para llevar a cabo estos controles a nivel nacional e internacional. Se trataron además, las tecnologías y herramientas a utilizar durante la elaboración del sistema así como conceptos básicos como la ingeniería de software, proceso unificado de desarrollo, lenguaje unificado de modelado, y proceso de desarrollo de software.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

2.1 Introducción

El objetivo fundamental de este capítulo es realizar de una descripción del sistema que se va a diseñar posteriormente. Para ello, se hará un análisis de los principales procesos del negocio que abarca el objeto de estudio enunciado en el capítulo anterior; de esta forma se podrán obtener las funcionalidades que debe tener el sistema a diseñar.

2.2 Objeto de estudio

Debido a la cantidad de estudiantes y profesores internos con que cuenta la UCI, es cada vez mayor el volumen de información que se maneja. Por este motivo se entorpece mucho la búsqueda de datos específicos a la hora de realizar informes, por ejemplo. Uno de los objetivos que se ha planteado la Universidad desde sus inicios ha sido la informatización de la sociedad, y qué mejor lugar para comenzar que el propio entorno universitario, donde se desarrollan a diario diversidad de actividades que deben ser estudiadas en vistas a agilizar los procesos que se llevan a cabo, y que son de vital importancia para el funcionamiento del centro.

El problema que se plantea, se refiere a la necesidad de una herramienta que permita gestionar toda la información referente a la entrega de los medios a estudiantes y profesores internos. Por ello, con este trabajo de diploma se pretende obtener un diseño que permita la implementación de un sistema que facilite dicha gestión. Actualmente estos procesos se realizan manualmente, lo que provoca muchas pérdidas, y desorden en la información. Se hace muy difícil controlar la entrega y recogida de artículos de aseo, de libros de texto, así como la pérdida y deterioro de medios básicos en los apartamentos. Resulta además muy engorroso todo el proceso de tramitación de baja a algún interno, pues debe comprobarse que no tiene deudas por concepto de pérdida o deterioro de alguno de los artículos (libros de texto, avituallamiento, préstamos en la biblioteca) que debe entregar antes de retirarse de la Universidad.

Durante el estudio realizado a los procesos que abarcan el campo de acción, se identificaron cinco áreas fundamentales:

- Área de beca o residencia: donde se controlan la gestión y entrega de los medios básicos que se encuentran distribuidos en los apartamentos, de los artículos de avituallamiento que se les entrega a los estudiantes y profesores internos cuando ingresan a la residencia de la Universidad y, por último, de los artículos del módulo de aseo que se entrega mensualmente a estudiantes (de forma gratuita) y profesores (quienes abonan una cantidad en moneda nacional)
- Área Docente: en esta se mantiene un control de los módulos docentes que son entregados semestralmente a los profesores y estudiantes.
- Área de Almacenes de Libros: es donde se almacenan los libros de texto que se entregarán a cada año de la carrera y que están disponibles durante todo el curso escolar; al finalizar este, los estudiantes deben devolver los libros extraídos del almacén.
- Área de la Biblioteca: aquí se controlan los préstamos que se realizan en la biblioteca de la Universidad, así como las devoluciones de los libros que han sido prestados.
- Área de Trámites de Baja: que es donde debe presentarse cada interno que haya sido baja de la Universidad, y donde se controla que no se hayan contraído deudas debido a pérdidas o deterioro de alguno de los recursos puestos a su alcance durante su estancia en el centro; en caso de haberlas, el interno debe ingresar una cantidad de dinero que se establece en un acta de responsabilidad material.

En cada una de estas áreas se detectó la necesidad de eliminar todo el papeleo que se genera diariamente y que entorpece, en ocasiones, el desarrollo de las actividades. En todos los casos se guarda la información utilizando tarjetas y modelos, que están sujetos a pérdidas y que dificulta la creación de reportes sobre el estado de los medios que se manejan en cada una de las áreas mencionadas, así como otros que reflejen diferentes tipos de información.

Para lograr un diseño que permita implementar un sistema capaz de suprimir esta problemática, es necesario estudiar cómo se llevan a cabo estos procesos de gestión de medios en general, que incluye los módulos docentes que se entregan a alumnos y profesores, el módulo de aseo, y el de avituallamiento, los medios básicos que se entregan con los apartamentos, así como los préstamos externos que se realizan en la biblioteca y la entrega de los libros de texto a estudiantes y profesores. Es importante resaltar que el control de todos estos medios es de gran necesidad para el proceso de ingreso y estadía de los internos,

así como para el proceso de tramitación de bajas de la Universidad antes mencionado, por lo que la generación de reportes que permitan conocer el estado de estos medios, forma parte de las funciones que debe tener el sistema a diseñar.

2.3 Modelo del Negocio

2.3.1 Actores del negocio

Actores del negocio	Descripción
Interno	Es toda persona, ya sea estudiante o profesor, que tiene asignado una residencia en la Universidad. Interviene en los procesos de entrega de medios básicos de los apartamentos, de los artículos de avituallamiento y en la realización de préstamos y devoluciones en la biblioteca. Es el principal beneficiado con la realización de estos procesos.
Estudiante	Es un estudiante de la universidad de cualquiera de los 5 años de la carrera. Participa en los procesos de entrega del módulo de aseo que realiza cada mes de forma gratuita.
Profesor	Es toda persona que se desempeñe como profesor de la Universidad, tenga o no tenga beca en ella. Participa en los procesos de entrega del módulo de aseo cada mes por el cual deben abonar una cantidad de dinero en moneda nacional.
Profesor Interno	Es una particularidad de los profesores, se refiere a los profesores que residen como becados en la universidad y que, por tanto, participan además en todos los procesos que tienen que ver con la residencia.
Usuario	Representa lo mismo a estudiantes como a profesores, ya sean internos o no lo sean.

2.3.2 Trabajadores del Negocio

Trabajadores del Negocio	Justificación
Becario	Es el responsable de la entrega de los artículos de avituallamiento cada vez que se presenta un interno a recoger el módulo que le corresponde.
Secretaria	Es la responsable de entregarles a los profesores el módulo

	con los artículos de aseo correspondiente al mes.
Instructora	Es la persona responsable de la entrega de los medios básicos de los apartamentos a los estudiantes y profesores que estarán becados en la Universidad. Además es la responsable de entregar el módulo de aseo a los estudiantes que residen en los apartamentos que atiende.
Encargado de baja	Es la persona encargada de tramitar la baja de un interno cuando se le es otorgada. En este proceso es el responsable de recibir el módulo de avituallamiento que se le entregó al interno.
Empleado Almacén	Es el encargado de la entrega y recogida de libros de texto a los profesores y estudiantes que los soliciten. También se encarga de la entrega de los módulos docentes.
Bibliotecaria	Es la encargada de realizar los préstamos externos de libros a los usuarios, ya sean estudiantes o profesores internos, de la biblioteca de la Universidad.

2.3.3 Diagrama de casos de uso del negocio

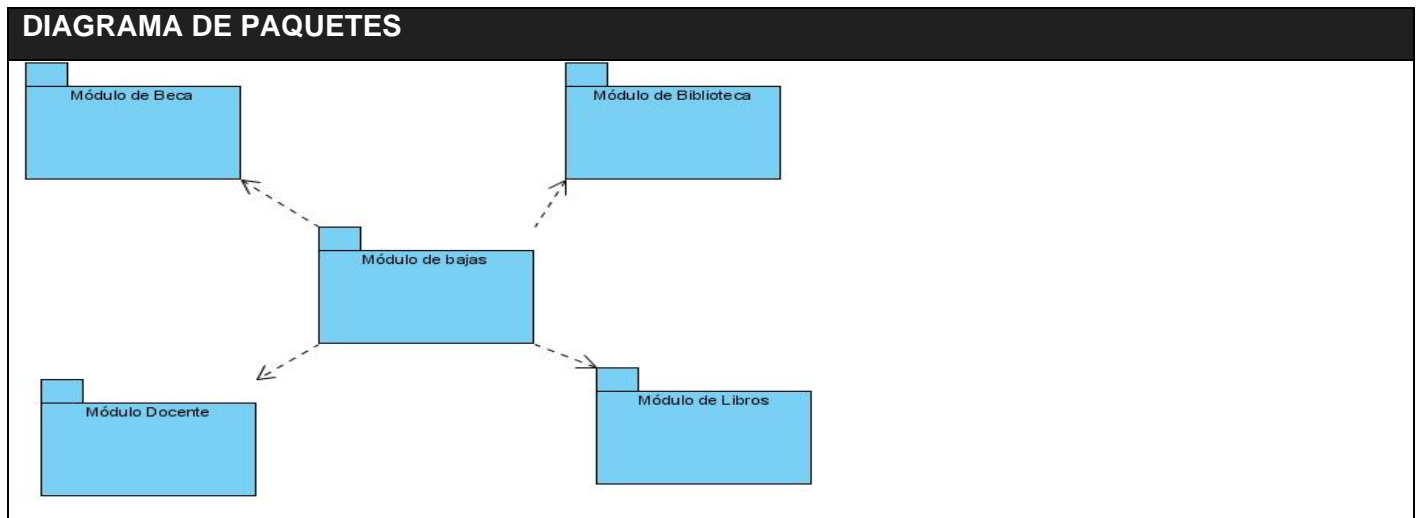


Figura 2.1 Diagrama de paquetes del negocio

MÓDULO DE LIBROS



Figura 2.2 Diagrama de Casos de Uso del negocio. Módulo de libros

MÓDULO DE BIBLIOTECA



Figura 2.3 Diagrama de Casos de Uso del negocio. Módulo de biblioteca

MÓDULO DE BECA

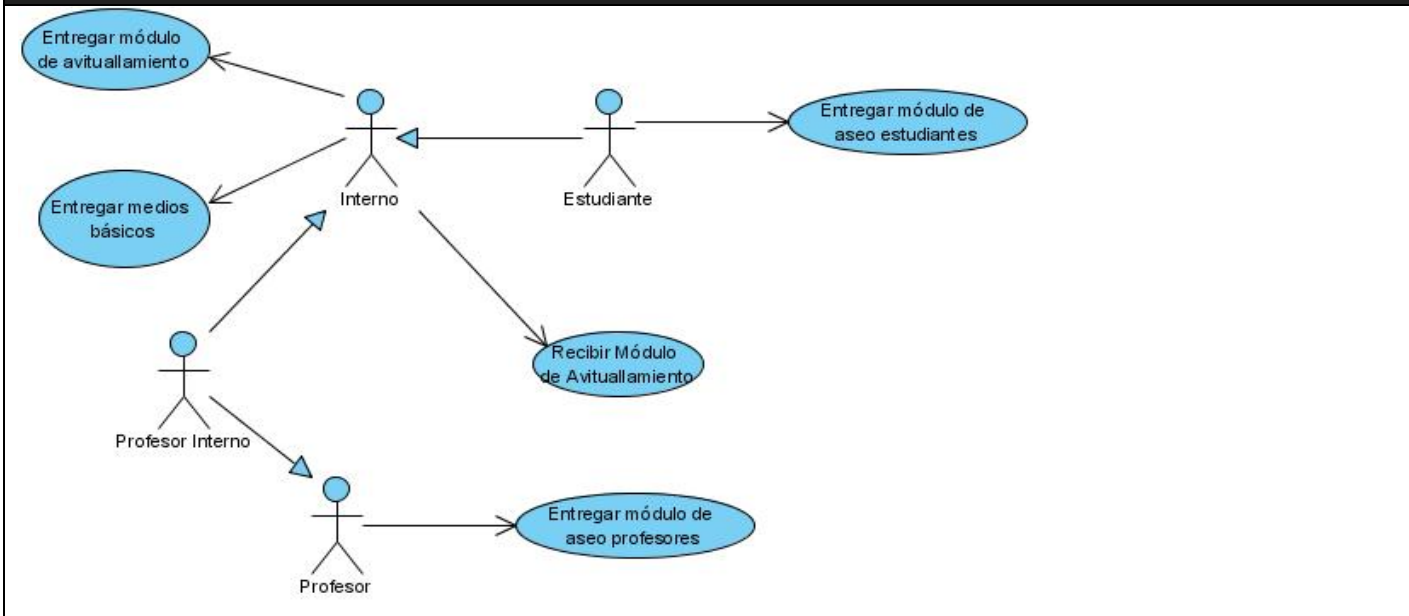


Figura 2.4 Diagrama de Casos de Uso del negocio. Módulo de beca

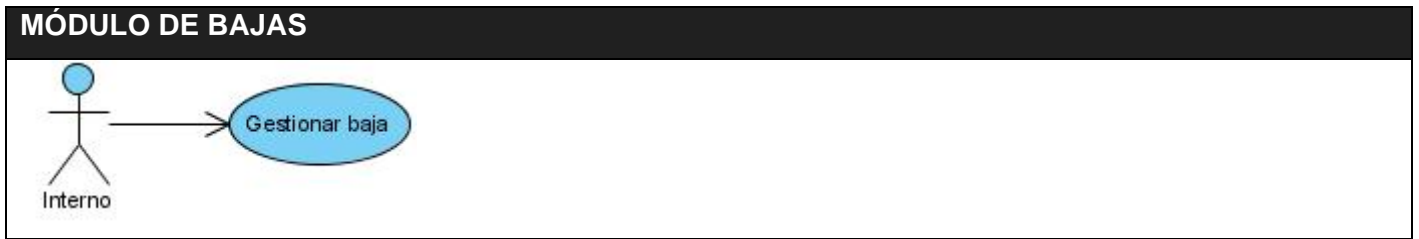


Figura 2.5 Diagrama de Casos de Uso del negocio. Módulo de bajas



Figura 2.6 Diagrama de Casos de Uso del negocio. Módulo Docente

2.3.4 Listado de casos de uso del negocio

Caso de uso	Actores	Trabajadores	Descripción
Recibir libro	Usuario	Empleado Almacén	Este caso de uso describe los pasos que se siguen cada vez que un usuario se presenta en el almacén de libros para entregar un libro que le ha sido entregado.
Entregar libro	Usuario	Empleado Almacén	Sucede cada vez que un usuario necesita pedir un libro de texto en el almacén de libros, entonces el empleado del almacén verifica todos los datos del usuario y si el libro está disponible, realiza la entrega del libro al usuario.
Realizar préstamo de libro	Interno	Bibliotecaria	Este proceso es el que se produce cuando un usuario de la biblioteca solicita un préstamo externo de uno a varios libros. La bibliotecaria realiza las comprobaciones correspondientes para realizar el préstamo, y si todo está en orden, se le da el libro al usuario.

Devolver libro	Interno	Bibliotecaria	Describe los pasos a seguir por la bibliotecaria cuando un usuario se presenta a realizar la devolución de un libro que estaba en préstamo externo.
Entregar módulo de aseo estudiantes	Estudiante	Instructora	Describe el proceso de entrega del módulo de aseo a los estudiantes. Este proceso es llevado a cabo por la instructora que se encarga del edificio donde residen los estudiantes.
Entregar módulo de avituallamiento	Interno	Becario	Este proceso es el que se realiza cada vez que ingresa un nuevo interno a la residencia de la Universidad. Consiste en entregarle a cada uno un módulo de avituallamiento (sábanas, toallas, etc.) que debe ser devuelto cuando al interno se le otorgue la baja de la Universidad.
Entregar medios básicos apto	Interno	Instructora	Este proceso describe la entrega de medios básicos de los apartamentos de la residencia cuando un nuevo interno es ubicado. La instructora realiza un inventario de todos los medios entregados y el interno debe hacerse responsable del cuidado de estos.
Entregar módulo de aseo profesores	Profesor	Secretaria	Describe el proceso de entrega del aseo a los profesores. La secretaria es la encargada de cobrar y entregar el aseo del mes a los profesores que se presenten a recogerlo.
Gestionar baja	Interno	Encargado de baja	Describe todas las comprobaciones que debe realizar el encargado de la baja cuando un interno se presenta con la baja del centro otorgada. Debe verificarse que el interno no ha contraído deudas por concepto de pérdida o deterioro de cualquier medio que le haya sido entregado. En caso de que exista alguna deuda de este tipo, debe saldarse inmediatamente.

Entregar módulo de materiales docentes	Usuario	Empleado almacén	Describe la entrega de los materiales docentes que les corresponde tanto a estudiantes como a profesores en cada semestre del curso.
--	---------	------------------	--

2.3.5 Diagramas de actividades (Anexos)

2.3.6 Diagrama de clases del modelo de objetos

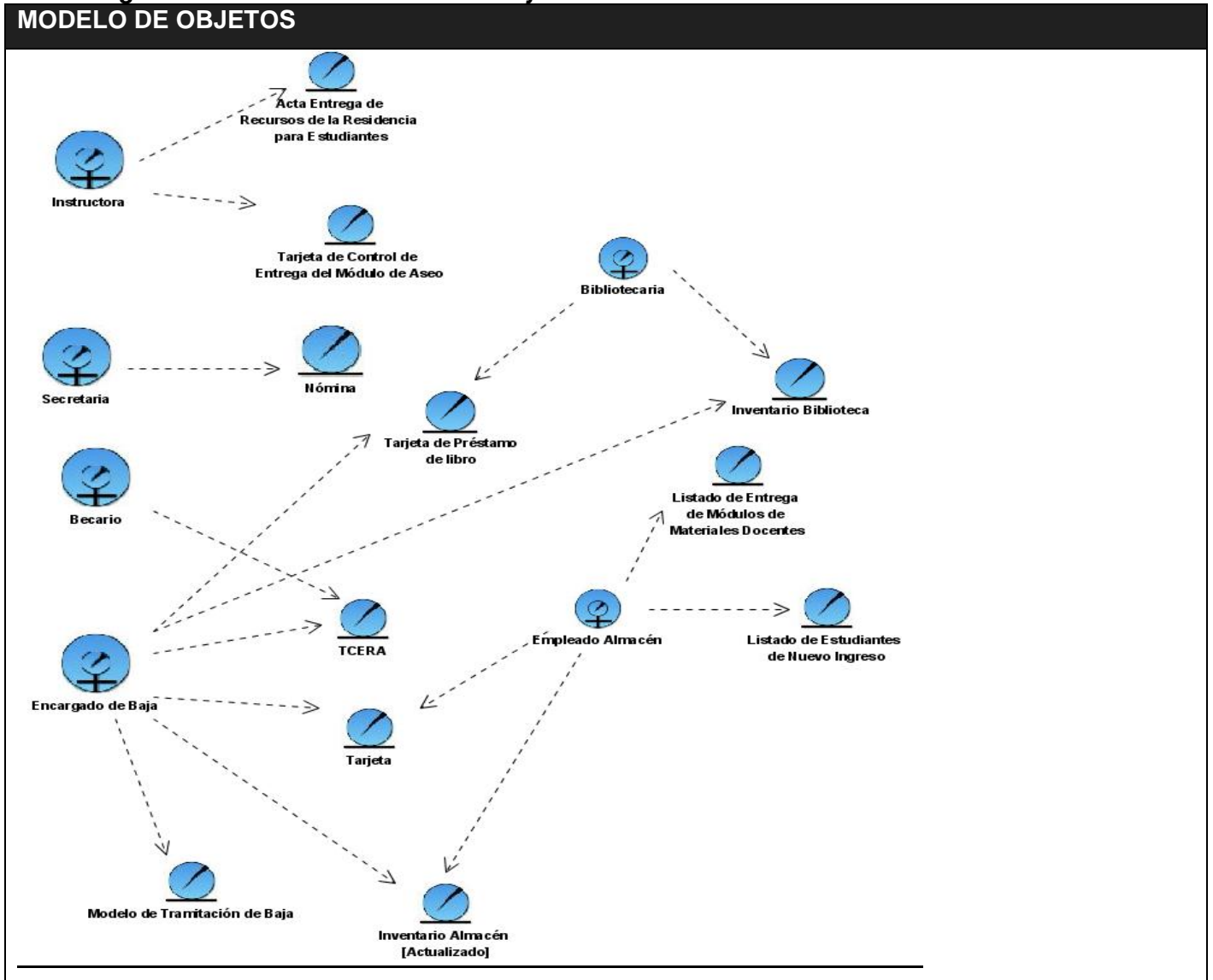


Figura 2.7 Modelo de objetos

2.4 Especificación de los requisitos del software

2.4.1 Requerimientos funcionales

1. Autenticar
2. Gestionar becados.
 - 2.1 Adicionar un interno a un apartamento.
 - 2.2 Eliminar un interno de un apartamento.
3. Gestionar medio básico
 - 3.1 Adicionar un medio básico al listado de medios básicos existentes.
 - 3.2 Eliminar un medio básico del listado de medios básicos existentes.
 - 3.3 Modificar los datos de un medio básico.
4. Registrar pérdida o ruptura de un medio básico
 - 4.1 Registrar la pérdida o ruptura de un medio básico
 - 4.2 Elaborar Acta de Responsabilidad Material.
5. Registrar pago de deuda de un medio básico
 - 5.1 Mostrar el listado de los estudiantes de un apartamento.
 - 5.2 Mostrar listado de estudiantes con deudas pendientes.
 - 5.3 Registrar pago de deudas
6. Gestionar apartamento
 - 6.1 Adicionar un medio básico a un apartamento.
 - 6.2 Eliminar un medio básico de un apartamento.
7. Entregar avituallamiento
 - 7.1 Buscar los datos de un interno.
 - 7.2 Mostrar listado de artículos.
 - 7.3 Gestionar la Tarjeta de Control de Entrega y Recepción de Avituallamiento de un interno.
 - 7.3.1 Crear Tarjeta de Control de Entrega y Recepción de Avituallamiento de un interno.
 - 7.3.2 Modificar la Tarjeta de Control de Entrega y Recepción de Avituallamiento de un interno.
 - 7.3.3 Eliminar la Tarjeta de Control de Entrega y Recepción de Avituallamiento de un interno.
8. Entregar módulo de aseo
 - 8.1 Buscar un interno por el número de su solapín.
 - 8.2 Mostrar el listado de artículos de aseo a entregarle al interno (estudiante).

8.3 Gestionar la Tarjeta de Control de Entrega del Módulo de Aseo Individual.

8.3.1 Crear la Tarjeta de Control de Entrega del Módulo de Aseo Individual.

8.3.2 Modificar la Tarjeta de Control de Entrega del Módulo de Aseo Individual.

8.3.3 Eliminar la Tarjeta de Control de Entrega del Módulo de Aseo Individual.

8.4 Gestionar la Nómina

8.4.1 Crear la Nómina.

8.4.2 Modificar la Nómina

9. Entregar libro

9.1 Buscar interno por numero de solapín

9.2 Mostrar un listado de los libros pendientes de un interno.

9.3 Mostrar un listado de libros disponibles para entregar.

10. Gestionar libro.

10.1 Incorporar lote de libros.

10.2 Dar baja a lote de libros.

11. Recibir libro

11.1 Buscar un interno por el número de su solapín

11.2 Mostrar un listado de los libros pendientes de un interno.

12. Registrar pago de deuda de libro.

12.1 Buscar un interno por su número de solapín

12.2 Mostrar un listado de los títulos a entregar y el importe correspondiente.

13. Entregar materiales

13.1 Buscar interno por número de solapín

13.2 Mostrar datos de los internos que no han recogido sus materiales docentes.

13.3 Mostrar un listado de materiales disponibles para entregar.

14. Gestionar materiales

14.1 Agregar un material.

14.2 Eliminar un material.

15. Dar baja a interno

15.1 Mostrar deudas de biblioteca.

15.2 Mostrar deudas de avituallamiento.

15.3 Mostrar deudas de libros.

15.4 Mostrar deudas de medios básicos.

15.5 Crear Boleta de Tramitación de Baja.

15.6 Consultar Boleta de Tramitación de Baja.

16. Gestionar Producto de Aseo.

16.1 Adicionar producto.

16.2 Modificar producto.

16.3 Eliminar producto.

17. Recibir avituallamiento

17.1 Buscar interno por número de solapín.

17.2 Registrar recibo de artículos de avituallamiento.

17.3 Registrar recibo de artículos de avituallamiento.

18. Gestionar libro.

18.1 Adicionar libro.

18.2 Eliminar libro.

19. Realizar préstamo.

19.1 Buscar un libro por su título.

19.2 Gestionar Tarjeta de Préstamo de Libros.

19.2.1 Crear Tarjeta de Préstamo de Libro.

19.2.2 Modificar Tarjeta de Préstamo de Libro.

19.3 Gestionar Tarjeta de Inventario de la Biblioteca.

23.3.1 Modificar tarjeta.

20. Devolver libro.

20.1 Buscar los datos de un libro por su título.

20.2 Mostrar listado de préstamos de un libro.

20.3 Registrar la devolución de un libro.

20.4 Registrar el pago de un libro en mal estado o perdido.

21. Gestionar Artículo de Avituallamiento

21.1 Adicionar un artículo de avituallamiento

21.2 Modificar artículo de avituallamiento

22.3 Eliminar artículo de avituallamiento

22. Gestionar usuarios

22.1 Agregar un usuario

22.2 Modificar los datos de un usuario

22.3 Eliminar a un usuario

23. Reportes

21.1 Reportes de medios básicos.

21.1.1 Listar medios básicos de un apartamento.

21.1.2 Listar medios básicos de un edificio.

21.1.3 Listar medios básicos de los apartamentos de un grupo.

21.1.4 Listar medios básicos de una residencia.

21.1.5 Listar medios básicos en mal estado de una residencia.

21.1.6 Listar medios básicos dado un precio

21.1.7 Mostrar cantidad de un medio básico dado.

21.1.8 Mostrar datos de un medio básico.

21.1.9 Listar medios básicos en buen estado de una residencia.

21.1.10 Listar medios básicos perdidos de una residencia.

21.1.11 Listar medios básicos de los apartamentos de una facultad.

21.1.12 Mostrar cantidad de un medio básico dado un medio y una residencia.

21.1.13 Mostrar cantidad de un medio básico dado un medio y un edificio.

21.1.14 Mostrar cantidad de un medio básico dado un medio y un apartamento.

21.1.15 Mostrar cantidad de un medio básico dado un medio y una facultad.

21.1.16 Mostrar medios básicos en mal estado de un edificio.

21.1.17 Mostrar medios básicos mal estado un apartamento.

21.1.18 Mostrar medios básicos en buen estado de un edificio.

21.1.19 Mostrar medios básicos en buen estado de un apartamento.

21.1.20 Mostrar medios básicos perdidos de un edificio.

21.1.21 Mostrar medios básicos perdidos de un apartamento.

21.1.22 Mostrar cantidad de un medio básico de los apartamentos de un grupo dado un medio y el grupo.

- 21.1.23 Mostrar medios básicos en mal estado de los apartamentos de una facultad.
- 21.1.24 Mostrar medios básicos en buen estado de una facultad.
- 21.1.25 Mostrar medios básicos perdidos de una facultad.
- 21.1.26 Mostrar medios básicos en mal estado de los apartamentos de un grupo.
- 21.1.27 Mostrar medios básicos en buen estado de los apartamentos de un grupo.
- 21.1.28 Mostrar medios básicos perdidos de los apartamentos de un grupo.
- 21.1.29 Mostrar actas de responsabilidad material dado un medio.
- 21.1.30 Mostrar actas de responsabilidad material por fecha.
- 21.1.31 Mostrar actas de responsabilidad material de pérdida.
- 21.1.32 Mostrar actas de responsabilidad material de rotura.
- 21.1.33 Mostrar actas de responsabilidad material de un interno.
- 21.1.34 Mostrar actas de responsabilidad material de un apartamento.
- 21.1.35 Mostrar actas de responsabilidad material de un edificio.
- 21.1.36 Mostrar actas de responsabilidad material de una facultad.
- 21.1.37 Mostrar actas de responsabilidad material de un grupo.
- 21.1.38 Mostrar actas de responsabilidad material de una residencia.
- 21.1.39 Mostrar actas de responsabilidad material por medida aplicada.
- 21.2 Reportes de biblioteca.
 - 21.2.1 Listar préstamos de un libro.
 - 21.2.2 Listar préstamos dado un interno.
 - 21.2.3 Listar libros de la biblioteca.
 - 21.2.4 Mostrar libros dado un precio.
 - 21.2.5 Listar libros disponibles para préstamos.
 - 21.2.6 Listar préstamos de la biblioteca.
 - 21.2.7 Mostrar cantidad de un libro dado.
 - 21.2.8 Mostrar libro más solicitado.
 - 21.2.9 Mostrar libro menos solicitado.
 - 21.2.10 Listar libros en mal estado.
 - 21.2.11 Listar libros perdidos.
 - 21.2.12 Listar libros en buen estado.

21.2.13 Listar préstamos en un rango de fecha.

21.2.14 Listar préstamos hechos a profesores.

21.2.15 Listar préstamos hechos a estudiantes.

21.2.16 Mostrar cantidad total de un título.

21.2.17 Listar libros dado un autor.

21.2.18 Listar libros dado un estante.

21.2.19 Mostrar datos de un libro.

21.2.20 Mostrar mejor lector interno.

21.2.21 Mostrar mejor lector estudiante.

21.2.22 Mostrar mejor lector profesor.

21.2.23 Mostrar cantidad prestada de un título.

21.2.24 Mostrar cantidad en almacén de un título.

21.3 Reportes de avituallamiento

21.3.1 Mostrar los artículos de un interno.

21.3.2 Listar artículos en mal estado.

21.3.3 Listar artículos dado un precio.

21.3.4 Mostrar precio total de artículos de un interno.

21.3.5 Listar artículos de una facultad.

21.3.6 Mostrar artículos de una residencia.

21.3.7 Mostrar artículos de un año.

21.3.8 Mostrar cantidad de un artículo entregado.

21.3.9 Mostrar cantidad de artículos entregados.

21.3.10 Mostrar datos de un artículo.

21.3.11 Mostrar precio total de un artículo de un interno.

21.4 Reportes de materiales docentes

21.4.1 Listar materiales entregados a un interno en un semestre.

21.4.2 Listar materiales entregados dado un semestre.

21.4.3 Listar materiales entregados dada una fecha.

21.4.4 Listar materiales entregados a un año.

21.4.5 Listar materiales entregados a una facultad.

- 21.4.6 Listar materiales a entregar en un semestre.
- 21.4.7 Listar materiales a entregar a un año.
- 21.4.8 Listar materiales entregados a un interno.
- 21.4.9 Listar materiales entregados a un interno año.
- 21.5 Reportes de baja.
 - 21.5.1 Listar bajas.
 - 21.5.2 Mostrar bajas por facultad.
 - 21.5.3 Mostrar bajas por año académico.
 - 21.5.4 Mostrar baja por grupo.
 - 21.5.5 Mostrar bajas de una residencia.
 - 21.5.6 Mostrar bajas por edificio.
 - 21.5.7 Mostrar bajas por apartamento.
 - 21.5.8 Mostrar bajas por provincia.
 - 21.5.9 Mostrar si a un interno se le dio baja.
- 21.6 Reportes de aseo
 - 21.6.1 Listar estudiantes que recibieron el aseo en un mes.
 - 21.6.2 Listar profesores que recibieron el aseo en un mes.
 - 21.6.3 Listar artículos entregados en un mes.
 - 21.6.4 Listar artículos entregados en un año.
 - 21.6.5 Mostrar cantidad de productos entregados por edificio.
 - 21.6.6 Mostrar cantidad de productos entregados por facultad.
 - 21.6.7 Mostrar cantidad de productos entregados por apartamento.
 - 21.6.8 Mostrar cantidad de productos entregados por residencia.
- 21.7 Reportes de libros.
 - 21.7.1 Listar libros entregados en un semestre.
 - 21.7.2 Listar libros entregados a un año.
 - 21.7.3 Mostrar libros entregados por facultad
 - 21.7.4 Mostrar libros entregados.
 - 21.7.5 Mostrar libros en almacén.
 - 21.7.6 Mostrar libros en almacén por título.

- 21.7.7 Mostrar libros entregados por título.
- 21.7.8 Mostrar libros entregados a un interno
- 21.7.9 Mostrar libros en mal estado.
- 21.7.10 Mostrar libros en buen estado.
- 21.7.11 Mostrar libros perdidos.
- 21.7.12 Mostrar libros que han sido pagados.
- 21.7.13 Mostrar total libros.
- 21.7.14 Mostrar libros dado un precio.
- 21.7.15 Mostrar libros entregados por título.

21.8 Reportes de administración

- 21.8.1 Mostrar todos los usuarios.
- 21.8.2 Mostrar usuarios por roles.
- 21.8.3 Mostrar usuario.

2.4.2 Requerimientos no funcionales

Apariencia o interfaz externa

- El sistema debe tener una interfaz fácil de usar y amigable para que pueda ser utilizada sin mucho entrenamiento por el usuario.
- Estará diseñado para la resolución deseada por el usuario.
- Empleo de imágenes y colores identificados con el negocio donde se implantará el sistema.

Usabilidad

- El sistema podrá ser utilizado por personas que tengan un conocimiento mínimo en el manejo de las computadoras.
- El software tendrá siempre visible la opción de Ayuda, lo que posibilitará una mejor explotación por parte de los usuarios de sus funcionalidades.
- El sistema estará disponible las 24 h del día.
- En los controles tipo lista que posean más de 20 posibles selecciones se debe emplear algún mecanismo que permita su rápida localización (búsqueda por teclado).

- Empleo de barras de progreso u otros para indicar el estado del proceso en cuestión.
- Barra de herramientas de acceso directo con las opciones más usadas.

Rendimiento.

- Los tiempos de respuesta y velocidad de procesamiento de la información serán rápidos, no mayores de 20 segundos para las recuperaciones.

Soporte.

- Se necesita un servidor de bases de datos que soporte grandes volúmenes de datos (postgresql).
- Se elaborará un paquete de instalación.

Portabilidad.

- El sistema será multiplataforma (Linux o Windows).

Seguridad.

- El usuario debe autenticarse antes de entrar al sistema.
- Garantizar el acceso controlado a la información. Este debe influir sobre cómo se presentan las interfaces para cada usuario dependiendo del nivel de acceso a la información.

Software

Para el cliente:

- Navegador Mozilla Firefox.
- Sistema operativo Windows 98 o superior, o Linux.

Para el servidor:

- Sistema operativo Windows Advancer Server (2000 o superior) o Linux en cualquiera de sus distribuciones.
- Un servidor Apache v2.0.58 con módulo PHP 5.1.2 disponible, este debe estar configurado con las extensiones “pgsql”, “pdo” y “pdo_pgsql”.
- Un servidor de bases de datos PostgresSQL v8.0.8.

Hardware

Para el servidor:

- Requerimientos mínimos: Procesador Pentium III a 1GHz de velocidad de procesamiento y 1Gb de memoria RAM.
- Al menos 40Gb de espacio libre en disco duro.
- Tarjeta de red.

Para el cliente:

- Requerimientos mínimos: Procesador Pentium II a 133Mhz con 128 Mb de memoria RAM.
- Tarjeta de red.

Restricciones para el diseño e implementación

- Utilizar los estándares establecidos de codificación, diseño, entre otros.
- Emplear como servidores Web y de bases de datos Apache y PostgreSQL respectivamente.
- Utilizar como lenguaje del lado del servidor al PHP v5.1.2 y del lado del cliente el Java Script y Ajax.

Confiabilidad.

- Deben establecerse los mecanismos necesarios para el restablecimiento del sistema ante fallos de comunicación u otros, los tiempos mínimos para ello no deben exceder las 6hrs.
- Deben montarse sistemas de respaldo eléctrico en los locales de los servidores para mantener la vitalidad de los servicios.

Ayuda.

- Debe contar con la ayuda necesaria para que el usuario final pueda estar informado acerca de las principales funcionalidades que brinda el sistema.
- La ayuda debe ser fácil de entender y utilizar.

2.5 Definición de los casos de uso

2.5.1 Definición de los actores

Actores	Justificación
Usuario	Es un usuario no autenticado en el sistema, que tiene acceso a la aplicación, pero que solo puede realizar tareas generales tales como autenticarse o ver algún reporte de acceso público.
Administrador	Es quien da o quita los permisos a los usuarios del sistema, puede agregar nuevos usuarios y tiene pleno acceso a todas las funcionalidades del sistema.
Administrador biblioteca	Se encarga de toda la gestión de los usuarios en la parte del sistema que se refiere a la gestión bibliotecaria. Es el encargado de dar los permisos a cada rol de los usuarios y tiene acceso a los reportes de los libros de la biblioteca.
Administrador Módulo docente	Es el usuario que tiene total acceso al módulo de la gestión de los materiales docentes. Se encarga de la gestión de los usuarios en esta área y tiene acceso a los reportes que tienen que ver con ella.
Administrador Beca	Es el encargado de administrar el sistema para la parte de la beca y que tiene acceso a toda la información de esta área.
Bibliotecaria	Es quien lleva a cabo los préstamos de libros y quien recibe las devoluciones y registra toda esta información en el sistema.
Encargado Baja	Es quien registra los datos de las bajas que se dan en la universidad que van a recoger su boleta.
Encargado Aseo	Es quien se encarga de registrar la entrega del aseo al personal de la Universidad en el sistema.
Encargado Avituallamiento	Es el encargado de registrar los datos de la entrega de los artículos de avituallamiento a los internos en el sistema, así como el estado en que se encuentran los mismos al ser devueltos cuando se va a realizar una baja.
Empleado Almacén	Es la persona que tiene acceso a la entrega de los materiales

	docentes y registra todos estos datos en el sistema y que además se encarga de registrar todos los libros de texto que se entregan a los estudiantes y profesores.
Instructor	Es la persona encargada de ubicar por apartamentos los medios básicos que va a tener este, también se encarga de registrar las pérdidas o rupturas de estos medios.
Económico	Es la persona encargada de registrar el pago de los medios que han sido objeto de rupturas por parte de un interno.

2.5.2 Listado de casos de uso

CU-1	Autenticar
Actor	Usuario
Descripción	Este caso de uso permite a un usuario no autenticado obtener los permisos a los que tiene acceso una vez que ha introducido una clave que lo identifique en el sistema.
Referencia	RF-1

CU-2	Obtener reporte
Actor	Usuario
Descripción	Este caso de uso tiene la responsabilidad de mostrar diferentes tipos de reportes en dependencia de los parámetros que sean especificados por el usuario
Referencia	RF-20

CU-3	Gestionar usuarios
Actor	Administrador
Descripción	Permite agregar nuevos usuarios al sistema, modificar los roles de

	cada uno de ellos y eliminarlos en caso de que ya no sean necesarios.
Referencia	RF-22

CU-4	Gestionar Libro de Biblioteca
Actor	Administrador Biblioteca
Descripción	Permite agregar un nuevo libro al sistema o eliminar uno que haya dejado de estar en existencia en la biblioteca.
Referencia	RF-18

CU-5	Gestionar Materiales
Actor	Administrador Modulo docente
Descripción	Permite agregar nuevos artículos al módulo de materiales docentes que se les entrega a estudiantes y profesores, y permite eliminar artículos que ya dejen de entregarse, en dependencia del módulo que se conforme para un semestre determinado.
Referencia	RF-14

CU-6	Gestionar libros
Actor	Administrador Módulo docente
Descripción	Este caso de uso permite que se incorporen nuevos libros de textos de las asignaturas, y que estarán disponibles para los estudiantes de los distintos años de la carrera y para los profesores que imparten estas asignaturas
Referencia	RF-10

CU-7	Gestionar medio básico
Actor	Administrador Beca
Descripción	Permite agregar nuevos medios básicos al listado de inventario de los apartamentos, así como eliminar alguno que ya no se vaya a entregar a los internos cuando ingresen a la Universidad.
Referencia	RF-3

CU-8	Gestionar internos
Actor	Administrador Beca
Descripción	Permite incorporar los datos de todos los becados de la universidad, modificar sus datos, así como eliminarlo una vez que se le de baja de la residencia.
Referencia	RF-2

CU-9	Gestionar producto de aseo
Actor	Administrador Beca
Descripción	Permite incorporar un nuevo tipo de producto a entregar en el módulo de aseo o eliminar un artículo que se dejará de entregar en dicho módulo.
Referencia	RF-16

CU-10	Gestionar Artículos de Avituallamiento
Actor	Administrador Beca
Descripción	Permite incorporar un nuevo tipo de artículo de avituallamiento a entregar o eliminar un artículo que se dejará de entregar a los becados.
Referencia	RF-21

CU-11	Dar baja a interno
--------------	--------------------

Actor	Encargado de baja
Descripción	Este caso de uso permite al encargado de las bajas verificar que el interno haya realizado la entrega de todos los medios básicos del apartamento donde residía y de los artículos de avituallamiento que le fueron entregados cuando ingresó en la Universidad. También se comprueba que no deba libros de préstamos en la biblioteca, ni libros de texto.
Referencia	RF-15

CU-12	Realizar Préstamo
Actor	Bibliotecaria
Descripción	Permite a la bibliotecaria realizar un préstamo en la biblioteca a un interno de la Universidad. Se registran las fechas de entrega del libro y la fecha en que debe ser devuelto a la biblioteca.
Referencia	RF-19

CU-13	Devolver Libro
Actor	Bibliotecaria
Descripción	Permite registrar la devolución de un libro que estaba en préstamo, en caso de que el libro sea devuelto en mal estado, se registra el pago por el daño causado.
Referencia	RF-20

CU-14	Entregar aseo
Actor	Encargado de aseo
Descripción	Este caso de uso permite al encargado del aseo registrar la entrega del módulo de aseo mensual a los internos que no lo hayan recibido aún.
Referencia	RF-8

CU-15	Entregar avituallamiento
Actor	Encargado de avituallamiento
Descripción	Este caso de uso da la posibilidad de registrar toas las entregas de artículos de avituallamiento a los internos que ingresan a la Universidad.
Referencia	RF-7

CU-16	Recibir avituallamiento
Actor	Encargado de avituallamiento
Descripción	Este caso de uso tiene la responsabilidad de registrar la devolución de los módulos de avituallamiento, que les fueron entregados a su ingreso, cuando un interno causa baja de la universidad.
Referencia	RF-17

CU-17	Entregar libro
Actor	Empleado de almacén
Descripción	Este caso de uso se inicia cuando el empleado del almacén indica que va a entregar un libro de texto a algún estudiante o profesor que se presenta para realizar la solicitud. El caso de uso verifica que el almacén cuente con el libro que se está solicitando, en caso de que no existan ejemplares disponibles, no se realiza la entrega.
Referencia	RF-9

CU-18	Registrar pago de deuda de libros
Actor	Empleado de almacén
Descripción	Permite registrar el pago de un libro cuando un usuario ha perdido o dañado alguno de los libros que le habían sido entregados.
Referencia	RF-12

CU-19	Entregar materiales
Actor	Empleado de almacén
Descripción	Este caso de uso permite registrar la entrega de materiales docentes a estudiantes y profesores en cada semestre. Como resultado se actualizan los inventarios del almacén y las tarjetas de control de entrega de materiales docentes.
Referencia	RF-13

CU-20	Recibir libro
Actor	Empleado de almacén
Descripción	Permite registrar la devolución de un libro de texto que había sido entregado a un estudiante o profesor.
Referencia	RF-11

CU-21	Gestionar apartamento
Actor	Instructor
Descripción	Este caso de uso permite asignar una cantidad de medios básicos a los apartamentos. Es decir, se registran los medios que va a tener un apartamento determinado. También permite eliminar un medio de un apartamento en caso de que sea necesario.
Referencia	RF-6

CU-22	Registrar pérdida o ruptura
Actor	Instructor
Descripción	Permite registrar la pérdida o la ruptura de un medio básico, como resultado se crea un Acta de Responsabilidad Material donde se especifica si el(los) implicado(s) deben pagar el medio básico, reintegrarlo, o si serán sometidos a comisión disciplinaria.
Referencia	RF-4

CU-23	Registrar pago de deuda
Actor	Económico
Descripción	Permite que cuando un interno se presente a pagar una deuda por concepto de pérdida o ruptura de algún medio básico, el pago de esta quede registrado.
Referencia	RF-5

2.5.3 Diagrama de casos de uso

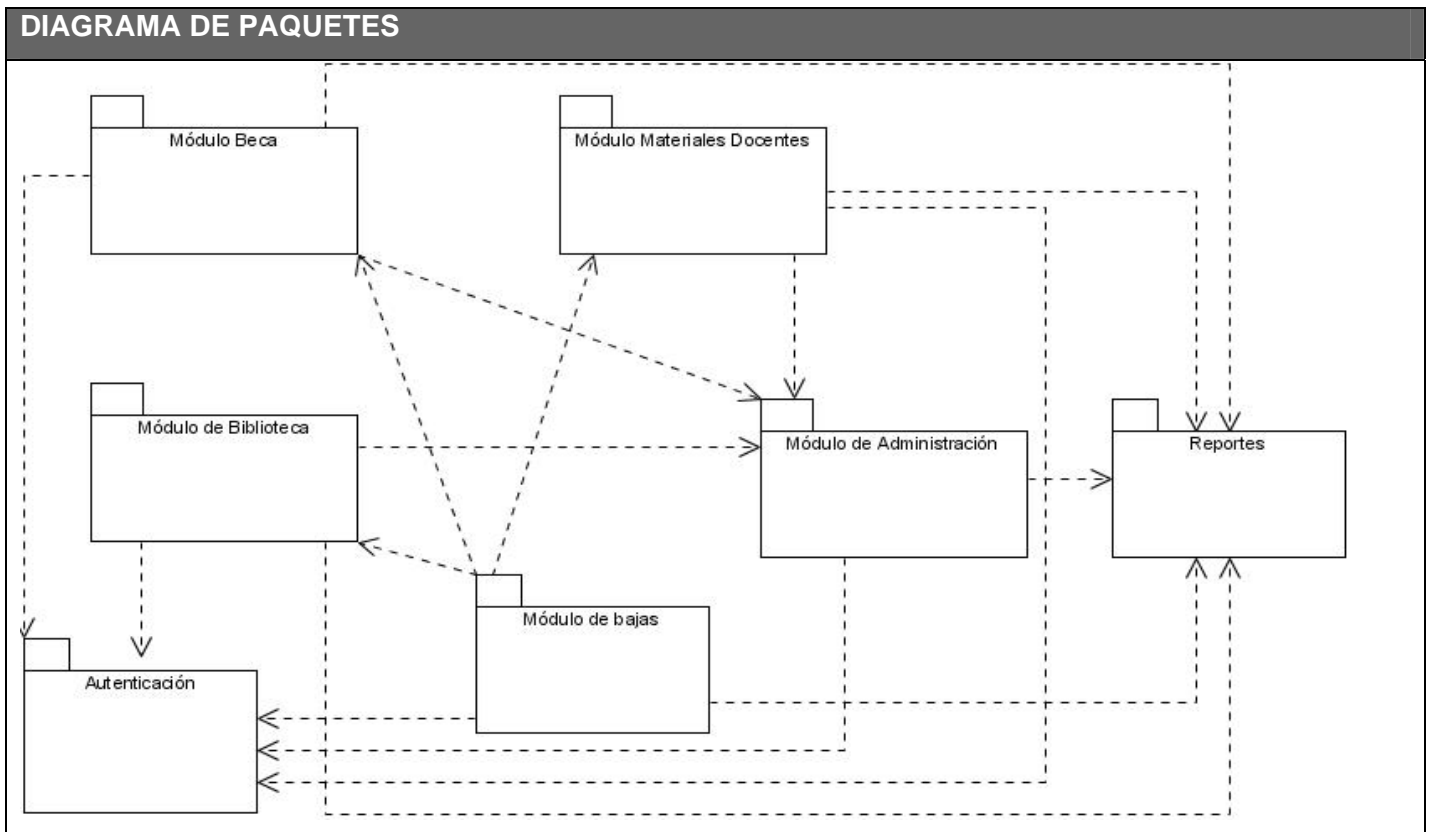


Figura 2.8 Diagrama de casos de uso del sistema. Diagrama de paquetes.



Figura 2.9 Diagrama de casos de uso del sistema. Módulo de Administración

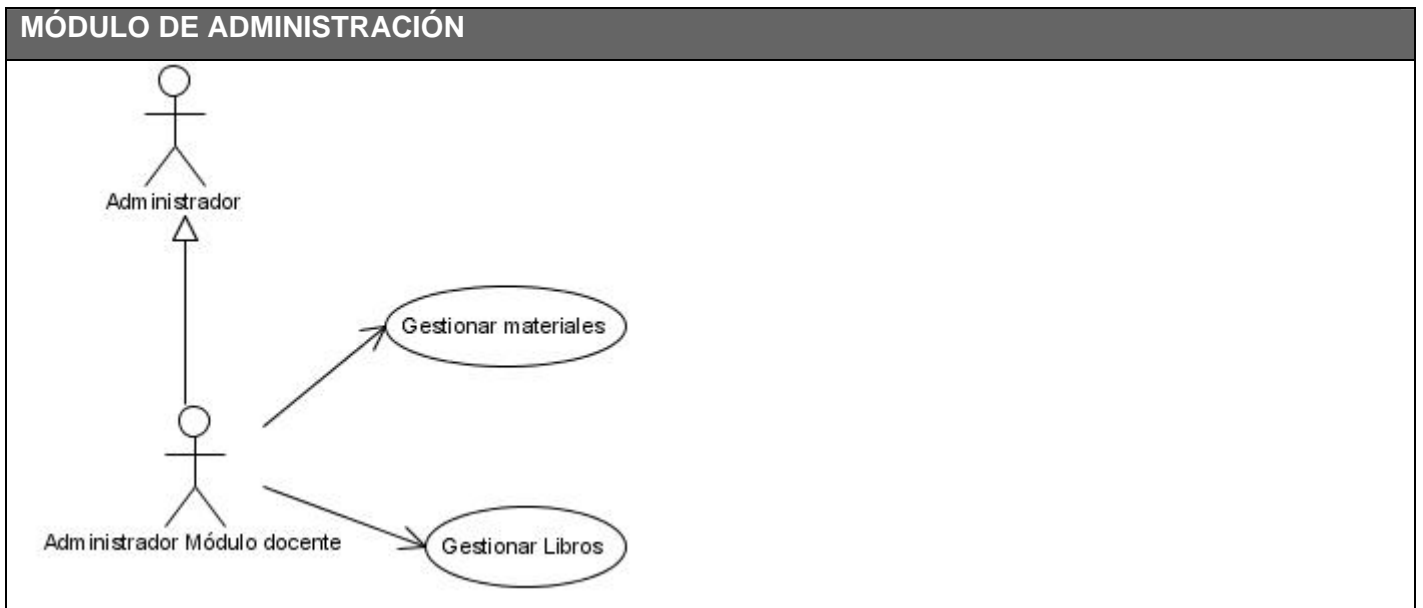


Figura 2.10 Diagrama de casos de uso del sistema. Módulo de Administración

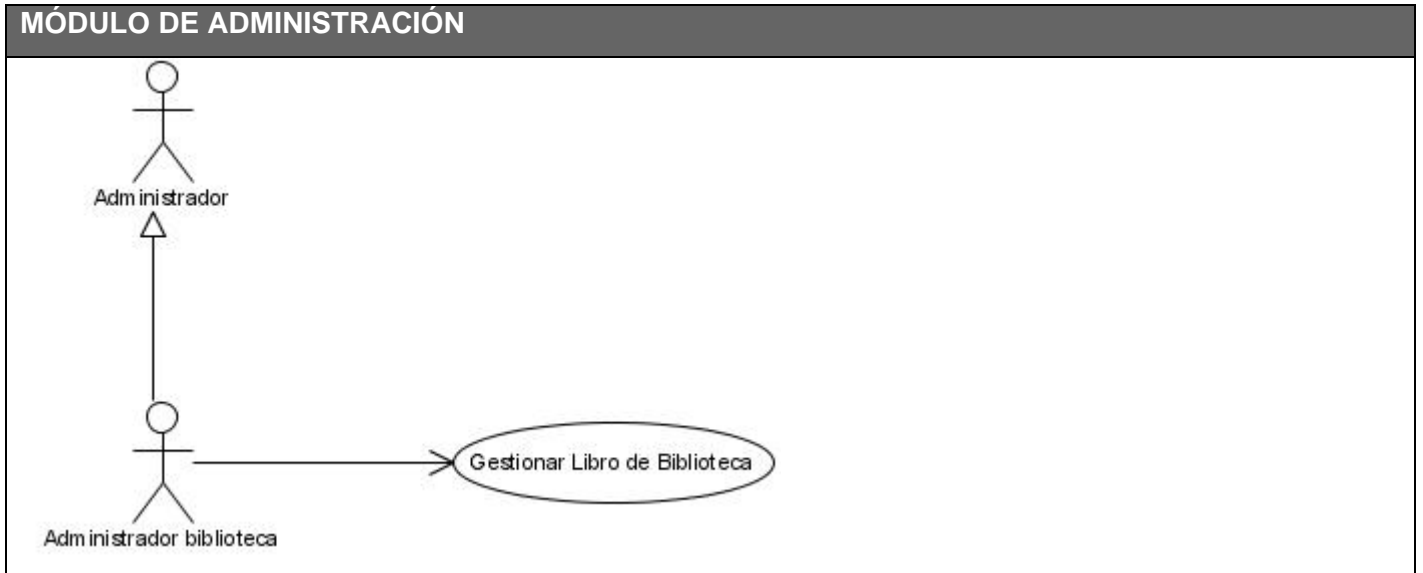


Figura 2.11 Diagrama de casos de uso del sistema. Módulo de Administración

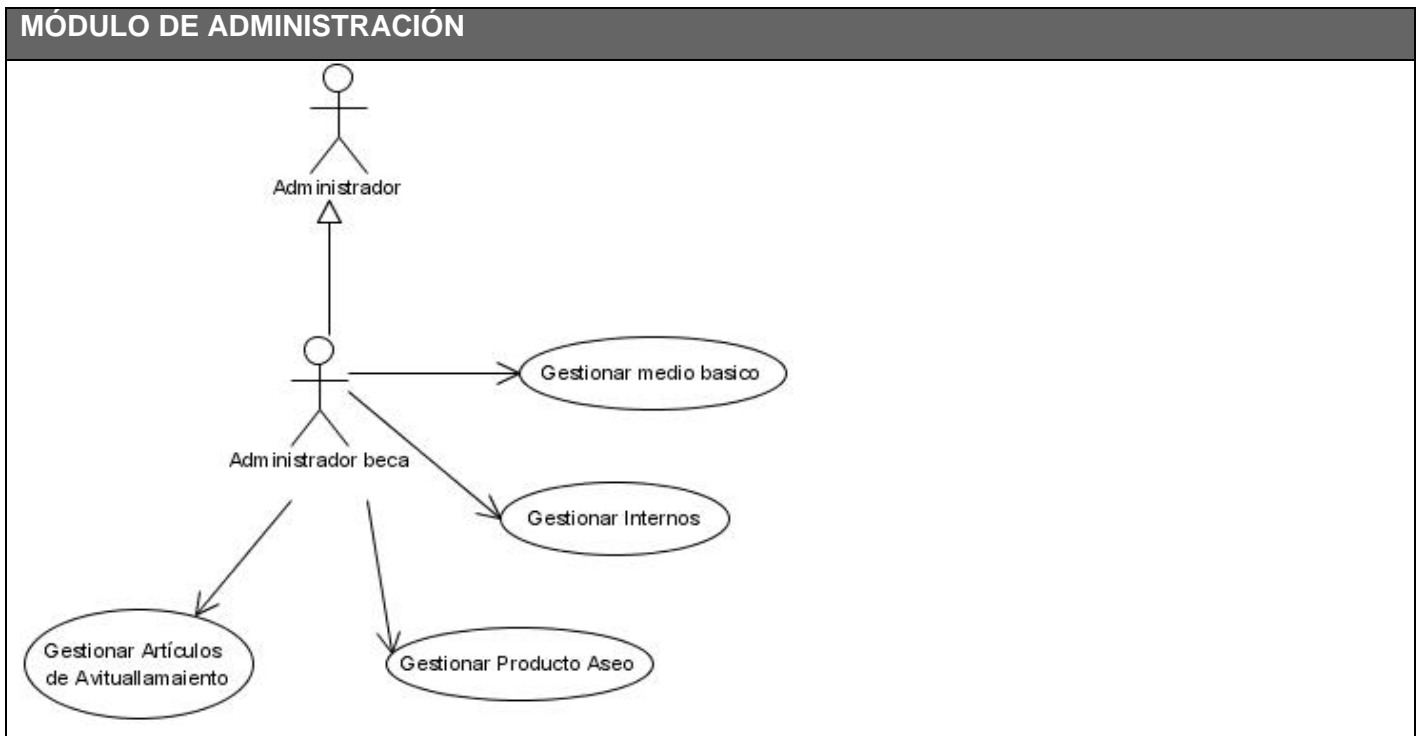


Figura 2.12 Diagrama de casos de uso del sistema. Módulo de administración.



Figura 2.13 Diagrama de casos de uso del sistema. Módulo de Biblioteca.

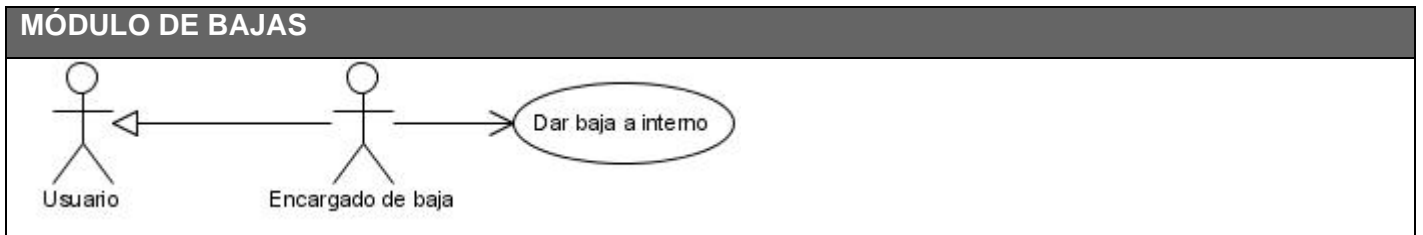


Figura 2.14 Diagrama de casos de uso del sistema. Módulo de Bajas.

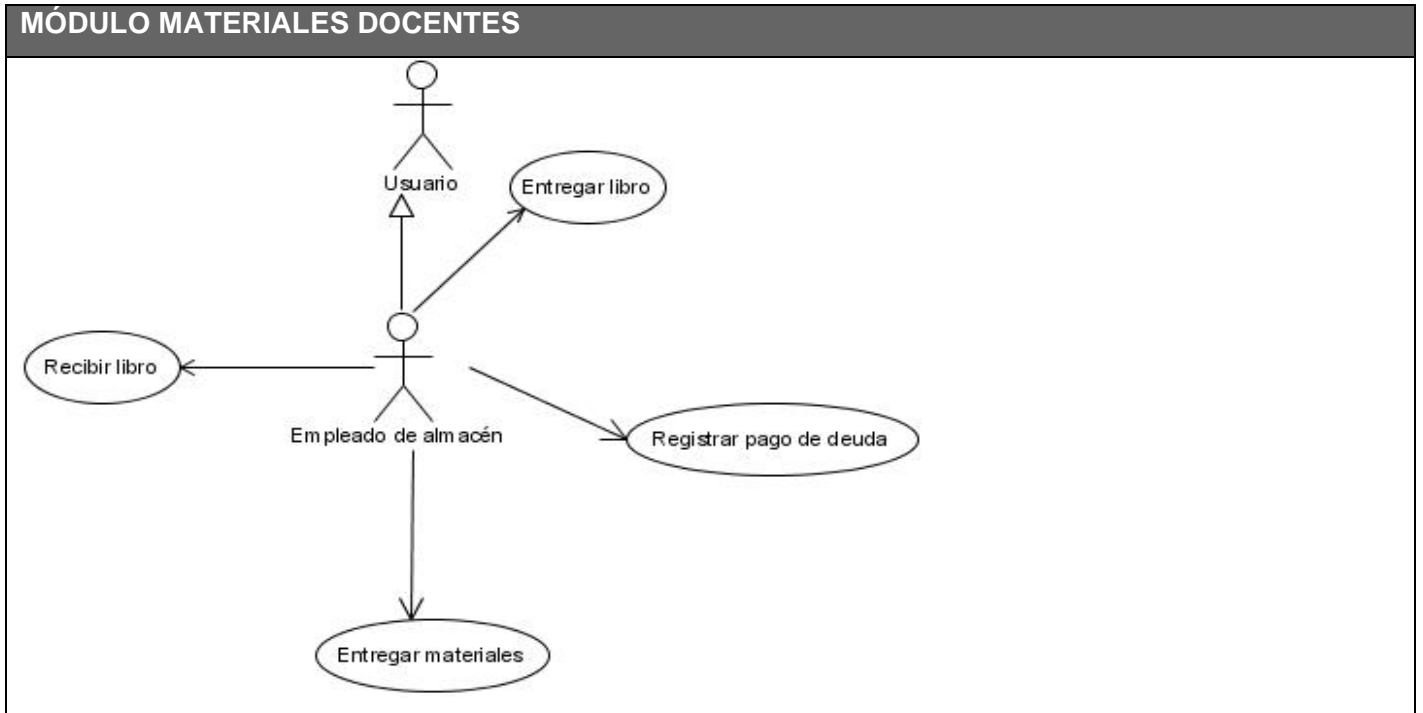


Figura 2.15 Diagrama de casos de uso del sistema. Módulo Materiales Docentes



Figura 2.16 Diagrama de casos de uso del sistema. Módulo Beca.

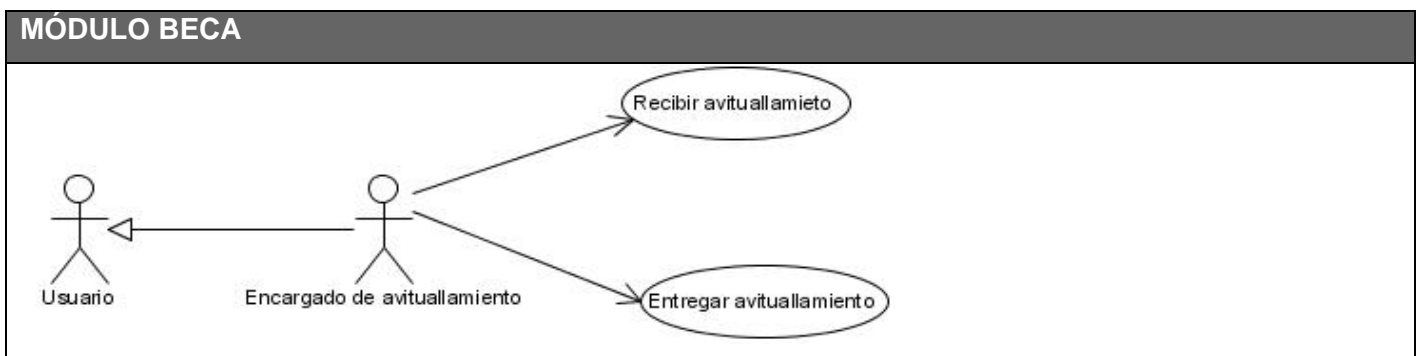


Figura 2.17 Diagrama de casos de uso del sistema. Módulo Beca.

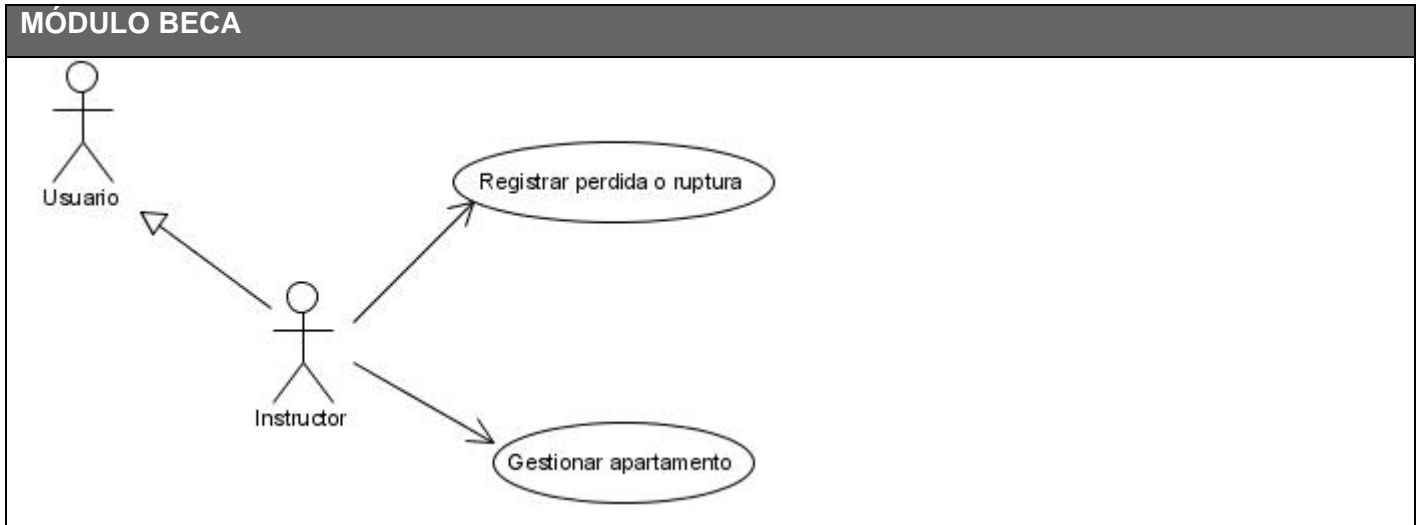


Figura 2.18 Diagrama de casos de uso del sistema. Módulo Beca.



Figura 2.19 Diagrama de casos de uso del sistema. Módulo Beca

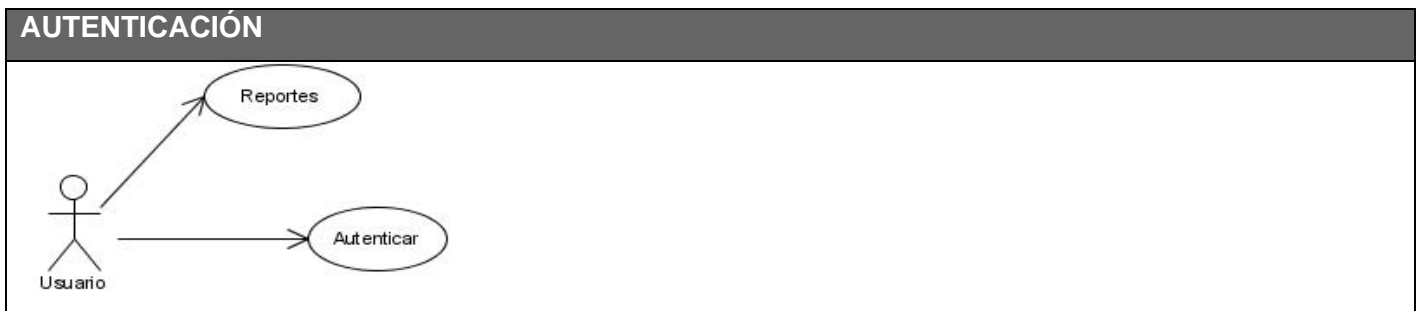


Figura 2.20 Diagrama de casos de uso del sistema. Autenticación

2.5.4 Descripciones textuales del sistema

Aquí se presenta una selección de algunas descripciones textuales de los casos de uso del sistema, ejemplificando los distintos tipos de procesos que se llevan a cabo, estos procesos son entregas, devoluciones, registrar pérdida o rotura, registro de pago de una deuda y el proceso de baja. El resto de las descripciones podrán verse en los anexos.

Caso de Uso:	Realizar préstamo	
Actores:	Bibliotecaria	
Resumen:	El caso de uso se inicia cuando la Bibliotecaria indica que se desea realizar el préstamo de un libro.	
Precondiciones:	El instructor debe haberse autenticado en el sistema.	
Referencias		
Prioridad	Primario	
Flujo Normal de Eventos		
Sección ""		
Acción del Actor	Respuesta del Sistema	
1. Indica que desea realizar préstamo de un libro.		
	2. Muestra la interfaz donde solicita que se introduzca el título del libro a buscar.	
3. Introduce el título del libro solicitado y selecciona el botón "Buscar"		
	4. Muestra mensaje "Hay libro disponible" "Ver flujo alterno"	
5. Pulsa botón "OK"		
	6. Muestra una interfaz con los datos del libro con un botón "Registrar Préstamo "	
7. Selecciona el botón "Registrar Préstamo"		
	8. Muestra la interfaz Registrar Préstamo	

	con los datos del libro obtenidos.
9. Registra los datos del interno, la fecha de inicio del préstamo y la fecha de devolución del libro.	
	10. Guarda los datos introducidos y actualiza en la base de datos la Tarjeta de Préstamo de Libros.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	4.1 Muestra mensaje “No hay libros disponibles”
1. Pulsa botón “OK”	
	2. Muestra nuevamente la interfaz donde se solicita que se introduzca el título del libro a buscar.

Caso de Uso:	Devolver libro
Actores:	Bibliotecaria
Resumen:	El caso de uso se inicia cuando la Bibliotecaria indica que se desea realizar la devolución de un libro.
Precondiciones:	La Bibliotecaria debe haberse autenticado en el sistema.
Referencias	
Prioridad	Primario
Flujo Normal de Eventos	
Sección “”	
Acción del Actor	Respuesta del Sistema
1. Indica que se desea realizar la devolución de un libro.	
	2. Muestra la interfaz donde solicitando que

	se entre el título del libro a devolver.
3. Introduce el título del libro a devolver y pulsa botón "Aceptar"	
	4. Muestra una interfaz donde se listan el título del libro y el número de solapín de los préstamos de ese libro con un botón "Ver datos del préstamo" al lado de cada uno de los préstamos listados.
5. Selecciona el préstamo a ver y pulsa el botón "Ver datos préstamo"	
	6. Muestra una interfaz con todos los datos del préstamo seleccionado y solicita que se seleccione el estado del libro a devolver en el RadioButton (Buen estado o Deteriorado o Perdido)
7. Selecciona la opción "Buen estado" del RadioButton y pulsa el botón "Registrar devolución" <i>"Ver flujo alterno"</i>	
	8. Muestra una interfaz con los datos del préstamo y el mensaje "No debe libros" con un botón "Aceptar"
9. Pulsa el botón "Aceptar"	
	10. Actualiza en la base de datos el Inventario Biblioteca y se elimina el préstamo de la base de datos.

Flujos Alternos

Acción del Actor	Respuesta del Sistema
7.1 Selecciona la opción "Deteriorado o Perdido" del RadioButton y pulsa botón "Registrar devolución"	

	8. Muestra una interfaz con los datos del préstamo, un mensaje con el precio del libro, un CheckBox "Registrar pago" y un botón "Aceptar"
9. Marca el CheckBox "Registrar pago" y pulsa el botón "Aceptar"	
	10. Muestra una interfaz con los datos del préstamo y el mensaje "No debe libros" con un botón "Aceptar"
11. Pulsa el botón "Aceptar"	
	12. Actualiza en la base de datos el Inventario Biblioteca y se elimina el préstamo de la base de datos.

Caso de Uso:	Registrar pago de deuda
Actores:	Empleado de Almacén
Resumen:	El caso de uso se inicia cuando el empleado del almacén selecciona la opción registrar el pago de la deuda.
Precondiciones:	El empleado del Almacén tiene que estar autenticado en el sistema
Referencias	
Prioridad	Primaria
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. Selecciona la opción "Registrar pago"	
	2. Solicita el número de solapín del interno.
3. Introduce el número de solapín solicitado	
	4. Realiza una búsqueda y verifica que el número de solapín es válido.

	5. Muestra el listado de títulos y la cantidad de cada uno de los libros que tiene pendientes a entregar con el importe correspondiente.
6. Selecciona los libros a pagar y oprime el botón "Saldar deuda".	
	7. Actualiza el listado de libros pendientes a entregar por el interno y la cantidad de libros disponibles de ese título.
	8. Muestra un mensaje indicando que la operación se ha realizado correctamente.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	4. Realiza una búsqueda e indica que el número de solapín introducido no es válido en un mensaje de error.

Caso de Uso:	Registrar pérdida o ruptura
Actores:	Instructor
Resumen:	El caso de uso se inicia cuando el instructor indica que desea Registrar pérdida o ruptura de medios básicos a un apartamento. Se registra el apartamento, los medios damnificados y los internos involucrados.
Precondiciones:	El instructor debe haberse autenticado en el sistema.
Referencias	
Prioridad	Primario
Flujo Normal de Eventos	
Sección "Registrar pérdida o ruptura de medio básico"	
Acción del Actor	Respuesta del Sistema
1. Indica que desea Registrar pérdida o ruptura.	
	2. Muestra la interfaz Registrar pérdida o

	ruptura.
3. Selecciona una de las opciones (pérdida o ruptura) y el apartamento e indica Aceptar.	
	4. Muestra listado de medios básicos e internos asignados al apartamento.
5. Selecciona el(los) medio(s) básico(s) a registrar por pérdida o rotura y internos involucrados e indica Registrar.	
	6. Guarda los datos introducidos.
	7. Muestra una interfaz para introducir los datos del "Acta de Responsabilidad Material"
8. Introduce los datos y pulsa el botón "Aceptar"	
	9. Guarda los datos introducidos y actualiza la BD.
Poscondiciones	Queda registrado el apartamento, el(los) medio(s) y el(los) interno(s) involucrados.

Caso de Uso:	Dar baja a interno
Actores:	Encargado de baja
Resumen:	El caso de uso se inicia cuando el Encargado de baja indica que desea dar baja a un interno, verifica que el interno no tiene deudas y luego emite la boleta de baja, finalizando así el caso de uso.
Precondiciones:	El Encargado de baja debe haberse autenticado en el sistema.
Referencias	
Prioridad	Primario

Flujo Normal de Eventos	
Sección ""	
Acción del Actor	Respuesta del Sistema
1. Indica que desea dar baja a un interno.	
	2. Muestra la interfaz de Dar baja a interno.

3. Introduce el número de solapín del interno e indica Aceptar.	
	4. Muestra un mensaje indicando que el interno no tiene deudas. <i>Ver flujo alternativo</i>
5. Selecciona la opción crear boleta de baja e introduce una breve descripción.	
	6. Guarda los datos introducidos y actualiza la BD.
Flujos Alternos	
Sección "Tiene deudas"	
Acción del Actor	Respuesta del Sistema
	4.1 Muestra las deudas que tiene el interno. Finaliza el caso de uso.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

3.1 Introducción

En el flujo de trabajo de análisis y diseño del sistema se profundiza en los casos de uso de forma que permitan reflejar una vista interna del sistema con el lenguaje que han de utilizar los desarrolladores. En esta vista interna se especifican mejor los casos de uso y se determinan las clases necesarias para llevar a cabo las funcionalidades en ellos contenidas. Esto sirve para que los desarrolladores comprendan como debe dársele forma al sistema, primero se esboza como llevar a cabo la funcionalidad dentro del sistema, y luego se diseña el mismo. La realización del análisis constituye una aproximación al diseño, en el mismo se comprenden y refinan los requisitos de software pero sin tener en cuenta el lenguaje o la plataforma en que se desarrollará la aplicación. Con la realización del diseño se obtiene ya lo anteriormente modelado en el análisis teniendo en cuenta el lenguaje a utilizar por los desarrolladores, quedando descrita la arquitectura a utilizar en la construcción del sistema y diseñada la Base de Datos a utilizar por el mismo. En este capítulo se abordarán aspectos relacionados con este flujo de trabajo y se presentarán los artefactos vinculados al mismo.

3.2 Análisis

En el modelo del análisis se estructuran los requisitos de manera que nos facilite su comprensión, preparación, su modificación y su mantenimiento en general. Aquí se refinan los requisitos profundizando en el dominio de aplicación y se le asignan funcionalidades a un grupo de objetos. A continuación se mostrarán los diagramas de clases del análisis para algunos casos de uso que se seleccionaron como primarios, el resto podrá verse en el Anexo II.

3.2.1 Modelo del análisis.

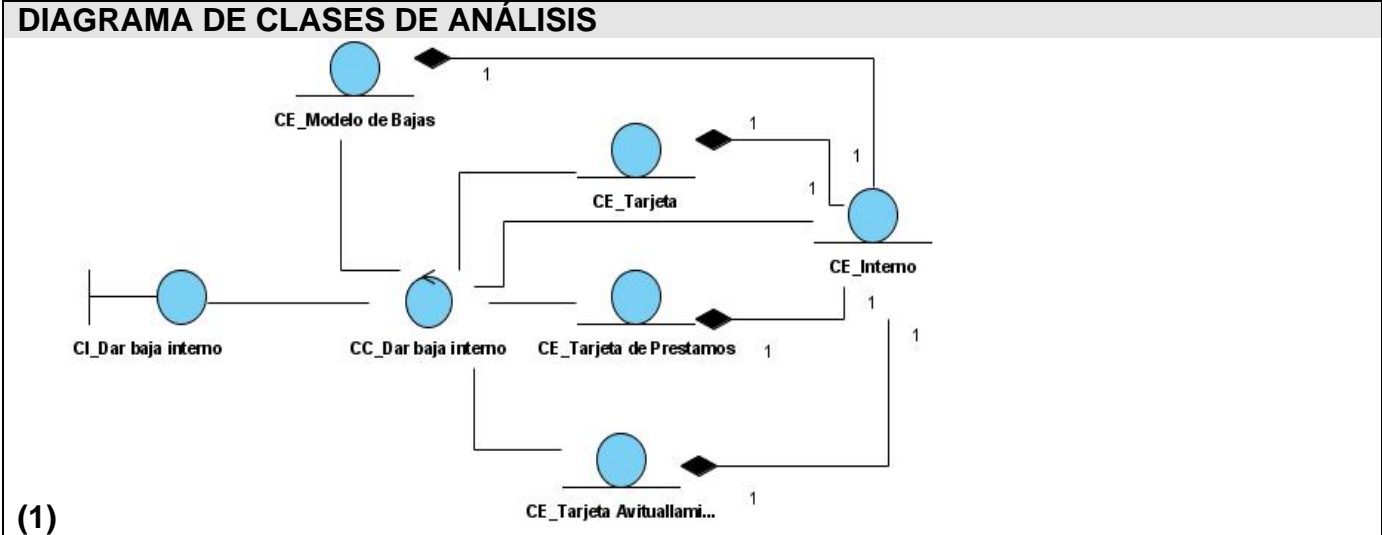


Figura 3.1 Diagrama de clases del análisis-CU Dar baja a interno.

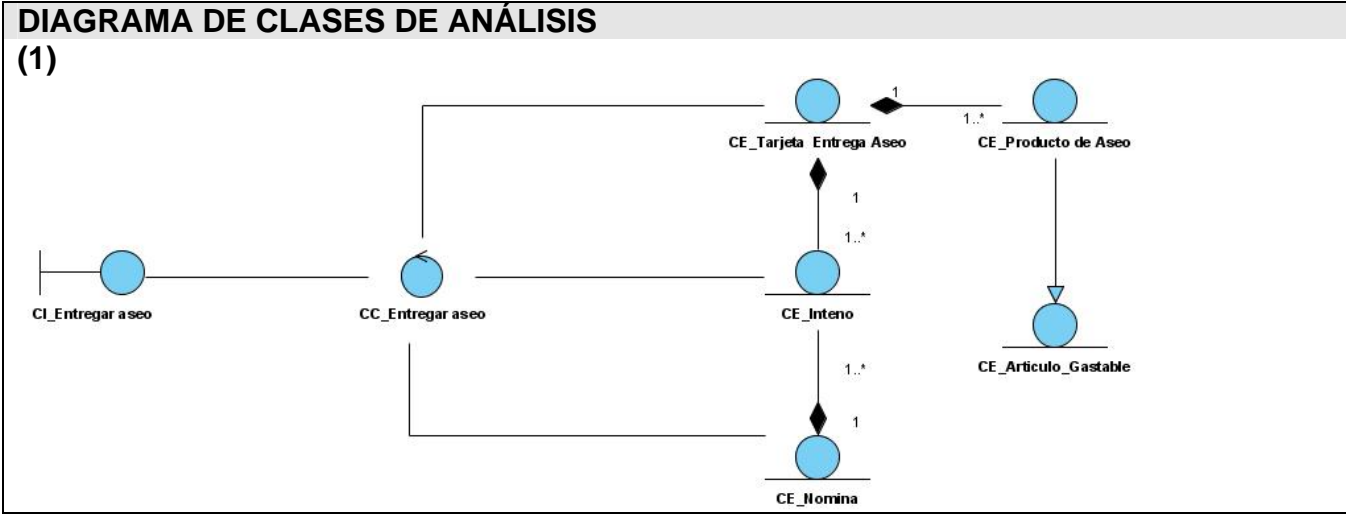


Figura 3.2 Diagrama de clases del análisis-CU Entregar Aseo.

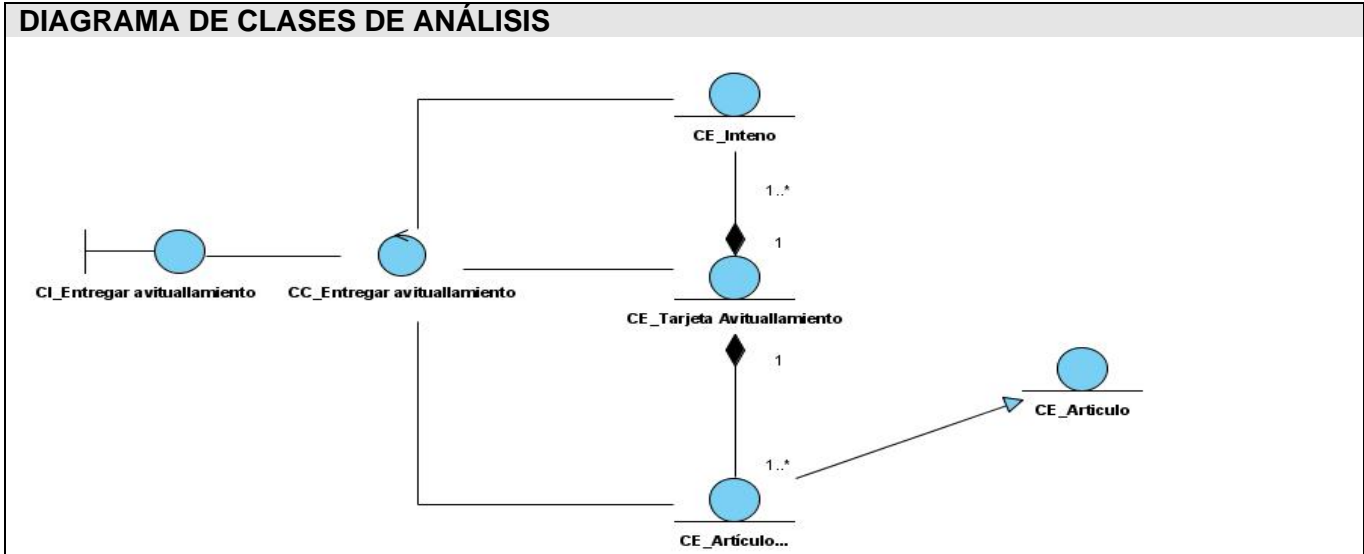


Figura 3.3 Diagrama de clases del análisis-CU Entregar Avituallamiento

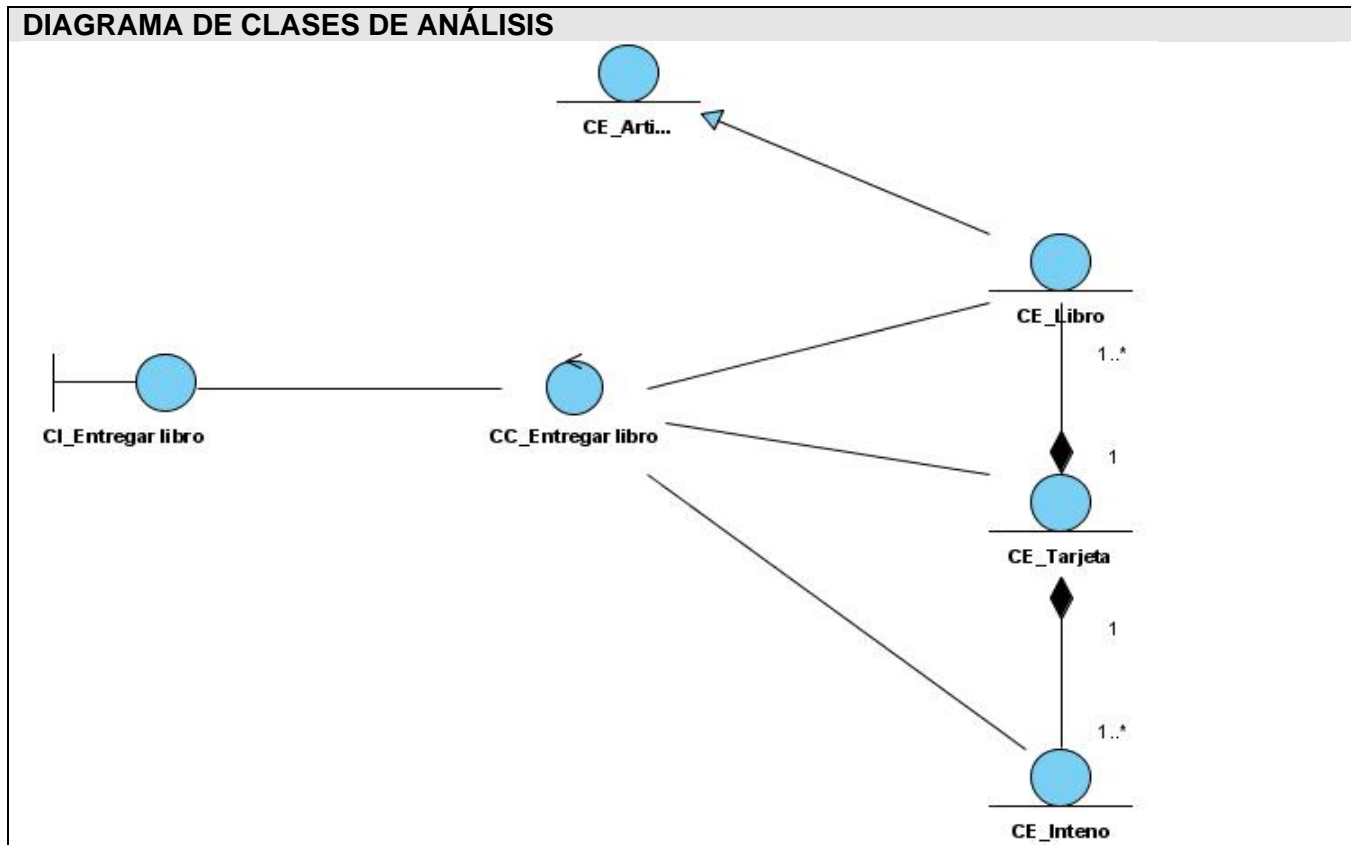


Figura 3.4 Diagrama de clases del análisis-CU Entregar Libro.

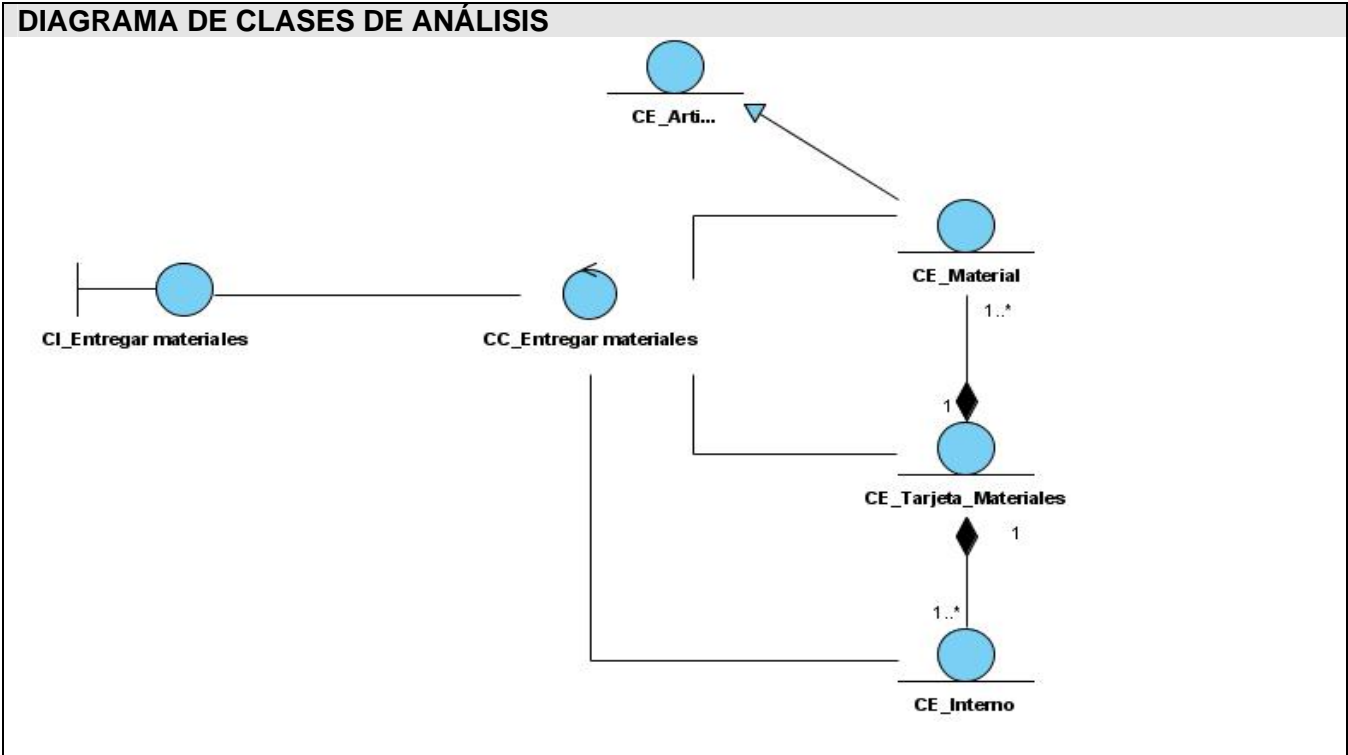


Figura 3.5 Diagrama de clases del análisis-CU Entregar Materiales.

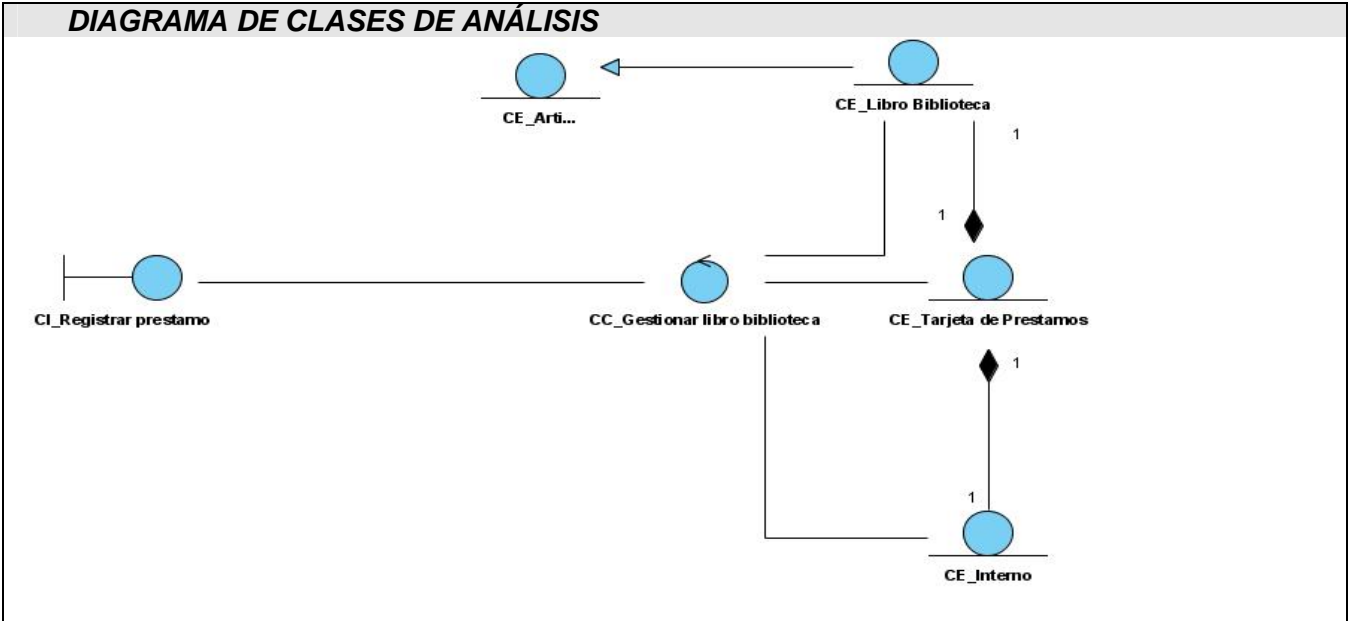


Figura 3.6 Diagrama de clases del análisis-CU Realizar Préstamo.

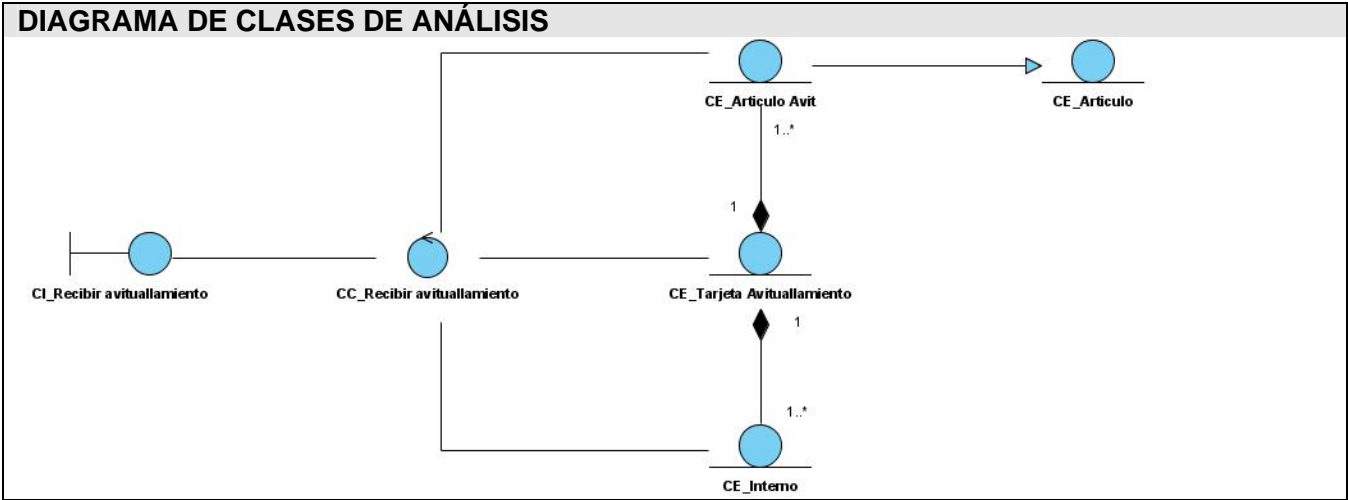


Figura 3.7 Diagrama de clases del análisis-CU Recibir Avituallamiento.

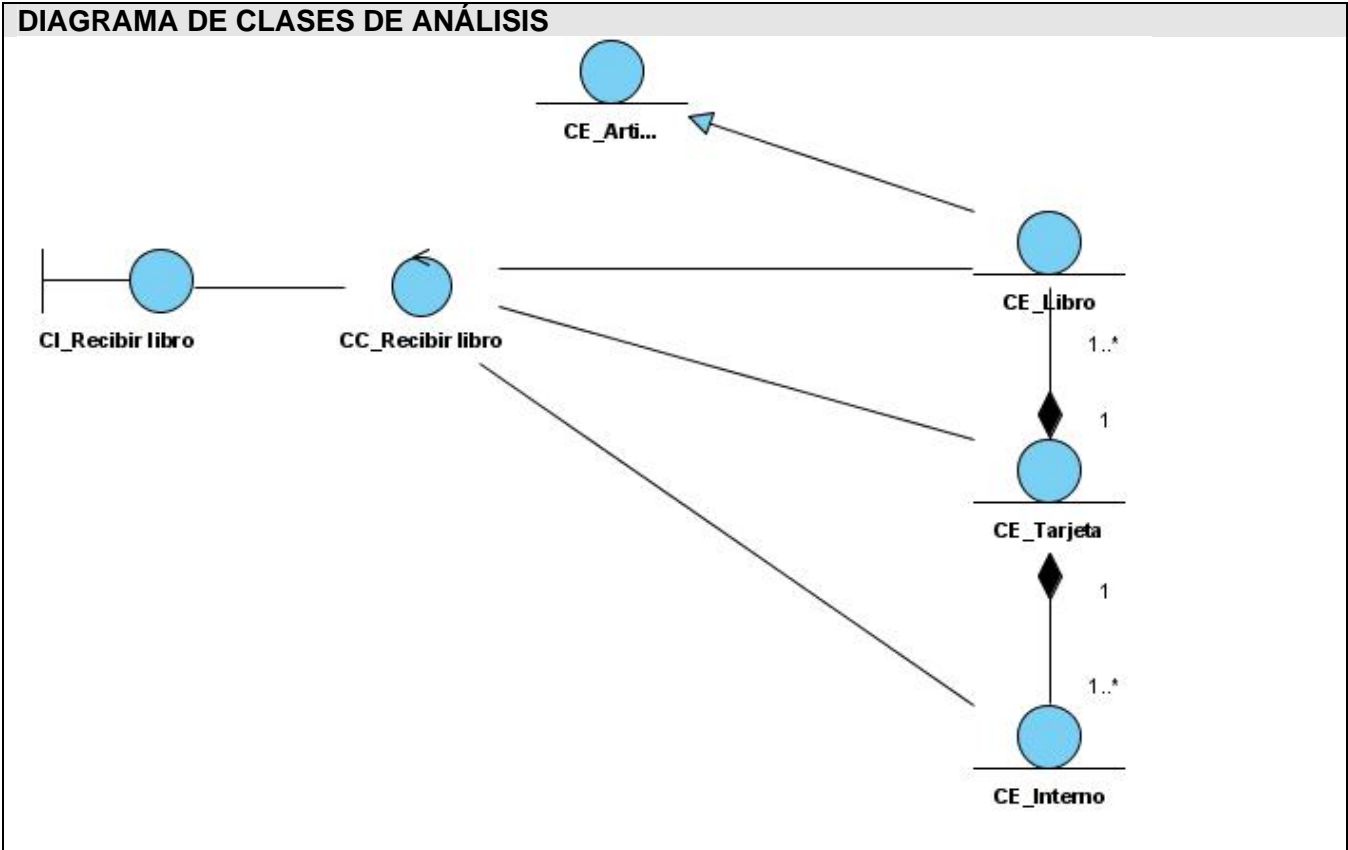


Figura 3.8 Diagrama de clases del análisis-CU Recibir Libro.

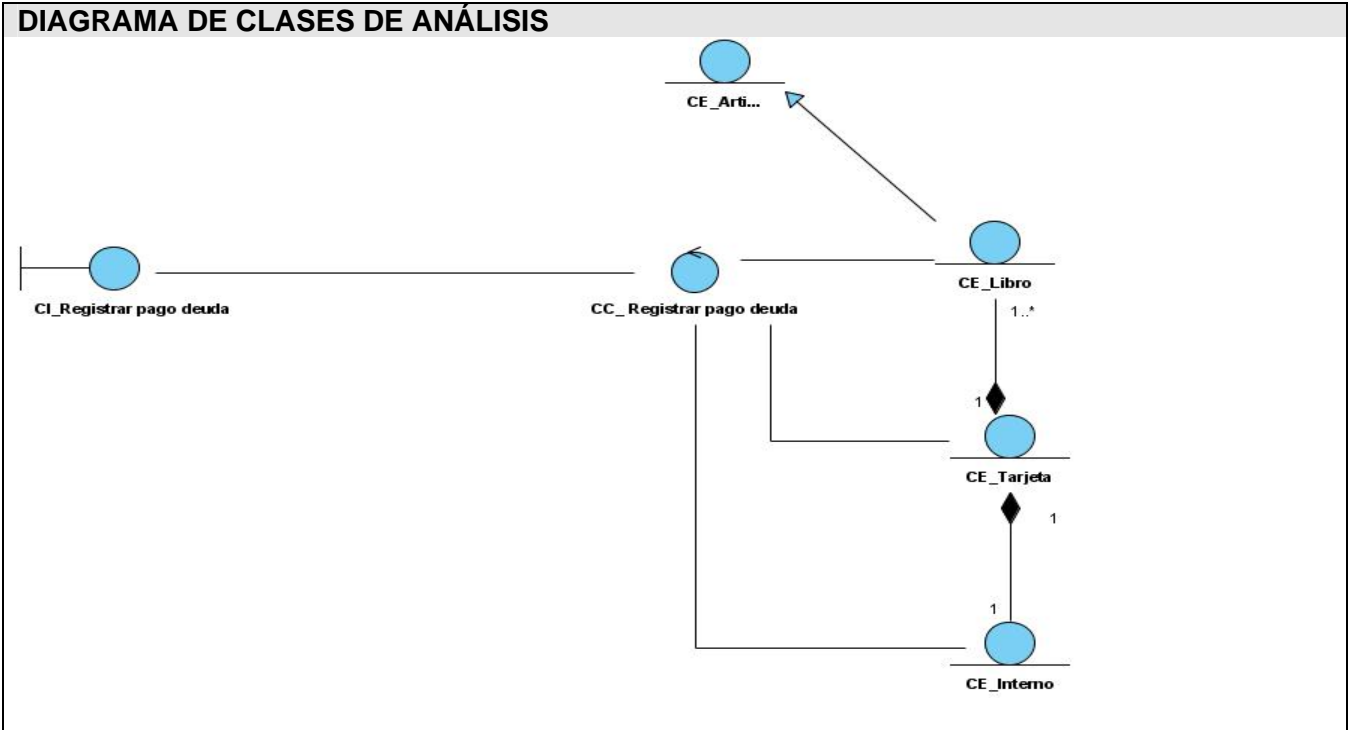


Figura 3.9 Diagrama de clases del análisis-CU Registrar pago de deuda de Libro.

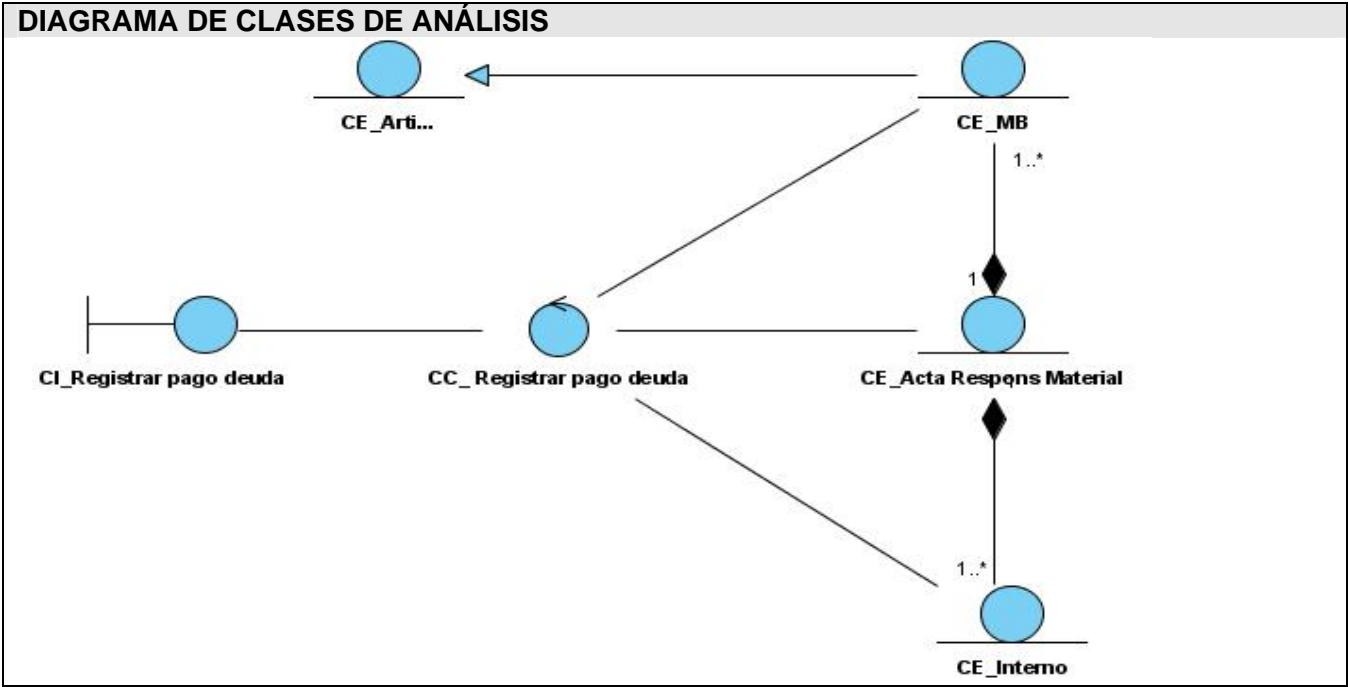


Figura 3.10 Diagrama de clases del análisis-CU Registrar pago de deuda de Medio Básico.

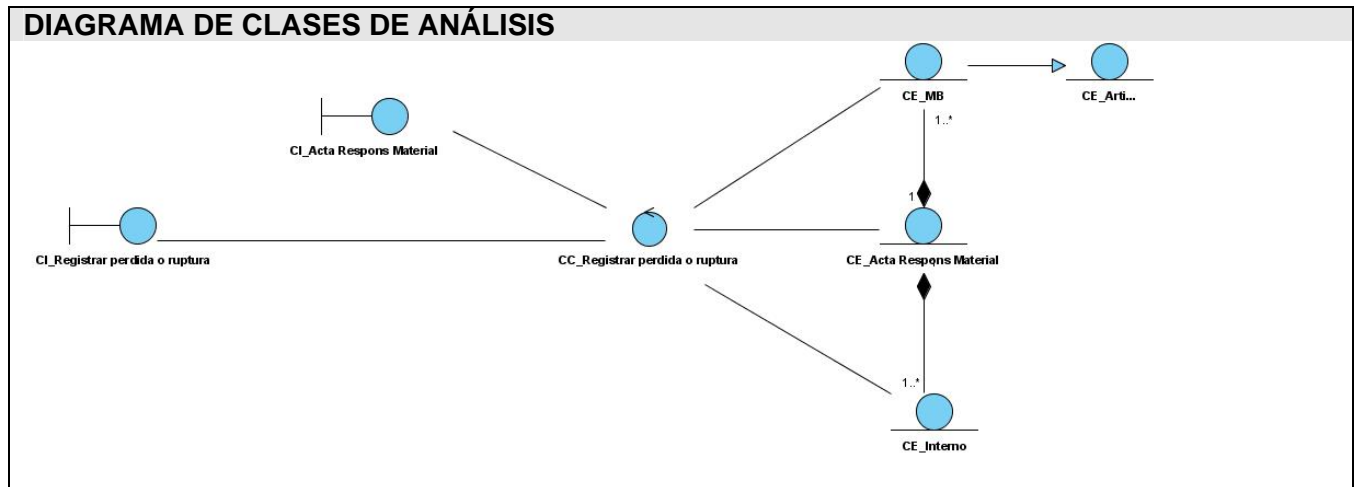


Figura 3.9 Diagrama de clases del análisis-CU Registrar pérdida o rotura de Medio Básico.

3.3 Diseño

Para desarrollar el diseño del sistema se realizaron los diagramas de clases elaborados en el modelo de análisis, enfocado al lenguaje PHP.

3.3.1 Diagramas de interacción.

Los diagramas de interacción (secuencia) podrán verse en el Anexo III.

3.3.2 Diagramas de clases del diseño.

En este epígrafe se presentarán los diagramas de clases del portal, autenticar, un diagrama de clases genérico para el caso de los reportes y un diagrama de clases genérico general, los cuales fueron diseñados para hacer más sencillo el diseño del resto de los diagramas de clases y para una mejor comprensión de los mismos, un diagrama de clases para el mecanismo de acceso a datos así como otros diagramas seleccionados como primarios, el resto de los diagramas podrán verse en el Anexo IV.

A continuación se muestra el diagrama de clases del caso de uso Autenticar unido al gestionar portal, puesto a que para un usuario poder acceder al portal debe haberse autenticado en el sistema, una vez autenticado el usuario y haberse comprobado la seguridad, entonces se muestra la página del portal.

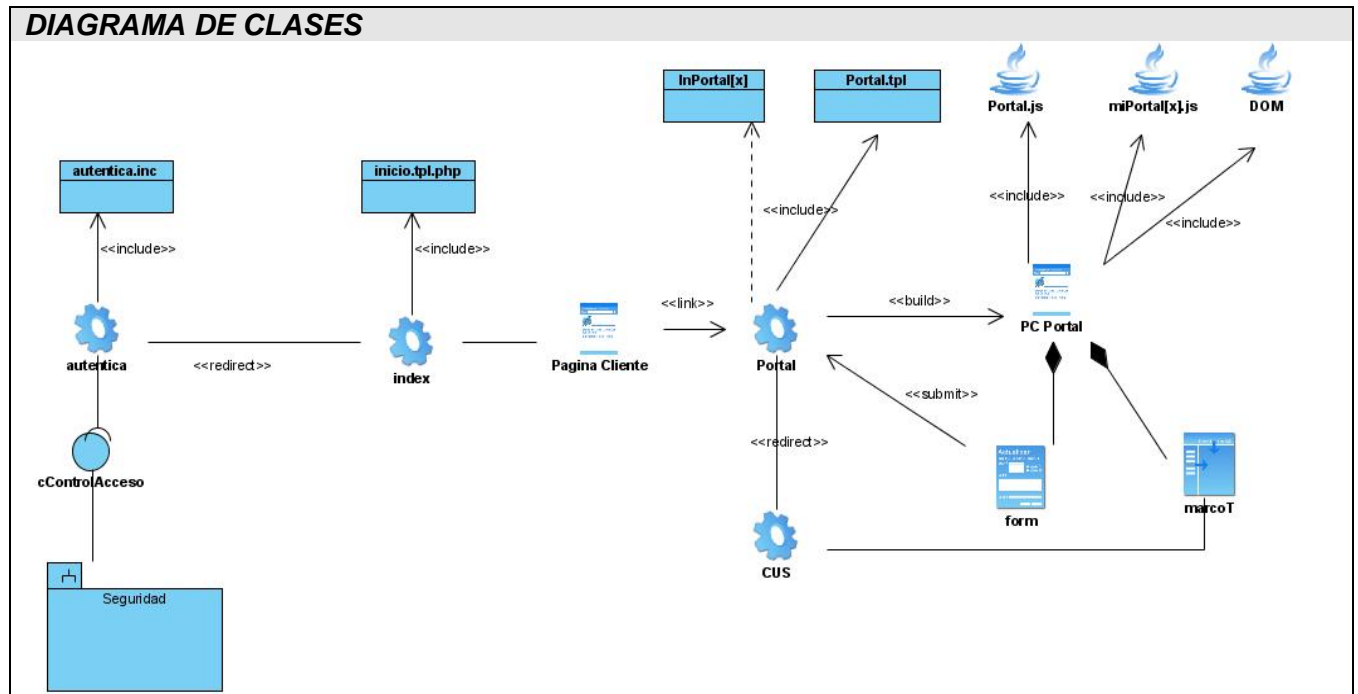


Figura 3.10 Diagrama de clases del diseño: Autenticar y Gestionar Portal.

A continuación se presenta el diagrama genérico de clases del diseño, donde aparecen una página servidora (SP_CUx), una página cliente (CP_CUx) con los ficheros js y los componentes y DOM, además de una clase lógica del negocio (L_Neg_CUx), que representan a las páginas de cada caso de uso en particular, los cuales están presentes en todos los diagramas de clases de cada caso de uso cambiando el nombre según el caso de uso que se esté representando. Además aparecen el frame ifMarco, la clase Grid y la clase Miscelánea, las cuales no están presentes en todos los diagramas de clases de todos los casos de uso, solo en aquellos que necesiten cargar información dinámicamente en una plantilla para el caso de la clase Miscelánea y cuando se utilice un Grid en la interfaz para el caso de la clase Grid.

La página DOM y los ficheros js y dhtmlx se incluyen siempre para todos los diagramas de clases de todos los casos de uso. La página del portal no está presente siempre en los diagramas de clases de todos los casos de uso puesto que estos pueden ser invocados tanto desde la página correspondiente a un determinado caso de uso que esté en ejecución como desde la página del portal. La clase Factoría Típica se utiliza siempre que se requiera hacer una modificación en alguna de las clases de datos o sobre una determinada tabla de la Base de Datos o cuando se desee instanciar un objeto determinado. Ver tablas de descripción de las clases y ver diagrama genérico figura 3.11.

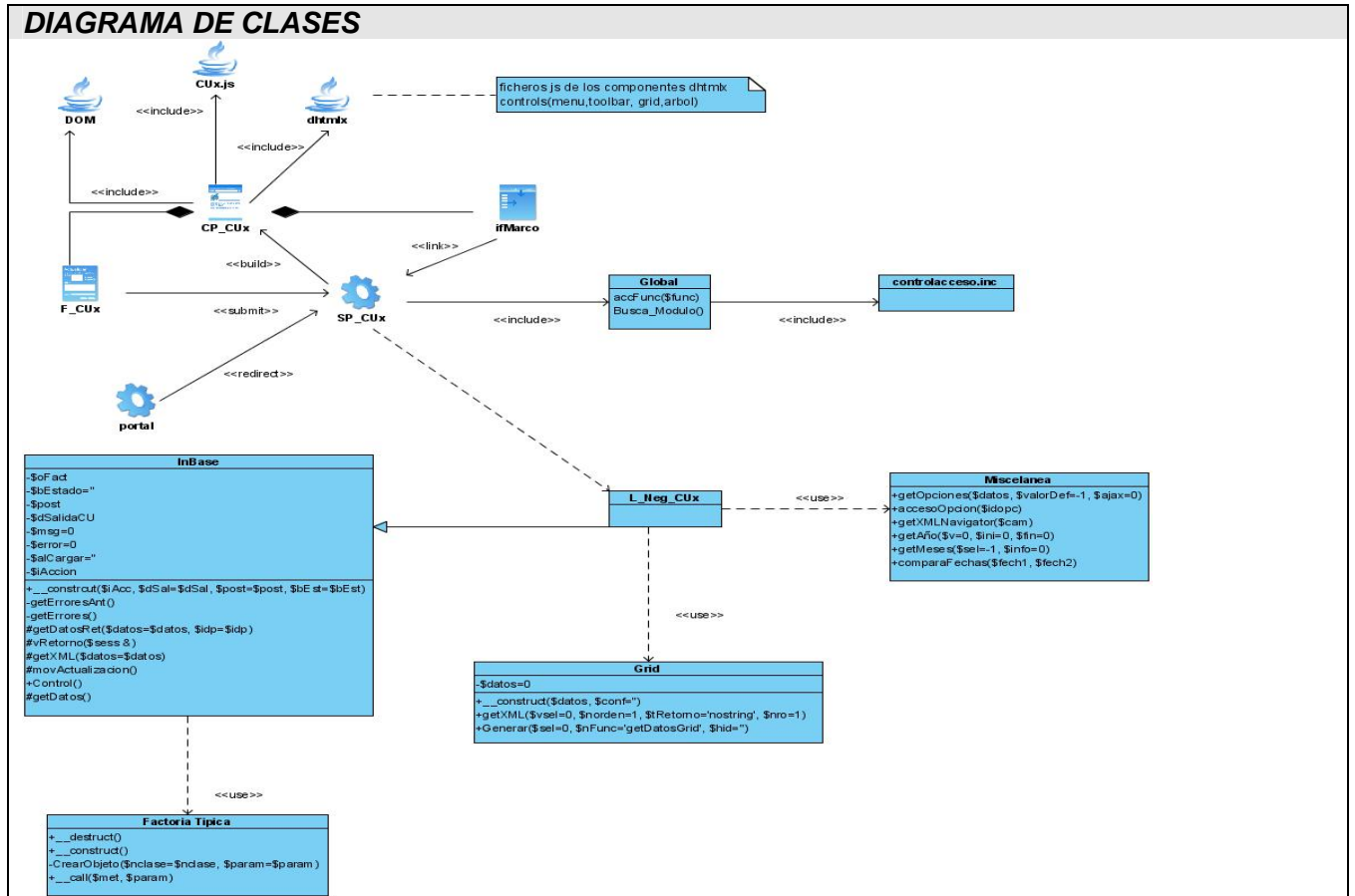


Figura 3.11 Diagrama de clases del diseño: Genérico.

El diagrama de clases del diseño genérico para el caso de uso de los reportes contiene una página servidora (SP_ReportesX), una página cliente (CP_ReportesX), un formulario (F_ReportesX), la página js (ReportesX.js) y la clase L_Neg_ReportesX que representan a las clases correspondientes a cada caso de uso que esté en ejecución.

Las clases Miscelánea y Grid son utilizadas en todos los diagramas de clases de todos los casos de uso de reportes, puesto a que en todos es necesaria la utilización de Grid así como de la incorporación de contenido dinámico en las plantillas. La página del portal puede estar presente o no en el resto de los diagramas, ya que no necesariamente los casos de uso de reportes tienen que ser invocados desde la página del portal, pues pueden ser invocados también desde la página de otro caso de uso en ejecución.

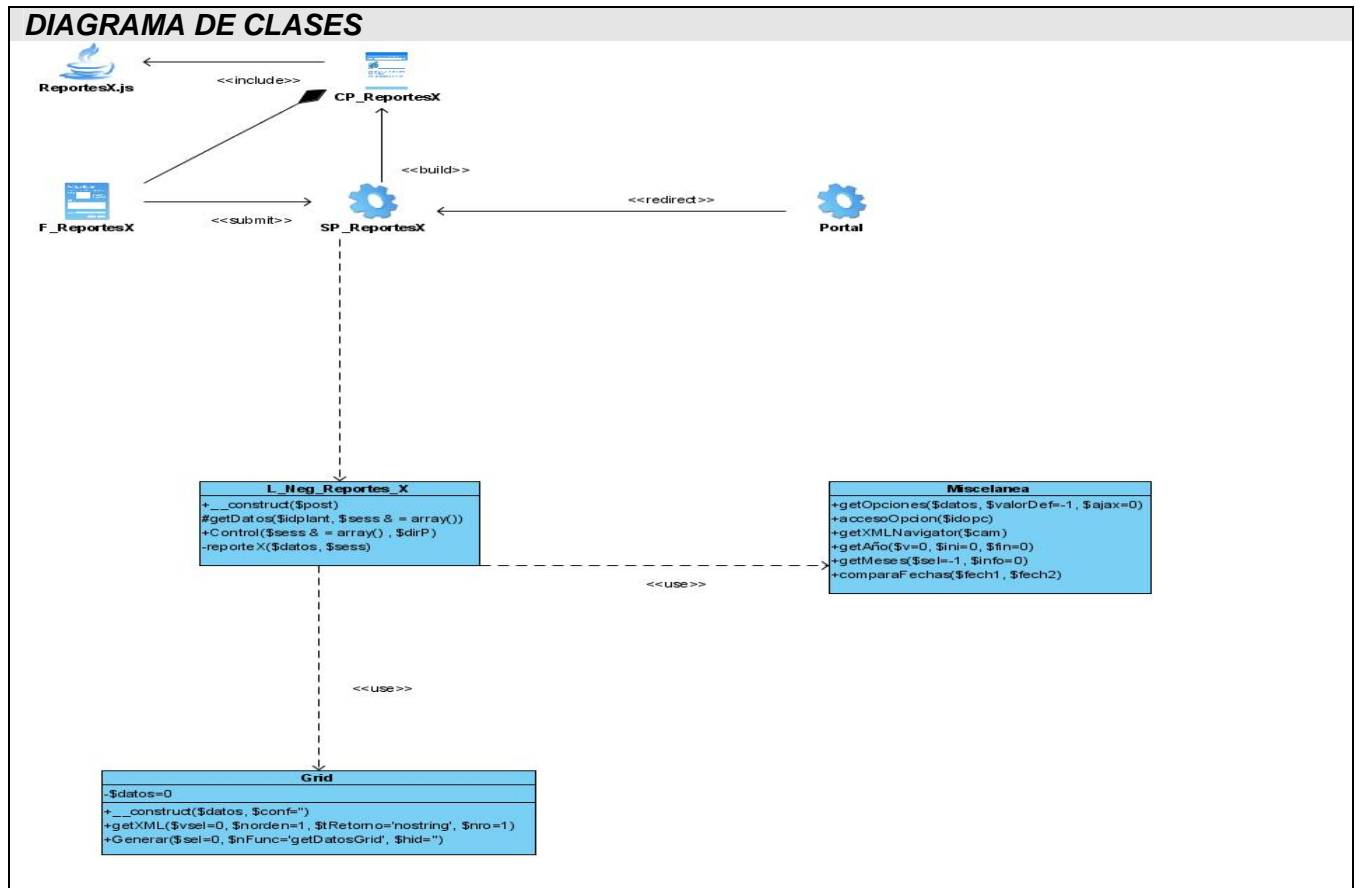


Figura 3.12 Diagrama de clases del diseño: Genérico Reportes

En el diseño de la mayoría de los casos de uso se encontraron operaciones o características comunes entre muchas de las clases diseñadas para cada caso de uso en particular, surgiendo la necesidad de implementar un mecanismo para simplificar el trabajo así como permitir una mejor comprensión de los mismos y a su vez la reutilización de código por los programadores, decidiéndose agruparlos según patrones básicos encontrados.

Se encontraron dos patrones básicos fundamentales, a uno se decidió llamarle patrón “Artículo” debido a que contiene las características básicas de la gran mayoría de los artículos y al otro patrón se le nombró “OtroArtículo”. El patrón “Artículo” es de fácil identificación, responde a la siguiente estructura: nombreArtículo, precioArtículo, estadoArtículo, cantidadArtículo tipoArtículo. Al mismo responden las clases “MB”, “Libro”, “Libro”, “Libro_Biblioteca” y “Artículo_Avituellamiento”.

El patrón “Artículo_Gastable” responde a la estructura: nombreArtículo, precioArtículo, cantidadArtículo y tipoArtículo, se decidió nombrarlo así debido a que recoge características básicas de los artículos sin importarle el estado del mismo ya que son artículos gastables y ese patrón responden las clases “Producto_Aseo” y “Material”.

Las operaciones básicas tienen un comportamiento similar en la mayor parte de los casos debido a la utilización de estos mecanismos. A continuación se presentan los diagramas de clases de los mismos.

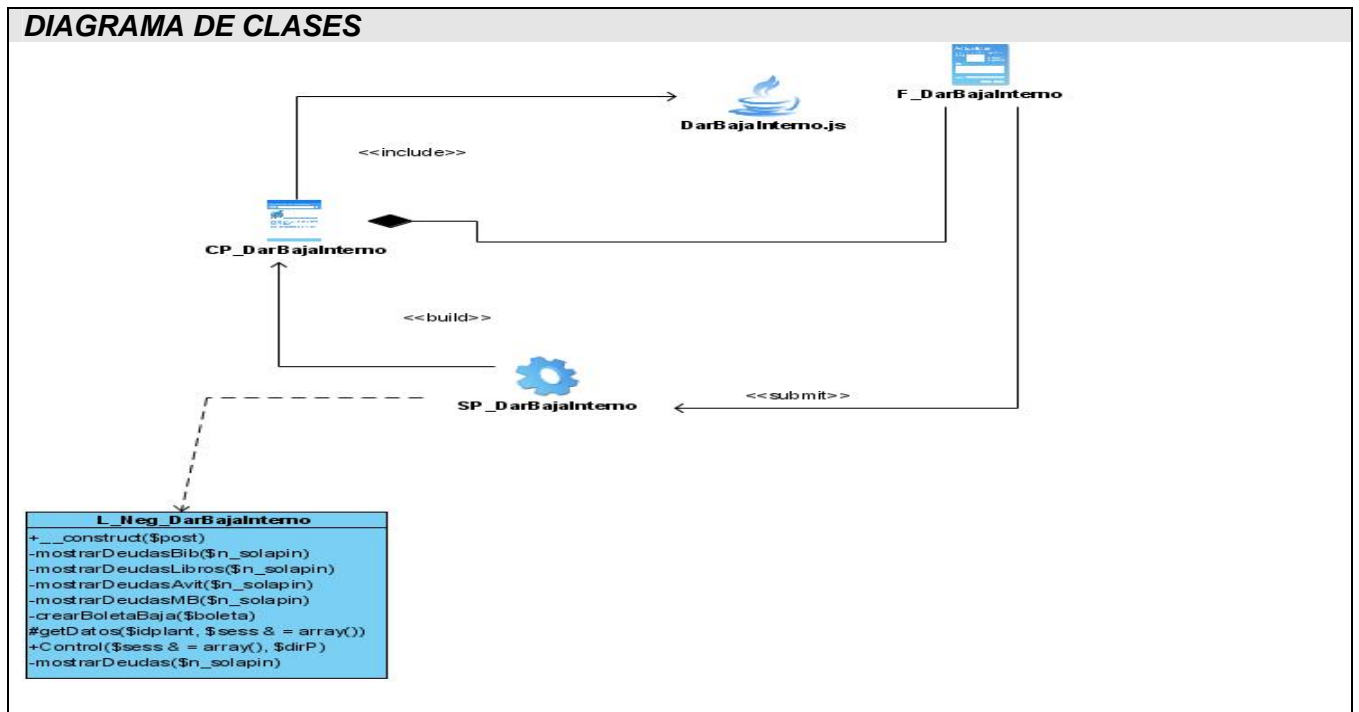


Figura 3.13 Diagrama de clases del diseño: Dar baja a interno.

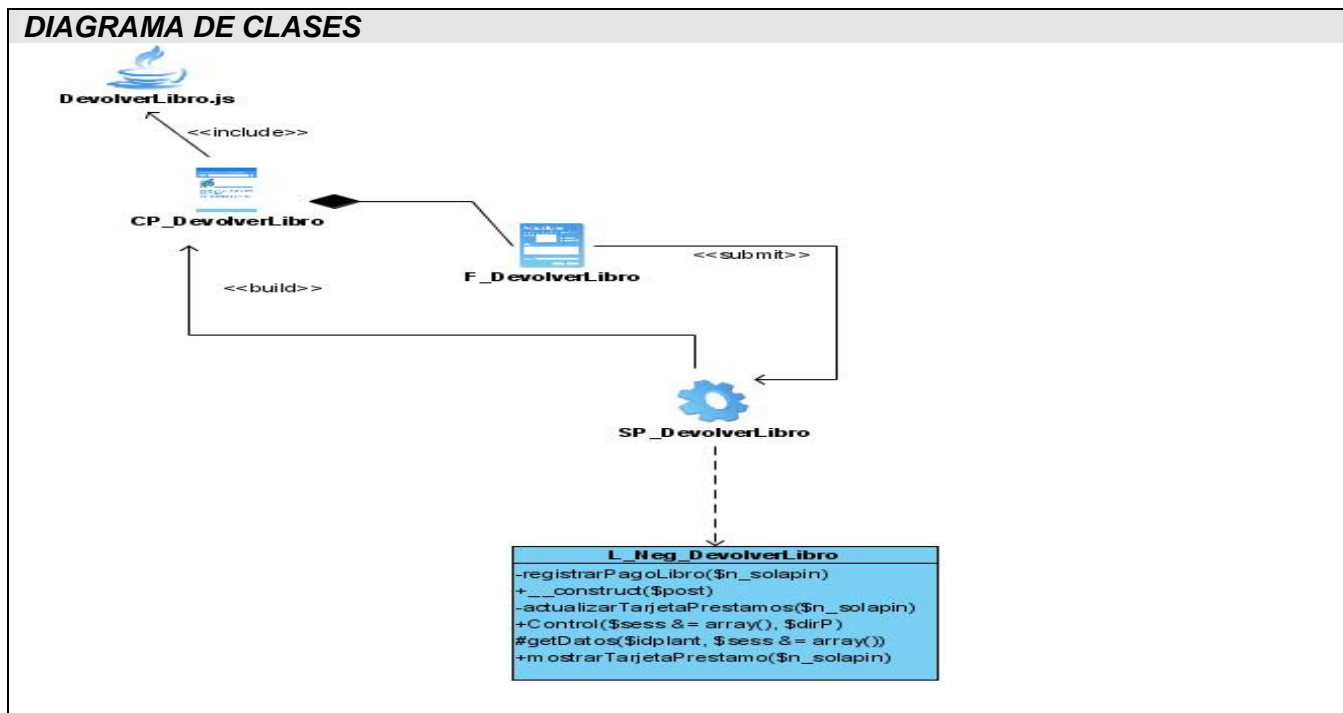


Figura 3.14 Diagrama de clases del diseño: Devolver libro.

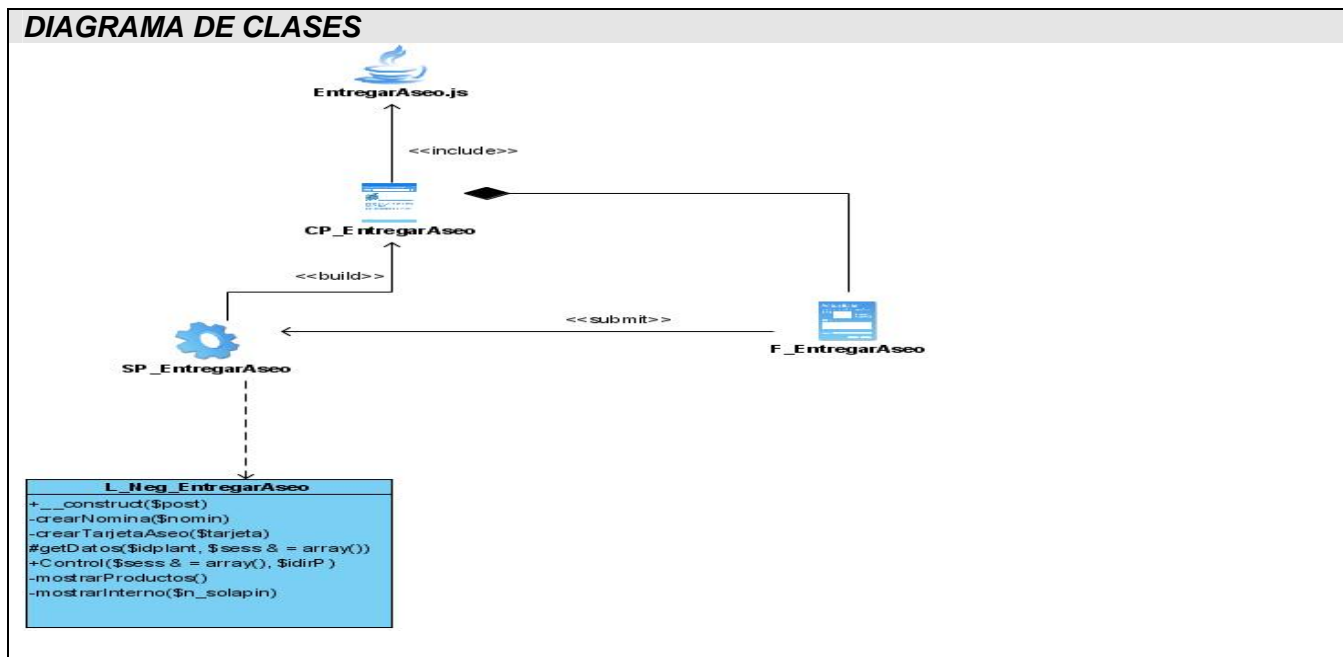


Figura 3.15 Diagrama de clases del diseño: Entregar Aseo.

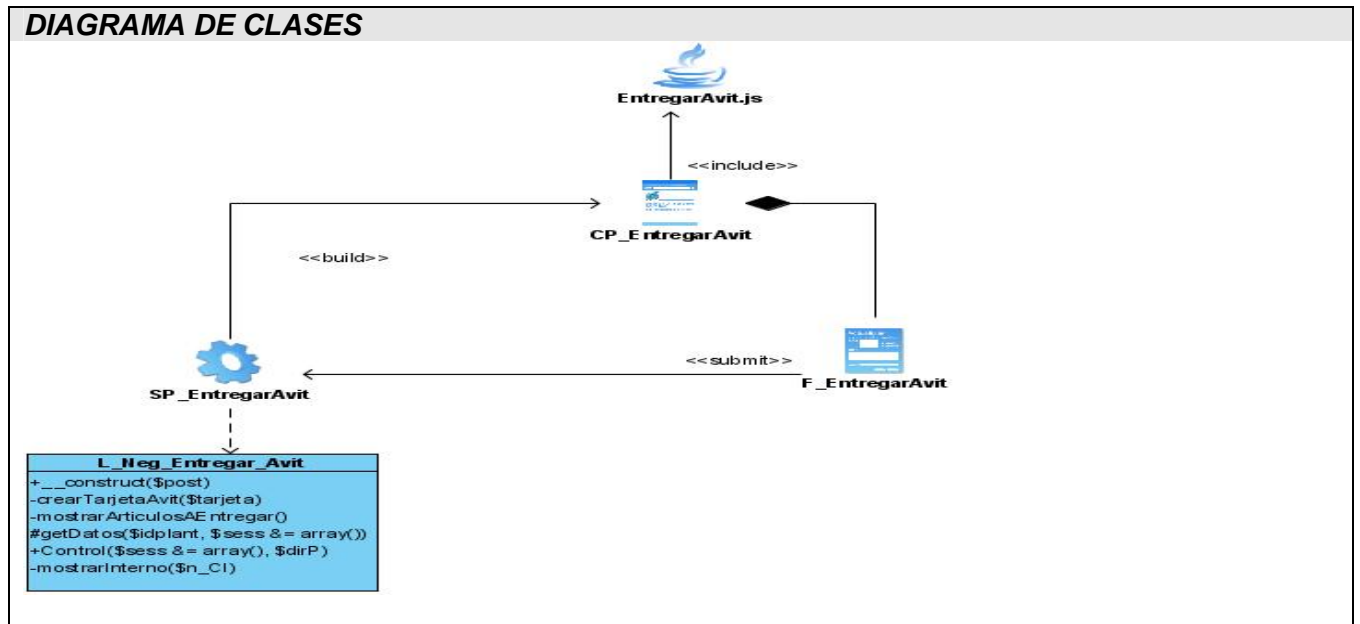


Figura 3.16 Diagrama de clases del diseño: Entregar Avituallamiento.

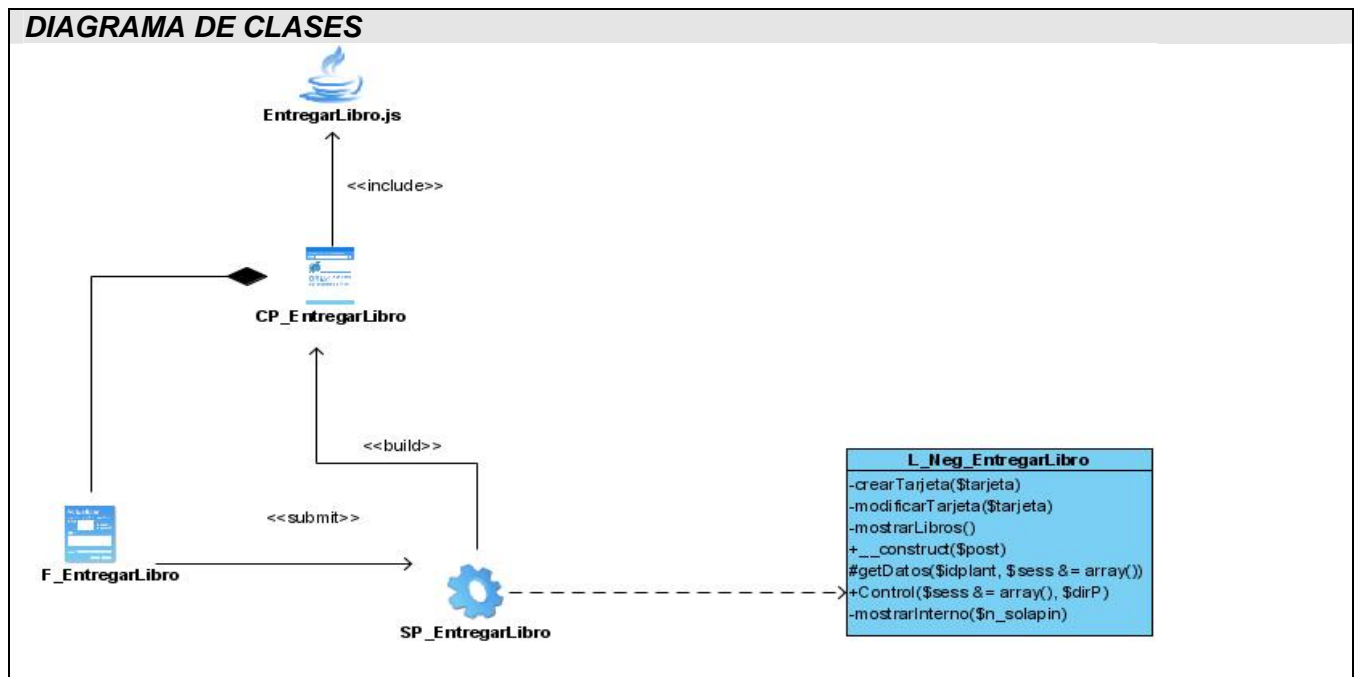


Figura 3.17 Diagrama de clases del diseño: Entregar Libro.

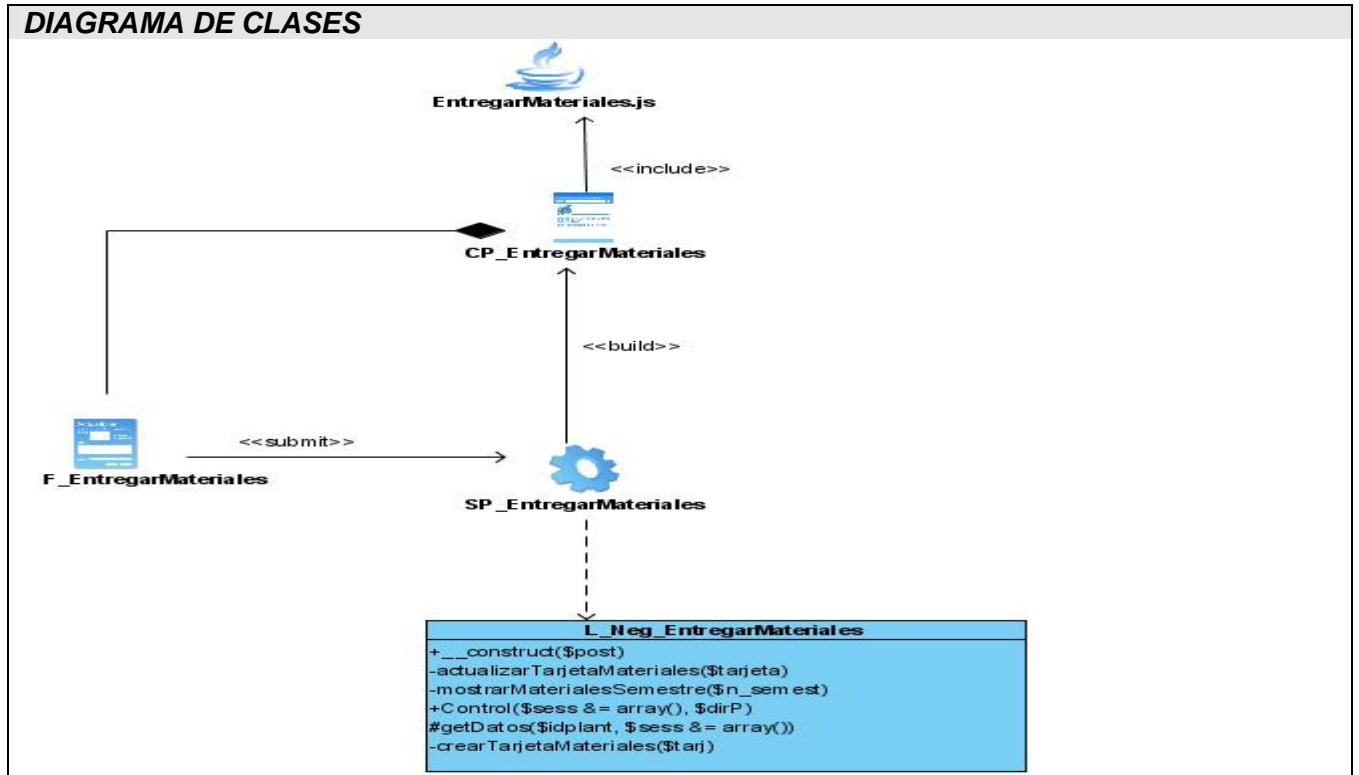


Figura 3.18 Diagrama de clases del diseño: Entregar Materiales.

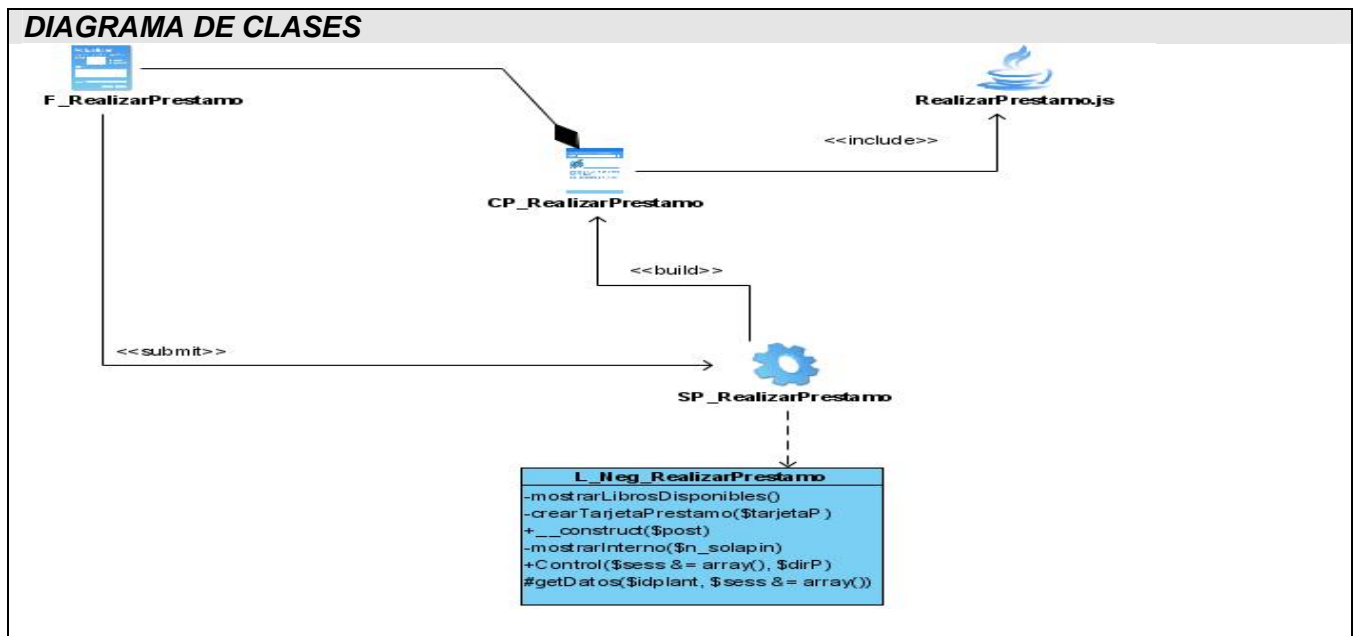


Figura 3.19 Diagrama de clases del diseño: Realizar Préstamo.

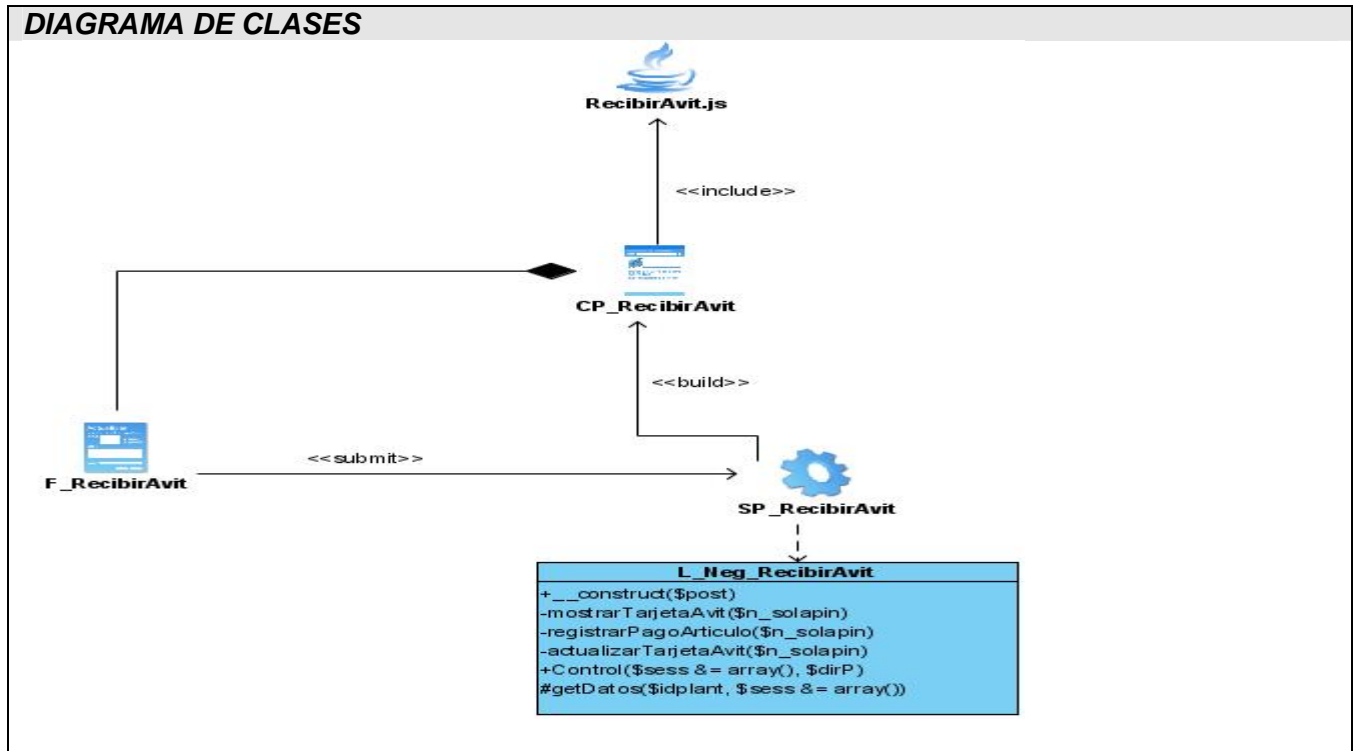


Figura 3.20 Diagrama de clases del diseño: Recibir Avituallamiento.

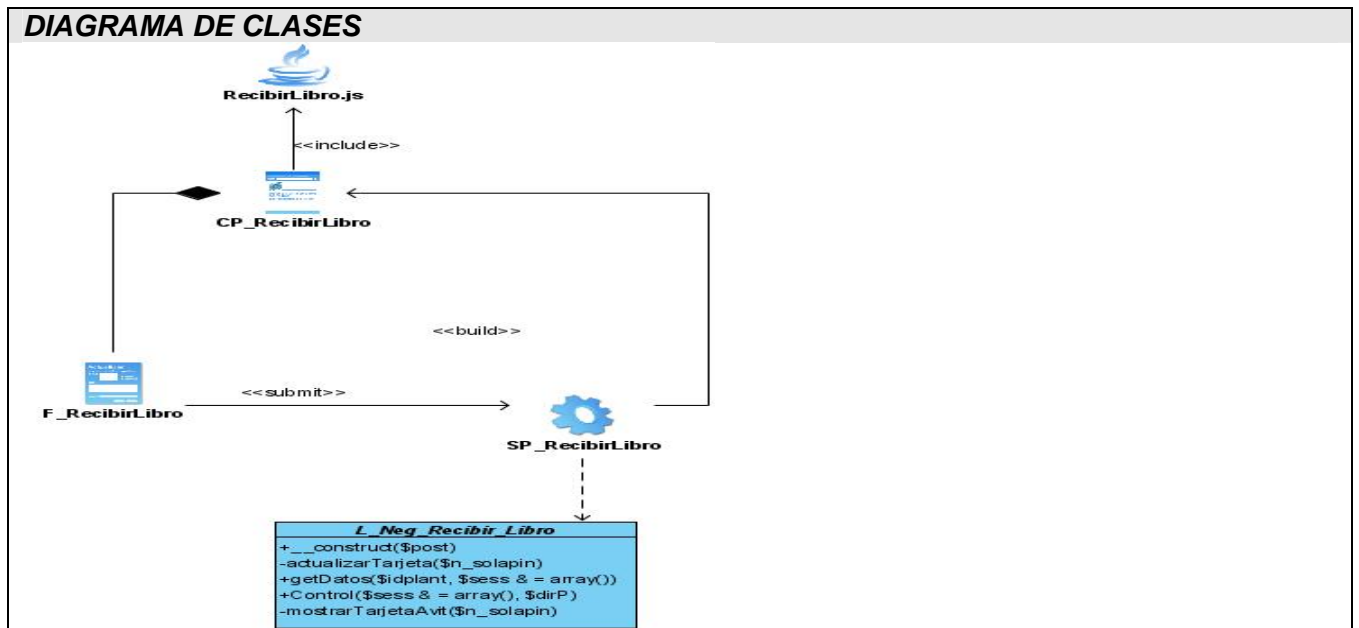


Figura 3.21 Diagrama de clases del diseño: Recibir Libros

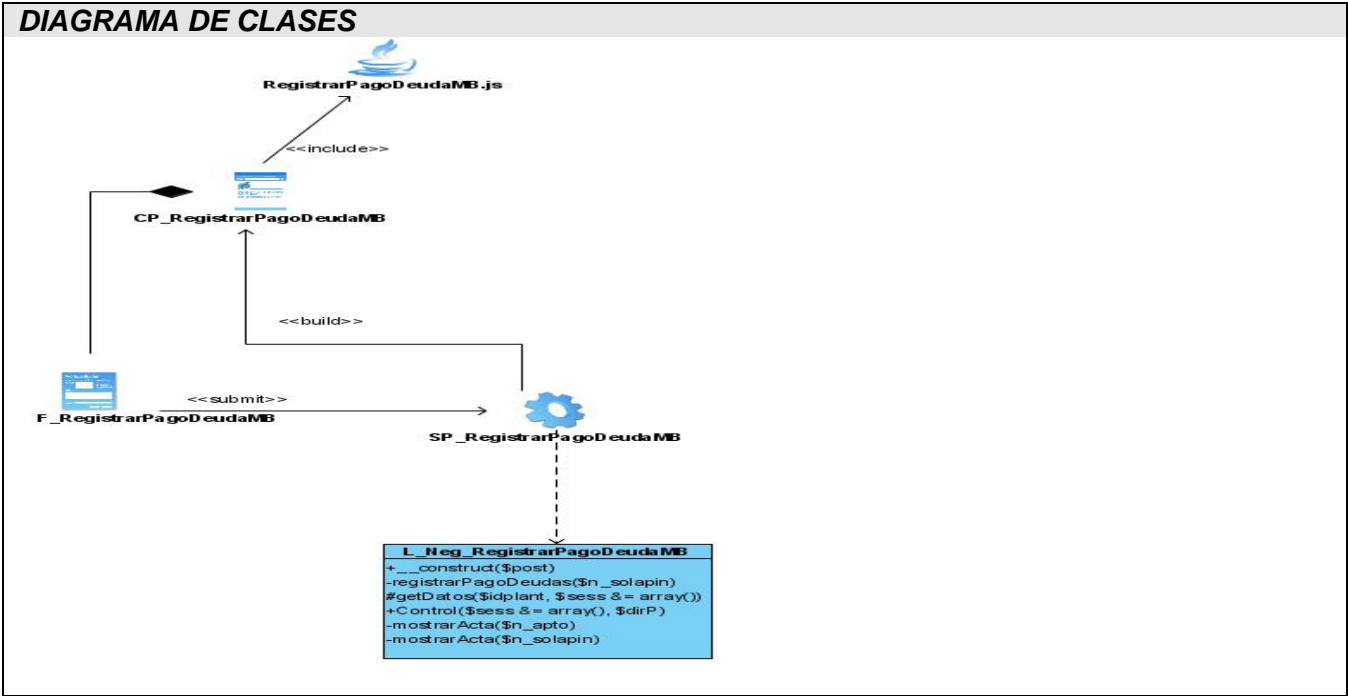


Figura 3.22 Diagrama de clases del diseño: Registrar Pago de Deuda de Medio Básico.

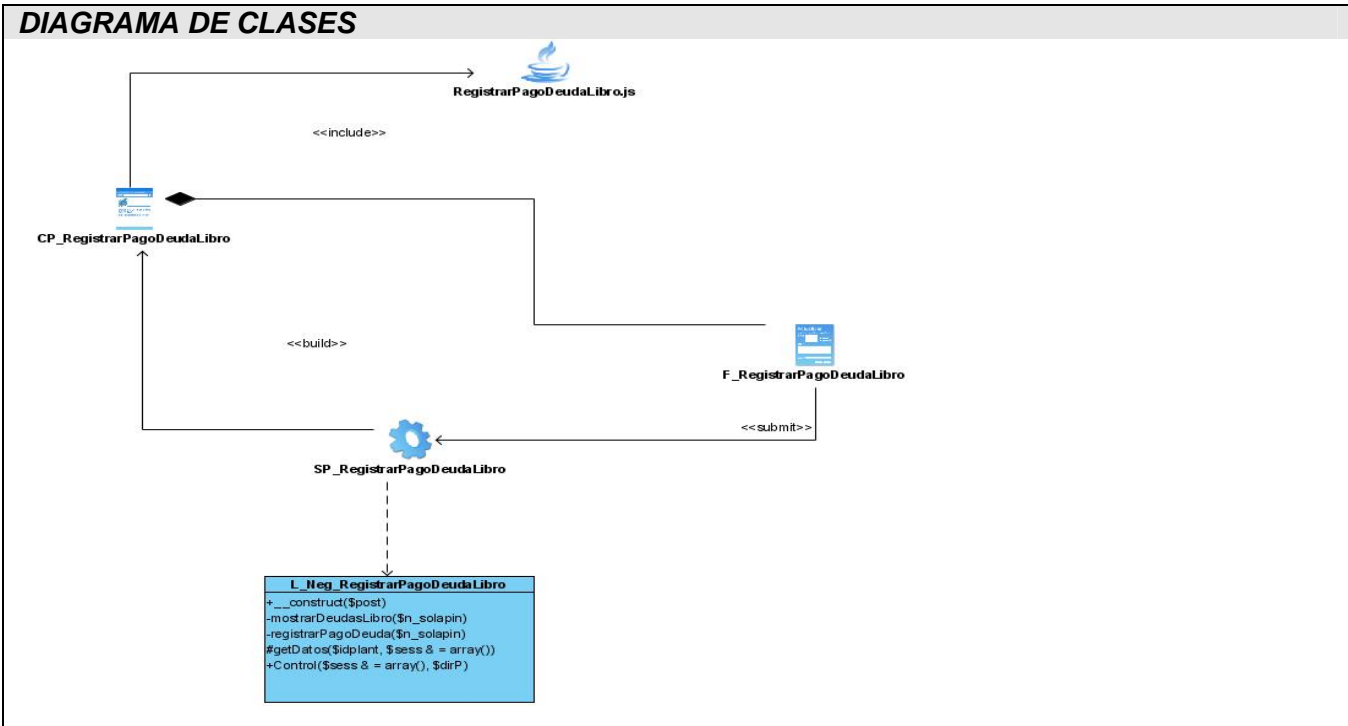


Figura 3.22 Diagrama de clases del diseño: Registrar Pago de Deuda de Libro.

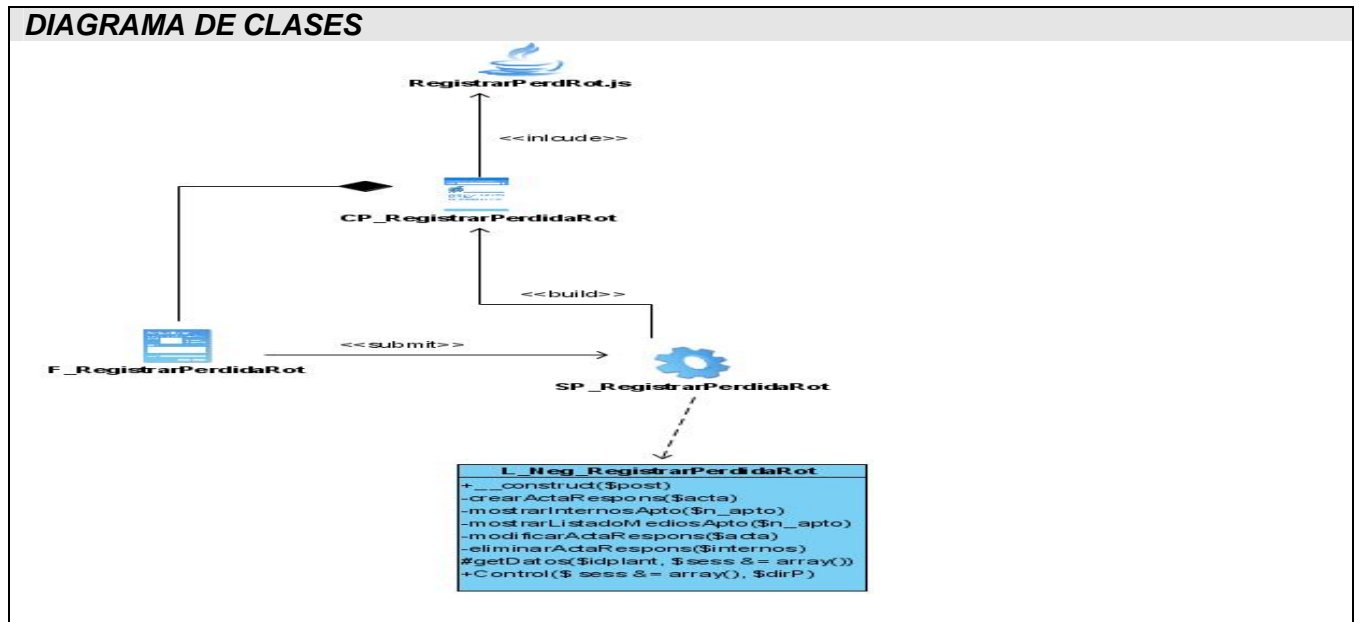
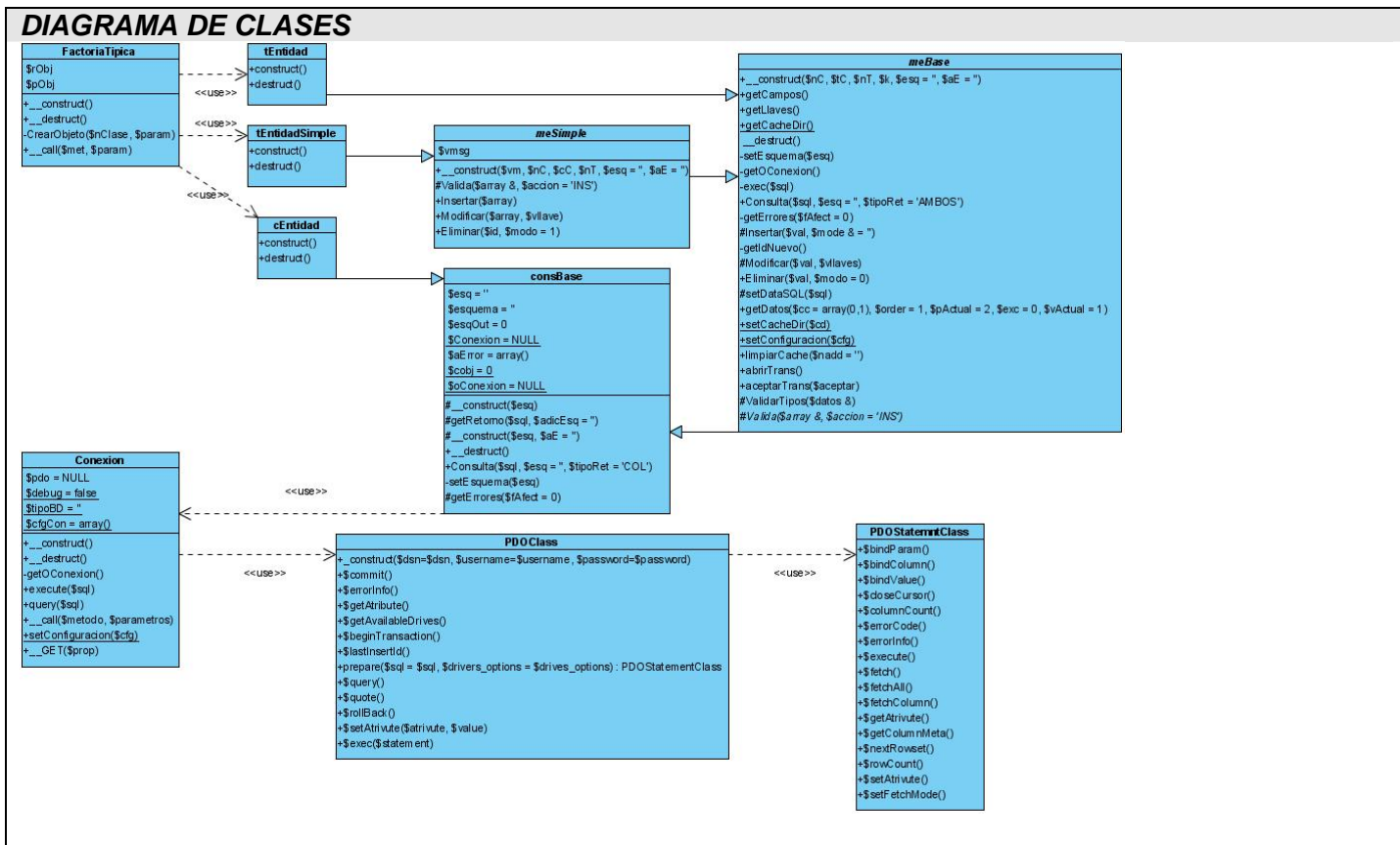


Figura 3.2 Diagrama de clases del diseño: Registrar Pérdida o Rotura de MB.

Repetir en cada modelo el empleo de clases y mensajes para el acceso a las entidades resulta una tarea bastante engorrosa además de tornar los diagramas menos comprensibles proporcionar mucho más trabajo y hacer más difícil la transición a la implementación. Para evitar esto y llegar a una solución correcta es necesaria la utilización de patrones, debido a que, además, es sabido que el acceso a la información almacenada en un modelo relacional desde un modelo orientado a objetos es uno de los principales problemas actuales de la construcción de software. Los patrones Factoría y Tabla Data Gateway (TDG) fueron utilizados, el patrón de creación Factoría para la clase Factoría Típica y el TDG para crear una clase para cada una de las entidades de la Base de Datos.

A continuación se muestra el diagrama de clases para el mecanismo de acceso a datos, donde a través clase Factoría Típica se crean y manipulan los objetos de las distintas entidades, tarjetas y artículos; esta clase es la clase que implementa la interfaz del modelo de persistencia con los demás subsistemas y constituye una puerta entre la capa de acceso a datos y la capa de lógica de negocio. La clase consBase es la base en toda la jerarquía de acceso a Datos, encapsula el objeto conexión y además implementa la operación de consulta. La clase Conexión es la encargada de establecer la conexión con el servidor de la Base de Datos a través de un objeto PDO de la librería de PHP. La clase meBase es una clase abstracta que hereda de consBase la operación de consulta e implementa además, las operaciones básicas a realizarse sobre el resto de las entidades (Insert, Delete y Update)

La clase meSimple es una clase abstracta que hereda de meBase y que a la vez es la base para la implementación de todas las típicas que respondan a nomencladores simples del modelo de persistencia dado. Redefine las operaciones básicas con la funcionalidad de Validación dada. Este conjunto de clases permiten el acceso y manipulación de los objetos de la persistencia utilizando los objetos brindados por el entorno PHP, PDO (PHP Data Object) y PDO_Statement.



3.3.3 Descripción de clases del diagrama de clases genérico.

Nombre: Global	
Tipo de clase: public	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	script_runtime (\$round)
Descripción:	Cálculo del tiempo de ejecución de un script
Nombre	__autoload(\$nomClase)
Descripción:	Función mágica autoload, garantiza la carga automática de las clases al instanciar

Nombre: Factoría Típica	
Tipo de clase: public	
Atributo	Tipo
\$rObj (Registro de objetos. Arreglo que contiene a los objetos que se hayan creado)	Objeto
\$pObj (Chequeo de objetos persistentes. Arreglo que contiene a los objetos persistentes que se hayan creado)	Objeto
Para cada responsabilidad:	
Nombre:	__construct()
Descripción:	Constructor de la clase
Nombre	__destruct()
Descripción:	Destructor de la clase
Nombre	CrearObjeto(\$nClase , \$param)
Descripción:	Se devuelve un objeto ya creado, en caso contrario se crea, se guarda en el registro de objetos y se devuelve. Es llamado únicamente desde el método __call.
Nombre	__call(\$met, \$param)
Descripción:	Función mágica que se ejecuta cuando se hace referencia a un método que no

	existe en la clase. Permite de forma transparente la interacción con el resto de las clases del modelo de persistencias. Implementa de forma explícita tres métodos base: Crear, pCrear, Destruir.
--	--

Nombre: InBase	
Tipo de clase: abstract	
Atributo	Tipo
\$dSalidaCU	String
\$oFact	Objeto
\$bEstado	string
\$post	array
\$msg	array
\$error	array
\$alCargar	string
\$iAccion	string
Para cada responsabilidad:	
Nombre:	__construct(\$iAcc, \$dSal, \$post , \$bEst)
Descripción:	Constructor de la clase.
Nombre	getErrores()
Descripción:	Permite dar salida a los mensajes de alerta o error registrados anteriormente en la clase.
Nombre	getDatosRet(& \$datos, \$idp)
Descripción:	Incorpora a \$datos chequeo errores
Nombre	vRetorno(& \$sess)
Descripción:	Verifica si se desea salir del CU
Nombre	getXML(& \$datos)
Descripción:	devuelve los datos en un XML
Nombre	movActualizacion(& \$sess, \$Elim)
Descripción:	Resuelve el tema del movimiento por un navegador.

Nombre: Miscelánea	
Tipo de clase: public	
Atributo: -	Tipo: -
Para cada responsabilidad:	
Nombre:	getOpciones(\$datos, \$valorDef = -1, \$ajax = 0)
Descripción:	Llenar las listas (select), devuelve las opciones de la misma en formato HTML.
Nombre	getAnno(\$v=0, \$ini=0, \$fin=0)
Descripción:	Formar listas de años. Devuelve las listas de los años correspondientes, en orden creciente o decreciente según se desee.
Nombre	getMeses(\$sel = -1, \$info = 0)
Descripción:	Obtención de la lista de meses.
Nombre	comparaFechas(\$fecha1, \$fecha2)
Descripción:	Comparar fechas.
Nombre	accesoOpcion(\$idopc)
Descripción:	Control de acceso a funcionalidades.
Nombre	getXMLNavegador (\$cam)
Descripción:	Obtención del xml del navegador.

Nombre: controlAcceso.inc	
Tipo de clase	
Atributo	Tipo
-	-
Para cada responsabilidad:	
Nombre:	AccFunc(\$func)
Descripción:	Verificar acceso a funcionalidades(VERIFICAR ESTO)
Nombre	Busca_Modulo()

Nombre: grid	
Tipo de clase: public	
Atributo	Tipo
\$datos	array
Para cada responsabilidad:	
Nombre:	__construct(\$datos, \$conf = "")
Descripción:	Constructor de la clase
Nombre	getXML(\$ysel = 0, \$norden = 1, \$tRetorno = 'nostring', \$nro = 1)
Descripción:	Método para retornar el XML para el llenado del grid.
Nombre	Generar(\$sel = 0, \$nFunc = 'getDatosGrid', \$hid = "")
Descripción:	Transforma el resultado de una consulta (array de datos) a una cadena JS asimilable por el grid que empleamos.

3.4 Arquitectura

El diseño del software contará con una arquitectura web en 3 capas con la cual se puede facilitar la integración de los sistemas, siguiendo el patrón arquitectónico Modelo-Vista-Controlador el cual permite la separación clara entre el modelo (lógica de negocio) y la vista (interfaz gráfica), gracias a un controlador que los mantiene desacoplados.

- **Capa de Cliente:** Conocida como capa de Presentación, es la que contiene las interfaces en las que el usuario interactúa con el sistema.
- **Capa de la Lógica de Negocio:** Es donde residen los programas que se ejecutan, recibiendo las peticiones del usuario y enviando las respuestas tras el proceso. 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 acceso a datos para el almacenamiento o recuperación de los mismos a través de ésta.
- **Capa de Almacenamiento:** mecanismo de almacenamiento persistente.



Figura 3.24 Representación de la Arquitectura

3.5 Diseño de la BD

3.5.1 Descripción de las tablas de la BD

Nombre: tNumGrupo		
Descripción: Almacena los datos de un grupo		
Atributo	Tipo	Descripción
idGrupo	Integer(10)	identificador
numGrupo	integer(10)	Número del grupo

Nombre: tNumFac		
Descripción: Almacena los datos de una facultad		
Atributo	Tipo	Descripción
idFac	Integer(10)	identificador
numFac	integer(10)	Número de la facultad

Nombre: tRol		
Descripción: Almacena los datos de un rol		
Atributo	Tipo	Descripción
idRol	Integer(10)	identificador
tipoRol	Varchar(10)	Tipo del rol

Nombre: tArea		
Descripción: Almacena los datos de un área		
Atributo	Tipo	Descripción
idArea	Integer(10)	identificador
tipoArea	Varchar(50)	Tipo del área

Nombre: Usuario		
Descripción: Almacena los datos de un usuario		
Atributo	Tipo	Descripción
nombreUsuario	Varchar(10)	Nombre del usuario
Contraseña	Varchar(30)	Contraseña del usuario
noSolapin	Varchar(10)	Número de solapín del usuario
noCl	Varchar(11)	Número de identidad del usuario
direccionPart	Varchar(50)	Dirección particular del usuario
pApellido	Varchar(10)	Primer apellido del usuario
sApellido	Varchar(10)	Segundo apellido del usuario
Nombre	Varchar(20)	Nombre

Nombre: Artículo_Gastable		
Descripción: Almacena los datos de un articulo gastable		
Atributo	Tipo	Descripción
precioArticuloGast	float(10)	Precio del artículo
nombreArticuloGast	Varchar(20)	Nombre del artículo
cantidadArticuloGast	Integer(10)	Cantidad del artículo

Nombre: tNumApto		
Descripción: Almacena los datos de un apartamento		
Atributo	Tipo	Descripción
idApto	Integer(10)	Identificador

numApto	Integer(10)	Número del apto
---------	-------------	-----------------

Nombre: tNumEdif		
Descripción: Almacena los datos de un edificio		
Atributo	Tipo	Descripción
idEdif	Integer(10)	Identificador
numEdif	Integer(10)	Número del edificio

Nombre: tNumResid		
Descripción: Almacena los datos de una residencia		
Atributo	Tipo	Descripción
idResid	Integer(10)	Identificador
numResid	Integer(10)	Número de la residencia

Nombre: tEstado		
Descripción: Almacena los datos de un estado		
Atributo	Tipo	Descripción
idEstado	Integer(10)	Identificador
tipoEstado	varchar(8)	Tipo del estado

Nombre: Artículo		
Descripción: Almacena los datos de un artículo		
Atributo	Tipo	Descripción
nombreArtículo	Varchar(20)	Nombre del artículo (identificador)
precioArtículo	Float(10)	Precio del artículo
cantidadArtículo	Integer(10)	Cantidad del artículo

Nombre: tAño		
Descripción: Almacena los datos de un año		

Atributo	Tipo	Descripción
Id	Integer(10)	Identificador
tipoAñoDocente	Integer(10)	Tipo del año docente

Nombre: tSemestre		
Descripción: Almacena los datos de un semestre		
Atributo	Tipo	Descripción
idSemestre	Integer(10)	Identificador
numSemestre	Integer(10)	Número del semestre

Nombre: tSemestreRecibido		
Descripción: Almacena los datos de un semestre donde se recibió un material o un libro, el semestre se considera como un número entero del 1 al 10.		
Atributo	Tipo	Descripción
idSemRec	Integer(10)	Identificador
Sem	Bit(1)	Indica si fue recibido en ese semestre

Nombre: Tarjeta de Avituallamiento		
Descripción: Tarjeta que almacena los Artículos de Avituallamiento entregados al interno		
Atributo	Tipo	Descripción
idTarjetaAv	Integer(10)	Número que identifica a la tarjeta, diferenciándola del resto

Nombre: Tarjeta		
Descripción: Tarjeta de Control de los Libros Docentes Entregados a los internos cada Semestre.		
Atributo	Tipo	Descripción
idTarjeta	integer(10)	Número que identifica a la tarjeta, diferenciándola del resto

Nombre: Tarjeta_Entrega_Aseo		
Descripción: Tarjeta de Control del Módulo de Aseo Entregado al interno mensualmente		
Atributo	Tipo	Descripción
idTarjetaA	integer(10)	Número que identifica a la tarjeta, diferenciándola del resto

Nombre: Inventario_Apto		
Descripción: Inventario de los Medios Básicos de un Apartamento.		
Atributo	Tipo	Descripción
idInvA	integer(10)	Número que identifica al inventario, diferenciándolo del resto.
Fecha	integer(10)	Fecha de realización de entrega de inventario.
Apto	integer(10)	Número del apartamento al que se le hace el inventario

Nombre: Tarjeta_Préstamo		
Descripción: Tarjeta de Control Para Préstamos de Libros en la Biblioteca		
Atributo	Tipo	Descripción
idTarjetaP	integer(10)	Número que identifica a la tarjeta de préstamos, diferenciándola del resto.
fechaInicio	integer(10)	Fecha de Inicio del Préstamo o extracción del Libro de la Biblioteca.
fechaDevolución	integer(10)	Fecha de Devolución o retorno del Libro a la Biblioteca.

Nombre: tTipoMedida		
Descripción: Tipo de Medida Disciplinaria Aplicada por la pérdida o ruptura de un medio por un interno.		
Atributo	Tipo	Descripción

Id	integer(10)	Número que identifica al Tipo de Medida, diferenciándolo del resto
nombre	varchar(30)	Nombre del Tipo de Medida Aplicada.

Nombre: Acta de Responsabilidad Material		
Descripción: Acta de registro de incidencias de pérdida o ruptura de medios del interno		
Atributo	Tipo	Descripción
Fecha	integer(10)	Fecha del Acta de Responsabilidad Material
idActa	integer(10)	Número que identifica al Acta de Responsabilidad Material, diferenciándola del resto.

Nombre: tTipo de Incidencia		
Descripción: Tipo de Incidencia de un interno en cuanto al cuidado de los Medios.		
Atributo	Tipo	Descripción
Id	integer(10)	Número que identifica al Tipo de Incidencia, diferenciándola del resto.
Tipo	varchar(20)	Tipo de incidencia cometida

3.6 Definiciones del diseño que se apliquen

Estándares en la interfaz de aplicación

La apariencia de la interfaz debe ser de fácil manejo, para lograr que el usuario cuando intercambie información con el sistema se sienta cómodo con la forma en que obtiene la misma.

Barra de Menú

- La barra de menú debe estar ubicada de manera horizontal.
- Los ítems de la barra deben agrupar un conjunto de operaciones, no una operación específica.

Diseño de la página principal.

La página principal debe llevar una imagen representativa del sitio, así como la barra de menú, vínculos a las demás páginas del sitio en la parte inferior izquierda de la barra de menú, en la parte inferior derecha de la barra de menú debe ir la información referente al usuario que está trabajando en el sitio así como la información del caso de uso en el que se encuentra trabajando el mismo y el tipo de acceso.

Hojas de Estilo

Se utilizarán color claro, verde claro y blanco para proporcionarle al usuario una interfaz fresca.

3.7 Tratamiento de errores

En el tratamiento de errores se utiliza la técnica de validación según la arquitectura utilizada. Existen dos tipos de validaciones: las validaciones en el cliente y las validaciones en el servidor.

Las **validaciones en el cliente** se realizan a través del código JavaScript, de manera que se eviten llamadas innecesarias a procedimientos en el servidor, pues al existir errores en el cliente se pueden corregir estos desde ahí mismo sin necesidad de que lleguen al servidor y que éste tenga que hacer las validaciones, por ejemplo campos sin marcar en un select, errores de tipos de datos o de nombres, en conclusión, las validaciones en el cliente se encargan de asegurar que se introduzcan los datos de manera correcta para luego hacer las llamadas al servidor.

Las **validaciones en el servidor** son importantes puesto a que en caso de que en el cliente se omitan validaciones para el caso de la introducción de información, en el servidor se efectúan, además de realizarse en el servidor también las validaciones lógicas, como la omisión de un parámetro en la llamada a un método o tratar de realizar asignaciones o obtenciones de información inexistente, en todos estos casos se muestra un mensaje al usuario con el error cometido de manera clara para que el mismo pueda entenderlo bien y así corregirlo.

3.8 Seguridad

En el sistema de control de seguridad intervienen las clases definidas por la arquitectura utilizada, están presentes las clases **controlacceso.inc** que contiene un método público `logueo` que permite saber si el usuario tiene acceso al sistema, en dependencia de lo que retorne el método se redirecciona el procesamiento a la página principal del sitio. A continuación se muestra el diagrama de control de seguridad.

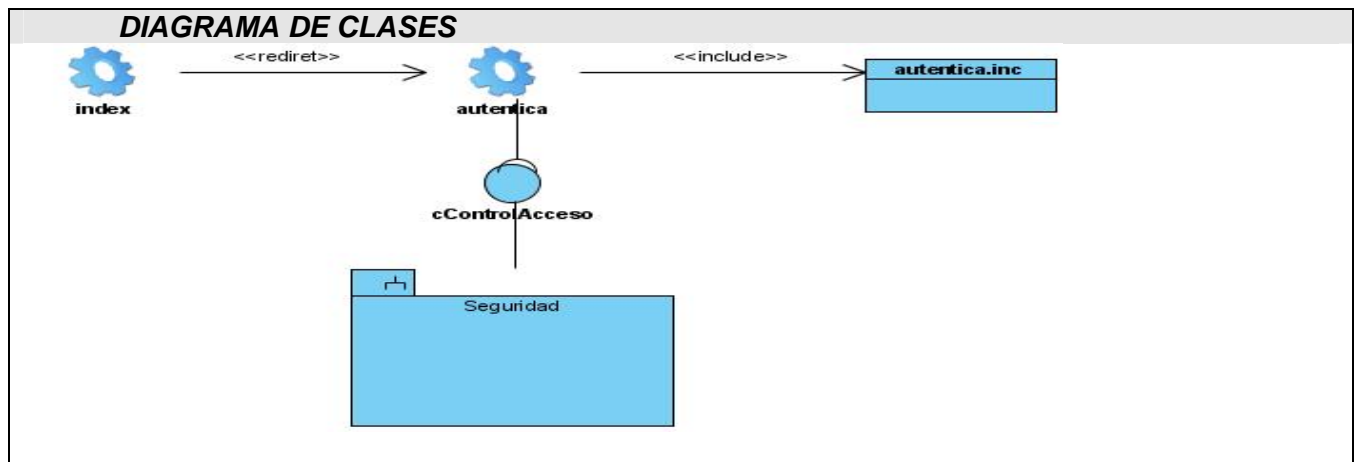


Figura 3.25 Diagrama genérico de seguridad.

3.9 Interfaz

La interfaz propuesta es una interfaz sencilla, con el uso de tonalidades claras como el verde claro y combinado con el blanco, de fácil manejo por el usuario y con barras de menú con opciones de fácil entendimiento para el mismo e interactiva, con el uso de Grids y Selects entre otros componentes para mostrar información al usuario, utilizando las tecnologías PHP y Ajax para el desarrollo de la misma.

3.10 Concepción de la ayuda

La ayuda es un tema de gran sensibilidad para los usuarios cuando se disponen a interactuar con el sistema, pues esta opción le permite conocer el funcionamiento del mismo y entenderlo mejor. El usuario final de la aplicación propuesta no tiene un alto dominio en la especialidad de uso de sistemas informáticos, por esta razón necesita estar informado todo el tiempo, de ahí la importancia de tener en cuenta este aspecto a la hora de concebir el software. Con el fin de resolver esta situación y proveer al usuario la información requerida el sistema contará con una ayuda en línea que ilustrará las funcionalidades del sistema de una forma breve y clara.

3.11 Diagrama de Despliegue

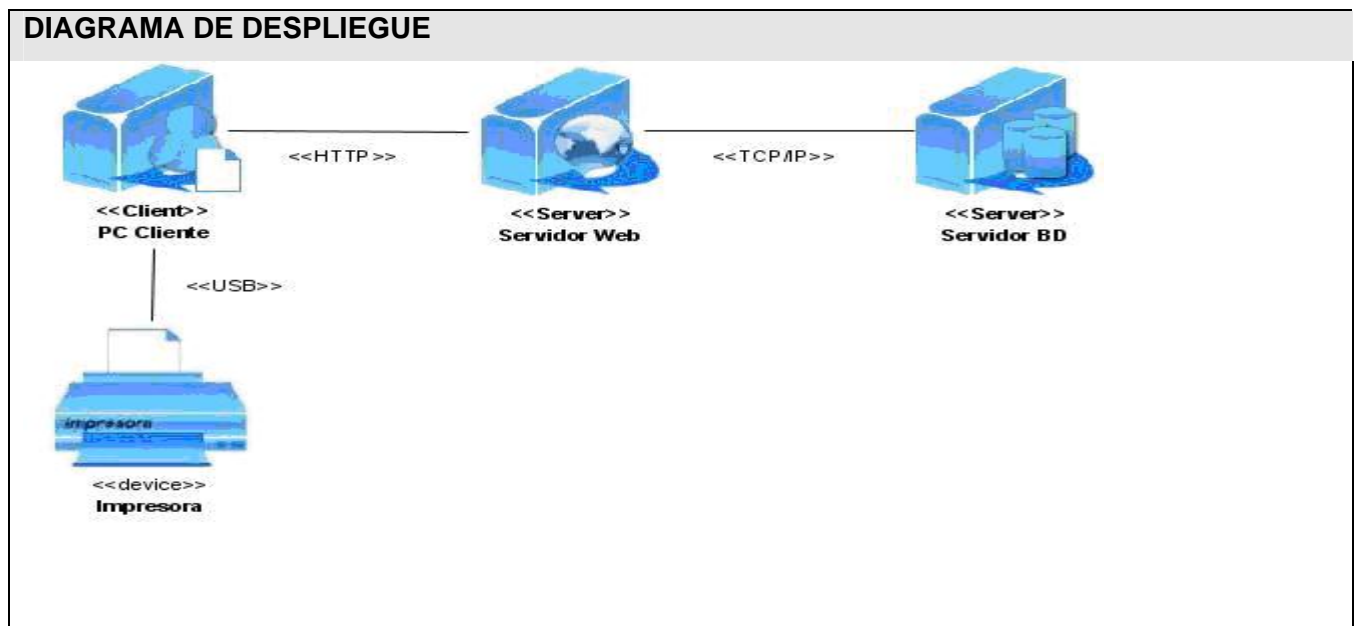


Figura 3.26 Diagrama de despliegue.

En el diagrama de despliegue se utilizan dispositivos como la impresora, con conexión USB a una PC Cliente, la que a su vez tiene una conexión mediante el protocolo HTTP al servidor web, donde corren los programas del lado del servidor, y éste a su vez, se conecta al servidor de base de datos mediante el protocolo TCP/IP.

3.12 Conclusiones.

En este capítulo se trataron los aspectos más significativos del análisis y diseño del sistema y los principales artefactos generados en este flujo de trabajo, entre los que se encuentran el modelo de análisis, los diagramas de clases del diseño de los casos de uso críticos o primarios, el diseño de la base de datos, el diagrama de despliegue así como otros temas de interés como definiciones del diseño aplicadas, tratamiento de errores, seguridad, interfaz y concepción de la ayuda los cuales tendrán una gran importancia y serán de gran ayuda para una futura implementación del sistema.

CONCLUSIONES

Con el desarrollo de este trabajo de diploma le ha dado cumplimiento al objetivo que se perseguía en primer lugar: obtener el diseño de un sistema que pusiera fin a los problemas de papeleo y desorganización de todos los procesos de gestión y control de medios que se entregan a estudiantes y profesores internos de la Universidad de las Ciencias Informáticas. Para lograrlo, se realizó un estudio de cómo se realizan los procesos de entrega de materiales docentes, módulos de aseo, artículos de avituallamiento, libros de texto, así como la forma en que se realizan los préstamos externos en la biblioteca de la Universidad, entre otros procesos. Esta información fue de gran importancia a la hora de proponer un diseño que tuviera en cuenta la problemática actual, y que permita que el sistema satisfaga las necesidades reales de la Universidad. Esta propuesta, garantizará un mejor flujo de información y sobre todo, ayudará a agilizar el proceso de la tramitación de bajas, que resulta tan engorroso en la actualidad.

RECOMENDACIONES

Una vez concluido este trabajo de diploma habiéndosele dado cumplimiento a los objetivos propuestos se recomienda:

- Continuar con la implementación del sistema para su posterior implantación en la universidad.
- Crear tarjetas automáticamente que se registre un interno se creen las diferentes tarjetas de control de los diferentes módulos con el fin de poder tener registrado el historial de entregas y devoluciones de artículos y medios para así poder determinar comportamientos futuros.
- Se realice la entrega del módulo de aseo individual por los instructores de los edificios para el caso de los profesores internos.
- Utilizar el servicio que brinda la Universidad para el caso de la autenticación de los estudiantes y profesores internos.

BIBLIOGRAFÍA

- PRESSMAN, R. *Ingeniería de Software. Un enfoque práctico*. La Habana. Félix Varela.2004.
- ROMÁN, A y MORALES, Y. *Propuesta de una guía para estandarizar la codificación en la Universidad de las Ciencias Informáticas*. Universidad de las Ciencias Informáticas.2007.
- SCHUMULLER, J. *Aprendiendo UML en 24 horas*.2000. [Disponible en: <http://bibliodoc.uci.cu/pdf/reg00004.pdf>]
- RUMBAUGH, JACOBSON, BOOCH. *Lenguaje Unificado de Modelado. Manual de Referencia*. 1998. [Disponible en: <http://bibliodoc.uci.cu/pdf/reg03050.pdf>]
- RUMBAUGH, JACOBSON, BOOCH. *El Proceso Unificado de Desarrollo de Software*.2000. [Disponible en: <http://bibliodoc.uci.cu/pdf/reg00060.pdf>]
- PRESSMAN, R. *Ingeniería del Software. Un enfoque práctico*. 2001. [Disponible en: <http://bibliodoc.uci.cu/pdf/reg02689.pdf>]
- LARMAN, C. *UML y Patrones*. 1999. [Disponible en: <http://bibliodoc.uci.cu/pdf/reg00061.pdf>]
- FERNANDEZ, E y LOPEZ, D. *Desarrollo de la capa de acceso a datos para los módulos de administración y nomencladores del sistema de gestión de inventario de almacenes SIGIA*. 2007. [Disponible en: <http://biblioteca.uci.cu/sbd/biuci/index.html>]
- PEREZ, C. *Desarrollo de la capa de presentación para los módulos de administración y nomencladores del sistema de gestión de inventario de almacenes SIGIA*. 2007. [Disponible en: <http://biblioteca.uci.cu/sbd/biuci/index.html>]
- LARRAMENDY, S y BERMUDEZ, Y. *Módulo para la gestión de medios básicos del proyecto Intranet del Centro Rector Universidad para Todos*. 2007. [Disponible en: <http://biblioteca.uci.cu/sbd/biuci/index.html>]

-
- PERERA, J. *Arquitectura de software para sistema gestión de inventarios*. 2007. [Disponible en: <http://biblioteca.uci.cu/sbd/biuci/index.html>]
 - MEDERO, J y MORALES, J. *Desarrollo de la capa de negocio para los módulos de administración y nomencladores del sistema gestión de inventario de almacenes SIGIA*. 2007. [Disponible en: <http://biblioteca.uci.cu/sbd/biuci/index.html>]
 - TEJERA, D y SANCHEZ, L. *Ingeniería de requisitos para el desarrollo del Sistema de Gestión de Inventario Almacén (SIGIA)*. 2007. [Disponible en: http://bibliodoc.uci.cu/TD/TD_0251_07.pdf]
 - DOMENECH, G y CORDERO, M. *Análisis del módulo Inventario/ Almacén del Sistema de Gestión de Inventarios y Almacenes SIGIA*. 2007. [Disponible en: <http://biblioteca.uci.cu/sbd/biuci/index.html>]
 - FIGUEROA, Y. *Diseño de la capa de datos del Sistema Gestión de Inventarios y Almacenes*. 2007. [Disponible en: <http://biblioteca.uci.cu/sbd/biuci/index.html>]
 - QUESADA, Y. *Sistema de contabilidad material para la actividad presupuestada en las fuerzas armadas revolucionarias. Módulo de Ajuste de Inventario de Medios Materiales*. 2007. [Disponible en: <http://biblioteca.uci.cu/sbd/biuci/index.html>]
 - GONZALEZ, H y LEZCANO, Y. *Sistema de Inventario Participativo de la UCI*. 2005. [Disponible en: <http://biblioteca.uci.cu/sbd/biuci/index.html>]
 - BRANDENBAUGH, J. *Aplicaciones JavaScript* 2000. [Disponible en: <http://bibliodoc.uci.cu/pdf/reg00009.pdf>]
 - STURM, J. *Desarrollo de soluciones XML*. 2000. [Disponible en: <http://bibliodoc.uci.cu/pdf/reg01281.pdf>]

GLOSARIO DE TÉRMINOS

- TCERA, Tarjeta_Avituallamiento: Tarjeta de Control de Entrega y Recepción de Avituallamiento.
- Artículo_Gastable: se refiere a aquellos artículos que no deben ser devueltos después de su utilización.
- Inv_Apto: se refiere al inventario de un apartamento, que no es más que el Acta de Entrega de Recursos y Activos de los Dormitorios de la Residencia Universitaria.
- Acta_Respons_Material: se refiere al acta que se elabora cuando se registra la pérdida o rotura de un medio básico de un apartamento, que no es más que el Acta de Análisis por Pérdidas o Roturas de Medios.
- Tarjeta: se refiere a la tarjeta de control de entrega de libros.
- Tarjeta_Materiales: se refiere a la tarjeta donde se registra la entrega de materiales docentes a un interno.
- Boleta_Baja: se refiere al Modelo de Tramitación de Bajas.
- Tarjeta_Entrega_Aseo: se refiere a la tarjeta de control de entrega del módulo de aseo individual.
- Tarjeta_Prestamo: se refiere a la tarjeta donde se registran los préstamos externos de libros de la biblioteca a un profesor o estudiante interno.