



Universidad de Cienfuegos “Carlos Rafael Rodríguez”
Facultad de Informática
Carrera de Ingeniería Informática

SIREU
Sistema Informático para la
Residencia Estudiantil Universitaria

Trabajo de diploma para optar por el título de Ingeniería en Informática

Autor:
Yoisel Martínez Gómez

Tutores:
Msc. Dailyn Sosa López.
Msc. Kadir Héctor Ortiz.

Cienfuegos, Cuba
Curso 2008 - 2009

Declaración de autoría

Yo, Yoisel Martínez Gómez declaro que soy el único autor del trabajo de diploma titulado “**Sistema Informático para la Residencia Estudiantil Universitaria**”, y autorizo al departamento de Informática de la facultad de Informática de la Universidad de Cienfuegos “Carlos Rafael Rodríguez”, para que haga el uso que estime pertinente con el mismo.

Para que así conste firmo la presente a los ____ días del mes de _____ del 2009.

Firma del autor

Yoisel Martínez Gómez

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

Firma del tutor

Msc. Dailyn Sosa López.

Firma del tutor

Msc. Kadir Héctor Ortiz.

Firma ICT

Agradecimientos

***A mi familia y mis amigos que sin los cuales nada de esto
hubiese sido posible a todos ustedes de todo corazón...***

-- muchas gracias--

Dedicatoria

A mi familia y amigos

Resumen

La revolución cubana como ejemplo de proliferación de libertad y cultura, ha venido desarrollando un amplio programa en materia de educación donde las universidades han jugado un rol decisivo en el desarrollo de todo nuestro pueblo, así como de los países amigos.

Es por ello que surgió la necesidad de diseñar e implementar un sistema para la Residencia Estudiantil de la Universidad de Cienfuegos "Carlos Rafael Rodríguez" donde se creó una herramienta que facilita la gestión de la información dentro de esta. En el momento de concebir este producto se analizaron sistemas desarrollados con anterioridad como son: **SACBEXT**, **BDEDUCACION** y **SIBEX**. En vista de los requerimientos hemos desarrollado un sistema basado en una tecnología cliente/servidor capaz de brindar soluciones que contribuyan a un mejor manejo de la información dentro de la residencia estudiantil.

El flujo de la información y el trabajo de la residencia estudiantil fueron las bases para el desarrollo del sistema que se propone.

El cual fue concebido bajo la utilización de la metodología RUP por su amplia difusión y facilidad de uso. Como herramienta de desarrollo se utilizó el Dreamweaver 8, el cual incluye facilidades de desarrollo para aplicaciones Web, incluyendo generación en código PHP, Java Script, HTML; lenguajes empleados en el desarrollo del software. Se utilizó como Sistema de Gestión de Base de Datos: PostgreSQL. Y como servidor Web Apache.

Índice

Introducción	2
Capítulo 1: Fundamentación teórica.	6
1.2 Conceptos asociados al dominio del problema.	6
1.2.1 Gestión.....	6
1.2.2 Gestión de la información.	6
1.2.3 Dispensarización.....	6
1.3 Descripción del objeto de estudio.....	6
1.3.1 - Objetivos estratégicos de la organización.....	8
1.4 Descripción de los sistemas existentes.	9
1.5 Metodologías.....	10
Lenguaje de Modelación Unificado (UML)	10
Proceso Unificado de Desarrollo (RUP).....	11
1.6 Tendencias actuales.....	13
1.6.1- Arquitectura de desarrollo de N Capas.	13
1.7 Tecnologías actuales.....	15
1.7.1-Tecnologías Web.....	15
1.7.2- Servicios Web.....	18
1.7.3- Lenguajes de programación Web	21
1.8 - Servidores de Aplicaciones Web.....	23
1.8.1- Apache.....	23
1.8.2- IIS	¡Error! Marcador no definido.
1.9- Sistemas Gestores de Base de Datos.....	23
1.10 Conclusiones	24
Capítulo 2: Modelo del negocio	25
2.1 Descripción del modelo de negocio.....	25
2.2 Reglas del negocio a considerar.	26
2.3 Modelo de casos de uso del negocio.	27
2.3.1 Actores del negocio.....	27
2.3.2- Diagramas de casos de uso del negocio.	28
2.3.3 – Trabajadores del negocio.....	29
2.3.4 – Descripción de los casos de uso del negocio.	29
2.3.5 – Diagramas de actividades del negocio.....	34
2.4 Modelo de objetos del negocio.....	38
2.5 Conclusiones	39
Capítulo 3 – Construcción de la Solución Propuesta.....	40
3.2 Descripción del sistema propuesto.....	40
3.2.1 – Requerimientos funcionales.....	40
3.2.2 – Requerimientos no funcionales.....	42
3.3 Modelo de casos de uso del sistema.....	45
3.3.1 – Actores del sistema.....	45
3.3.2 – Paquetes y sus relaciones	46
3.3.3 – Diagramas de casos de uso del sistema.....	47
3.3.4 – Descripción de los casos de uso del sistema.....	49
3.4 – Diseño de la base de datos.....	59

Índice

3.4.1 – Modelo lógico de datos.....	59
3.4.2 – Modelo físico de datos.....	60
3.5 – Diagrama de implementación.....	62
3.6 – Principios de diseño.....	62
3.6.1 – Estándares en la interfaz de la aplicación.....	62
3.6.2 – Tratamiento de errores.....	63
3.6.3 – Concepción General de la ayuda.....	63
3.7- Diagramas de Clases Web.....	63
3.7 – Conclusiones.....	64
Capítulo 4 Estudio de Factibilidad.....	65
4.1 – Planificación por puntos de función.....	65
4.3 – Determinación de los costos.....	69
4.4 – Beneficios tangibles e intangibles.....	73
4.5 – Análisis de costos y beneficios.....	73
4.6 – Conclusiones.....	73
Conclusiones:.....	74
Recomendaciones.....	75
Referencias Bibliográficas.....	76
Bibliografía.....	78
Glosario de términos.....	79
Anexos.....	80

Índice de tablas

Tabla 1: Comparación entre SGBD.	24
Tabla 2: Actores del Negocio.....	28
Tabla 3: Descripción de los Trabajadores del Negocio.....	29
Tabla 4: Descripción del caso de uso del negocio Solicitar Beca.....	30
Tabla 5: Descripción del caso de uso del negocio Solicitar cambio de cuarto.....	31
Tabla 6: Descripción del caso de uso del negocio Solicitar baja.	32
Tabla 7: Descripción del caso de uso del negocio Solicitar información.....	33
Tabla 8: Actores del sistema.	46
Tabla 9: Descripción Caso de Uso del Sistema Iniciar Sesión	51
Tabla 10: Descripción Caso de Uso del Sistema Cambiar Contraseña.....	52
Tabla 11: Descripción Caso de Uso del Sistema Gestionar Usuario.....	52
Tabla 12: Descripción Caso de Uso del Sistema Visualizar los datos de los estudiantes.	53
Tabla 13: Descripción Caso de Uso del Sistema Gestionar cuarto.	54
Tabla 14: Descripción Caso de Uso del Sistema Gestionar colchón.....	54
Tabla 15: Descripción Caso de Uso del Sistema Insertar información adicional del estudiante.....	55
Tabla 16: Descripción Caso de Uso del Sistema Visualizar información adicional del estudiante.....	56
Tabla 17: Descripción Caso de Uso del Sistema Visualizar mobiliario.....	56
Tabla 18: Descripción Caso de Uso del Sistema Gestionar mobiliario	57
Tabla 19: Descripción Caso de Uso del Sistema Visualizar trabajadores.	57
Tabla 20: Descripción Caso de Uso del Sistema Gestionar trabajador.	58
Tabla 21: Descripción Caso de Uso del Sistema Gestionar edificios.	58
Tabla 22: Diagramas de clases Web del Sistema	64
Tabla 23: Planificación. Entradas externas.....	66
Tabla 24: Planificación: Peticiones	67
Tabla 25: Planificación: Ficheros internos	68

Índice de Tablas

Tabla 26: Planificación: Punto de función.....	68
Tabla 27: Planificación: Miles de instrucciones fuentes.....	69
Tabla 28: Multiplicadores de esfuerzos.	70
Tabla 29: Costos totales.....	72

Índice de figuras

Fig. 1: Estructura organizativa de la Residencia Estudiantil de la UCF.	7
Fig. 2: Fases de RUP	13
Fig. 3 Arquitectura de tres capas.	14
Fig. 4: Diagramas de Casos de Uso del Negocio	28
Fig. 5: Diagrama de Actividad. Caso de Uso Solicitar Beca	34
Fig. 6 Diagrama de Actividad. Caso de Uso Solicitar Cambio de Cuarto.	35
Fig. 7: Diagrama de Actividad. Caso de Uso Solicitar Baja	36
Fig. 8: Diagrama de Actividad. Caso de Uso Solicitar información.....	37
Fig. 9 Diagramas de clases del modelo de objetos del negocio.	38
Fig. 10: Relación entre actores del sistema.....	46
Fig. 11: Diagrama de Casos de Usos por <i>Paquetes</i>	47
Fig. 12: Diagrama de casos de uso del sistema <i>Gestión de Información</i>	48
Fig. 13: Diagrama de casos de uso del sistema <i>Salida de Información</i>	49
Fig. 14: Diagrama de casos de uso del sistema <i>Paquete de Seguridad y Utilidades</i>	49
Fig. 15 Modelo Lógico de Datos	59
Fig. 16: Modelo físico de datos.....	61
Fig. 17: Diagrama de Implementación	62

Introducción

Actualmente las Tecnologías de Información y Comunicación (TIC), son consideradas como el conjunto de dispositivos y proceso lógicos y prácticos, basado en soportes físicos; que permiten agregar valores a los datos, difundirlos y transfórmalos en productos y servicios que garantizan el acceso a la información, al aprendizaje y a la generación de conocimientos.

Como parte de la política educacional que se ha desarrollado en nuestro país desde años anteriores se va evidenciando que el uso de las Tecnologías de la Informática y las Comunicaciones (TIC) tienen que ganar protagonismo con el objetivo de lograr que la información sea más viable y segura.

La Universidad de Cienfuegos (UCF), Carlos Rafael Rodríguez, no esta exenta de esta política nacional y en ella las TIC van ganado espacios en la automatización de sus procesos.

Este trabajo enmarca como **objeto de estudio** la Residencia Estudiantil de la Universidad de Cienfuegos, específicamente el flujo de la información dentro de dicha entidad siendo este el **campo de acción**.

La residencia de la UCF, es la encargada de llevar un control de toda la información referente a los estudiantes becados que se encuentran cursando sus estudios en nuestro centro (situación de salud, comportamiento en la residencia estudiantil, etc.), sus trabajadores, así como la disponibilidad de mobiliario con que cuenta la dicha entidad.

La utilización de las TIC en las acciones de la UCF y en particular en la residencia estudiantil sigue relativamente bajo, provocando el inadecuado aprovechamiento de las posibilidades que brindan dichas tecnologías. Muchos de los procesos que se llevan a cabo en la residencia, se manejan con el uso de los procesadores de textos, u hojas de cálculo; herramientas que impiden la obtención de interrelaciones entre la información allí manipulada y otras posibilidades que proporcionan los sistemas de gestión automatizada.

En la investigación realizada en la universidad se determinó que existe un sistema titulado SIBEX (Sistema de Gestión de la Información de Estudiantes Becarios

Introducción

Extranjeros) realizado en el año 2006 por la estudiante Ursula Vedora Willock, pero que dicho sistema solo recoge la información perteneciente a los estudiantes becado extranjeros dejando fuera del sistema a los estudiantes nacionales, trabajadores y mobiliario, que constituyen elementos claves en los procesos de la residencia estudiantil, lo cual limita su desempeño.

Toda la gestión y manipulación de la información relacionada con todos los datos de los estudiantes, ubicación del mobiliario dentro de la residencia, así como los datos personales de los trabajadores que en ella laboran, carece de inmediatez y seguridad, ya que se encuentra guardada en archivos que limitan su disponibilidad y exige además reservar espacios para su almacenamiento y consulta. Esto retarda el proceso de recopilación de información requerido para dar respuesta a todos los organismos y entidades organizativas que estén interesadas en dichos datos, como el Ministerio de Educación Superior (MES), las distintas facultades de la UCF, embajadas, entre otros interesados.

Toda esta situación permite identificar como **problema de investigación** la carencia en la Residencia Estudiantil de la UCF de una herramienta que facilite a los trabajadores el manejo de forma rápida y segura la información relacionada con los datos de los estudiantes, ubicación del mobiliario dentro de la residencia, y los datos personales de los trabajadores que en ella laboran.

El flujo preciso y confiable de la información en cualquier entidad es de vital importancia para un buen desempeño. Dada la necesidad de diseñar un sistema capaz de lograr que la información sea dinámica, confiable e interrelacionada, se define como **idea a defender** que la elaboración de la aplicación Web para la gestión de la información dentro la residencia estudiantil, garantizará la disponibilidad, seguridad y rapidez de la información para las entidades interesadas.

En correspondencia con lo anterior, se define como **objetivo general**: la elaboración de una aplicación Web que permita la gestión de la información relacionada con los procesos que tienen lugar dentro de la Residencia Estudiantil de la Universidad de Cienfuegos.

Del cual se desprenden los siguientes **objetivos específicos**:

- ✓ Analizar la gestión de la información de la Residencia Estudiantil de Cienfuegos referente a estudiantes becados, trabajadores y mobiliarios.
- ✓ Diseñar un sistema capaz de automatizar el proceso de gestión de la información relacionada con la residencia estudiantil.
- ✓ Implementar un sistema que cumpla con las características de lo analizado y diseñado.

Se precisa la realización de las siguientes **tareas** para cumplir con los objetivos propuestos:

- ✓ Entrevistas a todas las personas que están ligadas al proceso de gestión de la información en la residencia estudiantil.
- ✓ Análisis de los reportes que confeccionan los trabajadores de dicha entidad.
- ✓ Definición de todos los procesos que puedan ser automatizados.
- ✓ Realización de un diseño adecuado de la base de datos.
- ✓ Análisis de las tendencias y tecnologías a desarrollar para tener un mejor resultado.
- ✓ Confección de un único sistema con el objetivo de que sea capaz de gestionar toda la información necesaria referente a las necesidades de la residencia estudiantil.
- ✓ Establecimiento de niveles de acceso a la información por módulos.
- ✓ Confección de una ayuda del sistema para mejorar su manipulación por parte de sus usuarios.

Esta investigación proporciona a la Residencia Estudiantil una solución a la gestión de información requerida en sus procesos, pudiéndose mencionar como los principales **aportes prácticos** que:

- ✓ Provee a los trabajadores de la residencia de una herramienta que les permite el acceso rápido y fácil a la información disponible sobre los estudiantes, trabajadores y mobiliario.
- ✓ Brinda facilidades operativas a los trabajadores encargados de gestionar esta información.

Introducción

- ✓ Favorece significativamente en la disponibilidad con que pueden brindar la información a todo aquel que la solicite.

El presente Trabajo de Diploma se encuentra estructurado básicamente en resumen, Introducción, tres capítulos, conclusiones, recomendaciones y anexos. De los capítulos se muestra a continuación su contenido de forma resumida:

Capítulo 1: Fundamentación Teórica.

Se exponen los principales conceptos vinculados al campo de acción, un análisis de sistemas existentes y aparece el estudio realizado sobre las tendencias, metodologías y tecnologías actuales en las cuales se fundamenta la propuesta del objeto de automatización.

Capítulo 2. Modelo del Negocio.

Se describe el modelo del negocio, identificando los procesos involucrados en él y las reglas que lo caracterizan. Se realiza la descripción del modelo de casos de uso, identificando y describiendo los actores, trabajadores y casos de uso del negocio mediante el diagrama de casos de uso y el diagrama de actividades.

Capítulo 3: Construcción de la Solución Propuesta.

Se describe de modo general el funcionamiento del sistema. Se enuncian los requerimientos funcionales y no funcionales, definiéndose los actores y casos de uso, se realiza la descripción del modelo de casos de uso del sistema,

Finaliza el capítulo con la descripción del modelo de diseño e implementación del sistema, utilizándose los artefactos diagrama de clases web, los diagramas de modelo lógico y físico de datos y el diagrama de implementación.

Capítulo 4. Estudio de Factibilidad y Análisis de Resultados.

En este capítulo se muestran los resultados del análisis por Puntos de Función, donde se valora la factibilidad del sistema. Terminado el capítulo se analizan los resultados alcanzados con el sistema.

Capítulo 1: Fundamentación teórica.

En este capítulo se abordaran aspectos teóricos del tema que se va a analizar, exponiendo los principales conceptos asociados al dominio del mismo, así como las metodologías y tecnologías utilizadas.

1.1- Conceptos asociados al dominio del problema.

1.1.1 Gestión.

Gestionar es coordinar todos los recursos disponibles para conseguir determinados objetivos, implica amplias y fuertes interacciones fundamentalmente entre el entorno, las estructuras, el proceso y los productos que se deseen obtener.[1]

1.1.2 Gestión de la información.

La gestión de la información se puede definir como el conjunto de actividades realizadas con el fin de controlar, almacenar y, posteriormente, recuperar adecuadamente la información producida, recibida o retenida por cualquier organización en el desarrollo de sus actividades. [2]

1.1.3 Dispensarización.

Es el proceso organizado, continuo y dinámico de evaluación e intervención planificada e integral, con un enfoque clínico, epidemiológico y social, del estado de salud de los individuos y familias. Es un proceso coordinado y liderado por el Equipo Básico de Salud (EBS).[3]

1.3- Descripción del objeto de estudio.

Residencia Estudiantil de la Universidad de Cienfuegos.

La Residencia Estudiantil de la Universidad de Cienfuegos esta encaminada a reforzar la convivencia, educación formal, cuidado de la propiedad colectiva, hábitos de orden, limpieza, disciplina y protagonismo estudiantil en la solución de problemas de carácter

CAPÍTULO 1: Fundamentación Teórica

colectivo que contribuyan favorablemente a la formación integral de los becarios, así como elevar cualitativamente el trabajo educativo y político-ideológico de los becarios. Todo esto protagonizado por un profundo trabajo en unión con la FEU y la UJC para convertir la Residencia Estudiantil en una Comunidad Universitaria.

La Residencia Estudiantil de la UCF cuenta con la estructura organizativa que se muestra en la figura 1

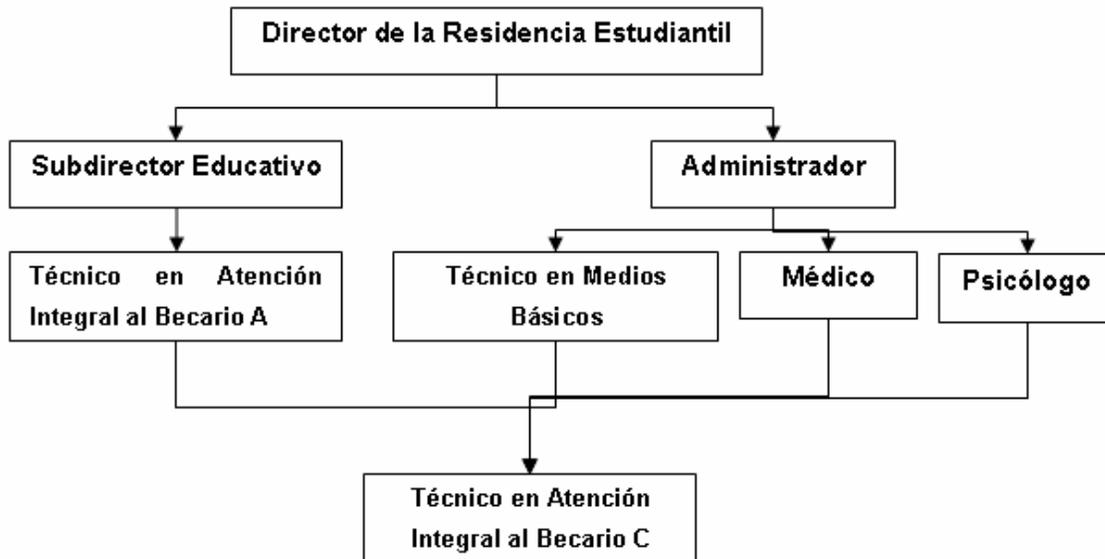


Fig. 1: Estructura organizativa de la Residencia Estudiantil de la UCF.

CAPÍTULO 1: Fundamentación Teórica

1.3.1 - Objetivos estratégicos de la organización.

La residencia estudiantil tiene como principal proceso la: ***gestión de la información de los estudiantes becados, de sus trabajadores así como de la ubicación y uso de su mobiliario.***

Cuando un estudiante ingresa a la universidad en alguna de las carreras que en ellas se estudia. Se dirige a la Residencia Estudiantil. La Técnico en Atención Integral al Becario con categoría A verifica sus datos generales y posteriormente se le da la ubicación en un cuarto de los edificios existentes. El cual es entregado por la Técnico en Atención Integral al Becario con categoría C.

El estudiante esta sujeto a un proceso de control en cual es monitoreado por dichos técnicos. Estos procesos son:

- ✓ Registrar Dispensarización: Es llevada a cabo cada curso por el Médico del Puesto Médico de la Universidad y debe ser registrada, actualizada y consultada por el Técnico A encargado de dicha labor en la residencia.
- ✓ Registrar Diagnóstico Psicosociopedagógico del estudiante: Durante la estancia del estudiante en la residencia estudiantil, el técnico A actualizará el diagnóstico psicosociopedagógico de cada estudiante. Este diagnóstico se basa fundamentalmente en el criterio del psicólogo y dicho técnico acerca de ciertos aspectos que permiten tener un mayor control del comportamiento del mismo y personalizar el acercamiento a cada uno de estos estudiantes.
- ✓ Registrar faltas cometidas por los becarios: Ante alguna falta de carácter leve, grave, menos grave o muy grave, la Dirección de la Residencia Estudiantil, conjuntamente con los Técnico en Atención Integral al Becario con categoría A y C respectivamente, levantarán un proceso investigativo con el objetivo de efectuar un señalamiento o aplicar una sanción según el Reglamento Disciplinario para las Residencias Estudiantiles, amparado por las resoluciones No. 34/2000 y 38/2000 respectivamente.
- ✓ Registrar Evaluación en Beca del estudiante: Se realiza al finalizar cada curso académico por el jefe de cuarto, conjuntamente con el jefe de piso y el técnico C del edificio. Dicha evaluación es registrada en la Oficina de Trabajo Educativo de

CAPÍTULO 1: Fundamentación Teórica

la Residencia Estudiantil y es solicitada por el técnico A para ser registrada, puesto que reviste gran importancia, ya que la misma tributa al proceso de integralidad del estudiante.

Todos estos procesos descritos son registrados, consultados y manejados por el Técnico en Atención Integral al Becario con categoría A en la elaboración de reportes que son solicitados por las instancias superiores. (Rectorado, MES, Embajadas, entre otras).

1.4 Descripción de los sistemas existentes.

A continuación se presentan los sistemas informáticos que han sido desarrollados hasta la actualidad y que están relacionados con el campo de acción:

SACBEXT: Sistema Automatizado de control de Becarios Extranjeros.

Fue desarrollado durante los años 1998-2000, para la manipulación de información de Estudiantes Extranjeros en la ONABE. El sistema gestiona en un alcance general la información de los estudiantes a nivel nacional, permitiendo así la gestión de esta información. SACBEXT logró mejorar el proceso de control y la rapidez en la toma de decisiones por la ONABE.

El sistema se destaca en las posibilidades brindadas para la obtención de resúmenes y reportes de la información almacenada y el envío por correo de los reportes generados por el sistema. Las exigencias presentes en ese período requirieron una gestión generalizada y resumida de la información pues SACBEXT no profundizó en los parámetros que podrían diferenciar y a la vez identificar con más precisión a los estudiantes extranjeros. La accesibilidad al sistema es limitada, ya que los Centros de Estudio donde radican los estudiantes y donde son mejor conocidos no tienen acceso directo a esta herramienta por cuestiones de seguridad que no fueron implementadas, aunque en general se puede afirmar que SACBEXT ha dado buenos resultados y ha estado en uso hasta la actualidad.

BDEDUCACION: Base de Datos Educación.

Proyecto que surgió con la intención de ofrecer mejoras a SACBEXT y que comenzó en el año 2003. Con el objetivo de perfeccionar el proceso de gestión, BDEDUCACION

CAPÍTULO 1: Fundamentación Teórica

rediseñó la estructura de la Base de Datos de Estudiantes Extranjeros (BDEE), introduciendo nuevos campos de información y algunos módulos. Estas modificaciones mejorarían el funcionamiento del sistema y darían nuevos resultados que permitirían una mejora en la manipulación de la información. Sin embargo, no se logró terminar el proyecto pues se estancó el proceso de corrección de los problemas existentes en SACBEXT y el sistema no fue culminado.

SIBEX: Sistema de Gestión de la Información de Estudiantes Extranjeros.

Este proyecto se desarrollo en el año 2006 en la Universidad de Cienfuegos por una estudiante de la facultad de informática con el objetivo de llevar le control de los estudiantes extranjeros que cursaban estudios en la Universidad de Cienfuegos. El sistema facilita la gestión de la información de modo que todo queda almacenado en una base de datos con la cual se puede interactuar mediante una interfaz Web con el objetivo de dar mayor seguridad y fiabilidad a la información que se procesa en la residencia estudiantil. Sibex no se encuentra en explotación por limitar las posibilidades a solo un grupo de estudiantes y no incluir otras facilidades de trabajo como es el control del mobiliario y los datos de los trabajadores.

1.5 Metodologías.

1.5.1-Lenguaje de Modelación Unificado (UML).

El Lenguaje Unificado de Modelado Unificado (UML - Unified Modeling Language) es un lenguaje que permite modelar, construir y documentar los elementos que forman un producto de software que responde a un enfoque orientado a objetos. Este lenguaje fue creado por un grupo de estudiosos de la Ingeniería de Software formado por: Ivar Jacobson, Grady Booch y James Rumbaugh en el año 1995. Desde entonces, se ha convertido en el estándar internacional para definir organizar y visualizar los elementos que configuran la arquitectura de una aplicación orientada a objetos. Con este lenguaje, se pretende unificar las experiencias acumuladas sobre técnicas de modelado e incorporar las mejores prácticas actuales en un acercamiento estándar.

CAPÍTULO 1: Fundamentación Teórica

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

Entre sus objetivos fundamentales se encuentran:

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

1.5.2- Proceso Unificado de Desarrollo (RUP).

El Proceso Unificado de Desarrollo, también conocido como Proceso Unificado de Rational (RUP, Rational Unified Process), es un proceso de desarrollo de software que utiliza UML, constituye una metodología estándar utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

Entre las características distintivas de esta metodología se pueden mencionar:

- ✓ Forma disciplinada de asignar tareas y responsabilidades (quién hace qué, cuándo y cómo)
- ✓ Pretender implementar las mejores prácticas en ingeniería de Software
- ✓ Desarrollo iterativo
- ✓ Administración de requisitos
- ✓ Uso de arquitectura basada en componentes
- ✓ Control de cambios
- ✓ Modelado visual del software
- ✓ Verificación de la calidad del software

CAPÍTULO 1: Fundamentación Teórica

RUP es un producto de Rational (IBM). Se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso. Incluye artefactos (que son los productos tangibles del proceso como por ejemplo, el modelo de casos de uso, el código fuente, etc.) y roles (papel que desempeña una persona en un determinado momento, una persona puede desempeñar distintos roles a lo largo del proceso).

RUP divide el proceso de desarrollo en ciclos, teniendo un producto final al concluir cada ciclo, en cada ciclo se analizan las fases (ver Figura.2) siguientes:

Inicio: se hace un plan de fases, se identifican los principales casos de uso y se identifican los riesgos.

Elaboración: se hace un plan de proyecto, se completan los casos de uso y se eliminan los riesgos

Construcción: se concentra en la elaboración de un producto totalmente operativo y eficiente y el manual de usuario

Transición: se implementa el producto en el cliente y se entrena a los usuarios.

Como consecuencia de esto suelen surgir nuevos requerimientos a ser analizados.[5]

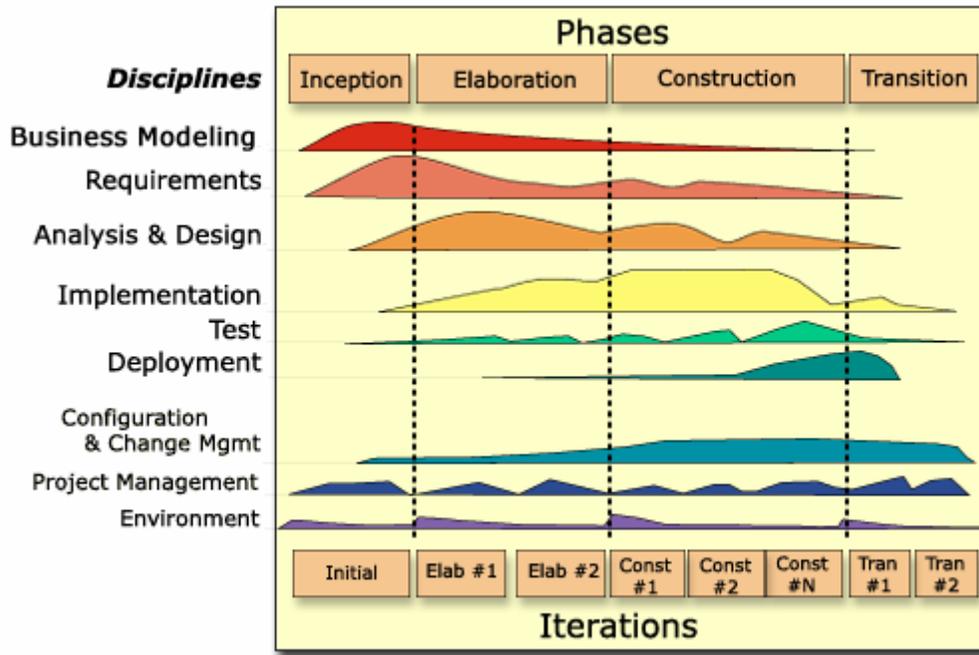


Fig. 2: Fases de RUP

Para la elaboración de los diagramas y otros modelos que propone RUP, se hará uso del Rational Rose, herramienta CASE que cubre todo el ciclo de vida de un proyecto: concepción y formalización del modelo, construcción de los componentes, transición a los usuarios y certificación de las distintas fases y entregables.

1.6 Tendencias actuales.

1.6.1- Arquitectura de desarrollo de N Capas.

Subdividir una aplicación en partes lógicas es un detalle muy provechoso. Dividir software de gran tamaño en partes más pequeñas puede hacer más simples los procesos de generarlo, reutilizarlo y modificarlo. También puede ser útil para acomodar diferentes tecnologías o diferentes organizaciones de negocio. No obstante pueden considerarse otras alternativas. La modularidad y la reusabilidad son aspectos útiles, pero pueden dar lugar a aplicaciones que no sean tan seguras, manejables o rápidas como podrían ser de otro modo.

CAPÍTULO 1: Fundamentación Teórica

Un ejemplo de la factorización de una aplicación es el modelo de N Capas, el cual constituye un modelo mejorado desde la perspectiva de más de dos capas, llamadas también niveles. Aunque, algunas veces, los niveles residen físicamente en máquinas diferentes debe enfatizarse en la distribución lógica de los mismos. Los nombres de estos niveles difieren de acuerdo a la fuente, no obstante es bastante extendido el uso de las siguientes referencias en el modelo de 3 capas:[6]

1. La capa de presentación. Generalmente es una interfaz gráfica que muestra los datos a los usuarios.
2. La capa de la lógica de negocios. Es responsable de procesar los datos recuperados y enviarlos a la capa de presentación.
3. La capa de datos. Almacena los datos de la aplicación en un almacén persistente, tal como una base de datos relacional o archivos XML.[7]

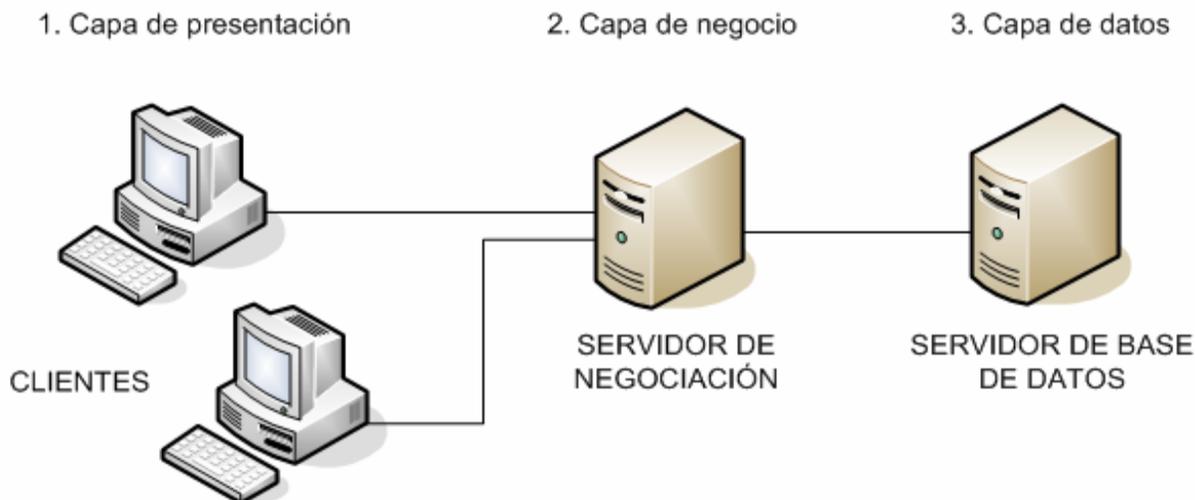


Fig. 3 Arquitectura de tres capas.

1.7 Tecnologías actuales.

1.7.1-Tecnologías Web.

Las tecnologías Web poseen una significación preponderante por el papel que está jugando la Internet en el mundo moderno. Esta plataforma WWW (World Wide Web) ha ido evolucionando paulatinamente para convertirse en un ambiente donde se implementan potentes aplicaciones cliente/servidor o arquitecturas de n capas, unido a ello han ido surgiendo nuevas tecnologías que se relacionan con el desarrollo Web lo que hacen a éste más interactivo e interesante. Entre las tecnologías utilizadas para la creación y mantenimientos de sitios Web, están las que funcionan del lado del cliente y las del lado del servidor. La diferencia entre éstas es grande:

Tecnologías del lado del cliente.

- HTML.
- CSS (Hojas de estilo en cascada).
- XML y derivados de XML.
- JavaScrip/DOM.

Están insertadas en la página HTML del cliente y son interpretadas y ejecutadas por el navegador. Es decir, su correcta funcionalidad depende del soporte de la versión del navegador a ser utilizado por el usuario visitante.

Tecnologías del lado del servidor.

- CGI y Perl.
- PHP.
- ASP.
- Java.
- ActiveX.

Pueden o no estar insertadas dentro de la página HTML. A diferencia del tipo anterior, estas tecnologías no dependen del navegador ya que son interpretadas y ejecutadas por el servidor.

HTML

HTML, no es un lenguaje de programación, es un lenguaje de especificación de contenidos para un tipo específico de documentos. Es decir, mediante HTML se puede especificar, usando un conjunto de etiquetas o tags, cómo va a representarse la

CAPÍTULO 1: Fundamentación Teórica

información en un navegador o browser. Se centra en la representación en la pantalla de la información.

HTML es un lenguaje muy sencillo que permite describir hipertexto, es decir, texto presentado de forma estructurada y agradable, con *enlaces (hyperlinks)* que conducen a otros documentos o fuentes de información relacionadas, y con *inserciones multimedia* como gráficos y sonidos. Contiene varias etiquetas (tags) las cuales son utilizadas por los desarrolladores para especificar la estructura lógica del contenido (títulos, párrafos de texto normal, enumeraciones, definiciones, citas, etc) así como los diferentes efectos que se quieren dar, tales como especificar los lugares del documento donde se debe poner cursiva, negrita, o un gráfico determinado.

Además el lenguaje HTML, permite a los desarrolladores crear documentos que pueden ser interpretados en ordenadores que tengan diferentes sistemas operativos. El HTML es un lenguaje de marcas. Los lenguajes de marcas no son equivalentes a los lenguajes de programación aunque se definan igualmente como "lenguajes". Son sistemas complejos de descripción de información, normalmente documentos, que se pueden controlar desde cualquier editor ASCII. Las marcas más utilizadas suelen describirse por textos descriptivos encerrados entre signos de "menor" (<) y "mayor" (>), siendo lo más usual que exista una marca de principio y otra de final.[8]

CSS (Hojas de estilo en cascada).

Las Hojas de Estilo en Cascada o CSS constituyen un lenguaje sencillo que complementa el de HTML, suponiendo un apoyo fundamental a la hora de diseñar páginas Web, porque permiten una mayor precisión en el ajuste de los elementos de diseño. Esta técnica consiste en separar el diseño del contenido, de manera que las indicaciones para conformar el diseño se agrupan en una hoja de estilo o archivo fuera del contenido del documento de la página HTML. Lo que hace fundamentalmente el código de las hojas de estilos es transformar las etiquetas del lenguaje HTML y conformarlas a las características que se quiera darle; pero también, y esto es lo importante, con este código se pueden crear etiquetas nuevas, que se introducen dentro del documento. Una de las ventajas de las hojas de estilos es que se puede

CAPÍTULO 1: Fundamentación Teórica

modificar algunas características de todos los documentos de un sitio Web desde un archivo, sin tener que modificarlas en cada uno de los documentos.[9]

XML

El Lenguaje Extensible de Marcas, abreviado XML (eXtensible Markup Language), describe una clase de objetos de datos llamados documentos XML y describe parcialmente el comportamiento de los programas de computadora que los procesan. XML es un "perfil de aplicación" o una forma restringida de SGML, el Lenguaje Estándar Generalizado de Marcación [ISO 8879]. Por construcción, los documentos XML son documentos SGML conformados.

XML fue desarrollado por un Grupo de Trabajo (originalmente conocido como "*SGML Editorial Review Board*" – Comité Editorial de Revisión del SGML) formado bajo los auspicios del Consorcio World Wide Web (W3C), en 1996. Fue presidido por Jon Bosak de Sun Microsystems con la participación activa de un Grupo Especial de Interés en XML (previamente conocido como Grupo de Trabajo SGML (*Standard Generalized Markup Language* – Lenguaje Estándar Generalizado de Marcado) también organizado en el W3C.

Características de XML:

- ✓ Es una arquitectura más abierta y extensible. No se necesita versiones para que puedan funcionar en futuros navegadores. Los identificadores pueden crearse de manera simple y ser adaptados en el acto en internet/intranet por medio de un validador de documentos (parser).
- ✓ Mayor consistencia, homogeneidad y amplitud de los identificadores descriptivos del documento con XML (los RDF Resource Description FrameWork), en comparación a los atributos de la etiqueta <META> del HTML.
- ✓ Integración de los datos de las fuentes más dispares. Se podrá hacer el intercambio de documentos entre las aplicaciones tanto en el propio PC como en una red local o extensa.

CAPÍTULO 1: Fundamentación Teórica

- ✓ Datos compuestos de múltiples aplicaciones. La extensibilidad y flexibilidad de este lenguaje nos permitirá agrupar una variedad amplia de aplicaciones, desde páginas web hasta bases de datos.
- ✓ Gestión y manipulación de los datos desde el propio cliente web.
- ✓ Los motores de búsqueda devolverán respuestas más adecuadas y precisas, ya que la codificación del contenido web en XML consigue que la estructura de la información resulte más accesible.
- ✓ Se permitirá un comportamiento más estable y actualizable de las aplicaciones web, incluyendo enlaces bidireccionales y almacenados de forma externa.
- ✓ El concepto de "hipertexto" se desarrollará ampliamente (permitirá denominación independiente de la ubicación, enlaces bidireccionales, enlaces que pueden especificarse y gestionarse desde fuera del documento, hiperenlaces múltiples, enlaces agrupados, atributos para los enlaces, etc. Creado a través del Lenguaje de enlaces extensible (XLL).
- ✓ Exportabilidad a otros formatos de publicación (papel, web, cd-rom, etc.). El documento maestro de la edición electrónica podría ser un documento XML que se integraría en el formato deseado de manera directa. **[10]**

1.7.2- Servicios Web

Un servicio Web es una entidad programable que proporciona alguna funcionalidad determinada, y es accesible a cualquier número de sistemas que usen las normas de Internet. Un servicio Web puede ser usado internamente por una aplicación o ser publicado hacia Internet. Estos servicios permiten la ejecución de sus funcionalidades sin importar la plataforma, sistema operativo, o lenguaje en el cual estén implementados.

Los servicios Web se pueden utilizar para intercambiar datos en redes de ordenadores como Internet. La interoperabilidad se consigue mediante la adopción de estándares abiertos. Las organizaciones OASIS y W3C son los comités responsables de la

CAPÍTULO 1: Fundamentación Teórica

arquitectura y reglamentación de los servicios Web. Para mejorar la interoperabilidad entre distintas implementaciones de servicios Web se ha creado el organismo WS-I, encargado de desarrollar diversos perfiles para definir de manera más exhaustiva estos estándares.

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

Los servicios Web brindan grandes ventajas dentro de las aplicaciones distribuidas como son:[11]

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

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

Los servicios web pueden ser utilizados a través de dos protocolos fundamentalmente: RPC (*Remote Procedure Call*) y SOAP (*Simple Object Access Protocol* – Protocolo de Acceso Simple a Objetos). El protocolo RPC tiene una dependencia de la plataforma y

CAPÍTULO 1: Fundamentación Teórica

la tecnología que se utiliza para ejecutar los servicios web, en cambio SOAP es un protocolo que permite la independencia de plataforma y tecnología.

SOAP

SOAP, siglas de *Simple Object Access Protocol*, es un protocolo estándar creado por Microsoft, IBM y otros, está actualmente bajo el auspicio de la W3C que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML (eXtensible Markup Language). SOAP es uno de los protocolos utilizados en los servicios Web. [10]

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

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

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

Razones para utilizar los Servicios Web XML

- ✓ Ofrecen un mecanismo sencillo para que las aplicaciones se comuniquen entre sí.
- ✓ Permiten que los componentes se compartan y que su funcionalidad trascienda a cualquier lugar y a cualquier persona.
- ✓ Son fáciles de distribuir y mantener.
- ✓ Ahorran tiempo y dinero al reducir la duración del ciclo de creación.

CAPÍTULO 1: Fundamentación Teórica

Permiten volver a utilizar el código que otras personas hayan desarrollado sin tener que descargar, copiar o instalar algo.

1.7.3- Lenguajes de programación Web.

ASP

Active Server Pages (ASP) es una tecnología del lado servidor de Microsoft para páginas Web generadas dinámicamente, que ha sido comercializada como un anexo a Internet Information Services (IIS). ASP ha pasado por cuatro iteraciones mayores, ASP 1.0 (distribuido con IIS 3.0), ASP 2.0 (distribuido con IIS 4.0), ASP 3.0 (distribuido con IIS 5.0) y ASP.NET (parte de la plataforma .NET de Microsoft). En el último ASP clásico, ASP 3.0, hay seis objetos integrados disponibles para el programador, Application, ASPError, Request, Response, Server y Session. Cada objeto corresponde a un grupo de funcionalidades frecuentemente usadas y útiles para crear páginas Web dinámicas. Las páginas pueden ser generadas mezclando código de scripts del lado del servidor (incluyendo acceso a base de datos) con HTML y código del lado del servidor.[12]

PHP

PHP acrónimo recursivo de "PHP: Hypertext Preprocessor" (Preprocesador de Hipertexto), es un lenguaje de programación interpretado, con licencia OpenSource. Fue originalmente diseñado en Perl, seguido por la escritura de un grupo de CGI binarios escritos en el lenguaje C por el programador Danés-Canadiense Rasmus Lerdorf en el año 1994 para mostrar su currículum vital y guardar ciertos datos, como la cantidad de tráfico que su página Web recibía.[13]

Su interpretación y ejecución se da en el servidor en el cual se encuentra almacenada la página, el cliente solo recibe el resultado de la ejecución. Cuando el cliente hace una petición al servidor para que le envíe una página Web, enriquecida con código PHP, el servidor interpretará las instrucciones mezcladas en el cuerpo de la página y las sustituirá con el resultado de la ejecución antes de enviar el resultado a la computadora del cliente. Permite el uso de las técnicas de Programación Orientada a Objetos. El

CAPÍTULO 1: Fundamentación Teórica

código PHP se incluye entre etiquetas especiales de comienzo y final que nos permitirán entrar y salir del modo PHP.

PHP tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos tales como UNIX, Linux, Windows y Mac OS X, y puede interactuar con los servidores de Web más populares. Además permite la conexión a numerosas bases de datos de forma nativa tales como: MySQL, Postgres, Oracle, ODBC, IBM DB2, Microsoft SQL Server y SQLite.

PHP frente a ASP y ASP.NET

Los motivos de ascenso de la utilización de PHP frente a otros lenguajes de de lado del servidor son:

- ✓ Libre y abierto (código fuente disponible, es gratuito). Además, posee un sinnúmero de bibliotecas que se pueden encontrar gratis en Internet.
- ✓ Multiplataforma: inicialmente fue diseñado para entornos UNIX por lo que ofrece más prestaciones en este sistema operativo, pero es perfectamente compatible con Windows.
- ✓ Posee soporte para varios servidores Web.
- ✓ Permite un soporte para la mayoría de los Sistemas Gestores de Bases de Datos.
- ✓ Se encuentra grandes cantidades de documentación. Ejemplos: manuales.
- ✓ Posee una sintaxis bastante clara y fácil de aprender.

- ✓ Es muy popular y se encuentran versiones orientadas a objetos.
- ✓ Es mantenido por una amplia comunidad de desarrolladores, lo que permite que la corrección de sus errores sean rápidos.
- ✓ Su rendimiento es reconocido por la mayoría de los desarrolladores y proyectos de software del mundo. **[14]**

1.8 - Servidores de Aplicaciones Web.

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

1.8.1- Apache.

El **servidor Apache** es un servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etcétera), Windows y otras, que implementa el protocolo HTTP/1.1 (RFC 2616) y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA HTTPd 1.3, pero más tarde fue reescrito por completo. El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation.

Apache presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido, carece de una interfaz gráfica que ayude en su configuración. En la actualidad, Apache es el servidor HTTP más usado, siendo el servidor HTTP del 70% de los sitios web en el mundo y creciendo aún su cuota de mercado.[15]

1.9- Sistemas Gestores de Base de Datos.

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

Un **Sistema Gestor de Base de Datos** (SGBD) es el software que permite la utilización y/o la actualización de los datos almacenados en una (o varias) base(s) de datos por uno o varios usuarios desde diferentes puntos de vista y a la vez.[16]

El objetivo fundamental de un SGBD consiste en suministrar al usuario las herramientas que le permitan manipular, en términos abstractos, los datos, o sea, de forma que no le sea necesario conocer el modo de almacenamiento de los datos en la computadora, ni el método de acceso empleado. En el mercado existen un sinnúmero de gestores de base de datos, todos con sus características que lo hacen una opción a escoger, pero la elección, la mayoría de las veces, depende del cliente y no del desarrollador.

CAPÍTULO 1: *Fundamentación Teórica*

A continuación mostramos una tabla (Tabla 1) comparativa de algunos de los SGBD en cuanto a si son Multiplataforma o no, Gratis o no y en cuanto a la disponibilidad del código fuente. Entre los sistemas de gestión de base de datos más utilizados en la capa de datos se encuentran SQLServer, MySQL, PostgreSQL, Oracle.

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

Tabla 1: Comparación entre SGBD. [17]

1.10 Conclusiones

En este capítulo se explica todo lo referente al proceso que tiene lugar dentro de la residencia estudiantil. Así como la necesidad de automatizar la gestión de la información dentro de esta.

Por lo que analizando las tendencias, metodologías y tecnologías actuales, que se expusieron en este capítulo, se decidió utilizar: metodología RUP y el lenguaje UML, por las ventajas que proporcionan y el alto nivel de aceptación que han tenido, Postgres se seleccionó como SGBD por su característica de ser Open Source y tener una buena seguridad en su información. Se escogieron Macromedia Dreamweaver 8 y Zend Studio para la implementación de la aplicación Web dinámica con HTML, CSS y JavaScript del lado del cliente y PHP del lado del servidor. Como servidor se utilizó Apache.

Capítulo 2: Modelo del negocio

Para desarrollar un sistema informático es necesario comprender los procesos que tienen lugar en la organización a la cual se le está realizando el estudio, con el objetivo de lograr una mejor comprensión del problema a resolver. El modelado del negocio se realiza con este fin. Esta técnica permite comprender los procesos del negocio de la organización.

En este capítulo se realiza un estudio sobre los procesos del negocio, identificando los actores y trabajadores que en él intervienen, además se listan las reglas que lo caracterizan y se describen los casos de usos.

2.1 Descripción del modelo de negocio

Un proceso de negocio es un grupo de tareas relacionadas lógicamente que se llevan a cabo en una determinada secuencia y manera y que emplean los recursos de la organización para dar resultados en apoyo a sus objetivos.[18]

Análisis del proceso.

Al ingresar un estudiante a la Universidad de Cienfuegos” Carlos Rafael Rodríguez” se dirige a la residencia estudiantil con el objetivo de solicitar una beca. El estudiante es atendido por un especialista el cual registra sus datos generales en una planilla y le hace entrega del carné de becado dándole una ubicación en un cuarto de la residencia estudiantil.

Situación Problemática.

Actualmente toda la información del estudiante en su ubicación dentro de la residencia estudiantil, así como la ubicación del mobiliario y los datos personales de los trabajadores se manipula de forma manual, siendo esto a veces muy engorroso para los responsables de dicha información, teniendo en cuenta que tienen que manejar un volumen considerable. Lo que provoca que el flujo de la información sea lento.

Procesos del Negocio

CAPÍTULO 2: Modelo del Negocio

- ✓ Solicitar beca: Proceso mediante el cual el estudiante solicita una beca para permanecer dentro de la residencia estudiantil.
- ✓ Solicitar cambio de cuarto: Proceso mediante el cual el estudiante solicita que se le traslade de cuarto dentro de la residencia estudiantil.
- ✓ Solicitar baja: Proceso mediante el cual el estudiante solicita la baja de la residencia estudiantil.
- ✓ Solicitar información: Proceso mediante el cual se le solicita información a la residencia estudiantil, referente a sus trabajadores, mobiliarios y estudiantes

2.2 Reglas del negocio a considerar.

Las reglas del negocio describen políticas que deben cumplirse o condiciones que deben satisfacerse, por lo que regulan algún aspecto del negocio. **[18]**

Después de identificar los procesos de negocio se definen las siguientes reglas del negocio:

- ✓ El estudiante puede solicitar residencia una vez matriculado en alguna carrera de la universidad y solo así se lo asigna un cuarto.
- ✓ La solicitud de cambio de cuarto es autorizada por la dirección de la residencia estudiantil.
- ✓ La solicitud de baja de la residencia es autorizada por la dirección de la residencia estudiantil.
- ✓ Solo se efectúa la baja si el estudiante ha entregado su colchón
- ✓ El estudiante solo se puede asignar a un cuarto dentro de la residencia estudiantil.
- ✓ Al estudiante solo se le puede entregar un colchón.

CAPÍTULO 2: Modelo del Negocio

- ✓ Cuando las instancias superiores solicitan alguna información, esta se considera de carácter urgente y debe ser elaborada en un margen de tiempo relativamente corto.
- ✓ Un mobiliario solo puede pertenecer a un cuarto dentro de la residencia estudiantil.

2.3 Modelo de casos de uso del negocio.

El modelo de Casos de Uso del Negocio es un modelo que describe los procesos de negocio de una empresa en términos de casos de uso y actores del negocio en correspondencia con los procesos del negocio y los clientes, respectivamente. El modelo de casos de uso del negocio presenta un sistema (en este caso, el negocio) desde la perspectiva de su uso y esquematiza cómo proporciona valor a sus usuarios. [19]

El modelo de Casos de Uso del Negocio es definido a través de tres artefactos: el diagrama de casos de uso del negocio, la descripción de los casos de uso del negocio y el diagrama de actividades de cada caso de uso del negocio.

2.3.1 Actores del negocio.

Un actor del negocio es cualquier individuo, grupo, entidad, organización, máquina o sistema de información externos; con los que el negocio interactúa. Lo que se modela como actor es el rol que se juega cuando se interactúa con el negocio para beneficiarse de sus resultados. [19]

La tabla muestra los actores y una descripción de su rol:

Nombre del Actor	Justificación
Estudiante	Es quien se beneficia e inicia la mayoría de los procesos del negocio, como la solicitud de beca, el registro de los datos personales, la solicitud de traslado de cuarto, la solicitud de la baja de la residencia estudiantil.
Directivo	Es el que inicia y se beneficia del proceso de solicitud de información a la residencia estudiantil.

CAPÍTULO 2: Modelo del Negocio

Tabla 2: Actores del Negocio

2.3.2- Diagramas de casos de uso del negocio.

Un proceso de negocio es un grupo de tareas relacionadas lógicamente que se llevan a cabo en una determinada secuencia y manera y que emplean los recursos de la organización para dar resultados en apoyo a sus objetivos. Un caso de uso del negocio representa a un proceso de negocio, por lo que se corresponde con una secuencia de acciones que producen un resultado observable para ciertos actores del negocio. Desde la perspectiva de un actor individual, define un flujo de trabajo completo que produce resultados deseables. [18]

Para comprender los procesos de negocio se construye el diagrama de casos de uso del negocio en el que aparece cada proceso del negocio relacionado con su actor.

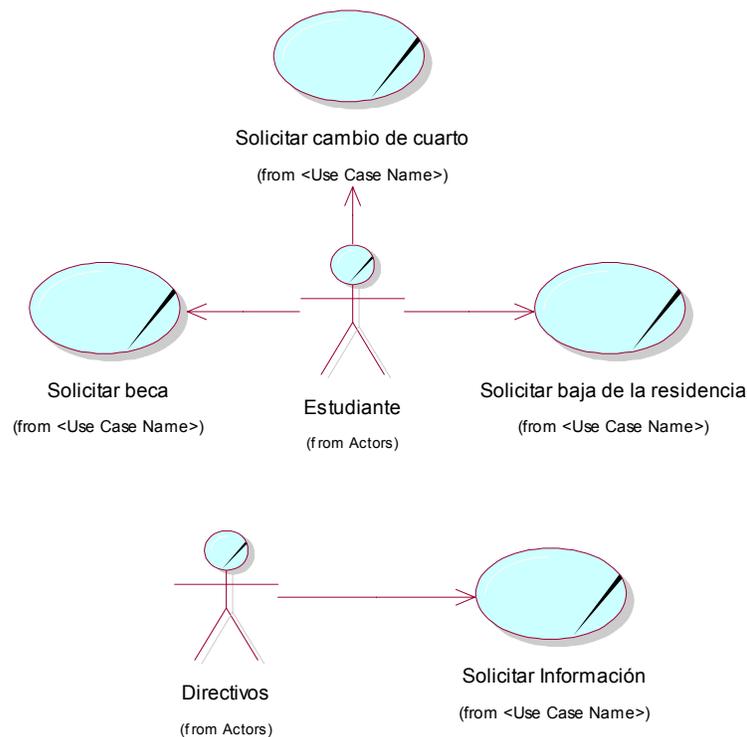


Fig. 4: Diagramas de Casos de Uso del Negocio

CAPÍTULO 2: Modelo del Negocio

2.3.3 – Trabajadores del negocio.

Un trabajador del negocio es una abstracción de una persona (o grupo de personas), una máquina o un sistema automatizado; que actúa en el negocio realizando una o varias actividades, interactuando con otros trabajadores del negocio y manipulando entidades del negocio. Representa un rol.[19]

Trabajador	Descripción
Especialista	Encargado de registrar y actualizar los datos del estudiante, así como de asignarlos en un cuarto de la residencia. Elabora informes como resultado de solicitudes efectuadas por algún <i>Directivo</i> .
Administrador	Encargado de controlar la ubicación y el estado del mobiliario dentro de la residencia, así como entregar los medios al estudiante.
Jefe de la Residencia.	Encargado de analizar las solicitudes de beca, cambio de cuarto de los estudiantes y baja de la residencia estudiantil.

Tabla 3: Descripción de los Trabajadores del Negocio

2.3.4 – Descripción de los casos de uso del negocio.

Caso de Uso del Negocio	Solicitar beca
Actores	Estudiante(Inicia)
Propósito	Asignar el estudiante a un cuarto de la residencia estudiantil.
Resumen El caso de uso se inicia cuando el estudiante arriba a la residencia estudiantil una vez matriculado en una de las carreras de la universidad con anterioridad. El especialista los recibe y lo ubica en un cuarto de la residencia estudiantil. El caso de uso finaliza cuando el estudiante queda ubicado en el cuarto.	
Acción del Actor	Respuesta del negocio
1. El estudiante se presenta en la	

CAPÍTULO 2: Modelo del Negocio

Prioridad	Media
Flujos Alternos	
Línea 3	Si la persona no tiene acceso a la información el jefe de la residencia notifica el rechazo de la solicitud culminando el caso de uso.
Mejoras	El proceso se realizará de forma automatizada, utilizando la información que se encuentra almacenada en el por el sistema.

Tabla 7: Descripción del caso de uso del negocio Solicitar información.

2.3.5 – Diagramas de actividades del negocio.

El diagrama de actividad es un grafo que contiene los estados en que puede hallarse la actividad a analizar. Cada estado de la actividad representa la ejecución de una sentencia de un procedimiento, o el funcionamiento de una actividad en un flujo de trabajo. En resumen describe un proceso que explora el orden de las actividades que logran los objetivos del negocio.[19]

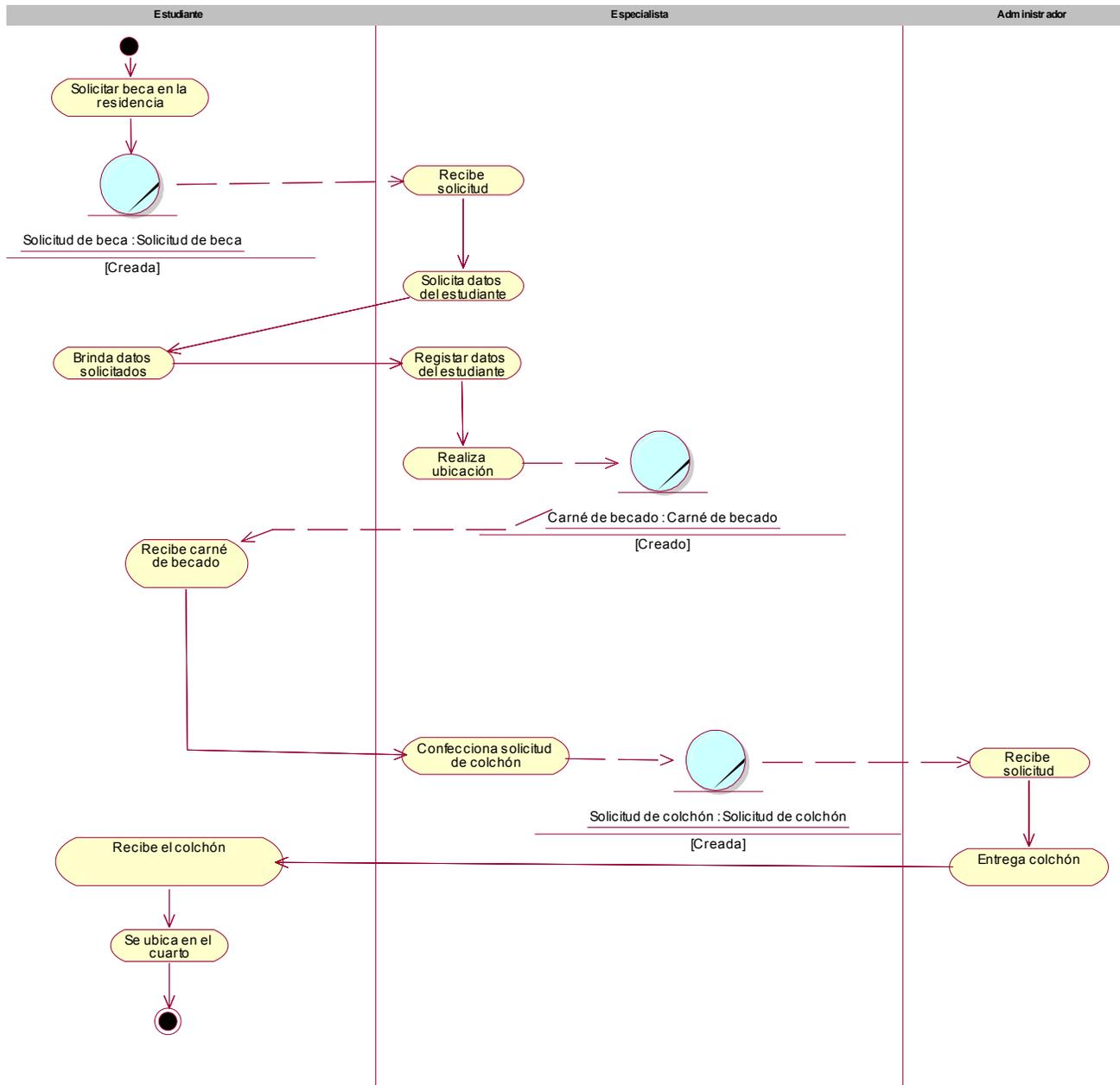


Fig. 5: Diagrama de Actividad. Caso de Uso Solicitar Beca

CAPÍTULO 2: Modelo del Negocio

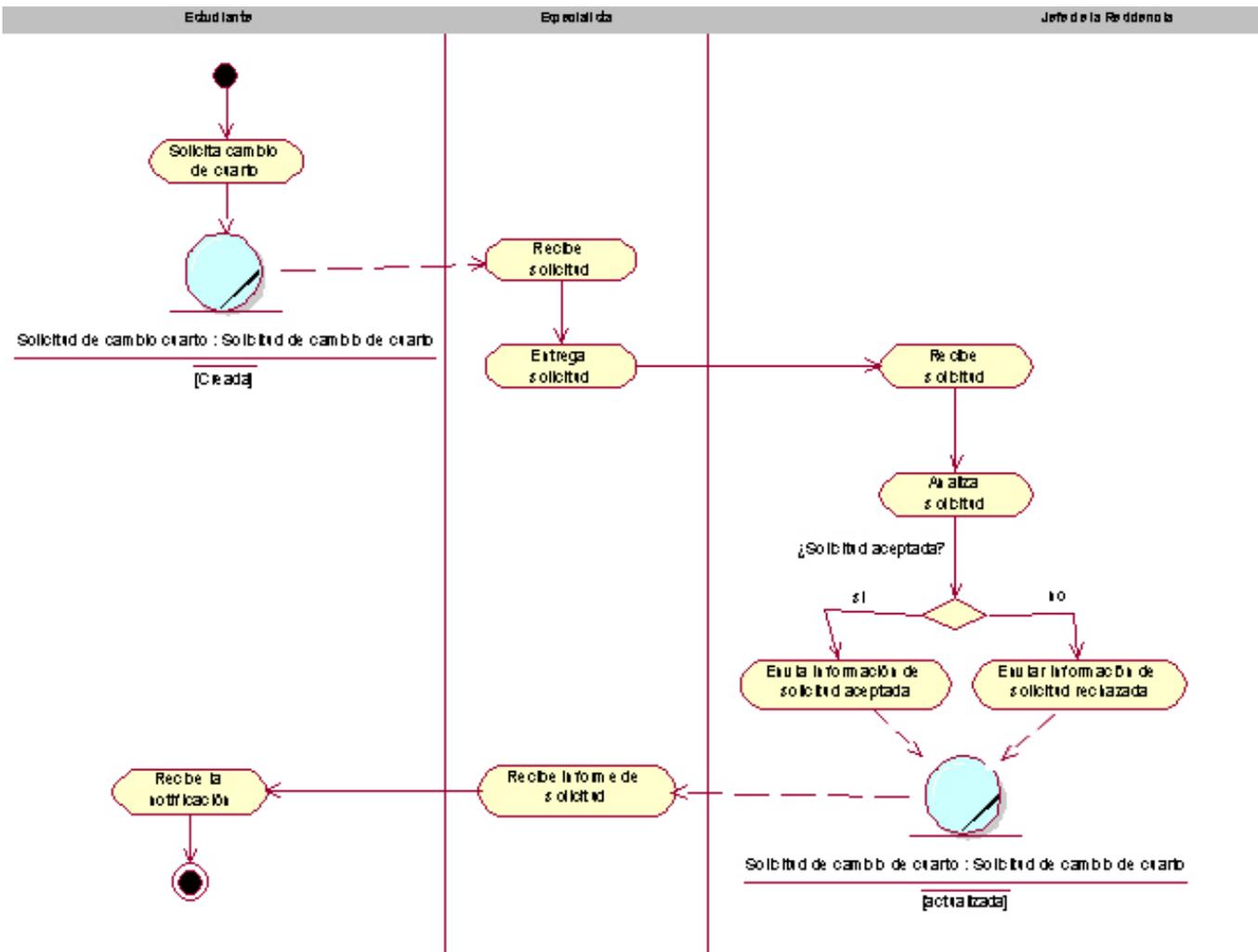


Fig. 6 Diagrama de Actividad. Caso de Uso Solicitar Cambio de Cuarto.

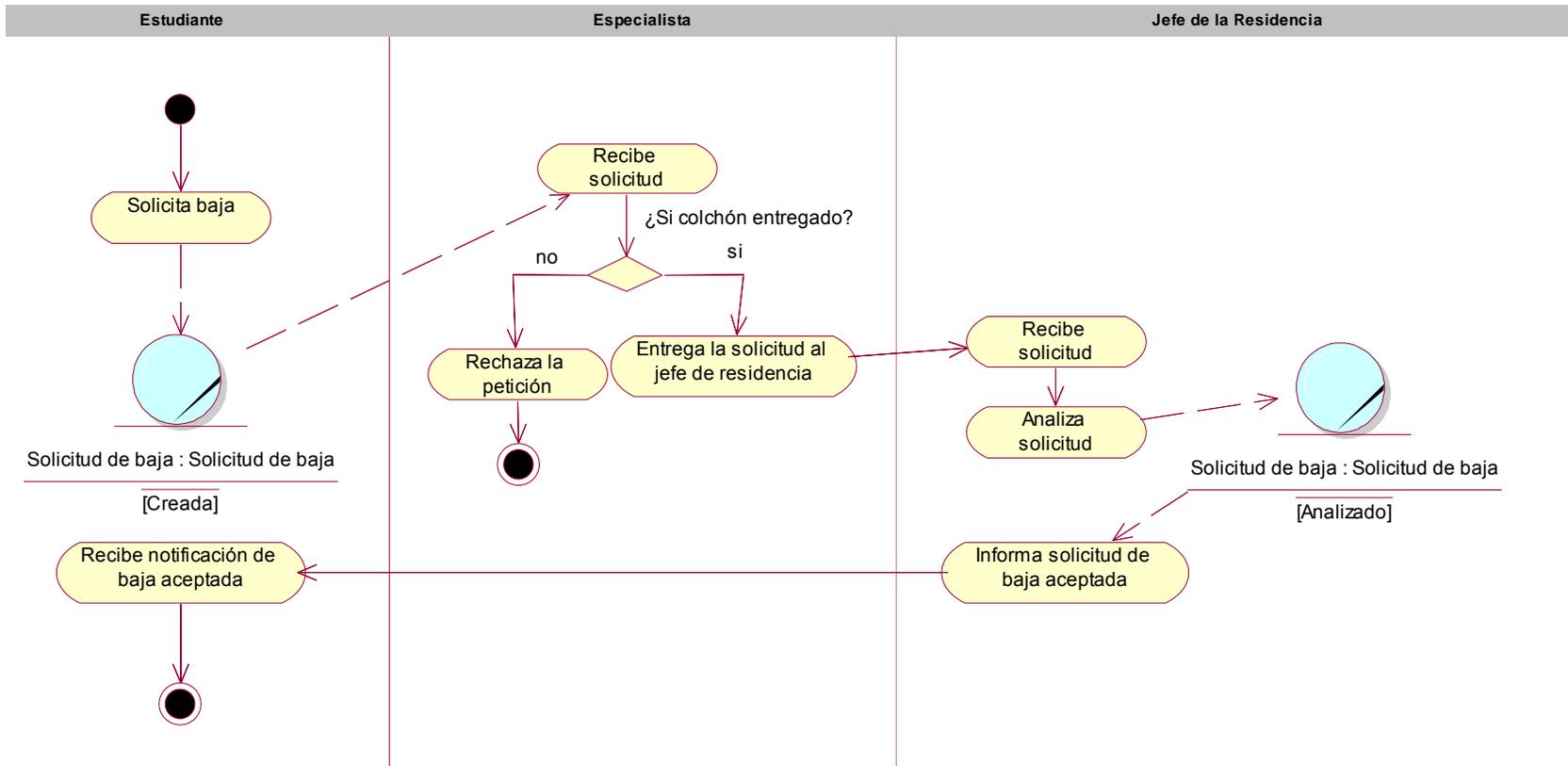


Fig. 7: Diagrama de Actividad. Caso de Uso Solicitar Baja

CAPÍTULO 2: Modelo del Negocio

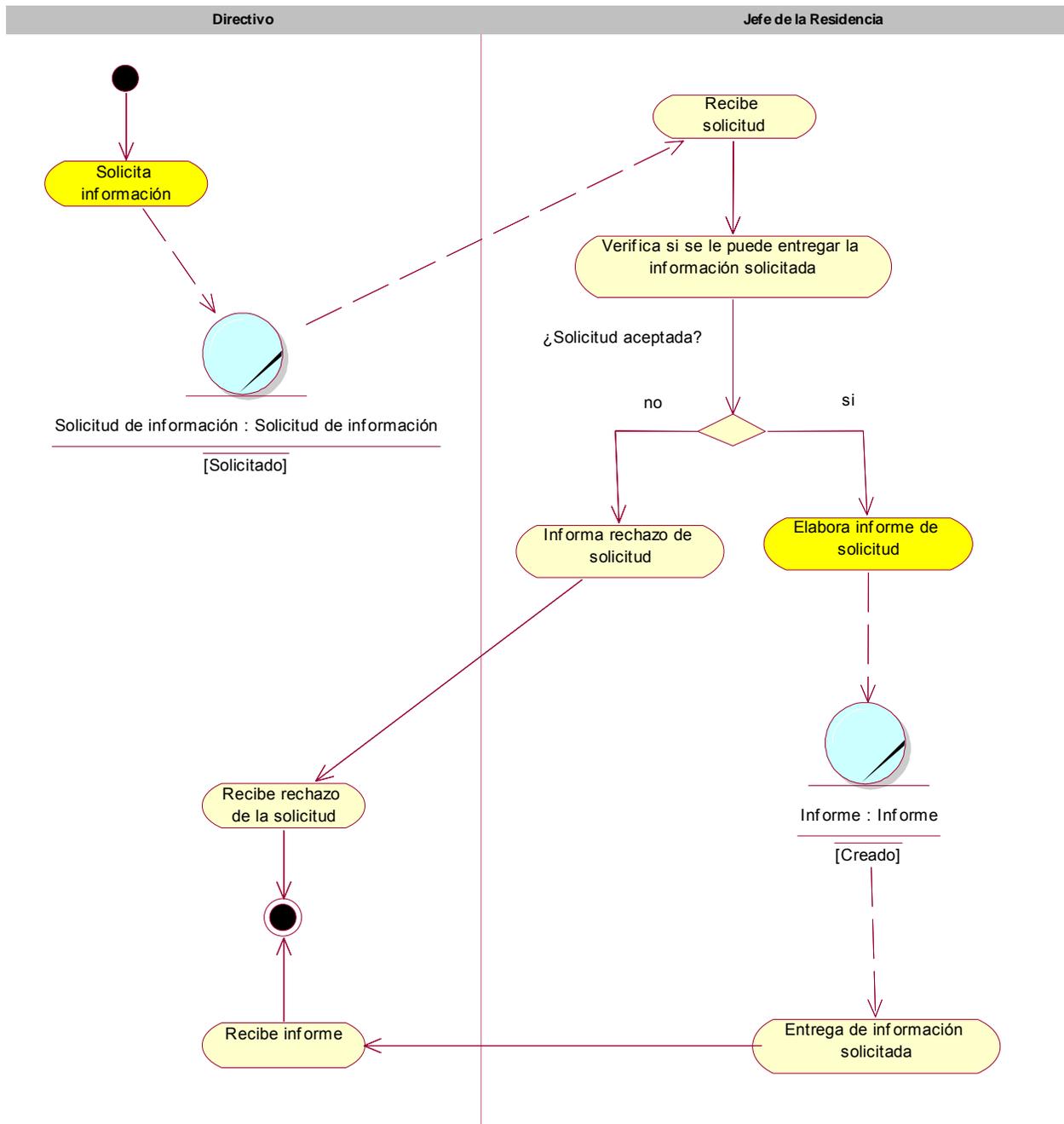


Fig. 8: Diagrama de Actividad. Caso de Uso Solicitar información.

2.4 Modelo de objetos del negocio.

El diagrama de clases, como artefacto que se construye para describir el modelo de objetos del negocio, muestra la participación de los trabajadores y entidades del negocio y la relación entre ellos.

Las entidades de negocio representan a los objetos que los trabajadores del negocio toman, inspeccionan, manipulan, producen o utilizan durante la realización de los casos de uso de negocio. [18]

En la siguiente figura se muestra el diagrama del modelo de objetos correspondiente al negocio que se modela.

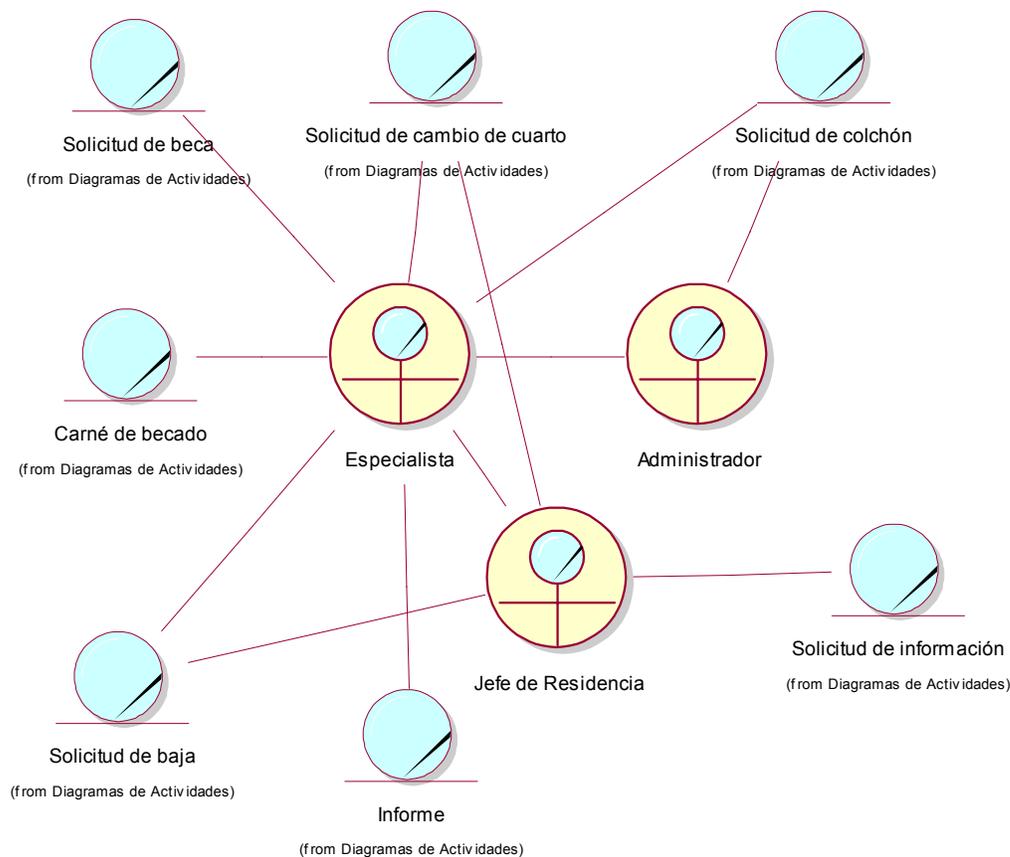


Fig. 9 Diagramas de clases del modelo de objetos del negocio.

2.5 Conclusiones

En este capítulo fueron descritos los procesos que tiene lugar en la residencia estudiantil de la Universidad de Cienfuegos, identificando a su vez los actores, trabajadores y objetos del negocio, así como su relación en esos procesos (casos de uso). Esta descripción fue realizada mediante el modelo del negocio, para lo cual se elaboraron los modelos de casos de uso y de actividad, lográndose una mejor comprensión de los procesos de la residencia estudiantil de la UCF, dando paso al modelado del sistema.

Capítulo 3 – Construcción de la Solución Propuesta.

En este capítulo se identifican los requisitos funcionales y no funcionales del sistema que dará solución al problema planteado; quiénes interactuarán con él (actores del sistema) y las distintas funcionalidades que ofrecerá a cada uno de los actores (casos de uso), utilizando para esto el artefacto de la metodología RUP modelo del sistema.

3.1 Descripción del sistema propuesto.

El sistema que se propone como solución a la problemática expuesta en capítulos anteriores lleva por nombre SIREU (Sistema Informático para la Residencia Estudiantil). SIREU está concebido en su arquitectura como una aplicación Web utilizando un modelo de tres capas y con la visión de código libre (Open Source).

Con este software se automatizará gran parte de la información que se maneja en la Residencia Estudiantil de la Universidad de Cienfuegos. Aunque su uso puede extenderse a otras universidades del país.

El sistema proporciona la gestión de la información relacionada con los estudiantes becados, la ubicación y el estado del mobiliario de la residencia estudiantil así como la información de los trabajadores que en ella laboran. Esta dividido en tres módulos, el primer módulo relacionado con la gestión de la información de los estudiantes al que tendrá acceso los especialistas y el jefe de la residencia estudiantil, un segundo módulo relacionado con la información del mobiliario de la residencia y un tercero que engloba la información de los trabajadores con las opciones administrativas del sistema.

3.1.1 – Requerimientos funcionales.

Los requerimientos funcionales permiten expresar una especificación más detallada de las responsabilidades del sistema que se propone. Ellos permiten determinar, de una manera clara, lo que debe hacer el mismo. [19]

Todo esto basándose en las necesidades de los usuarios y clientes.

CAPÍTULO 3: Construcción de la Solución Propuesta

Los requerimientos funcionales del sistema propuesto son los siguientes:

1. Autenticar usuario.
2. Cambiar contraseña.
3. Modificar datos usuario.
4. Visualizar usuarios del sistema.
5. Eliminar usuario del sistema.
6. Visualizar estudiante por facultad.
7. Visualizar estudiante por Carné de Identidad.
8. Visualizar estudiante por nombre.
9. Visualizar estudiante por país.
10. Visualizar datos generales del estudiante.
11. Asignar estudiante a un cuarto.
12. Eliminar estudiante de un cuarto.
13. Mover estudiante de cuarto.
14. Asignar colchón a estudiante.
15. Retirar colchón al estudiante.
16. Modificar colchón al estudiante.
17. Insertar evaluación del estudiante.
18. Visualizar evaluación del estudiante.
19. Imprimir evaluación del estudiante
20. Insertar diagnostico psicológico del estudiante.
21. Visualizar diagnostico psicológico del estudiante.
22. Imprimir diagnostico psicológico del estudiante.
23. Insertar sanciones del estudiante.
24. Visualizar sanciones del estudiante.
25. Imprimir sanciones del estudiante.
26. Insertar dispensarización del estudiante.
27. Visualizar dispensarización del estudiante.
28. Imprimir dispensarización del estudiante.
29. Visualizar mobiliario por cuarto.

CAPÍTULO 3: Construcción de la Solución Propuesta

30. Visualizar mobiliario por número de inventario.
31. Insertar mobiliario.
32. Eliminar mobiliario.
33. Modificar mobiliario.
34. Visualizar trabajadores por cargo.
35. Visualizar trabajadores por departamento.
36. Insertar datos de un trabajador.
37. Eliminar los datos trabajador
38. Modificar los datos del trabajador.
39. Otorgar permisos al trabajador.
40. Insertar datos de edificios.
41. Visualizar datos de edificios.
42. Modificar datos de edificios.
43. Insertar datos de cuartos.
44. Visualizar datos de cuartos.
45. Modificar datos de cuartos.

3.1.2 – Requerimientos no funcionales.

Los requisitos no funcionales especifican propiedades del sistema, como restricciones del entorno o de la implementación, rendimiento, dependencias de la plataforma, factibilidad de mantenimiento, extensibilidad y fiabilidad; con las cuales debe cumplir.[19]

Apariencia o interfaz externa.

El sistema debe presentar una interfaz amigable, sencilla, agradable, legible y de fácil uso, de tal forma que en lugar de convertirse un problema para el usuario signifique una ventaja. El sistema además debe indicarle al usuario dónde está y qué puede hacer desde ese lugar.

CAPÍTULO 3: Construcción de la Solución Propuesta

Usabilidad.

El sistema propuesto es usado por el jefe de la residencia, la especialista y el administrador de la residencia personas que pueden o no, tener habilidades en el trabajo con la computadora, debido a que contiene una ayuda para informar al usuario sobre los aspectos más importantes en la utilización del mismo. El sistema proporcionará un mejor desempeño del personal involucrado en la confección de los planes, facilitando además el acceso a la información sin un costo elevado.

Rendimiento.

El sistema propuesto debe ser rápido en el procesamiento de la información así como a la hora de dar respuesta a la solicitud de los usuarios, además debe permitir el acceso simultáneo a los datos por diferentes usuarios. Todo esto depende en gran medida del uso que se le de a los recursos que se disponen en el modelo Cliente/Servidor y de la velocidad de las consultas en la Base de Datos. El sistema deberá recuperarse en un corto período de tiempo ante cualquier falla.

Seguridad.

El sistema propuesto debe garantizar la protección de la información de acceso no autorizado, utilizando la autenticación para garantizar el cumplimiento de esto. En la implementación del sitio Web se harán validaciones de la información tanto en el cliente como en el servidor para contribuir a la seguridad del sistema. El sistema impondrá un estricto control de acceso que permitirá a cada usuario tener disponible solamente las opciones relacionadas con su actividad.

Requisitos de Rendimiento

Aunque no se requiere una velocidad de respuesta comparada con los sistemas de tiempo real, se debe garantizar la rapidez de respuesta del sistema ante las solicitudes de los usuarios.

CAPÍTULO 3: Construcción de la Solución Propuesta

Como la aplicación está concebida para 3 capas, debe garantizarse que los tiempos de respuestas sean generalmente rápidos al igual que la velocidad de procesamiento de la información. Se seguirá una lógica de negocios en la comunicación entre el cliente y el servidor Web de modo tal que cada cual procesará lo que le corresponde, puesto que es posible mencionar que existen funciones que son más óptimas ejecutarlas en el cliente, otras por cuestiones de seguridad, o de acceso a los datos, deben realizarse en el servidor. Se realizará una parte del proceso en el cliente y en dependencia de esto se concluirá en el servidor, lo que facilitará un tiempo de respuesta más rápido, una mayor velocidad de procesamiento, y un mayor aprovechamiento de los recursos. El software estará disponible las 24 horas del día y debe recuperarse ante una falla lo más pronto posible aunque es válido destacar que una caída momentánea no alterará significativamente los procesos de gestión.

Requisitos de Soporte

✓ **Del lado del Servidor:**

Se requiere una computadora que cuente con un servidor Web Apache, con soporte para PHP versión 4.0 o superior. Además, se requiere de un servidor de base de datos Postgres. Todo lo anterior para una eficiencia óptima, aunque todo el conjunto puede estar en una sola máquina.

✓ **Del lado del cliente:**

Por parte del cliente se requiere un navegador que interprete HTML y las funciones básicas de JavaScript, con cualquier sistema operativo.

Requerimientos políticos, culturales y legales.

La herramienta propuesta deberá responder a los intereses de la Constitución de la República de Cuba, asimismo no existirán prioridades en el servicio según el nivel social, cultural o étnico.

Requisitos de Ayuda y Documentación en Línea.

Debe disponerse de una ayuda sobre las principales opciones del sistema.

CAPÍTULO 3: Construcción de la Solución Propuesta

Requisitos de Software

La aplicación debe poderse ejecutar en entornos *Windows* y/o *Linux* (Multiplataforma). Del lado del servidor se utilizará Apache como servidor Web, del lado del cliente cualquiera de los exploradores existentes en el mercado.

Requerimiento de Hardware

✓ Servidor:

La máquina servidora debe tener como mínimo las siguientes características de hardware: Procesador Pentium III 450 MHz o superior, 256 Mb de memoria RAM (incluye la utilizada por el Sistema Operativo) y 5Gb de capacidad en disco duro.

✓ Cliente:

Las computadoras situadas en los puestos de trabajo de los usuarios requerirán como mínimo un procesador Pentium II, 64 Mb de memoria RAM. Estas máquinas deberán estar conectadas en red con el servidor.

3.3 Modelo de casos de uso del sistema.

El modelo de casos de uso permite que los desarrolladores de software y los clientes lleguen a un acuerdo sobre los requisitos, es decir, sobre las condiciones y posibilidades que debe cumplir el sistema. Describe lo que hace el sistema para cada tipo de usuario y proporciona la entrada fundamental para el análisis, el diseño y las pruebas.[19]

3.3.1 – Actores del sistema.

Los actores representan terceros fuera del sistema que colaboran con este. Suelen corresponderse con trabajadores o actores en un negocio.[19]

Actor	Descripción
Especialista	Este actor es el encargado de gestionar la información de los estudiantes becados, de insertar y actualizar las

CAPÍTULO 3: Construcción de la Solución Propuesta

	evaluaciones, sanciones, de asignarlos a un cuarto de la residencia, así como asignarle un colchón.
Administrador	Este actor es el encargado de gestionar toda la información relacionada con la ubicación y el estado del mobiliario en la residencia estudiantil.
Jefe de la Residencia	Este actor tiene control total sobre el sistema, es quien crea los usuarios y le da permiso según el nivel de acceso a la información.
Usuario del Sistema	Este actor es un usuario genérico que va a realizar las acciones comunes entre los otros actores.

Tabla 8: Actores del sistema.

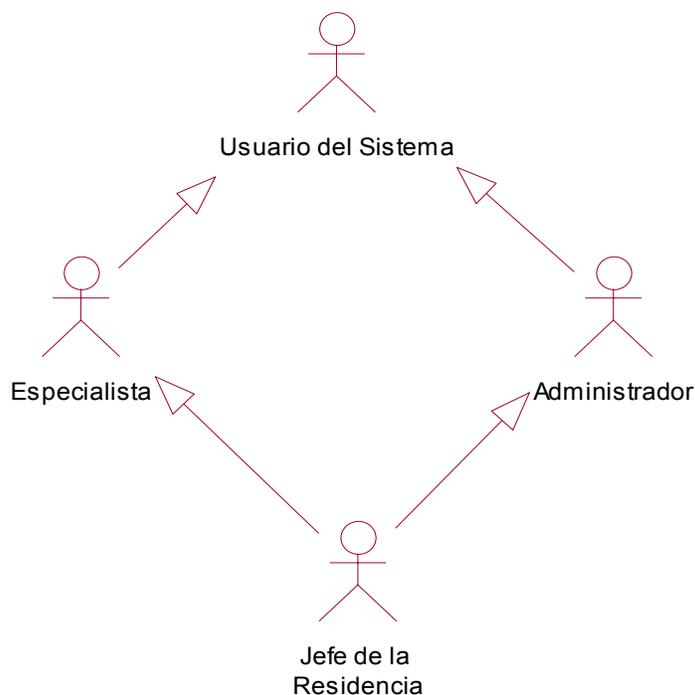


Fig. 10: Relación entre actores del sistema.

3.3.2 – Paquetes y sus relaciones

Los paquetes son un mecanismo de organización de elementos que subdividen el modelo en otros más pequeños que colaboran entre sí. Este particionamiento debe

CAPÍTULO 3: Construcción de la Solución Propuesta

hacerse sobre la base de los requerimientos funcionales y el dominio del problema; y debe ser reconocible por las personas con conocimiento del dominio.[18]

A partir de lo anteriormente expuesto se propone el siguiente diagrama de paquetes:

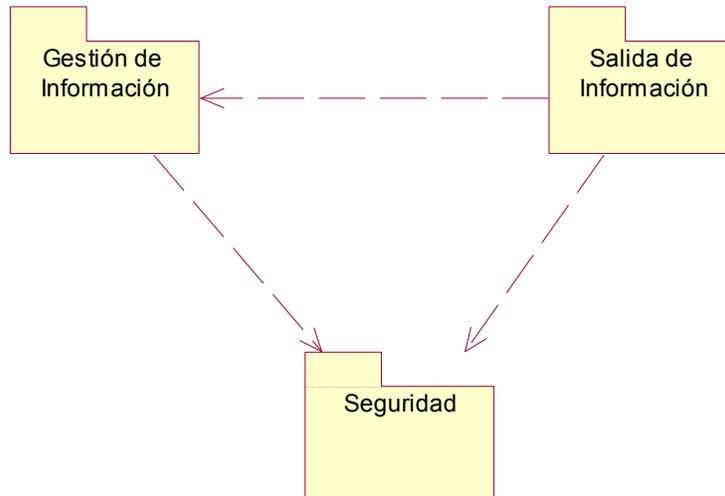


Fig. 11: Diagrama de Casos de Usos por Paquetes.

Paquete Gestión de Información.

Se agrupan todas las funcionalidades relacionadas con la actualización (inserción, modificación o eliminación) de los datos.

Paquete Salida de Información.

Se agrupan todas las funcionalidades relacionadas con, la visualización de los datos, filtrados según diferentes criterios de búsqueda, la obtención e impresión de reportes.

Paquete de Seguridad.

Se agrupan todas las funcionalidades relacionadas con la autenticación y cambio de contraseña de los distintos tipos de usuarios del sistema.

3.3.3 – Diagramas de casos de uso del sistema.

Los casos de usos constituyen un instrumento para dividir la complejidad del sistema en varios pedazos más manejables que facilitarán su comprensión y análisis, estos deben estar basados en la forma en que el sistema es visto por un usuario.

CAPÍTULO 3: Construcción de la Solución Propuesta

A continuación se muestra el diagrama de casos de uso para cada uno de los paquetes del sistema.

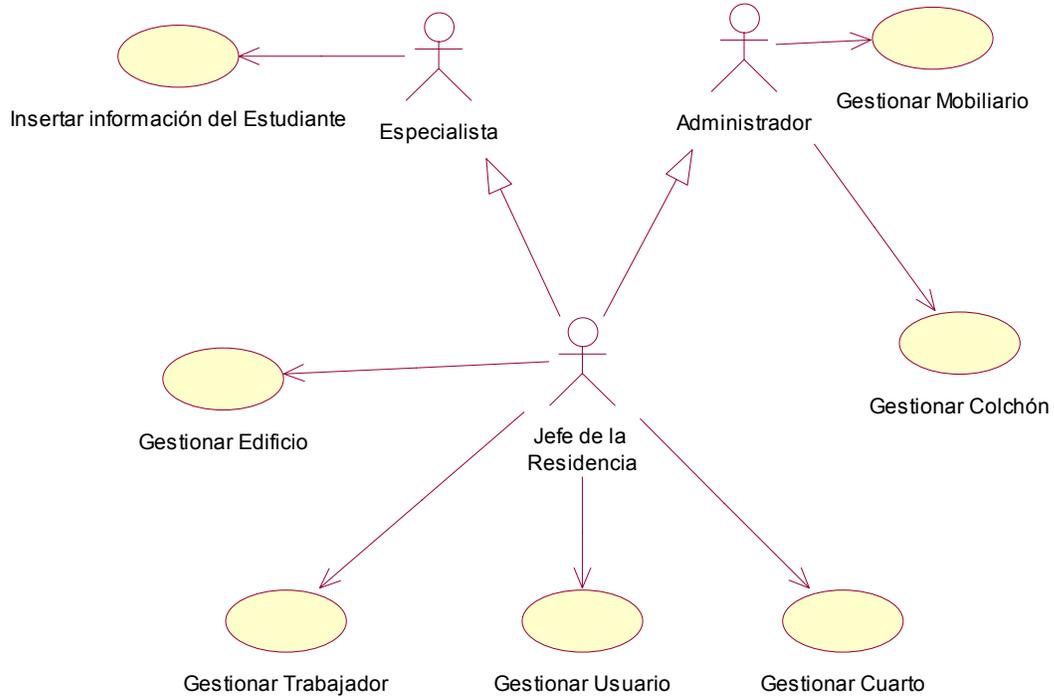


Fig. 12: Diagrama de casos de uso del sistema *Gestión de Información*

CAPÍTULO 3: Construcción de la Solución Propuesta

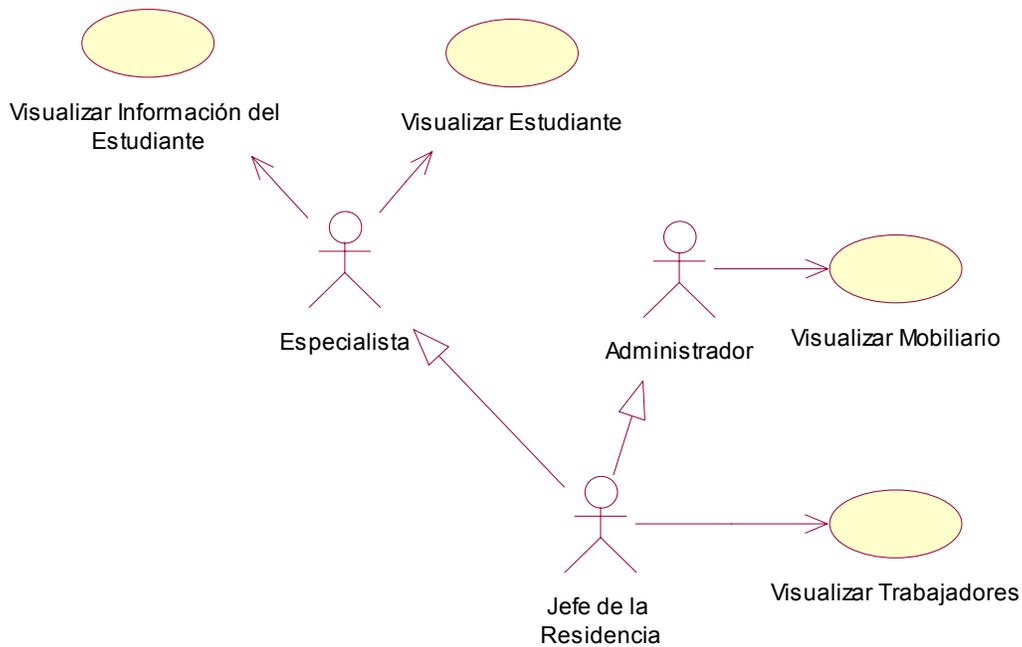


Fig. 13: Diagrama de casos de uso del sistema *Salida de Información*



Fig. 14: Diagrama de casos de uso del sistema *Paquete de Seguridad y Utilidades*

3.3.4 – Descripción de los casos de uso del sistema.

Cada forma en que los actores usan el sistema se representa con un Caso de Uso. Los Casos de Uso son “fragmentos” de funcionalidad que el sistema ofrece para aportar un resultado de valor para sus actores. Un Caso de Uso especifica una secuencia de acciones que el sistema puede llevar a cabo interactuando con sus actores, incluyendo alternativas dentro de la secuencia. [19]

CAPÍTULO 3: Construcción de la Solución Propuesta

Para este sistema se definieron los siguientes casos de usos:

1. Iniciar sesión.
2. Cambiar contraseña
3. Gestionar usuario.
4. Visualizar estudiantes.
5. Gestionar cuarto.
6. Gestionar colchón.
7. Insertar información adicional del estudiante.
8. Visualizar información adicional del estudiante.
9. Visualizar mobiliario.
10. Gestionar mobiliario.
11. Visualizar trabajadores.
12. Gestionar trabajador
13. Gestionar edificios.

Los cuales se agrupan en los paquetes antes mencionados como se muestra a continuación:

Paquete: **Gestión de Información**

1. Gestionar usuario.
2. Gestionar cuarto.
3. Gestionar colchón.
4. Insertar información adicional del estudiante.
5. Gestionar mobiliario.
6. Gestionar trabajador
7. Gestionar edificios.

Paquete: **Salida de Información**

8. Visualizar estudiantes.
9. Visualizar información adicional del estudiante.
10. Visualizar mobiliario.

CAPÍTULO 3: Construcción de la Solución Propuesta

11. Visualizar trabajadores.

Paquete: **Seguridad**

12. Iniciar sesión.

13. Cambiar contraseña

A continuación se muestran las descripciones de esos casos de uso.

Caso de uso	Iniciar sesión.
Actores	Usuario del sistema (inicia).
Propósito	Registrarse en el sistema.
Resumen	<p>El caso de uso inicia cuando el usuario del sistema desea iniciar sesión, para esto debe introducir su cuenta y contraseña. El sistema verifica en la base de datos si la información es correcta, el usuario podrá acceder a las opciones del sistema que le corresponde, en caso contrario el sistema muestra un mensaje de error denegando el acceso y finalizando así el caso de uso.</p>
Referencias	R1
Precondiciones	La cuenta y la contraseña del usuario deben ser correctas.
Post-condiciones	El usuario puede acceder a toda a la información a la que tiene permiso.
Prototipo	Ver Anexo A.1

Tabla 9: Descripción Caso de Uso del Sistema Iniciar Sesión

Caso de uso	Cambiar contraseña.
Actores	Usuario del sistema (inicia).
Propósito	Cambiar contraseña de entrar al sistema.
Resumen	<p>El caso de uso se inicia cuando un usuario registrado desea cambiar su contraseña el sistema le brinda la posibilidad de introducir la nueva contraseña, si los datos son</p>

CAPÍTULO 3: Construcción de la Solución Propuesta

correctos la contraseña es cambiada en caso contrario se muestra un mensaje de error, finalizando así el caso de uso.	
Referencias	R2
Precondiciones	Debe existir información almacenada de este usuario.
Post-condiciones	Queda actualizada la contraseña del usuario.
Prototipo	Ver Anexo A.2

Tabla 10: Descripción Caso de Uso del Sistema Cambiar Contraseña

Caso de uso	Gestionar usuario.
Actores	Jefe de la Residencia (inicia).
Propósito	Gestionar la información relacionada al grupo de trabajadores que tendrán permisos a introducir y modificar datos en el sistema.
Resumen	El caso de uso inicia cuando el jefe de la residencia decide crear los usuarios del sistema a partir del grupo de trabajadores. Para ello puede insertar, modificar y eliminar los datos referentes a estos usuarios. De forma tal que el sistema permita actualizar la información con facilidad, culminando así el caso de uso.
Referencias	R3, R4 , R5 y R6
Precondiciones	Debe existir al menos un administrador del sistema para realizar las acciones de crear, modificar y eliminar los usuarios.
Post-condiciones	Se actualiza la información referente a los usuarios: Si acción: insertar, inserta un usuario. Si acción: modificar, se modifican los datos del usuario. Si acción: eliminar, se elimina el usuario.
Prototipo	Ver Anexo A.3

Tabla 11: Descripción Caso de Uso del Sistema Gestionar Usuario

Caso de uso	Visualizar estudiantes.
Actores	Especialista (inicia).

CAPÍTULO 3: Construcción de la Solución Propuesta

Propósito	Visualizar los datos de los estudiantes.
Resumen	El caso de uso se inicia cuando la especialista desea visualizar los datos de un estudiante o un listado de estudiantes haciendo uso de un criterio de búsqueda como lo puede ser: por facultad, por nombre, por país, por carné de identidad. El sistema muestra el listado de estudiantes, culminando así el caso de uso.
Referencias	R7,R8,R9,R10 y R11
Precondiciones	Debe existir información del estudiante almacenada con anterioridad en el sistema SIGENO de el cual se extrae toda la información.
Prototipo	Ver anexo A.4

Tabla 12: Descripción Caso de Uso del Sistema Visualizar los datos de los estudiantes.

Caso de uso	Gestionar cuarto.
Actores	Especialista (inicia).
Propósito	Gestionar la información relacionada con la asignación de estudiantes por cuartos.
Resumen	El caso de uso se inicia cuando la especialista se dispone a ubicar a los estudiantes en los cuarto de la residencia estudiantil. Con lo que se puede asignar a un cuarto, moverlo de cuarto o eliminarlo de un cuarto. De esta forma el sistema permitirá mantener la información actualizada, finalizando así el caso de uso.
Referencias	R12,R13 y R14
Precondiciones	Debe existir información del estudiante almacenado con anterioridad en el sistema SIGENO de el cual se extrae toda la información y debe estar creada la estructura de la residencia estudiantil en el sistema SIREU.
Post-condiciones	Se actualiza la información referente a la ubicación de los estudiantes en la residencia estudiantil. Si acción: insertar, se asigna el estudiante a un cuarto. Si acción: modificar, se modifica la ubicación del

CAPÍTULO 3: Construcción de la Solución Propuesta

	estudiante en la residencia estudiantil. Si acción: eliminar, se elimina el estudiante de la residencia.
Prototipo	Ver Anexo A. 5

Tabla 13: Descripción Caso de Uso del Sistema Gestionar cuarto.

Caso de uso	Gestionar colchón.
Actores	Especialista (inicia).
Propósito	Gestionar la información relacionada con la asignación de colchones a los estudiantes.
Resumen	El caso de uso se inicia cuando la especialista se dispone a realizar la asignación de colchones por estudiante. Con lo que se puede asignar, modificar y eliminar el identificador de un colchón a un estudiante. De esta forma el sistema permitirá que la información esté actualizada, finalizando así el caso de uso.
Referencias	R15,R16 y R17
Precondiciones	El estudiante debe estar asignado a un cuarto de la residencia estudiantil.
Post-condiciones	Se actualiza la información referente a la asignación de colchones por estudiantes. Si acción: insertar, se asigna un identificador de colchón a un estudiante. Si acción: modificar, se modifica el identificador de colchón del estudiante. Si acción: eliminar, se elimina el identificador de colchón del estudiante.
Prototipo	Ver Anexo A.6

Tabla 14: Descripción Caso de Uso del Sistema Gestionar colchón.

Caso de uso	Insertar información adicional del estudiante.
--------------------	--

CAPÍTULO 3: Construcción de la Solución Propuesta

Actores	Especialista (inicia).
Propósito	Insertar información adicional a los estudiantes para mantener actualizada sus evaluaciones, sanciones, dispensarización y su diagnóstico psicológico.
Resumen El caso de uso se inicia cuando el especialista decide insertar información adicional a un estudiante, como es el caso de sus evaluaciones, sanciones, dispensarización y su diagnóstico psicológico. Con lo cual quedan actualizados dichos datos, finalizando así el caso de uso.	
Referencias	R18,R21,R24 y R27
Precondiciones	El estudiante debe tener asignado un cuarto de la residencia estudiantil.
Post-condiciones	Se actualizan las evaluaciones, sanciones, dispensarización y el diagnóstico psicológico del estudiante.
Prototipo	Ver Anexo A.7

Tabla 15: Descripción Caso de Uso del Sistema Insertar información adicional del estudiante.

Caso de uso	Visualizar información adicional del estudiante.
Actores	Especialista (inicia).
Propósito	Visualizar información adicional de los estudiantes como lo son: sus evaluaciones, sanciones, dispensarización y su diagnóstico psicológico.
Resumen El caso de uso se inicia cuando el especialista visualiza la información adicional de un estudiante, como es el caso de sus evaluaciones, sanciones, dispensarización y su diagnóstico psicológico. Con lo que se muestra una ficha del estudiante según la opción que se halla elegido que permite imprimirla como un reporte de ser necesario, finalizando así el caso de uso.	
Referencias	R19,R20,R22,R23,R25,R26,R28 y R29

CAPÍTULO 3: Construcción de la Solución Propuesta

Precondiciones	El estudiante debe tener asignado un cuarto de la residencia estudiantil y la información adicional debe haberse insertado con anterioridad.
Post-condiciones	Se visualizan las evaluaciones, sanciones, dispensarización y el diagnóstico psicológico del estudiante con la opción de imprimirla a modo de reporte.
Prototipo	Ver Anexo A.8

Tabla 16: Descripción Caso de Uso del Sistema Visualizar información adicional del estudiante.

Caso de uso	Visualizar mobiliario
Actores	Administrador (inicia).
Propósito	Visualizar información referente al mobiliario de la residencia estudiantil.
Resumen El caso de uso inicia cuando el administrador visualiza la disponibilidad del mobiliario de la residencia, filtrando si lo desea en función de un criterio de búsqueda, finalizando así el caso de uso.	
Referencias	R30, y R31
Precondiciones	Los datos a visualizar deben haber sido insertados
Prototipo	Ver Anexo A. 9

Tabla 17: Descripción Caso de Uso del Sistema Visualizar mobiliario.

Caso de uso	Gestionar mobiliario.
Actores	Administrador (inicia).
Propósito	Mantener actualizado la información referente al mobiliario de la residencia estudiantil.
Resumen El caso de uso se inicia cuando el administrador se dispone a actualizar la información relacionada con el mobiliario de la residencia estudiantil, el cual se puede insertar, modificar o eliminar de un cuarto de la residencia estudiantil.	
Referencias	R32,R33 y R34

CAPÍTULO 3: Construcción de la Solución Propuesta

Precondiciones	La estructura de la residencia debe estar creada.
Post-condiciones	Se actualiza la información referente al mobiliario de la residencia estudiantil. Si acción: insertar, se insertan los datos del mobiliario. Si acción: modificar, se modifica los datos del mobiliario. Si acción: eliminar, se elimina los datos del mobiliario.
Prototipo	Ver Anexo A.10

Tabla 18: Descripción Caso de Uso del Sistema Gestionar mobiliario

Caso de uso	Visualizar trabajadores
Actores	Jefe de la residencia (inicia).
Propósito	Visualizar información referente a los trabajadores de la residencia estudiantil.
Resumen El caso de uso inicia cuando el jefe de la residencia visualiza los datos de los trabajadores de la residencia estudiantil o si lo desea en función de un criterio de búsqueda, finalizando así el caso de uso.	
Referencias	R35 y R36
Precondiciones	Los datos a visualizar deben ser previamente llenados.
Post-condiciones	--
Prototipo	Ver Anexo A.11

Tabla 19: Descripción Caso de Uso del Sistema Visualizar trabajadores.

Caso de uso	Gestionar trabajador.
Actores	Jefe de la Residencia (inicia).
Propósito	Mantener actualizado la información referente a los trabajadores de la residencia estudiantil.
Resumen El caso de uso se inicia cuando el jefe de la residencia se dispone a actualizar la información relacionada con los trabajadores de la residencia estudiantil, cuyos datos se pueden insertar, modificar o eliminar.	

CAPÍTULO 3: Construcción de la Solución Propuesta

Referencias	R37,R38 y R39
Precondiciones	La estructura de la residencia debe estar creada.
Post-condiciones	Se actualiza la información referente al trabajador de la residencia estudiantil. Si acción: insertar, se insertan los datos del trabajador. Si acción: modificar, se modifica los datos del trabajador. Si acción: eliminar, se elimina los datos del trabajador.
Prototipo	Ver Anexo A.12

Tabla 20: Descripción Caso de Uso del Sistema Gestionar trabajador.

Caso de uso	Gestionar edificios.
Actores	Jefe de la Residencia (inicia).
Propósito	Mantener actualizado la información referente a la estructura de la residencia estudiantil.
Resumen	
El caso de uso se inicia cuando el jefe de la residencia se dispone a actualizar la información relacionada con la estructura de la residencia estudiantil, cuyos datos se pueden insertar, modificar, finalizando así el caso de uso.	
Referencias	R32,R33 y R34
Precondiciones	El administrador del sistema debe estar registrado
Post-condiciones	Se actualiza la información referente a la estructura de la residencia estudiantil. Si acción: insertar, se insertan los datos de los edificios. Si acción: modificar, se modifica los datos del edificio.
Prototipo	Ver Anexo A.13

Tabla 21: Descripción Caso de Uso del Sistema Gestionar edificios.

CAPÍTULO 3: Construcción de la Solución Propuesta

3.4 – Diseño de la base de datos.

3.4.1 – Modelo lógico de datos.

Representación gráfica del diagrama de clases persistentes.

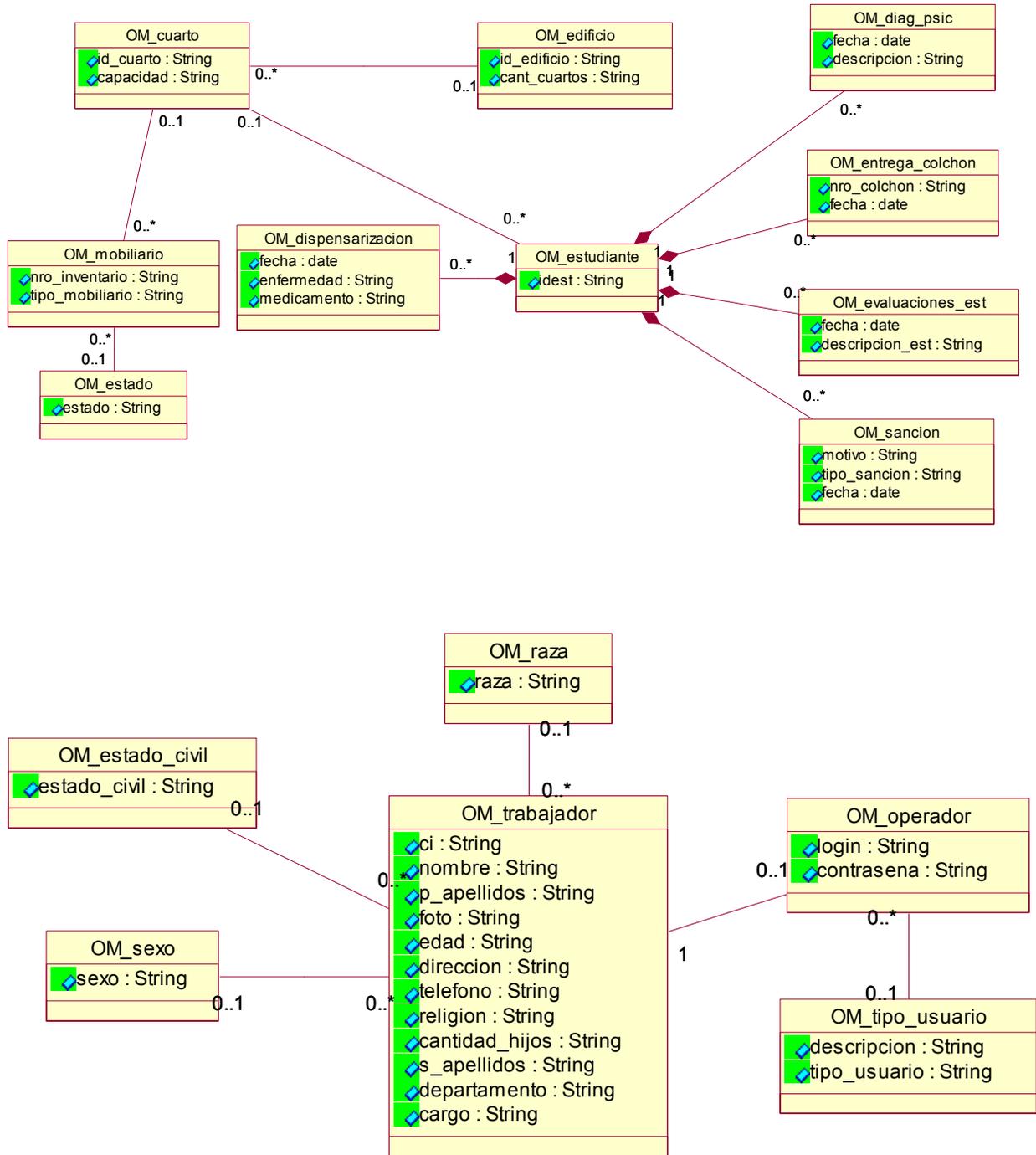
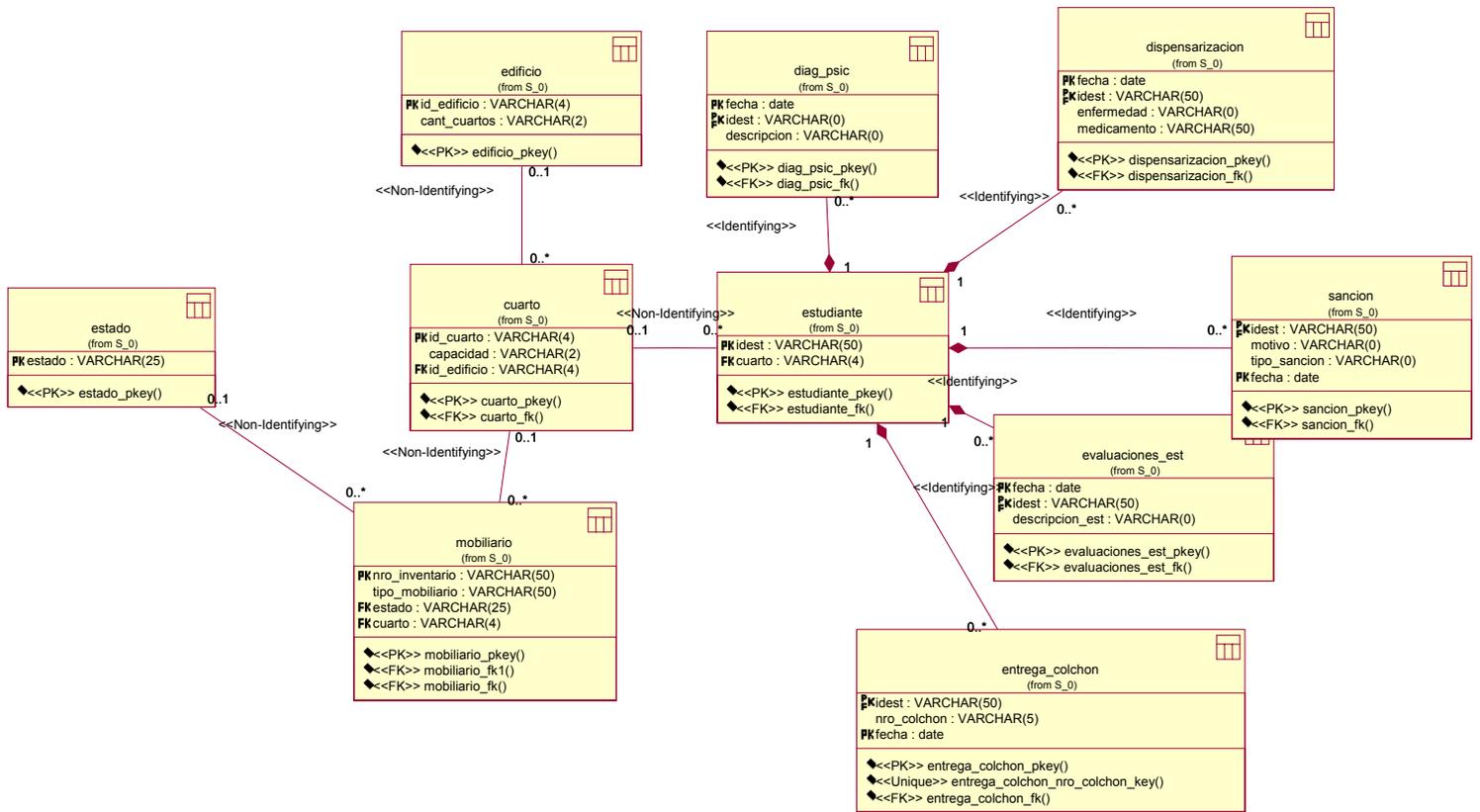


Fig. 15 Modelo Lógico de Datos

CAPÍTULO 3: Construcción de la Solución Propuesta

3.4.2 – Modelo físico de datos.

Representación gráfica del diagrama del modelo físico de datos.



CAPÍTULO 3: Construcción de la Solución Propuesta

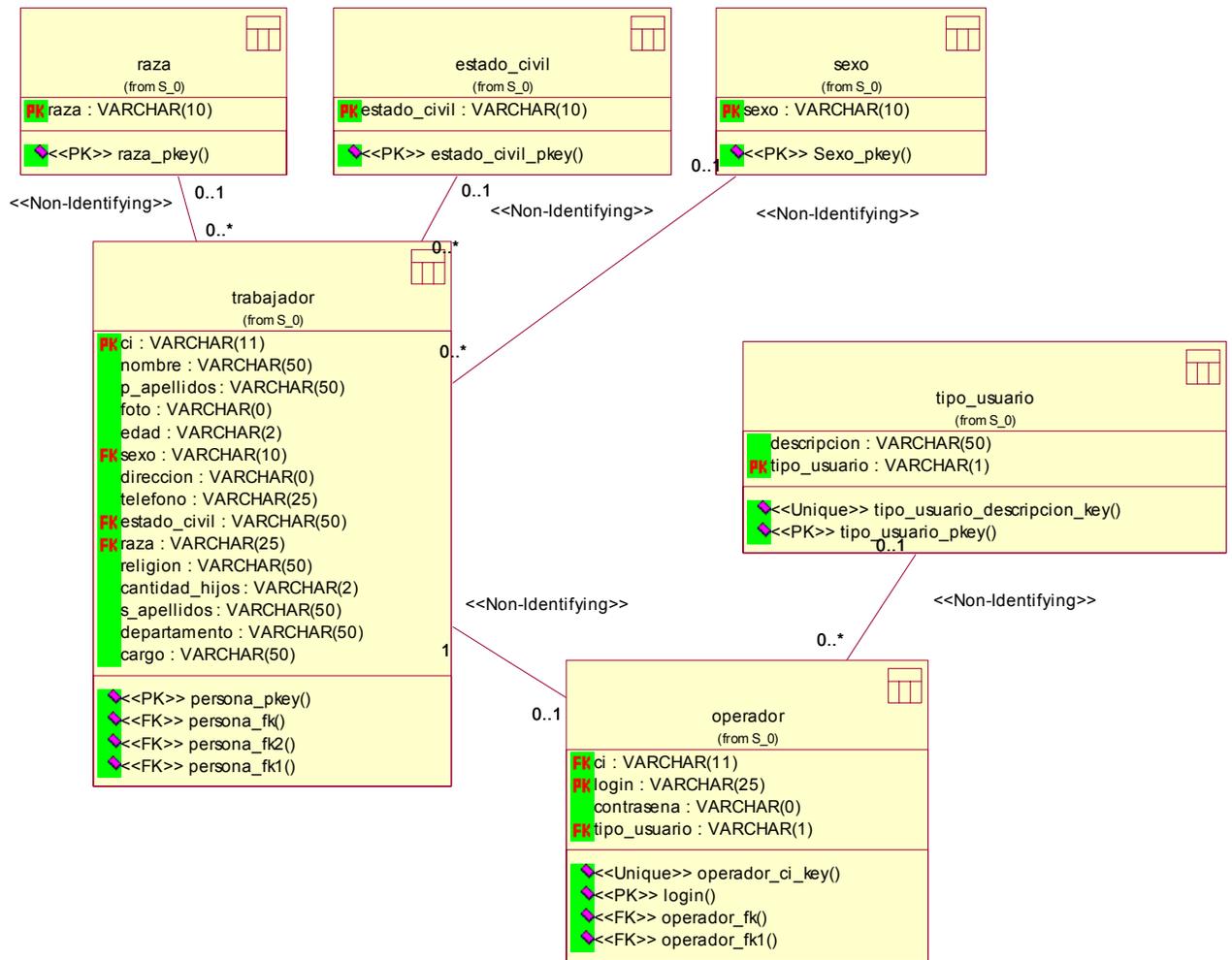


Fig. 16: Modelo físico de datos.

3.5 – Diagrama de implementación.

Representación gráfica de los diagramas de implementación.

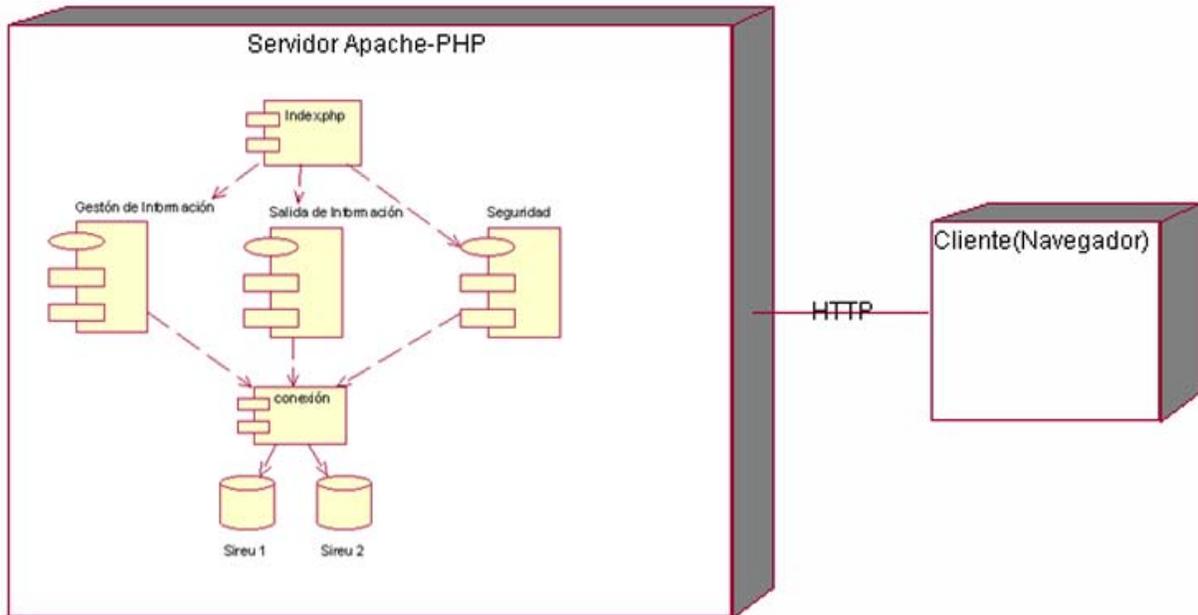


Fig. 17: Diagrama de Implementación

3.6 – Principios de diseño.

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

3.6.1 – Estándares en la interfaz de la aplicación.

La primera impresión del usuario cuando visita una aplicación Web la brinda el diseño de la interfaz. Es por ello que, para lograr la apariencia adecuada y que el usuario se sienta confortable, se tienen en cuenta varios aspectos, sobre todo relacionados con tipografía, colores, gráficos, navegación, composición del sitio, etc., que a continuación

CAPÍTULO 3: Construcción de la Solución Propuesta

se detallan. En el sistema, el diseño de la interfaz está basado en páginas Web, se utilizan las tonalidades de azul por ser suaves y refrescantes. El vocabulario manejado es lo menos técnico posible, acercándose al utilizado por los usuarios.

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

3.6.2 – Tratamiento de errores.

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

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

3.6.3 – Concepción General de la ayuda.

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

. Cada aspecto de la ayuda ha sido diseñado con el objetivo de expresar explícitamente cómo y en qué orden debe operar el usuario.

3.7- Diagramas de Clases Web.

Un diagrama de clases presenta las clases del sistema con sus relaciones estructurales y de herencia.

CAPÍTULO 3: Construcción de la Solución Propuesta

En el anexo B se pueden encontrar el diagrama de clases web correspondiente a caso de uso del sistema.

Casos de Uso	Diagrama Clases Web
Iniciar sesión.	Anexo B.1
Cambiar contraseña	Anexo B.2
Gestionar usuario	Anexo B.3
Visualizar estudiantes	Anexo B.4
Gestionar cuarto	Anexo B.5
Gestionar colchón	Anexo B.6
Insertar información adicional del estudiante	Anexo B.7
Visualizar información adicional del estudiante.	Anexo B.8
Visualizar mobiliario.	Anexo B.9
Gestionar mobiliario.	Anexo B.10
Visualizar trabajadores	Anexo B.11
Gestionar trabajador	Anexo B.12
Gestionar edificios.	Anexo B.13

Tabla 22: Diagramas de clases Web del Sistema

3.7 – Conclusiones

En este capítulo se ha descrito los requerimientos funcionales y no funcionales, del sistema.

Mediante el modelo del sistema, se identificaron los actores, casos de uso y sus relaciones.

Con el modelo de diseño se expusieron las consideraciones asumidas en cuanto a este, a la vez que se proporcionó, mediante los diagramas de clases web, modelo físico y modelo lógico de datos una representación de las relaciones existentes entre clases.

Finalmente con el modelo de implementación quedó clara la distribución física del sistema.

Capítulo 4-Estudio de Factibilidad.

En el presente capítulo se realiza el estudio que concluye si es o no factible la realización del proyecto, basado en un previo análisis de costos y beneficios. Se utilizó para el cálculo de la estimación del esfuerzo, el tiempo de desarrollo y el costo del proyecto el método de puntos de fusión. Se deben obtener primero las instrucciones fuentes. Analizándose para esto las cantidades de entradas, salidas, peticiones, archivos lógicos e interfaces externas preliminares que tiene el sistema.

4.1 – Planificación por puntos de función.

Nombre de la entrada externa	Cantidad de ficheros	Cantidad de Elementos de datos	de Clasificación(Bajo, Medio y Alto)
Cambiar contraseña	1	4	Baja
Insertar usuario	1	4	Baja
Modificar datos del usuario	1	4	Baja
Eliminar usuario del sistema	1	4	Baja
Asignar estudiante a un cuarto	1	2	Baja
Eliminar estudiante de un cuarto	1	2	Baja
Mover estudiante de cuarto	1	2	Baja
Asignar colchón a estudiante	2	3	Baja
Retirar colchón al estudiante	2	3	Baja
Modificar colchón al estudiante	2	3	Baja
Insertar evaluación del estudiante	2	3	Baja
Insertar diagnóstico psicológico del estudiante	2	3	Baja
Insertar sanciones del estudiante	2	4	Baja
Insertar dispensarización del	2	4	Baja

CAPÍTULO 4: Estudio de Factibilidad y Análisis de Resultados

estudiante			
Insertar mobiliario	1	4	Baja
Eliminar mobiliario	1	4	Baja
Modificar mobiliario	1	4	Baja
Insertar datos de un trabajador	1	14	Baja
Eliminar los datos trabajador	1	14	Baja
Modificar los datos del trabajador	1	14	Baja
Insertar datos de edificios	1	2	Baja
Modificar datos de edificios	1	2	Baja
Insertar datos de cuartos	1	3	Baja
Modificar datos de cuartos	1	3	Baja
Totales por clasificación			
Baja	Media		Alta
24	0		0

Tabla 23: Planificación. Entradas externas

Nombre de la petición	Cantidad de ficheros	Cantidad de Elementos de datos	de Clasificación(Bajo, Medio y Alto)
Autenticar usuario	1	4	Baja
Visualizar usuarios del sistema	1	4	Baja
Visualizar evaluación del estudiante	1	3	Baja
Visualizar diagnostico psicológico del estudiante	1	3	Baja
Visualizar sanciones del estudiante	1	4	Baja
Visualizar dispensarización del estudiante	1	4	Baja
Visualizar mobiliario por cuarto	1	4	Baja

CAPÍTULO 4: Estudio de Factibilidad y Análisis de Resultados

Visualizar mobiliario por número de inventario	1	4	Baja
Visualizar trabajadores por cargo	1	14	Baja
Visualizar trabajadores por departamento	1	14	Baja
Visualizar datos de edificios	1	2	Baja
	1	3	Baja
Totales por clasificación			
Baja	Media		Alta
12	0		0

Tabla 24: Planificación: Peticiones

Nombre del fichero interno	Cantidad de record	Cantidad de Elementos de datos	Clasificación(Bajo, Medio y Alto)
Cuarto	1	3	Baja
Diag_psic	1	3	Baja
Dispensarización	1	4	Baja
Edificio	1	2	Baja
Entrega_colchón	1	3	Baja
Estado	1	1	Baja
Estado_civil	1	1	Baja
Estudiante	1	2	Baja
Evaluaciones_est	1	3	Baja
Mobiliario	1	4	Baja
Operador	1	4	Baja
Raza	1	1	Baja
Sanciones	1	4	Baja
Sexo	1	1	Baja

CAPÍTULO 4: Estudio de Factibilidad y Análisis de Resultados

Tipo_usuario	1	2	Baja
Trabajador	1	14	Baja
Totales por clasificación			
Baja	Media		Alta
16	0		0

Tabla 25: Planificación: Ficheros internos

Elementos	Bajos	X Peso	Medios	X Peso	Altos	X Peso	Subtotal de puntos de función
Ficheros lógicos internos	16	7	0	10	0	10	112
Ficheros de interfaces externas	0	5	0	7	0	15	0
Entradas externas	24	3	0	4	0	6	72
Salidas externas	0	4	0	5	0	7	0
Peticiones	12	3	0	4	0	6	36
Total							220

Tabla 26: Planificación: Punto de función.

CAPÍTULO 4: Estudio de Factibilidad y Análisis de Resultados

Características	Valor		
Puntos de función desajustados	220		
Lenguaje	SQL	PHP	JavaScript
Instrucciones fuentes por puntos de función	37	44	58
Por ciento de la aplicación en cuanto a requerimientos funcionales	3%	87%	10%
Instrucciones fuentes	244	8421	1276
Total de Instrucciones fuentes	9941		

Tabla 27: Planificación: Miles de instrucciones fuentes

4.2 – Determinación de los costos.

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

Cálculo de:	Valor	Justificación
RCPX	1,00	Base de datos es moderada, no se requiere de amplia documentación (Nominal)
RUSE	1,00	Se implementa código reusable para el aprovechamiento de este en toda la aplicación. (Nominal)
PDIF	1,00	No tiene restricciones en cuanto al tiempo de ejecución ya que el software podrá estar trabajando muchas horas. EL Software no tiene limitación de memoria impuesta. La plataforma de aplicación tiene gran estabilidad. (Nominal)
PERS	0,83	Hay poco movimiento del personal (Alto)
PREX	0,87	El equipo tiene buen dominio y posee conocimiento del lenguaje de programación. Con una experiencia

CAPÍTULO 4: Estudio de Factibilidad y Análisis de Resultados

		de aproximadamente 3 años. (Alto)
FCIL	0,87	Se utilizan herramientas de programación como: Macromedia Dreamweaver 8, Zend Development Studio 5.5, así como la herramienta CASE Rational Rose 2003 para la documentación, empleando como notación UML. (Alto)
SCED	1,00	La planificación se hace con moderada frecuencia. (Nominal)
PREC	3,72	El equipo de desarrollo posee una comprensión considerable de los objetivos del producto, no tiene experiencia en la realización de software de este tipo. (Nominal)
FLEX	3,04	El sistema cuenta con alguna flexibilidad en relación con las especificaciones de los requerimientos preestablecidos y a las especificaciones de interfaz externa. (Nominal)
TEAM	2,19	El equipo que va a desarrollar el software no tiene problema con la cooperación entre los miembros que lo componen. (Alto)
RESL	4,24	Teniendo en cuenta la alta experiencia que existe en el país acerca de este tipo de estudios existen algunos factores de riesgo. (Nominal)
PMAT	1,56	El grupo de desarrollo domina las áreas de prácticas claves de desarrollo. (Muy alto)

Tabla 28: Multiplicadores de esfuerzos.

CAPÍTULO 4: Estudio de Factibilidad y Análisis de Resultados

Multiplicador de esfuerzos

$$EM = \prod_{i=1}^7 E_{mi} = RCPX * RUSE * PDIF * PERS * PREX * FCIL * SCED$$

$$EM = \prod_{i=1}^7 E_{mi} = 1,00 * 1,00 * 1,00 * 0,83 * 0,87 * 0,87 * 1,00 = 0,63$$

Factores de escala

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

$$SF = \sum SFi = 3,72 + 3,04 + 2,19 + 4,24 + 1,56 = 14,75$$

Valores de los coeficientes

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

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

$$E = 0,91 + 0,01 * 14,75$$

$$E = 1,0575$$

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

$$F = 0,24 + 0,2 * (1,0575 - 0,91)$$

$$F = 0,2695$$

Esfuerzo

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

$$PM = 2,94 * (9)^{1,0575} * 0,63$$

$$PM = 2,94 * 10,21 * 0,63$$

$$PM = 18,91$$

Cálculo del tiempo de desarrollo

$$TDEV = C * PM^F$$

$$TDEV = 3,67 * (18,91)^{0,2695}$$

$$TDEV = 3,67 * 2,2083$$

$$TDEV = 8,10 \approx 8 \text{ meses}$$

Cálculo de la cantidad de hombres

CAPÍTULO 4: Estudio de Factibilidad y Análisis de Resultados

$$CH = PM / TDEV$$

$$CH = 18,91/8,10$$

$$CH = 2,3 \approx 2 \text{ hombres}$$

Costo

$$CHM = CH * \text{Salario medio}$$

$$CHM = 2 \text{ hombres} * \$ 275$$

$$CHM = 550\$/\text{mes}$$

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

$$\text{Costo} = \$550 * 18,91$$

$$\text{Costo} = \$10400 \text{ costo del proyecto.}$$

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

Cálculo de:	Valor
Esfuerzo(PM)	18,91
Tiempo de desarrollo	8 Meses
Cantidad de hombres	2 Personas
Costo	10400 Pesos
Salario medio	275
RCPX	1
RUSE	1
PDIF	1
PREX	0,87
FCIL	0,87
SCED	1

Tabla 29: Costos totales.

CAPÍTULO 4: Estudio de Factibilidad y Análisis de Resultados

4.4 – Beneficios tangibles e intangibles.

La creación de este sistema ofrece determinados beneficios para la Residencia Estudiantil de la Universidad de Cienfuegos que se evidencian en el ahorro de materiales de oficina, la gestión más rápida y confiable de la información relacionada con los estudiantes becados, los datos personales de los trabajadores y los datos del mobiliario de la residencia, se logra una mayor integridad en los datos.

4.5 – Análisis de costos y beneficios.

El desarrollo de un producto informático tiene asociado un costo, y el llevarlo a cabo o no está en dependencia de los beneficios que el mismo pudiese reportar. Los beneficios pueden ser económicos y de orden social. Este sistema como producto del presente trabajo de diploma no implica costo alguno para la Residencia Estudiantil de la Universidad de Cienfuegos, y si permite mejorar la gestión de la información de la residencia. El desarrollo del software no supone grandes gastos de recursos, ni tampoco de tiempo. Tomando como base el estudio de factibilidad realizado y el conjunto de beneficios que brinda el sistema propuesto, se determina que el mismo es factible.

4.6 – Conclusiones

En este capítulo se realizó el estudio de factibilidad correspondiente al desarrollo del proyecto, utilizando el método de estimación por puntos de función. Se realizó el análisis entre los costos y los beneficios que reporta la aplicación y se concluye que su realización es económicamente factible.

Conclusiones:

La carencia en la Residencia Estudiantil de la UCF de una herramienta que facilite a los trabajadores el manejo de forma rápida y segura la información relacionada con los datos de los estudiantes, ubicación del mobiliario dentro de la residencia, y los datos personales de los trabajadores que en ella laboran, fue el problema que dio inicio a la presente investigación, que finalizó con el desarrollo de SIREU, un sistema para la gestión de información en la residencia.

Según los objetivos trazados es posible concluir:

1. Se realizó un análisis de los procesos de la gestión de la información de la Residencia Estudiantil de Cienfuegos referente a estudiantes becados, trabajadores y mobiliarios.

Se realizó un estudio de los principales conceptos asociados al dominio del problema, y fueron seleccionadas las metodologías, herramientas y tecnologías factibles a utilizar.

Al identificarse los procesos que intervienen en el campo de acción que enmarca este trabajo, se obtuvo una mejor comprensión de los problemas existentes.

2. Se diseñó un sistema capaz de automatizar el proceso de gestión de la información en la residencia estudiantil.

Se realizó y documentó el diseño e implementación que describe la metodología RUP para este tipo de aplicaciones, utilizando varios artefactos que evidenciaban características del sistema.

3. Se implementó un sistema que cumple con las características de lo analizado y diseñado.

Finalmente se implementó el sistema propuesto para la Residencia Estudiantil, permitiendo una gestión de información más rápida, eficiente y confiable, disminuyendo los costos materiales y humanos que hasta hoy se veían implicados.

Recomendaciones

Aun cuando los objetivos trazados al inicio de este trabajo han sido logrados, pudiera convertirse en la fase inicial de un proyecto mucho más ambicioso. Por tanto hacemos las siguientes recomendaciones:

- ✓ Poner a prueba el sistema durante un período, para comprobar su desempeño.
- ✓ Profundizar en el análisis de los procesos de la gestión de información en la Residencia Estudiantil, con el objetivo de agregarle nuevas funcionalidades al sistema y así enriquecer su valor práctico y profesional.
- ✓ Extender su uso a todas las entidades de este tipo en el país.
- ✓ Conservar la idea de que el sistema debe ser desarrollado sobre la base de la multiplataforma lo que permitirá adaptarse a las nuevas tecnologías de la informática moderna.

Referencias Bibliográficas

- [1] Enciclopedia Universal Ilustrada Europeo Americana.

- [2] Phil Bartle, “Información para la gestión y gestión de la información,” Mar. 2009; <http://www.scn.org/mpfc/modules/mon-miss.htm>.

- [3] Ricardo Batista Moliner, “La dispensarización: una vía para la evaluación del proceso salud-enfermedad , Mar. 2009; http://www.bvs.sld.cu/revistas/mgi/vol17_2_01/mgi01201.htm.

- [4] Xavier Ferrá Grau, “Desarrollo orientado a objetos con UML,” Mar. 2008; <http://www.clikear.com/manuales/uml/introduccion.asp>.

- [5] Zaguero, “Administración de Proyectos de Software. Trabajo práctico grupal: Ciclos de Vida de proyectos. Grupo 4. Ciclo vida RUP , Mar. 2008; <http://www.zohowriter.com/public/27201/38205>.

- [6] Oscar Muñoz, “Arquitectura de aplicaciones Web. Conferencia de Seminarios Especiales I, 2004.

- [7] Raúl Cordero Carrasco, “Introducción al diseño y a la programación orientada a objetos,” Mar. 2009; <http://www.nielsoft.com/Seminario/3capas/introduccion.ppt>.

- [8] Joaquín Rodríguez, “Tutorial de HTML,” Feb. 2009; <http://html.programacion.net>.

- [9] Alejandro Valero, “Curso Práctico para aprender a crear páginas Web,” Feb. 2009; <http://fresno.cnice.mecd.es/~avaler3/presenta.htm>.

- [10] Ene. 2009; <http://es.selfhtml.org/introduccion/tecnologias/xml.htm>.

- [11] Stefan Munz, "XML y derivados de XML," Ene. 2009;
<http://es.selfhtml.org/introduccion/tecnologias/webservices.htm>.
- [12] Alex Morales, "Conceptos básicos de ASP," Ene. 2009;
http://www.programacion.com/asp/tutorial/asp_basics/
- [13] Stefan Munz, "Tecnologías Web/ PHP," Ene. 2008;
<http://es.selfhtml.org/introduccion/tecnologias/php.htm>
- [14] Janet Rodríguez Febles, "Sistema Informático para la Gestión Integral de Comedores (SISCOMED), 2005.
- [15] "Tutoriales sobre Apache," Feb. 2008; <http://www.naninet.com.ar/apache>.
- [16] Rosa María Matos, "Introducción al trabajo con Base de Datos. Asignatura de Sistemas de Gestión de Base de Datos," 2004.
- [17] Dailyn Sosa López y Kadir Hector Ortiz, "Desarrollo de un prototipo de intranet para la facultad de un Centro de Educación Superior Módulo docente," 2006.
- [18] Anaisa Hernández González, "Modelo del Negocio: material para uso docente," 2005.
- [19] James et al., El Proceso Unificado de Desarrollo de Software, La Habana: Félix Varela, 2004.

Bibliografía

Jacobson, J.Rumbaugh, y G.Booch, *El Lenguaje Unificado de Modelado*, Madrid: Addison Wesley, 2000.

Jacobson, J.Rumbaugh, y G.Booch, *El proceso unificado del desarrollo del software*, Madrid: Pearson Education. S.A. , 2000.

Abraham Silberschatz, Henry F. Korth, y S. Sudarshan, *Fundamentos de Bases de Datos*, McGRAW-HILL/INTERAMERICANA DE ESPAÑA, .

Timoty Budd, *Introducción a la programación orientada a objetos*.

“Metodologías De Desarrollo De Software”;
http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html.

“PHP en castellano”; <http://www.programacion.com/php/>.

“PostgreSQL: The world's most advanced open source database”; <http://www.postgresql.org/>.

Craig Larman, *UML y Patrones*, Prentice Hall Iberoamericana, 1999.

“Vivir en la Sociedad de la Información”; <http://www.oei.es/revistactsi/numero1/trejo.htm>.

Glosario de términos

ASP: Páginas de Servidor Activas (siglas en Inglés)

CSS: Página de Estilos de Código (siglas en Inglés)

DOM: Modelo de Objetos del Documento (siglas en Inglés)

IE: Internet Explorer (Navegador Web)

MES: Ministerio de Educación Superior

NN: Netscape Navigator (Navegador Web)

PHP: Lenguaje Preprocesado de Hipertexto (siglas en Inglés)

RUP: Proceso Unificado de Desarrollo (Metodología de desarrollo de software)

SGBD: Sistema de Gestión de Base de Datos.

SOAP: Simple Object Access Protocol (Protocolo de Acceso Simple a Objetos).

SACBEXT: Sistema Automatizado de Control de Becarios Extranjeros.

SGBD: Sistemas de Gestión de Base de Datos.

SIBEX: Sistema de Gestión de la información de Estudiantes Becarios Extranjeros.

SIREU: Sistema Informático para la Residencia Estudiantil Universitaria.

TIC: Tecnologías de la Informática y la Comunicaciones

UCF: Universidad de Cienfuegos

UML: Lenguaje Unificado de Modelado (siglas en Inglés)

W3C: Consorcio Internacional de la World Wide Web (www)

XML: Lenguaje de Marcado Ampliable o Extensible (siglas en Inglés)

Anexos

Anexos A: Prototipos

Anexo A.1 Iniciar Sesión

Autentificarse

Usuario

Contraseña

Anexo A.2 Cambiar Contraseña

CAMBIAR CONTRASEÑA

Cambio de Contraseña

Contraseña nueva

Confirmar contraseña

Anexo A.3 Gestionar Usuario

INSERTAR USUARIO

Insertar Usuario

Nombre y Apellidos

C.Identity

Login

Contraseña

Confirmar Contraseña

Tipo de Usuario

LISTA DE USUARIOS

C.Identidad	Login	Nivel de Acceso	Editar
65030514583	Especialista	1	Eliminar Modificar
78030514583	Angel	2	Eliminar Modificar

MODIFICAR USUARIO

Modificar Usuario

Nombre y Apellidos

C.Identidad

Usuario

Contraseña

Confirmar Contraseña

Tipo de Usuario

Anexo A.4 Visualizar Estudiante

BUSCAR ESTUDIANTE

CI
 NOMBRE
 PAÍS

Estudiante	País	Información
Charles James Hudson	Jamaica	Ver Ficha Editar
Anthony Baker Phillip	Jamaica	Ver Ficha Editar
Karalaise Jhonson	Jamaica	Ver Ficha Editar
Kadian Manique Weir	Jamaica	Ver Ficha Editar

Anexo A.5 Gestionar Cuarto

ASIGNAR CUARTO A ESTUDIANTE

Seleccione el número de cuarto...

Lista de Cuartos

Asignar

FICHA DEL ESTUDIANTES

CI:	85030514583
Nombre:	Yoisel
Primer Apellidos:	Martinez
Segundo Apellido:	Gomez
Fecha de Nacimiento:	1985-03-05
Provincia:	
Municipio:	

La página en http://localhost dice:

Esta seguro que desea eliminar el estudiante del cuarto, se borrará toda la información editada.

Aceptar Cancelar

Tipo de Estudiante:	Becado Nacional
Facultad:	Informática
Carrera:	Ingeniería Informática
Raza:	B
Cuarto:	231
Número de Colchón:	

Eliminar Modificar

Insertar

MODIFICAR CUARTO A ESTUDIANTE

Seleccione el número de cuarto...

Lista de Cuartos

Mover

Anexo A.6 Gestionar Colchón

ASIGNAR COLCHÓN AL ESTUDIANTE

--Yoisel Martinez Gomez--

Fecha de Entrega 

Número del Colchón

FICHA DEL ESTUDIANTES

CI:	85030514583
Nombre:	Yoisel
Primer Apellidos:	Martinez
Segundo Apellido:	Gomez
Fecha de Nacimiento:	1985-03-05
Provincia:	
Municipio:	
Dirección:	
País:	
Sexo:	
Teléfono:	
Estado Civil:	
Email:	
Tipo de Estudiante:	Becado Nacional
Facultad:	Informática
Carrera:	Ingeniería Informática
Raza:	B
Cuarto:	231
Número de Colchón:	E-690

[Eliminar](#) [Modificar](#)
[Eliminar](#) [Modificar](#)

La página en http://localhost dice: ✖

 Esta seguro que desea retirarle el colchón al estudinte

MODIFICAR COLCHÓN AL ESTUDIANTE

--Yoisel Martinez Gomez--

Fecha de Entrega 

Número del Colchón

Anexo A.7 Insertar Información Adicional a Estudiantes.

EVALUACIÓN DEL ESTUDIANTE

--Yoisel Martinez Gomez--

Fecha:  12

Evaluación:

[Ir al Editar](#)

DIAGNÓSTICO PSICOLÓGICO DEL ESTUDIANTE

--Yoisel Martinez Gomez--

Fecha:  12

Descripción:

[Ir al Editar](#)

SANCIONES DEL ESTUDIANTE

--Yoisel Martinez Gomez--

Fecha 12

Motivo de la Sanción

Tipo de Sanción

[Ir al Editar](#)

DISPENSARIZACIÓN DEL ESTUDIANTE

--Yoisel Martinez Gomez--

Fecha 12

Enfermedad

Medicamento

[Ir al Editar](#)

Anexo A.8 Visualizar Información Adicional a Estudiantes.

EVALUACIÓN DEL ESTUDIANTE

--Yoisel Martinez Gomez--

Fecha	Descripción
-------	-------------

[Ir al Editar](#)

DIAGNÓSTICO PSICOLÓGICO DEL ESTUDIANTE

--Yoisel Martinez Gomez--

Fecha	Descripción
-------	-------------

Ir al Editor

SANCIONES DEL ESTUDIANTE

--Yoisel Martinez Gomez--

Fecha	Motivo de la Sanción	Tipo de Sanción
-------	----------------------	-----------------

Ir al Editor

DISPENSARIZACIÓN DEL ESTUDIANTE

--Yoisel Martinez Gomez--

Fecha	Enfermedad	Medicamento
-------	------------	-------------

Ir al Editor

Anexo A.9 Visualizar Mobiliario

BUSCAR MOBILIARIO

Seleccione el criterio de búsqueda...

- Seleccionar todo...
- Cuarto
- Número de Inventario

Anexo A.10 Gestionar Mobiliario

INSERTAR MOBILIARIO

Insertar Mobiliario...

Número de Inventario

Tipo de Mobiliario

Estado Físico

Número de Cuarto

LISTAR MOBILIARIO

Inventario	Tipo	Estado Físico	Cuarto	Editar
41165	silla	Bueno	231	Eliminar Modificar
42156	mesa	Bueno	231	Eliminar Modificar
				Eliminar Modificar
				Eliminar Modificar
				Eliminar Modificar
				Eliminar Modificar

La página en http://localhost dice:

Esta seguro que desea eliminar los datos del mobiliario.

EDITAR MOBILIARIO

Editar Mobiliario...

Número de Inventario

Tipo de Mobiliario

Estado Físico

Número de Cuarto

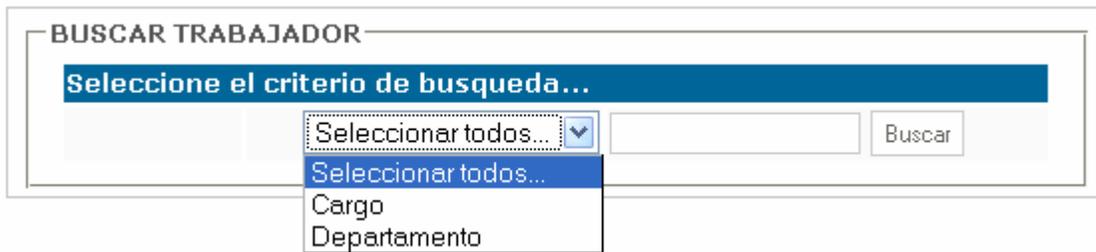
Anexo A.11 Visualizar Trabajador

BUSCAR TRABAJADOR

Seleccione el criterio de búsqueda...

Seleccionar todos...
Carga
Departamento

Buscar



Anexo A.12 Gestionar Trabajador

INSERTAR TRABAJADOR

Número de CI:

Nombre:

Primer Apellido:

Segundo Apellido:

Departamento:

Cargo:

Dirección:

Edad:

Religión:

Teléfono:

Estado Civil:

Cantidad de Hijos:

Sexo:

Raza:

Enviar Datos



EDITAR TRABAJADOR

Número de CI:	75030514583
Nombre:	Mary
Primer Apellido:	Martinez
Segundo Apellido:	Gomez
Departamento:	Administrativo
Cargo:	Tecnico
Dirección:	cienfuegos
Edad:	45
Religión:	Ateo
Teléfono:	584789
Estado Civil:	Casado <input type="button" value="v"/>
Cantidad de Hijos:	2
Sexo:	Femenino <input type="button" value="v"/>
Raza:	Blanca <input type="button" value="v"/>

LISTAR TRABAJADOR

Nombre	Departamento	Cargo	Editar
Mary	Administrativo	Tecnico	Eliminar Modificar Permisos
Angel	Administrativo	Administrador	Eliminar Modificar Permisos

La página en http://localhost dice: ✖

 Esta seguro que desea eliminar los datos del trabajador.

Anexo A.13 Gestionar Edificios

INSERTAR EDIFICIO

Insertar Edificio

Número del Edificio

Cantidad de Cuartos

Insertar

MODIFICAR EDIFICIO

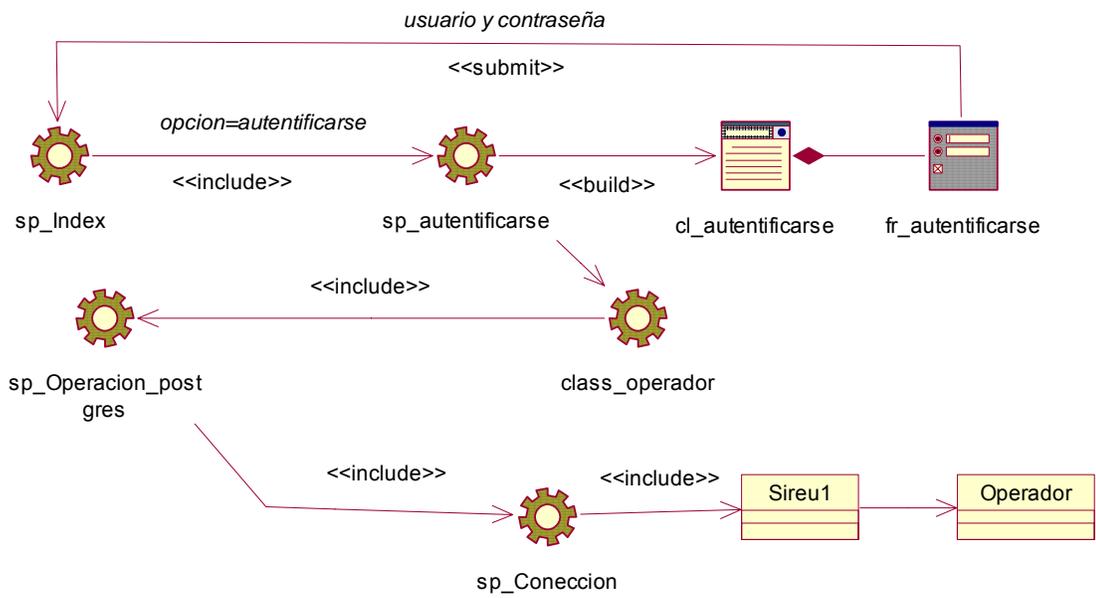
Modificar Edificio

Número del Edificio

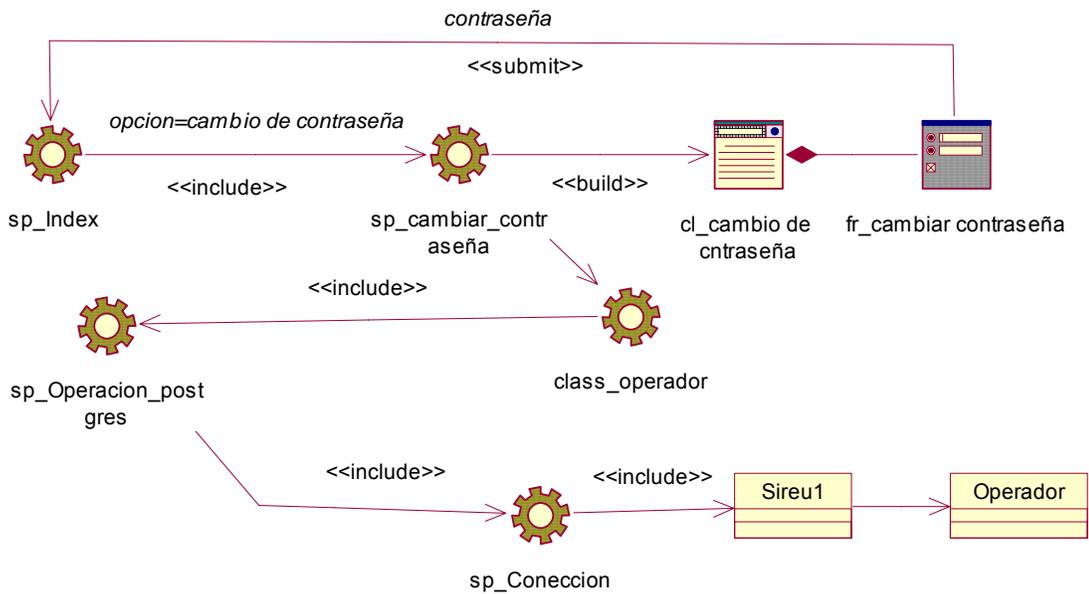
Cantidad de Cuartos

Actualizar

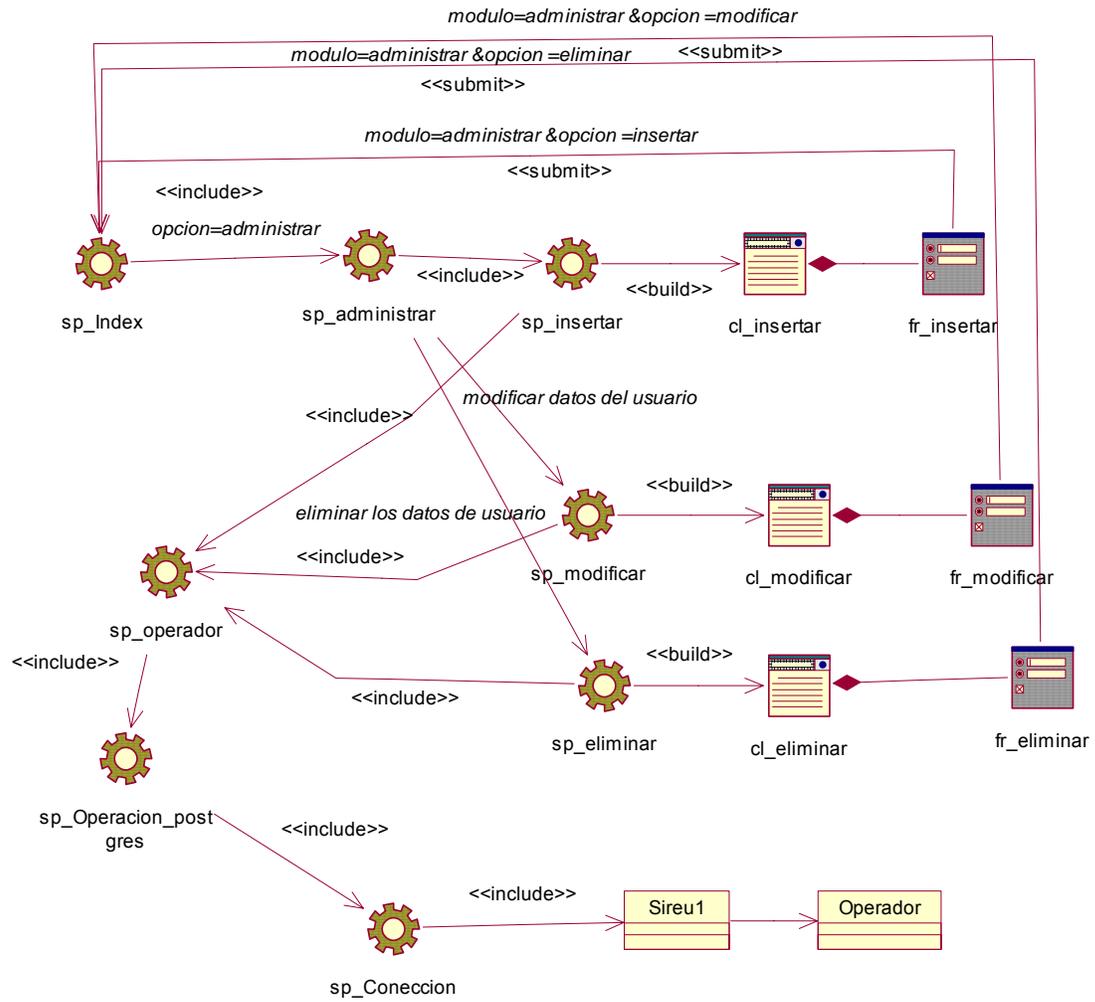
Anexo B.1 Iniciar Sesión



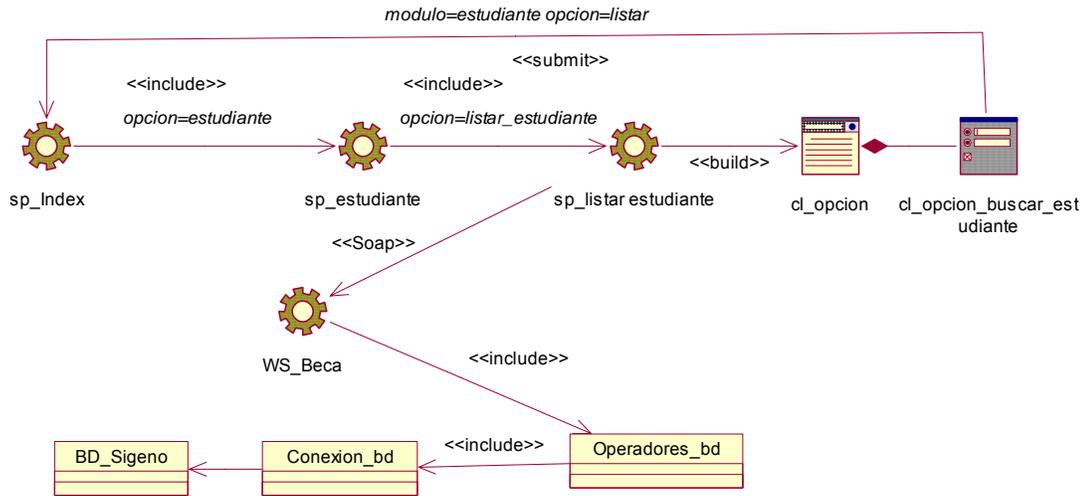
Anexo B.2 Cambiar Contraseña



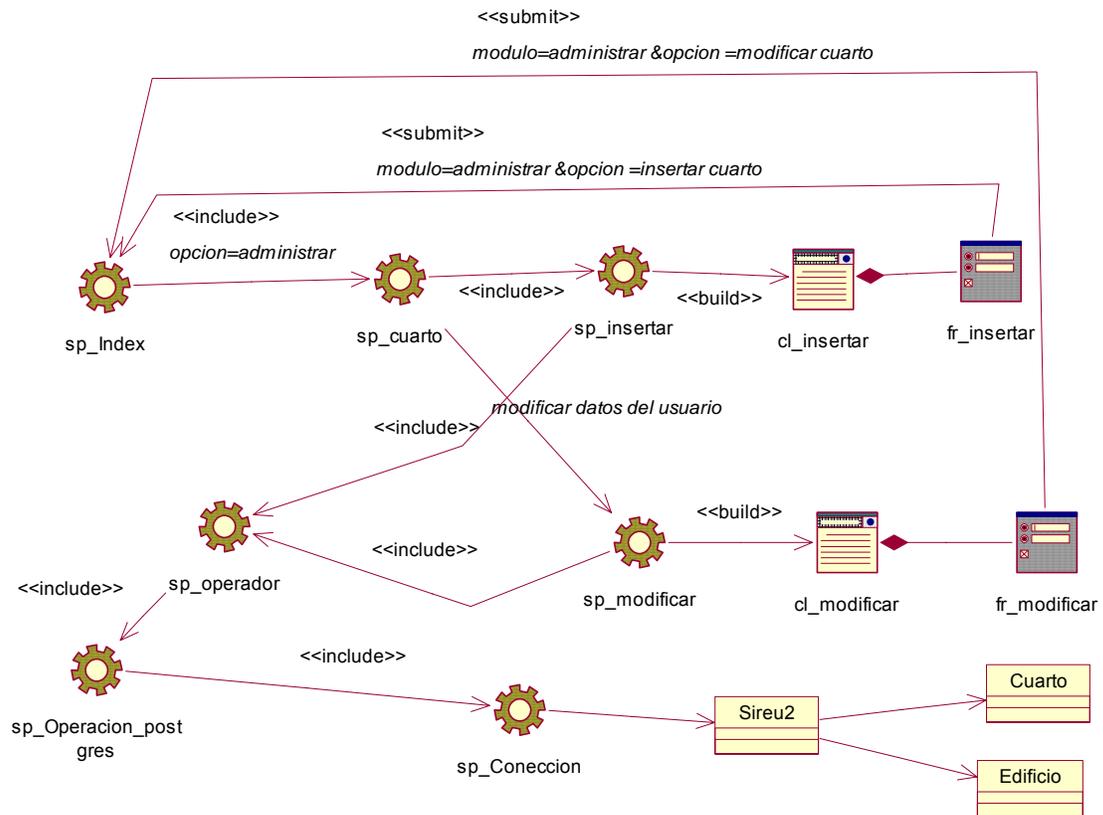
Anexo B.3 Gestionar Usuario.



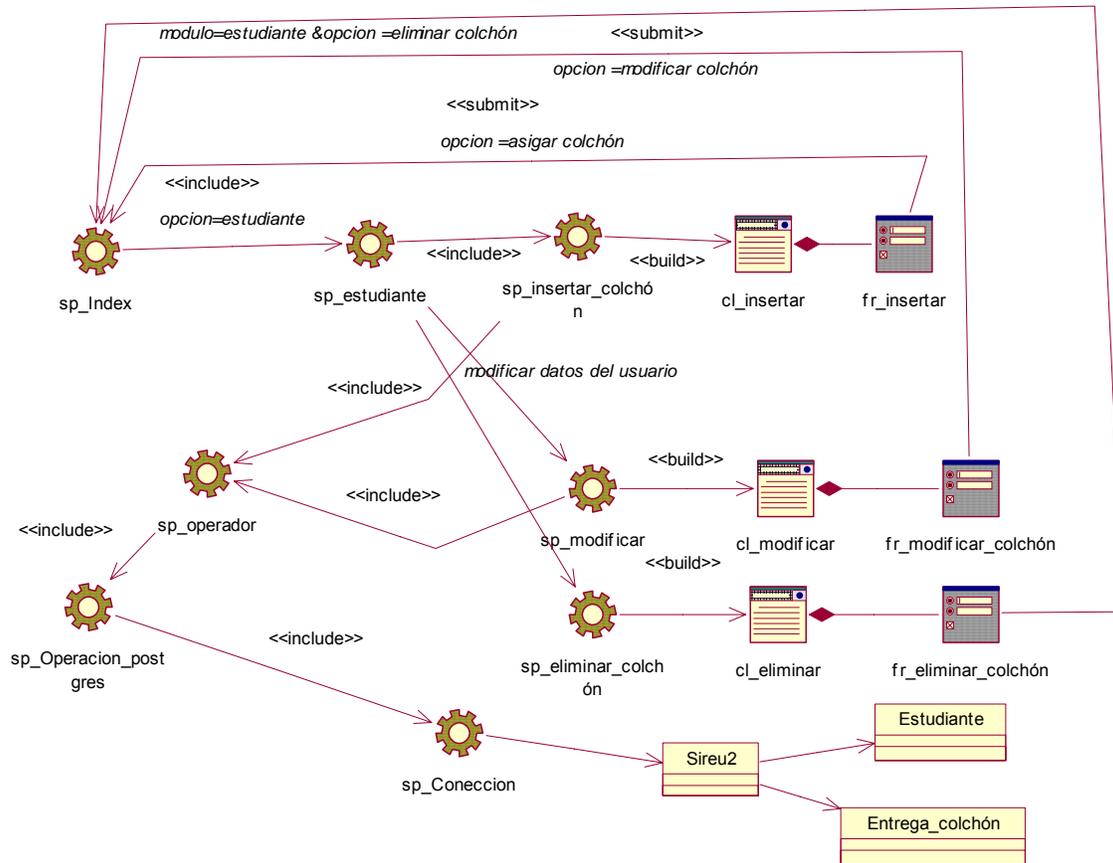
Anexo B.4 Visualizar Estudiante



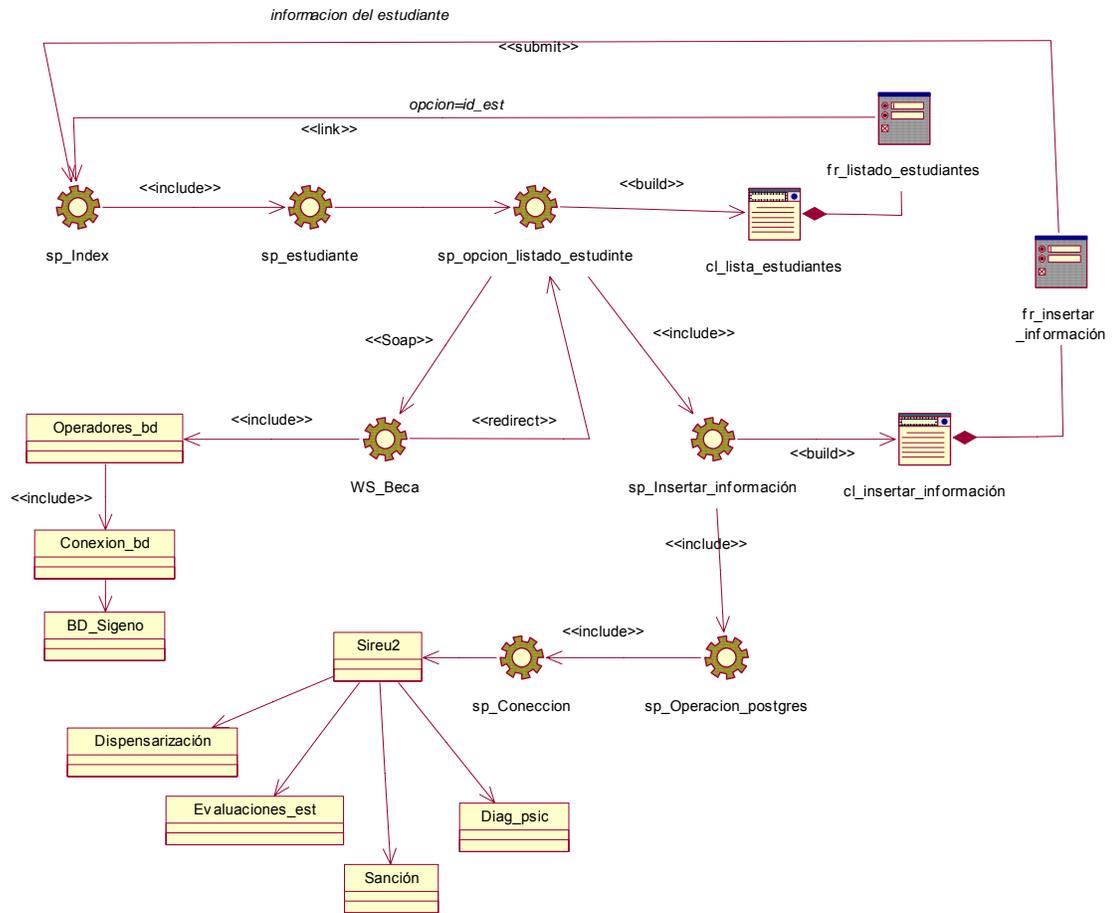
Anexo B.5 Gestionar Cuarto



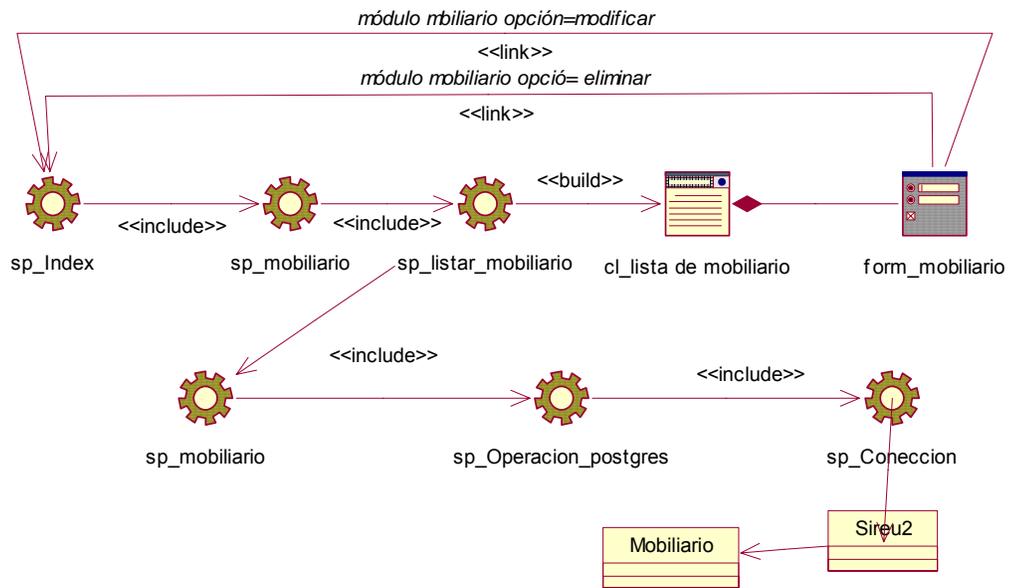
Anexo B.6: Gestionar Colchón



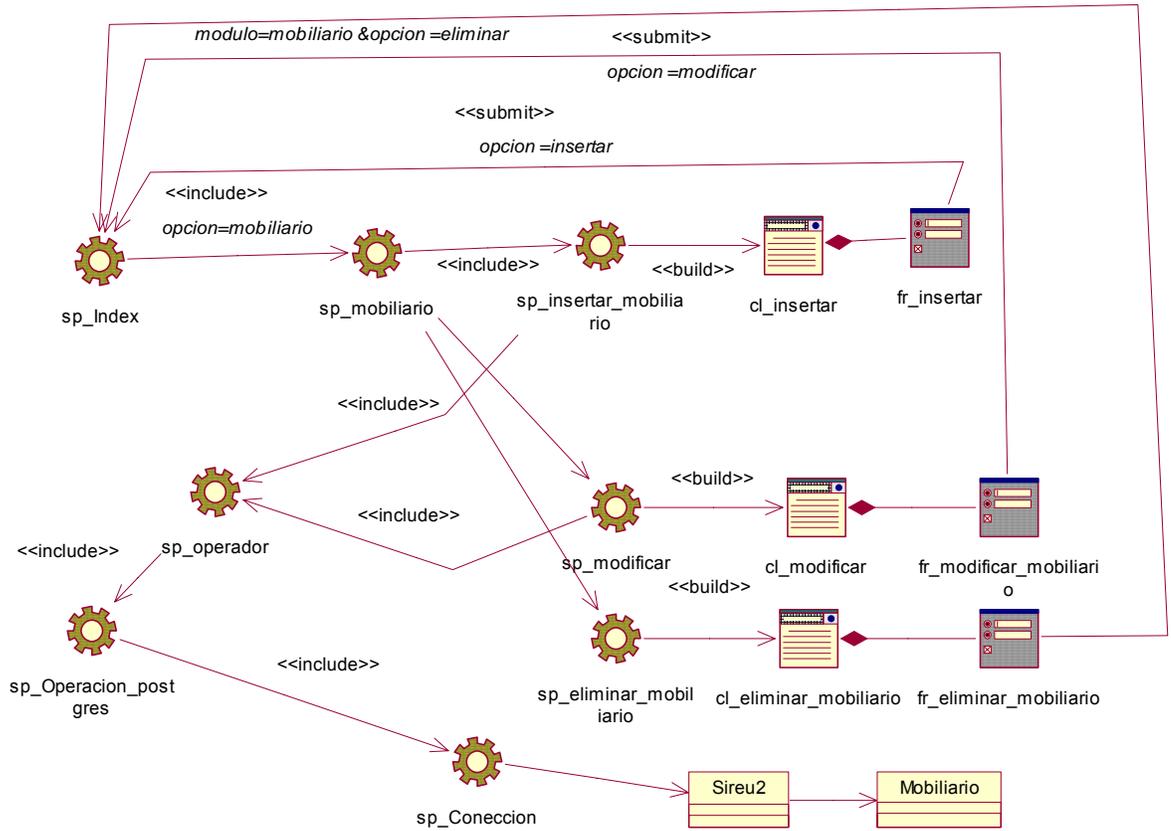
Anexo B.7 Insertar Información Adicional del Estudiante



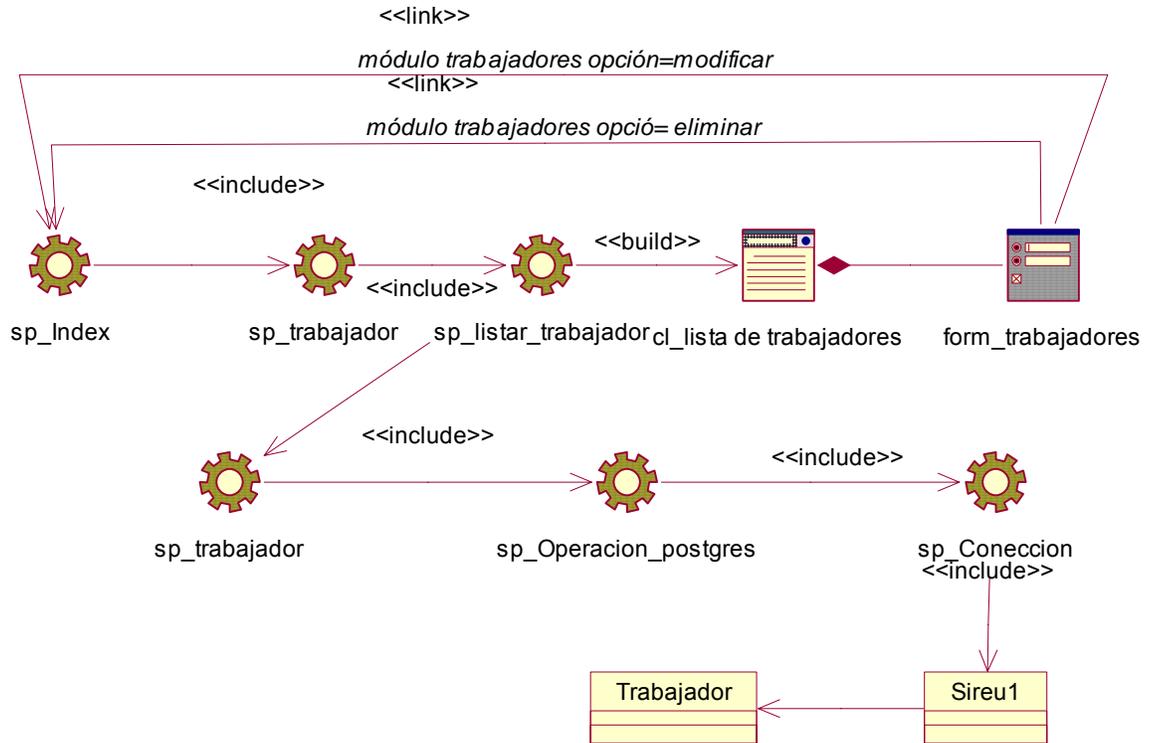
Anexo B. 9 Visualizar mobiliario



Anexo B.10 Gestión del Mobiliario



Anexo B.11 Visualizar trabajadores



Anexo 12 Gestionar Trabajador

