

005.12
Carl
M
TD-0161-06

TD-0161-06



INSTITUTO SUPERIOR POLITÉCNICO
"JOSÉ ANTONIO ECHEVERRÍA"
FACULTAD DE INGENIERÍA INDUSTRIAL
CENTRO DE ESTUDIOS DE INGENIERÍA DE SISTEMAS (CEIS)

MODULO DE GESTION PARA LA RESIDENCIA DE LA FACULTAD TRES EN LA UNIVERSIDAD DE LAS CIENCIAS INFORMATICAS (UCI)

Versión 1.0

Trabajo de Diploma para optar por el Título de Ingeniería Informática

Autor: Rolando Sacher Camacho Pupo

Tutor: Lic. Ridosbey Milian Iglesias

CUJAE, junio del 2006
Ciudad de la Habana

A mis padres y a mi hermano, por lo que representan para mí...

A mi tía Ruth y Elio por cargar conmigo

A mi abuela y abuelo, que ya no está

A mi novia por poner las tildes...

Y a mi esfuerzo.

Rolo

Agradecimientos

Antes que nada dejar bien claro que siempre quedarán algunos fuera, por mis descuidos, no obstante no se preocupen, esto es solo un papel, yo siempre estaré ahí para agradecerles de frente lo que quizás olvide aquí. *

- A mi madre, por ser parte de mí, por estar tan dentro, por cantarme: "... un Unicornio Azul ayer se me perdió...". Gracias por soportarme cada minuto que respiro, por aguantarme cada vez que necesito levantar mi autoestima, por reírte incansablemente... VIEJA, NO TE VAYAS!!!
- A mi padre, por ser mi guía profesional y personal, por ubicarme cada vez que me disparato, incluso por cada vez que lo hago bien e increíblemente "necesito" ser reubicado. Por reírte y contarme tus secretos de guerra. Por creer en mí y exigirme tanto. Gracias mi viejo.
- A mi hermano, que esta de más, recordarle que sin sus peleas yo hoy no fuese ni la mitad de lo que soy. Por aun quererme como antes. Por acordarse solamente de mí, cuando su "mente visitó otros lugares". Por darme ese nombre que siempre se dice en casa: *Nani*.
- A mi tía Ruth, por creer ser mi madre y por jugar tan bien ese papel.
- A Elio, por ser tan resabioso y aun así, mostrar muestras de cariño en estos 5 años.
- A mi abuela, por creer aun que hoy puedo ser mejor. Por predestinarme "ser alguien grande". Por decirme a diario que puedo casarme.
- A mi abuelo, que debe andar por ahí tratando de ver si sigo al pie de la letra todo lo que me explicó y aconsejó durante años.
- A Marina y Montero, mis vecinos.
- A mis amigos de Sancti Spíritus: Alain López ("La Kabia"), por hacerme creer que aun queda gente linda y que de ellos es el "general" de la especie; a Roberto Rodríguez ("Pata de Plomo" alias el Masacote) por brindar un espacio cuando más lo necesite en años anteriores; a Kemell Hernández por su Cake de cumpleaños, sus muecas, su guitarra; a Roberto (El Robe) por

Agradecimientos

aun escribirme y por hacerme participe de su amistad desde tiempos remotos; Liz María Zamora Pardillo, gracias por nacer casi casi conmigo y mantenerte todos estos años “junto a mi”; Arturo Arias, mi guía inicial en esta decisión de graduarme de Ingeniero Informático; Duniesky (El Duni), por ser el nuevo del grupo y darnos ejemplo de amistad. De igual forma créanse merecedores de todos los alagos del mundo, por formarme como soy: Yosel López (El Yusi), Shamil Estrada (El Negrón), Yoangel Rodríguez (El Topo), Héctico, Rusellita, María del Carmen (Carmencita), Maily Hernández, Dayana, Maximiliano Trujillo (Loco antiguo) y a todos esos otros, que he conocido allí.

- A estos compañeros habaneros les agradezco ayudarme con mi carrera en estos 5 años y por casi siempre estar cuando me era necesario: Noslen González (te debo bastante conocimiento), Henrik Santander (a ti parte de mis notas), Yuset Calzadilla (reconozco que me ayudaste un mundo), Yerquis Trujillo (de Sancti Spíritus es verdad, pero tus llantos los escuche aquí), Billy Yero (el hombre más largo del mundo), Yasset Pérez (ejemplo de loco rosado), Yaniel Álvarez (hazme el favor, no llegues a viejo), Misael Alfaro (no te preocupes, ahora vas pa’ la UCI), Rolando Pérez Pinto (F E O), Norges González Gacita (TRAIDOR!!!), Emilio Peña (jejeje, tu sabes...), Julio Cesar Camps (por tutorarme dentro de Microsoft), Adrián Capdevila (mil gracias por muchas cosas), Dunia Trujillo y su pareja Yandell (valen mil pesos), Jorge Michel Díaz (el Dracfov, tu apodo lo dice todo ingenioso amigo) y a toda la retreta restante de gente que sabe admiro dentro de la CUJAE.
- A Oriallys Sánchez Fidalgo, por crear en mi la seguridad que me ha venido faltando desde hace años; por hacerme escribir su nombre en esta hoja; por lograr mi sonrisa y hacerme repensar que vale la pena caminar más por el sendero que me “enseñaste”.
- A XEON, por aguantarme tanto tiempo a su lado, y ser una de las PCs más estables del mundo.
- A mi aguante ante las adversidades de becado: hambre, soledad, falta de higiene, viajes interprovinciales, desprecio regional, entre tantas otras.

Resumen

El presente trabajo pretende un recorrido por el proceso ingenieril de elaboración del "Módulo de Gestión para la Residencia" (Módulo), el mismo comprende diferentes aspectos:

Estudio del problema a resolver en un dominio de Residencia con características específicas para un sistema estudiantil emergente y de veloz crecimiento. Lo que implica el estudio del ciclo de vida de cada uno de los elementos que forman parte de la Residencia de forma que puedan ser totalmente gestionables.

Examen detallado de la solución a la problemática, donde se arriban a un conjunto de conclusiones, como la arquitectura de la base de datos que soportará la información; las características funcionales que deberá poseer la aplicación para que sea efectiva su explotación dentro de un sistema integrado de gestión, agregándosele la capacidad para insertarse dentro de este medio con un diseño de base de datos capaz de adaptarse al del sistema, sin entorpecer su autonomía.

Además se abordarán aspectos relacionados con las herramientas computacionales empleadas en la construcción del Módulo, en aspectos como su implementación y gestión de su base de datos.

Este proceso será guiado por una metodología de desarrollo cuyo hilo conductor lo ocupa el Proceso Unificado de Software (RUP), con los distintos artefactos implicados, expresados a través del Lenguaje de Modelo Unificado (UML).

En la actualidad no se conoce ningún software que automatice la gestión de la Residencia íntegramente dentro de la UCI. Por lo que este Módulo tendrá un impacto significativo para las tendencias de automatización de la información y podrá asumirse su uso en diferentes facultades, gracias a su modelación.

INTRODUCCIÓN	1
CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA.....	8
1.1 INTRODUCCIÓN	8
1.2 OBJETO DE ESTUDIO.....	9
1.2.1 <i>Objetivos estratégicos de la organización.....</i>	<i>9</i>
1.2.2 <i>Flujo Actual de los Procesos</i>	<i>10</i>
1.2.3 <i>Análisis Crítico de la Ejecución de los Procesos</i>	<i>11</i>
1.3 PROCESOS OBJETOS DE AUTOMATIZACIÓN	12
1.4 SISTEMAS AUTOMATIZADOS EXISTENTES VINCULADOS AL CAMPO DE ACCIÓN.....	14
1.5 FUNDAMENTACIÓN DE LOS OBJETIVOS	16
1.6 TENDENCIAS Y TECNOLOGÍAS ACTUALES	17
1.6.1 <i>Sistemas de Gestión</i>	<i>18</i>
1.6.1.1 Funcionalidades de los Sistemas de Gestión.....	19
1.6.1.2 Beneficios del Uso de Sistemas de Gestión.....	21
1.6.2 <i>Fundamentación de la Metodología Utilizada</i>	<i>22</i>
1.6.2.1 Metodología RUP	22
1.6.2.2 Notación UML	23
1.6.3 <i>Fundamentación del Lenguaje, Gestor de Base de Datos y Plataforma de Desarrollo</i>	<i>25</i>
1.6.3.1 ASP.NET.....	26
1.6.3.2 Visual Studio .NET	28
1.6.3.3 SQL Server 2000.....	31
1.7 CONCLUSIONES.....	33
CAPÍTULO II: MODELO DEL NEGOCIO	34
2.1 INTRODUCCIÓN.....	34
2.2 MODELO DEL NEGOCIO ACTUAL.....	35
2.3 REGLAS DEL NEGOCIO A CONSIDERAR	35
2.4 ACTORES DEL NEGOCIO	36
2.5 TRABAJADORES DEL NEGOCIO.....	36
2.6 DIAGRAMA DE CASO DE USO DEL NEGOCIO.....	37
2.7 CASOS DE USO DEL NEGOCIO	38
2.7.1 <i>Caso de uso: Control Residentes Estudiantiles</i>	<i>38</i>
2.7.2 <i>Caso de uso: Control de Apartamentos.....</i>	<i>40</i>
2.8 MODELO DE OBJETOS	42
2.9 <i>Conclusiones.....</i>	<i>43</i>

CAPÍTULO III: REQUISITOS Y DESCRIPCIÓN DEL SISTEMA PROPUESTO	44
3.1 INTRODUCCIÓN.....	44
3.2 <i>Definición de Requisitos no Funcionales</i>	44
3.2.1 <i>Apariencia o Interfaz Externa</i>	45
3.2.2 <i>Usabilidad</i>	45
3.2.3 <i>Rendimiento</i>	45
3.2.4 <i>Soporte</i>	46
3.2.5 <i>Portabilidad</i>	46
3.2.6 <i>Seguridad</i>	47
3.2.7 <i>Confiabilidad</i>	47
3.2.8 <i>Ayuda y Documentación en Línea</i>	48
3.2.9 <i>Requerimientos de Software</i>	48
3.2.10 <i>Requerimientos de Hardware</i>	48
3.3 CONCEPCIÓN GENERAL DEL SISTEMA	48
3.4 ACTORES DEL SISTEMA A AUTOMATIZAR.....	51
3.5 MODELO DE CASOS DE USO DEL SISTEMA	52
3.5.1 <i>Paquete: Control de Disponibilidades</i>	54
3.5.2 <i>Paquete: Control de Guardias</i>	54
3.5.3 <i>Paquete: Reportes</i>	55
3.5.4 <i>Paquete: Control de Residentes</i>	56
3.5.5 <i>Paquete: Seguridad</i>	56
3.6 DESCRIPCIÓN DE LOS CASOS DE USO DEL SISTEMA.....	57
3.7 CONCLUSIONES.....	72
CAPÍTULO IV: DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA	73
4.1 INTRODUCCIÓN.....	73
4.2 DIAGRAMA DE CLASES DEL DISEÑO	74
4.2.1 <i>Paquete: Seguridad</i>	74
4.2.2 <i>Paquete: Control de Disponibilidades</i>	75
4.2.3 <i>Paquete: Control de Residentes</i>	76
4.2.4 <i>Paquete: Reportes</i>	77
4.2.5 <i>Paquete: Control de Guardias</i>	78
4.2.6 CONSIDERACIONES POR CAPAS PARA LA APLICACIÓN.....	79
4.3 PRINCIPIOS DE DISEÑO	79
4.3.1 <i>Interfaz de Usuario</i>	79

Introducción

La Universidad de las Ciencias Informáticas (UCI) cuenta como característica el estar compuesta por varias facultades docentes que a su vez están encargadas del manejo de cada uno de los estudiantes que se encuentran asociados a su plantilla docente. Estos mismos cuentan con un espacio dentro de la Residencia de la UCI que también es controlada por la Facultad que lo atiende docentemente.

Dentro de la institución (UCI), existen algunas reglamentaciones que deben ser acatadas en diferentes ramas, como la docencia y la residencia. Ambas ramas tienen un gran peso para la evaluación del estudiante anualmente, para que pueda seguir sus estudios el próximo año en la UCI. Específicamente en el área de la Residencia se lleva un control, basado en una serie de parámetros, sobre los estudiantes, para poder determinar su situación dentro de la misma. De esta forma la Facultad tiene el control de los miembros estudiantiles de la Residencia.

La Residencia de la Facultad está estructurada por una serie de edificios de apartamentos, que son asignados por la UCI, los cuáles tienden a variar anualmente en su ubicación. De esta forma a la Facultad se le dificulta el control de cada apartamento, al inicio de cada curso.

Cada uno de los apartamentos del edificio residencial, cuenta con varias habitaciones: dormitorios, baño, cocina, sala, comedor, etc. En cada habitación existen una serie de recursos, para hacerlas habitables y brindar algunas comodidades en el transcurso del curso escolar a sus integrantes. Estos recursos son controlados por la Facultad, su estado físico y ubicación son necesarios para conservar la integridad de cada una de las habitaciones además el conocimiento de estos pudiese influir en la evaluación de cada uno de los residentes (estudiantes) que conviven en el apartamento.

Si se desea lograr la “salud” de los medios de la Residencia, la Facultad debe controlar que cada residente mantenga la limpieza y cuidado de su apartamento; asignarle una “Cuartería”, para el cuidado de parte de su edificio en horario diurno,

así como una “guardia estudiantil”, con el mismo sentido de la “Cuartelería” pero en horario nocturno.

Para lograr el control de la Residencia, por parte de la Facultad, se necesitan fundamentalmente, la información relacionada a dos aspectos fundamentales: los residentes y apartamentos. Dicha información es de “difícil acceso” para el Vise-Decano que atiende la rama, pues regularmente se necesita acceder a diferentes aspectos evaluativos que evidencian el comportamiento de cada residente o el estado de los apartamentos. Actualmente el Vise-Decano debe “esperar” por los diferentes informes de las Instructoras, encargadas del funcionamiento de cada edificio de la Residencia, en aras de lograr el control deseado.

El procedimiento de confección de los informes, anteriormente referenciados, se cataloga como lento, al igual que su llegada al Vise-Decano, lo cual implica un nivel de control medio, pues la información puede variar en demasía, entre la entrega de un informe y otro.

Existen disímiles tipos de informes: sobre el estado de los apartamentos, sobre el inventario de los medios básicos ubicados en apartamentos, sobre las evaluaciones de los residentes estudiantiles en tareas referentes al apartamento o a la realización de las diferentes guardias, entre otros.

La problemática más grande de estos informes radica en poder generarlos de manera rápida y eficiente. La instructora es la responsable mayoritaria de la confección de estos informes.

Para poder controlar los apartamentos se necesita que la Instructora pase por cada uno de ellos y haga un informe sobre el estado físico actual de: las habitaciones y medios básicos para poder decidir si es habitable, de la misma manera debe informar sobre la cantidad de residentes actuales de cada apartamento, así como la disponibilidad de estos para aceptar otros residentes, el sexo de los habitantes y el inventario de todo lo contenido en el apartamento.

Para el control de los residentes estudiantiles es necesario que la Instructora emita una evaluación sobre diferentes aspectos que rigen el comportamiento de los mismos:

asistencia a las diferentes guardias programadas por la Residencia y la limpieza de los apartamentos. Estas evaluaciones posteriormente deben ser revisadas por la Psicopedagoga para que la misma, emita su propia evaluación, lo que en general contribuiría a la construcción de un informe general para cada residente realizado por dos miembros del personal de la Residencia. Teniendo el segundo que esperar por la actividad del primero y además revisar que ningún residente se encuentre en desacuerdo con la evaluación emitida sobre él, lo que implicaría una re-evaluación.

El éxito del control anterior acarrea que dos situaciones problemáticas funcionen correctamente. Pues resulta difícil confeccionar los calendarios de guardias y de limpieza de cuartos. Cada apartamento necesita que se definan bien los estudiantes que deben limpiar diariamente y en caso de que haya más ó menos residentes que los días de la semana es necesario definir con exactitud como se distribuirán a esos residentes "sobrantes". Igualmente las guardias necesitan cada una un calendario con las características específicas de si mismas, para distribuir a cada uno de los residentes. Una gran dificultad es que deben saberse la ubicación de cada uno de los residentes y tenerse en cuenta los disímiles problemas que puedan surgir que atenten contra la realización de cada una de estas, así como la redistribución de residentes, acarreada por una de estas dificultades. Lo anterior es muy complicado pues todo se encuentra sobre material físico de oficina, entiéndase: hojas, folios, etc.

El manejo de estos materiales de oficina es muy lento y difícil por la carencia de flexibilidad de los medios a cambios, con el agravante de tener que confeccionar todo desde el inicio ante varios cambios por el deterioro de los mismos.

Comentados las formas de control principales, se destaca que ocurren situaciones problemáticas similares en la gestión de los residentes, en la estructura de ubicaciones y disponibilidades. Pues cada inserción, modificación o eliminación de cualquiera de estos aspectos en la Residencia, se dificulta por el soporte físico en que se encuentra la información, siendo estos mismos, la base para cada uno de los procesos de control descritos anteriormente.

La forma en que se controla la Residencia es hoy por hoy, una vía retrograda, en cuanto a manejo y mantenimiento de la información, pues carece de todo tipo de

automatización y por ende recae sobre los factores de poca velocidad en el flujo de información y en el manejo de la misma.

Los problemas a resolver, abarcados en este documento, para desarrollar las distintas soluciones a las situaciones problemáticas abarcadas anteriormente, son los siguientes:

- Eliminar los medios físicos en los que se encuentra almacenada la información actual y migrar la misma hacia formato digital, siendo montada en una base de datos, con vistas a su automatización, para el logro exitoso de los diferentes procesos de la Residencia.
- Eliminar la vía en que se actualizan los datos sobre toda la información referente a la Residencia y crear un sistema que permita la gestión de toda la información.
- Eliminar el tiempo en que transita la información desde la Residencia a la Facultad. Para esto es necesario lograr la conectividad entre ambas instituciones mediante el sistema de gestión propuesto, el cual deberá brindar su información vía Web a través de la Intranet de la UCI.
- Crear los medios para hacer más viable la creación de los calendarios de guardia y de limpieza de apartamentos, o sea automatizar estos procesos, para brindar una mayor flexibilidad a los mismos ante los diferentes cambios que puedan ocurrir y evitar los errores que se pudiesen cometer al generarlos de forma manual.
- Crear los mecanismos para automatizar la gestión de los residentes y de los edificios, junto a cada uno de sus apartamentos. Para afrontar los diferentes cambios que puedan surgir cada nuevo inicio de curso.
- Construir las vías para el control de los inventarios de cada uno de los apartamentos, pudiéndose así tener una mayor regulación sobre los medios básicos ubicados en cada uno de ellos.
- Brindar la posibilidad de generar, visualizar y enviar cualquier tipo de reportes referentes a cada uno de los medios técnicos, personal residente, apartamentos o cualquier información devengada por el control de la Residencia.
- Automatizar nuevas formas de evaluación para los residentes, eliminando las tradicionales que se emitían de forma manual. Brindando así los medios para

una mejor visualización de las mismas por parte de los residentes y agilizar el trabajo de Instructoras y Psicopedagogas.

La necesidad de este trabajo de diploma radica en proporcionar un Sistema Informático que sea capaz de gestionar toda la información generada en la Residencia y contribuir a la automatización de todos sus procesos, desde la gestión de su material humano y material, hasta la emisión de diferentes tipos de reportes, que garanticen un control más flexible, exhaustivo, veloz y eficiente que el actual. Lográndose un salto cualitativo y tecnológico en la administración de la Residencia de la Facultad 3.

Como objeto de estudio se seleccionaron:

- Las diferentes vías de control de la Residencia, haciendo énfasis en los referentes a los de residentes y apartamentos.
- Tipos y formas de evaluar las Instructoras y Psicopedagogas a los residentes, así como las vías de conocimiento de estos últimos sobre las mismas.
- Diseño de las características de la información archivada actualmente en formato duro de material de oficina referente a la Residencia.
- Las diferentes vías para la creación de Reportes ó Informes acerca de cada uno de los aspectos de la Residencia y el formato de los mismos.
- Roles de cada uno de los trabajadores de la Residencia, así como el Vise-Decano, en la administración de dicha entidad.

El campo de acción se determinó sobre lo siguientes puntos:

- Formas de creación de los diferentes calendarios generados en la Residencia para responder a las diferentes guardias y limpieza de cuarto.
- Aspectos a ser considerados para la asignación de un residente a una disponibilidad en la Residencia.
- Factores que influyen para determinar el concepto de "habitable" para un apartamento.
- Factores que permiten definir el estado físico de los mediõs básicos.

- Elementos que pudiesen incidir negativamente en los calendarios de guardias y limpieza de apartamentos. Haciendo que estos deban ser redefinidos.

El desarrollo de estas investigaciones partieron de la siguiente hipótesis:

- Con el estudio de técnicas de diseño de base de datos y con el empleo de herramientas, interfaces de programación de aplicaciones y lenguajes de programación de alto nivel, se puede realizar un diseño estándar de la estructura de las tablas de una base de datos y la implementación de las políticas de seguridad para la conformación de un sistema de gestión sobre un ambiente Web, que brinde acceso a todas las funcionalidades del mismo.

El objetivo general de este trabajo de diploma lo constituye el facilitar la creación de un Sistema de Gestión, que implique garantizarle a sus usuarios la posibilidad de acceder a todos los aspectos relacionados con la información, disponible en su base de datos, sobre la Residencia. Sin embargo como objetivos específicos se encuentran: estandarizar el diseño e implementar los componentes del sistema de forma que permita a los desarrolladores facilitar el proceso de creación de un nuevo servicio y desarrollar la aplicación Web bajo la concepción de que formará parte de un grupo de módulos que serán ubicados bajo la interfaz brindada por el Portal de la Facultad 3.

Para cumplir los objetivos propuestos se desarrollaron las siguientes tareas:

- Estudio de las técnicas de diseño de base de datos.
- Estudio de las posibilidades que brinda el lenguaje de programación C# y la tecnología ASP.NET para la implementación de aplicaciones Web.
- Diseño e implementación de la aplicación Web que forma parte del Sistema de Gestión.
- Estudio de los algoritmos necesarios para la creación de los calendarios específicos necesitados dentro de los servicios del Sistema de Gestión.

Este documento esta dividido en introducción, cinco capítulos, conclusiones, recomendaciones, bibliografía y anexos. Los capítulos siguen la siguiente organización:

- Capítulo I: Fundamentación Teórica. Abordá conceptos claves para el entendimiento de este trabajo de diploma. Se realiza la descripción y análisis del flujo actual de los procesos. Además de la fundamentación de los objetivos, tecnologías y herramientas utilizadas en el desarrollo de este proyecto.
- Capítulo II: Modelo del Negocio. Se describe el negocio donde se va a desarrollar la aplicación al igual que un conjunto de reglas por las cuales se rige el mismo. De esta forma, se analizan y describen los actores, trabajadores y casos de uso de dicho negocio.
- Capítulo III: Requisitos y Descripción del Sistema Propuesto. Se realiza una descripción de los distintos requisitos del sistema, así como de los actores y casos de uso. Se definen paquetes, dentro de los cuales se agrupan los casos de uso de acuerdo a sus funcionalidades.
- Capítulo IV: Descripción de la Solución Propuesta. Se realizan, por paquetes, los diferentes diagramas de clases del diseño. Así como el diseño de la Base de Datos y el Modelo de Despliegue, junto a algunos principios del diseño del proyecto.
- Capítulo V: Estudio de Factibilidad. Se realiza un estudio de la factibilidad del sistema basado en la planificación por casos de uso. Se analizan los beneficios del proyecto desarrollado, así como un análisis sobre la relación costo/beneficio.

Capítulo I: Fundamentación Teórica

1.1 Introducción

Una residencia es, por una vertiente, una actividad económica que requiere disponer de un sistema de facturación y gestión, pero también es una actividad asistencial que necesita mantener actualizado un historial y una serie de documentos (registros de inventarios, hojas de seguimiento y planificación) requeridos por la administración.

Esta doble realidad hace indispensable disponer de una herramienta ágil que permita manejar toda la información que se genera en las dos vertientes de actividad.

La decisión de informatizar la gestión de una residencia y la elección de la aplicación más adecuada para llevar a cabo la misma requiere plantearse las siguientes cuestiones:

- No todos los usuarios tienen un alto nivel técnico, de esta forma se necesita disponer de una aplicación fácil de utilizar.
- Se necesita un programa que contemple la documentación que exige la inspección y control. Si los modelos que utiliza la aplicación se ajustan a los exigidos nos facilitará la relación con la Administración.
- Se necesita poder garantizar la privacidad de determinados datos. O sea, que cada miembro del equipo pueda consultar los datos que le son relevantes pero no aquellos otros que no son de su contenido profesional.
- Se necesita un buen servicio técnico que asegure el soporte necesario en todo momento, así como la posibilidad de adaptar el programa a la cambiante realidad de un sector tan dinámico como el nuestro.

No todos los programas de gestión residenciales son iguales ni cumplen con estos requisitos. Razón que nos lleva a optar por una aplicación basada en los beneficios de un sistema de gestión, con las características que se expondrán posteriormente y bajo las tendencias más usadas a nivel internacional y nacional, apoyadas en la necesidad de satisfacer los requerimientos específicos de la Facultad 3 de la Universidad de las Ciencias Informáticas.

1.2 Objeto de Estudio

1.2.1 Objetivos estratégicos de la organización

La Universidad de las Ciencias Informáticas cuenta actualmente con 10 facultades docentes, cada una especializada en diferentes ramas, con el objetivo de poder crear un amplio desarrollo tecnológico que le permita responder ante los diversos proyectos en los que pudiese participar en el futuro.

Específicamente dentro de la Institución, la Facultad 3, la cual responde, organizada y funcionalmente, a las necesidades y demandas de la UCI, en referencia a un grupo plenamente identificado, el Turismo. De esta forma establecen como sus objetivos estratégicos:

1. El Desarrollo de Sistemas Informáticos vinculados a su segundo perfil, el Turismo.
2. Formar a sus estudiantes bajo las bases de su segundo perfil.

Con el fin de lograr estos objetivos, pone a disposición de los clientes, apoyada en los recursos tecnológicos y económicos de que dispone, el trabajo de estudiantes y profesionales debidamente capacitados.

Para poder responder de manera eficiente al desarrollo de los recursos humanos, la Facultad consta de 3 departamentos:

1. Ciencias Básicas
2. Especialidad Informática
3. Ciencias Humanísticas

En estos departamentos se diseñan los pasos a seguir por la Facultad para lograr sus objetivos, con la participación de un claustro de profesores conformados, según su grado científico, de la siguiente forma: dos (2) Doctores; tres (3) Másteres y 31 Licenciados e Ingenieros.

Además, la Facultad cuenta con una Residencia para sus estudiantes y profesores, de la cual tiene como objetivo, la automatización de la misma, persiguiendo con esto:

- I. Facilitar el Proceso de Control para la Residencia Estudiantil.

- II. Gestionar toda la información de inventario en el área de residencia.
- III. Agilizar el proceso de distribución de apartamentos a los residentes estudiantiles.
- IV. Brindar información precisa del proceso en la Residencia, a directivos y otras instituciones, de manera rápida y clara.

1.2.2 Flujo Actual de los Procesos

Actualmente, dentro de la Residencia estudiantil, se realizan varios procesos, los cuales procedemos a su explicación:

- Cada año la Residencia, asociada a la Facultad 3, generalmente cambia su estructura. Lo cual obliga a reestructurar todo el control residencial de los estudiantes, pues el año anterior la Residencia de la Facultad pudo tener asignada, varios edificios y para el año entrante esta asignación cambia (no específicamente en un 100%), lo cual obliga reubicar, a las nuevas instalaciones, a los estudiantes, anteriores inquilinos de los edificios “afectados”.
- De la misma forma cada inicio de curso infiere una planificación de tareas a realizar por los estudiantes:
 - Guardia Estudiantil
 - Cuartelería
 - Limpieza de apartamento

Estas planificaciones se hacen de forma manual, por la lista de estudiantes de cada Edificio y apartamento, según corresponda a las necesidades de la tarea.

- Para el control de la Residencia, se designan varias personas, de ahora en adelante Instructoras, las cuales son asignadas a uno o varios edificios, para lograr una correcta inspección y dirección de los mismos. Su tarea principal es: diariamente llevar el control de las habitaciones de los edificios a los

cuales fue asignada. Para esto revisa cada apartamento, en el cual evalúa la limpieza del mismo, al estudiante que se encuentre de limpieza en el día en curso.

- De forma esporádica, en toda la Residencia, se lleva un control de los medios básicos, de cada una de las habitaciones de los diferentes edificios, por parte de la Instructora, encargada de realizar dicha tarea. Este control de inventario, se hace de forma manual, en el cual la Instructora revisa, la existencia de los medios y su estado físico.
- Diariamente, excepto los domingos, las Instructoras se encargan de evaluar la Cuartelería al estudiante de turno, en parámetros como la limpieza de las escaleras y del área verde correspondiente.
- Se lleva el control diario de la asistencia al estudiante asignado a la Guardia Estudiantil, en horario nocturno.

1.2.3 Análisis Crítico de la Ejecución de los Procesos

Todos estos procesos responden a una coyuntura marcada por la carencia de una aplicación completamente adaptada y desarrollada a las condiciones de la Residencia, así como a las diferentes variaciones que pudiese sufrir la misma, en el transcurso evolutivo de la UCI.

Los principales problemas que presentan estos procesos son los siguientes:

- A raíz del cambio estructural que sufre la Residencia anualmente, es obligatorio reestructurar de forma manual y engorrosa la nueva ubicación de los estudiantes afectados por dichos cambios. De forma que no existe una salida rápida para reubicar a dichos estudiantes, en cada uno de las nuevas vacantes que pudiesen aparecer, debido a la baja de algunos estudiantes en los diferentes apartamentos, no afectados, o en las nuevas habitaciones adquiridas tras la reestructuración. Lo cual implica, recorrer de forma manual y en folletos físicos, la disponibilidad de la Residencia para poder hacer dichos cambios.

- Aparentemente la planificación de las diferentes tareas a efectuar por los estudiantes, pudiese parecer sencilla, pero una vez deben solicitarse todos los registros físicos de los estudiantes y definir manualmente, los días y horarios para la ejecución de cada una de las tareas (Guardia Estudiantil, Cuarterería, Limpieza de Apartamento). Lo cual crea un “modus operandi” nada flexible ante situaciones como las de correr o reubicar nuevos días y horarios, a los estudiantes por diversas razones, específicamente: salida de la UCI, enfermedad u otras actividades.
- En el caso específico del trabajo de las Instructoras, la falta de automatización implica el control de cada uno de las actividades que desarrollan de forma manual y archivado físico en folios. Lo cual impide el posterior desarrollo ágil del conocimiento de los mismos y la “pobre” cantidad de estadísticas, que esta forma de actuar puede aportar, en comparación con lo que puede brindar un buen sistema de gestión automatizado, que por demás elimina el alto gasto de recursos, en hojas y otros medios, para mantener el total control de sus actividades. Todo lo anterior retrasa la supuesta solución de la Facultad de lograr una mayor velocidad y control sobre la residencia con el empleo de Instructoras, atacándose de esta forma el objetivo de agilizar más la “comunicación”: Residencia – Facultad.

1.3 Procesos Objetos de Automatización

Tras el estudio del flujo de procesos, se pretende automatizar:

- El control de las ubicaciones y disponibilidades de la residencia, tanto las posibles asignadas como las ya existentes.
- El control de cada uno de los estudiantes de la Residencia, entendiéndose la gestión de los mismos, nuevos ingresos, bajas y modificaciones dentro de la entidad, con el objetivo de tener un mayor control sobre los mismos.
- La ubicación y reubicación de los estudiantes, en cada inicio de curso, aprovechando las bondades sobre control de los mismos y de los edificios de

la Residencia, se automatizará la antigua forma, creando una nueva con amplio poder de flexibilidad y agilidad en la ejecución de la misma.

- La Limpieza de Apartamentos, se automatizará, utilizando el conocimiento brindado por el control de los estudiantes, sobre los integrantes de cada apartamento, logrando de forma automática la asignación semanal de los mismos con flexibilidad en caso de reasignación por diversas causas.
- La “Cuartería” contará con la posibilidad de “saber” los diferentes estudiantes de cada edificio, para asignar a cada uno de ellos el día que deben asistir a la misma, excepto los domingos, con la misma flexibilidad de reasignación presente en la “Limpieza de Apartamentos”.
- La Guardia Estudiantil, será automatizada de la misma forma que las dos tareas anteriores (Limpieza de Apartamentos y Cuartería), lo cual implica la misma flexibilidad de reubicación y cambios, en la misma, ante las diferentes afectaciones que esta pudiese presentar por ser realizada en horarios nocturnos.
- Se automatizará la forma en que las Instructoras evalúan cada una de sus tareas:
 - Evaluar la limpieza de los apartamentos diariamente.
 - Evaluar la asistencia y desarrollo de la Cuartería diariamente.
 - Chequeo de los medios básicos de cada apartamento.
- En aras de lograr un mayor dominio sobre la gestión de los medios básicos referentes a la Residencia, se automatizará todo el control de los mismos, para lograr una mayor flexibilidad, debido su grado de importancia material.

1.4 Sistemas Automatizados existentes vinculados al Campo de Acción

Muchas veces, en el proceso de elegir la herramienta informática que maneje la administración de su empresa o institución, habrá cierta frustración, pues los denominados "sistemas enlatados" (fabricados en serie por empresas) no cubren totalmente las expectativas, le faltan utilidades, lo que indica que no cumplirá en un 100% con las necesidades aunque poseen servicio técnico post-venta. La otra alternativa es encarar un desarrollo a medida que podría abarcar todo lo que usted requiere de un Sistema de Gestión, pero resulta sumamente costoso, demora demasiado tiempo en su construcción, período de prueba y puesta a punto, y no posee el soporte técnico adecuado. [2]

En la actualidad, por las condiciones antes mencionadas, los Sistemas de Gestión para Residencias, no existen de manera genérica para su implantación. Existen empresas que desarrollan software para residencias, partiendo de un software principal que ajustan a las necesidades de los diferentes clientes.

Actualmente en nuestro país, al igual que en el extranjero, no existe ninguna solución que responda totalmente a las necesidades de la entidad. Pues la aplicación que se precisa necesita ser a la medida. Actualmente las soluciones existentes no llenan todas las necesidades que pretenden en la Residencia de la Facultad 3.

No obstante existen algunas variantes de aplicaciones para la gestión residencial, pero generalmente desarrolladas para geriátricos debido al auge de los mismos en el sistema social actual. Lo cual no implica que universidades u otras instituciones semejantes a la UCI, no cuenten con un sistema de gestión que responda a los intereses automáticos de su residencia, pero usualmente forman parte de un sistema integral de gestión, concebidos en forma de módulos.

Algunos ejemplos de software desarrollado para residencias son los siguientes:



Figura 1

- **UNYGES:** Es una herramienta informática que se crea específicamente para cada cliente, en aras de facilitar la gestión administrativa de residencias con un corte médico - geriátrico. (Ver anexo #1)



Figura 2

- **RESIGES:** Herramienta informática para la Gestión de Residencias de Mayores y Centros de día.

Con esta herramienta se puede llevar la gestión de los residentes, dispone de una completa ficha médica de cada residente, puede hacer una gestión efectiva del personal mediante el generador de cuadrantes incluido en el programa, tiene a los familiares informados mediante los informes generados por el programa, lleva la gestión de facturación y dispone de un gran número de informes y estadísticas que permiten hacer una radiografía continua de la marcha de su Residencia.

RESIGES se adapta a las necesidades concretas de cada residencia; no es un programa cerrado en el que no se admiten cambios o modificaciones. Ha sido adaptado al uso de cientos de residencias, con cambios específicos para cada Comunidad Autónoma, tipo de centro, necesidades de gestión, etc. (Ver Anexo #2)

Si bien de los dos ejemplos planteados anteriormente, la solución que propone RESIGES, parece viable no se puede obviar que la contratación de una de estas empresas desarrolladoras de aplicaciones a la medida, representa un gran gasto de recursos, pudiéndose llegar a un monto por encima de los miles de dólares. Por lo que no se consideró esta opción. De esta forma se optó por el desarrollar un sistema que no incurriese en los altos costos del mercado y que cumpliera con todas las características necesarias, sin olvidar los niveles de calidad y profesionalidad exigidos por la Facultad 3.

1.5 Fundamentación de los Objetivos

Como principal objetivo tenemos la creación de un Sistema de Gestión, que automatice las actividades de la Residencia y que sea capaz de brindar toda una gama de reportes que respondan a los intereses de la Facultad, para visualizar la situación actual de la Residencia y de sus inquilinos, los cuales precisan de una buena actitud en la misma, para su permanencia dentro de la Institución.

Para garantizar la buena implantación de este Sistema de Gestión, primero se debe lograr que las personas que actualmente se encuentran inmersos en la realización de los procesos de la Residencia, comprendan la importancia del mismo, y el cambio tecnológico que este implica, para llevar adelante los beneficios que otorga a corto plazo. De esta forma se logrará eliminar la antigua estructura, conformada por "material de oficina", que tan obsoleta resulta, en los días actuales y más en una institución de este tipo, puntera de tecnologías, en nuestro país.

Es imprescindible lograr una buena comunicación entre la Residencia y sus Instructoras para mantener el control de los medios básicos residentes y de las evaluaciones de todas las actividades en las que estas se ven implicadas, para así lograr el flujo real de gran parte de la información entre la actividad real y el Sistema de Gestión. Manteniendo de esta forma un gran nivel de actualización para el Sistema y con ello la eficiencia requerida en cada unos de sus reportes y actividades controladas por el mismo.

Se planificarán adecuadamente las actividades para evitar que coincidan con otros intereses que tengan una “mayor” relevancia a entidades superiores, como la Facultad o la Institución (UCI). Esto permitirá que la antigua planificación de las actividades para los estudiantes de la Residencia no sean afectadas, o en caso de afectación tenga una mejor viabilidad.

Como una respuesta a lo anterior, se generarán avisos cuando ocurran afectaciones a los residentes, por vía de correo electrónico o específicamente en la aplicación visual del Sistema de Gestión. De esta forma se opta por la opción de brindar conocimiento a sus usuarios de su situación actual o de futuros cambios, logrando que el Sistema de Gestión convierta de este servicio un “lugar” de acceso habitual para la mantención del buen estado evaluativo del estudiante y que no ocurran situaciones anteriores, donde las evaluaciones eran discutidas por los estudiantes por falta de conocimiento sobre su estado evaluativo o de comunicación ante un reajuste en las actividades.

Existe una alternativa intermedia que cubre tanto las expectativas de gestión como de costos. Son los sistemas modulares; cada módulo es, en sí, un programa especializado en una determinada tarea; con la posibilidad de optimizar sus costos de adquisición seleccionando los módulos que realmente se requieren para armar la gestión administrativa que se necesite. Debido a que este Sistema de Gestión forma parte de todo un proceso de automatización de la Facultad 3, se garantizará la construcción del mismo, de una forma modular y con gran independencia del sistema al cual pertenecerá en un final. Para garantizar su futura inserción, se debe diseñar una propuesta robusta y con gran autonomía, capaz de “aceptar” las futuras características del proceso de automatización de la Facultad y con una flexibilidad capaz de soportar la inserción, dentro de su concepción de cambios estructurales en cuanto a su definición.

1.6 Tendencias y Tecnologías Actuales

En la era de la información, de la explosión de sus tecnologías, hemos alcanzado un desarrollo imprevisible. Se habla constantemente sobre la sociedad de la información. En este contexto, debe entenderse que las tecnologías de la información y las

telecomunicaciones (TIC) no son más que un medio para transmitir y gestionar datos, información y conocimiento.

En el mundo de los servicios de información, las ideas de la calidad no deben vincularse sólo con la tenencia de recursos tecnológicos. Si el usuario/cliente no recibe los servicios como él los solicita y los espera, el fracaso es seguro. La gestión de la información se vincula con la generación y la aplicación de estrategias, el establecimiento de políticas, así como con el desarrollo de una cultura organizacional y social dirigida al uso racional, efectivo y eficiente de la información en función de los objetivos y metas de las compañías u organizaciones en materia de desempeño y de calidad.

Las estrategias actuales para la gestión de la información y el conocimiento deben responder a los nuevos tipos de demandas, resultantes de la aparición de tendencias gerenciales más modernas en las organizaciones. En la creación de los nuevos sistemas de gestión de la información es imprescindible considerar las fuentes factográficas (datos), documentales y no documentales, los sistemas informáticos, la cultura de información, los modelos de comunicación, entre otros elementos. Según los requerimientos de los procesos internos de trabajo y los flujos de información propios, todos ellos deben propiciar la gestión del conocimiento organizacional y la implementación de sistemas de gestión para la evaluación de los resultados y los proyectos de la institución. [1]

1.6.1 Sistemas de Gestión

En los nuevos modelos de negocio la gestión de la información, la documentación y el conocimiento se perfila como un componente estratégico de primera magnitud. La orientación tradicional del carácter táctico de los proyectos de gestión de la información o de la documentación cambia cuando se considera un verdadero componente de la estrategia empresarial. Este cambio no sólo afecta a las empresas que se crean para actuar específicamente en el ámbito del e-business, sino que, al menos en el ámbito teórico es una realidad hasta en las empresas más "tradicionales". La profunda revolución tecnológica que estamos viviendo, ha sido el motor de este cambio. Por esta

razón muchas veces los empresarios y directivos simplifican su actuación frente a la nueva realidad, centrándola en la compra e instalación de herramientas informáticas de última generación que deberían dar resultados a corto plazo.

Aunque la tan mentada "oficina sin papeles" esta lejos de ser una realidad, lo que tenemos es la "oficina con menos papeles". En esto ha influido mucho en la rápida aceptación del correo electrónico como medio de trabajo. En muchas organizaciones documentos que se guardaban y utilizaban hasta hace poco en papel ya no existen físicamente. En este contexto cualquier práctica de gestión de la información debe tener en cuenta la realidad mixta (documentos papel/electrónicos) en la que todavía nos moveremos durante bastante tiempo.

La gestión de las principales actividades de una entidad a través de un sistema informático permite crear, almacenar, administrar y archivar documentos, registros de inventarios, procedimientos, hojas de seguimiento y planificación; asociados a cada uno de ellos. Así los usuarios, de dicho sistema informático, pueden consultar sus intereses específicos y tomar, posteriormente, las acciones necesarias. [3]

1.6.1.1 Funcionalidades de los Sistemas de Gestión

Existen varios servicios que debe cumplir un sistema de gestión para satisfacer las necesidades requeridas por sus usuarios. A continuación plasmaremos algunos que debe ofrecer todo sistema de gestión:

- Proporcionar a los usuarios la capacidad de almacenar datos en la base de datos, acceder a ellos y actualizarlos. Esta es la función fundamental de un sistema de gestión y por supuesto, debe ocultar al usuario la estructura física interna (la organización de los ficheros y las estructuras de almacenamiento).
- Proporcionar un mecanismo que garantice que todas las actualizaciones correspondientes a una determinada transacción se realicen, o que no se realice ninguna.

- Proporcionar un mecanismo que asegure que la base de datos se actualice correctamente cuando varios usuarios la están actualizando concurrentemente.
- Proporcionar un mecanismo capaz de recuperar la base de datos en caso de que ocurra algún suceso que la dañe.
- Proporcionar un mecanismo que garantice que sólo los usuarios autorizados pueden acceder a la base de datos. La protección debe ser contra accesos no autorizados, tanto intencionados como accidentales. [14]
- Proporcionar los medios necesarios para garantizar que tanto los datos de la base de datos, como los cambios que se realizan sobre estos datos, sigan ciertas reglas. La integridad de la base de datos requiere la validez y consistencia de los datos almacenados. Se puede considerar como otro modo de proteger la base de datos, pero además de tener que ver con la seguridad, tiene otras implicaciones. La integridad se ocupa de la calidad de los datos. Normalmente se expresa mediante restricciones, que son una serie de reglas que la base de datos no puede violar.
- Permitir que se mantenga la independencia entre los programas y la estructura de la base de datos.

Los sistemas de gestión son paquetes de software muy complejos y sofisticados que deben proporcionar los servicios comentados anteriormente. No se puede generalizar sobre los elementos y funcionalidades que componen un sistema de gestión, pues varían mucho unos de otros. [7]

Los sistemas de gestión permiten la automatización de tareas mediante la programación de actividades que se desean que ahora realicen las computadoras, mejorando en gran medida cualquier tipo de proceso de datos, así como su rapidez y fiabilidad. Permiten además la instantaneidad e interconexión para difundir información. Los procesos de interacción, su frecuencia y el valor de la información intercambiada constituyen la viveza del sistema. La interactividad es una consecuencia de que las máquinas sean programables y podamos definir su comportamiento, determinando las

respuestas que deben dar ante las distintas acciones que ejecuten sobre ellas los usuarios. [18]

1.6.1.2 Beneficios del Uso de Sistemas de Gestión

Para la concepción de todo sistema hay ciertos pasos a seguir de forma general, que garantizan el éxito final. Inicialmente se debe determinar las necesidades existentes, en cuanto a los tipos de datos que deben existir para garantizar que los procesos se lleven a cabo bajo condiciones totalmente controladas. Posteriormente, conocer la situación actual del negocio, comparando lo que existe con las necesidades encontradas anteriormente. Una vez establecidas, se está en condiciones de elaborar los elementos generales necesarios para la elaboración del sistema, o sea diseñarlo, que implica definir jerarquías, autoridad y responsabilidades a cada nivel, estructuras y formatos de los datos. Entonces se está en condiciones de elaborar el sistema, módulo a módulo, realizando las pruebas necesarias en cada parte; cuando cada una de ellas funcione de la manera esperada se implantará el sistema, pues solo en su implantación misma se puede realizar su prueba, mantenimiento y mejoras. [12]

Los sistemas de gestión, en la actualidad, están en completo auge como una vía para alcanzar todos sus beneficios y estar a la altura de la mayoría de las empresas e instituciones del mundo, teniendo en cuenta el gran desarrollo de las tecnologías digitales. A la hora de implantar un sistema de gestión es primordial diseñar una política de integración total con la red local en el que se encuentra. Cuando se concibe el sistema se debe realizar una estimación aproximada de la cantidad de datos que se tratará y el crecimiento futuro que tendrán con el transcurso del tiempo, para diseñar una correcta política de almacenamiento, formatos y tiempo de vida de los archivos implicados.

El uso de estos sistemas de gestión nos brinda la posibilidad de tener todos los datos en un solo sistema, o sea que no necesitamos “indagar” por diferentes vías para obtener un resultado, pues se localiza en un solo lugar. La transferencia de datos es sencilla además de una ausencia de problemas de interconexión de datos entre los

diferentes instrumentos y programas. Incluso brinda la posibilidad de numerosas aplicaciones en múltiples tareas.

Es un sistema de fácil manejo, abierto y con la total capacidad de concebirse modularmente, lo cual brinda ventajas: sencillez e idoneidad para un uso frecuente; buen “precio” para un uso diario; admite soluciones existentes, los usuarios se benefician de un desarrollo continuo y posibilita responder a todos los interés cambiantes de los usuarios.

Estas características hacen que los sistemas de gestión formen parte de las opciones a optar por parte de las diferentes entidades que necesiten el control y supervisión de sus intereses.

1.6.2 Fundamentación de la Metodología Utilizada

El objetivo de un proceso de desarrollo es garantizar la calidad del software a través de una mayor transparencia y control sobre el proceso.

En la actualidad a la producción de software muchas veces no se le aplica una adecuada metodología y paradójicamente coincide el éxito de un software en la utilización correcta de una metodología, capaz de cumplir con los objetivos del sistema.

1.6.2.1 Metodología RUP

El Rational Unified Process (RUP) es un proceso de software de ingeniería habilitado para Internet que enriquece la productividad en equipo y proporciona prácticas óptimas de software a todos los miembros del equipo. RUP es un mentor en línea fácil de usar que hace el proceso práctico brindando amplias guías, plantillas y ejemplos para todas las actividades críticas del desarrollo electrónico. El contenido específico de negocios electrónicos de RUP proporciona guías explícitas para áreas tales como modelado de negocios, arquitectura Web y pruebas para la Red. RUP es un framework ajustable, que puede adaptarse fácilmente a la forma en que usted trabaja. [13]

RUP divide el proceso de desarrollo en ciclos, donde se obtiene un producto al final de cada ciclo. Cada ciclo se divide en cuatro fases: concepción, elaboración, construcción, y transición. Cada fase concluye con un hito bien definido donde deben tomarse ciertas decisiones.

Las características principales que lo distingue son:

- Proceso Dirigido por Casos de Uso:
 - Capturar, definir y validar casos de uso.
 - Realizar casos de uso.
 - Verificar que se satisfacen casos de uso.
- Proceso Iterativo e Incremental
 - El ciclo de vida iterativo se basa en la evolución de prototipos ejecutables que se muestran a los usuarios y clientes.
 - En el ciclo de vida iterativo a cada iteración se reproduce el ciclo de vida en cascada a menor escala.
 - Los objetivos de una iteración se establecen en función de la evaluación de las iteraciones precedentes.
 - Las actividades se encadenan en una mini-cascada con un alcance limitado por los objetivos de la iteración. [6]
- Proceso Centrado en la Arquitectura
 - Arquitectura de un sistema es la organización o estructura de sus partes más relevantes. Una arquitectura ejecutable es una implementación parcial del sistema, construida para demostrar algunas funciones y propiedades.
 - RUP establece refinamientos sucesivos de una arquitectura ejecutable.

1.6.2.2 Notación UML

El Unified Modeling Language (UML) fue desarrollado en conjunto por Grady Booch, Ivar Jacobson, y Jim Rumbaugh en la Rational Software Corporation, con contribuciones de otros excelentes metodólogos, vendedores de software, y muchos usuarios. Basado en el amplio uso del Booch, OMT, y métodos de Jacobson, el UML es

la evolución de estos y otros métodos para el moldeado de procesos de negocios, objetos y componentes. El UML brinda el lenguaje de aplicación de modelado para:

- Modelado de proceso de negocios con casos de uso.
- Modelado de clases y objetos.
- Modelado de componentes.
- Modelado de distribución y despliegues.

Es el lenguaje estandarizado en la industria para especificar, visualizar, construir y documentar los dispositivos de sistemas de software. Simplifica el complejo proceso de diseño de software, haciendo "planos" para su construcción. [17]

El UML estándar está compuesto por tres partes: bloques de construcción (tales como clases, objetos, mensajes), relaciones entre los bloques (tales como asociación, generalización) y diagramas (por ejemplo, diagrama de actividad). Los perfiles del UML son las extensiones a las notaciones estándares del UML usando los mecanismos de extensión del UML: los estereotipos, los valores etiquetados y las restricciones. Presenta nueve diagramas estándares: diagrama de casos de uso, de clases, de secuencia, de colaboración, de actividad, de estados, de implementación (componentes), de despliegue, si fuera necesario, el diagrama de objetos se puede crear usando los diagramas de colaboración. Las diferencias existen solamente con relación al soporte de una característica específica durante la creación de los diagramas UML y los perfiles extendidos del UML.

UML es un lenguaje estándar con el que es posible modelar todos los componentes del proceso de desarrollo de aplicaciones. Sin embargo, hay que tener en cuenta un aspecto importante del modelo: no pretende definir un modelo estándar de desarrollo, sino únicamente un lenguaje de modelado. Otros métodos de modelaje como OMT (Object Modeling Technique) o Booch sí definen procesos concretos. En UML los procesos de desarrollo son diferentes según los distintos dominios de trabajo; no puede ser el mismo el proceso para crear una aplicación en tiempo real, que el proceso de desarrollo de una aplicación orientada a gestión, por poner un ejemplo.

Con él sus protagonistas se propusieron que:

El método debía ser capaz de modelar no sólo sistemas de software sino otro tipo de sistemas reales de la empresa, siempre utilizando los conceptos de la orientación a objetos (OO).

Crear un lenguaje para modelado utilizable a la vez por máquinas y por personas.

Establecer un acoplamiento explícito de los conceptos y los artefactos ejecutables.

Manejar los problemas típicos de los sistemas complejos de misión crítica

UML no define un proceso concreto que determine las fases de desarrollo de un sistema, las empresas pueden utilizar UML como el lenguaje para definir sus propios procesos y lo único que tendrán en común con otras organizaciones que utilicen UML serán los tipos de diagramas. UML es un método independiente del proceso. Los procesos de desarrollo deben ser definidos dentro del contexto donde se van a implementar los sistemas. [16]

1.6.3 Fundamentación del Lenguaje, Gestor de Base de Datos y Plataforma de Desarrollo

Las soluciones creadas para software específicos, usualmente cuentan con una gran robustez, pero fallan al no llevar un buen tratamiento de los datos y problemáticas que dieron origen a la solución, lo cual no garantiza un cambio radical en la concepción de una nueva solución. Lo cual tiende a no satisfacer las necesidades de información de la organización; necesidades acentuadas con el transcurso del tiempo.

Para la concepción de una buena solución informática, debe tenerse en cuenta, la decisión sobre que medios usar para sustentar a la misma. En el caso de nuestra aplicación, tiene la característica de ser un Sistema de Gestión para la Residencia. Este sistema vera sus frutos gracias a la interacción y explotación eficiente de un buen gestor de base de datos, asociado a un práctico lenguaje de programación. Pues en la actualidad, existen un buen número de gestores de base de datos y de lenguajes de programación, con la característica de “combinarse”, independientemente del desarrollador.

Existen varios patrones eficientes para el desarrollo de aplicaciones. Por lo que se decide optar por la utilización una de las alternativas mas viables, productivas y confiables del mundo tecnológico actual: SQL Server 2000 (Gestor de Base de Datos) y ASP.NET (Tecnología para la programación) sustentada bajo el empleo de Visual Studio 2005.

1.6.3.1 ASP.NET

ASP.NET es más que una nueva versión de las páginas Active Server (ASP); proporciona un modelo de desarrollo Web unificado que incluye los servicios necesarios para que los programadores creen aplicaciones Web para la empresa, proporciona también un modelo de programación y una estructura nuevos para crear aplicaciones más escalables y estables que ayuden a proporcionar mayor protección.

ASP.NET es un entorno compilado basado en .NET. Se pueden crear aplicaciones en cualquier lenguaje compatible con .NET, como Visual Basic .NET, C# y JScript .NET. Además, .NET Framework está disponible en su totalidad para cualquier aplicación ASP.NET. Los programadores pueden aprovechar fácilmente las ventajas de estas tecnologías, que incluyen el entorno Common Language Runtime administrado, seguridad de tipos, herencia, etc.

ASP.NET se ha diseñado para funcionar sin problemas con editores HTML WYSIWYG y otras herramientas de programación como Microsoft Visual Studio .NET. Todo esto, además de hacer más fácil la programación Web, ofrece todas las ventajas de estas herramientas, con una GUI que los programadores puede utilizar para ubicar controles de servidor en una página Web e integrar completamente la compatibilidad con la depuración.

A la hora de crear una aplicación ASP.NET, los programadores pueden utilizar formularios Web Forms o servicios Web XML o combinarlas de la manera que más les convenga. Las dos características son compatibles con la misma infraestructura, que permite utilizar esquemas de autenticación, almacenar en caché datos que se utilizan

con frecuencia y personalizar la configuración de la aplicación, entre otras muchas cosas.

Los formularios Web Forms permiten crear páginas Web basadas en formularios muy eficaces. Al crear estas páginas, se pueden usar controles de servidor ASP.NET para crear elementos comunes de la interfaz de usuario y programarlos para que realicen las tareas comunes. Estos controles permiten crear con rapidez un formulario Web Forms a partir de componentes integrados reutilizables o personalizados, con un código de página simplificado.

Un servicio Web XML proporciona los medios para obtener acceso a la funcionalidad del servidor de manera remota. Con los servicios Web XML, las empresas pueden exponer interfaces de programación a sus datos o lógica empresarial, que, a su vez, pueden obtener y manipular las aplicaciones de cliente y servidor. Los servicios Web XML permiten el intercambio de datos en escenarios cliente-servidor o servidor-servidor, utilizando estándares como los servicios de mensajería HTTP y XML para que los datos pasen los servidores de seguridad. Los servicios Web XML no están ligados a ninguna tecnología de componentes ni a ninguna convención de llamada a objetos concretas. En consecuencia, pueden tener acceso a los servicios Web XML los programas escritos en cualquier lenguaje, utilizando cualquier modelo de componentes y que se ejecuten en cualquier sistema operativo.

El acceso a bases de datos desde aplicaciones ASP.NET es una técnica utilizada con frecuencia para mostrar datos a los usuarios que visitan un sitio Web. ASP.NET hace que tener acceso a bases de datos con esta finalidad sea más fácil que nunca. También permite administrar la base de datos desde el código.

ASP.NET proporciona un modelo sencillo que permite que los programadores Web escriban que el código se ejecuta en el nivel de la aplicación. Los programadores pueden escribir dicho código en el archivo de texto Global.asax o en una clase compilada implementada como ensamblado. Esta lógica puede incluir eventos del nivel de la aplicación, pero los programadores tienen la posibilidad de extender este modelo para que se ajuste a las necesidades de la aplicación Web.

ASP.NET aprovecha las mejoras de rendimiento de .NET Framework y Common Language Runtime. Además, se ha diseñado para que ofrezca un rendimiento sensiblemente mejor que ASP y otras plataformas de programación Web. Todo el código de ASP.NET se compila, en lugar de interpretarse, lo que permite realizar enlaces en tiempo de diseño, establecer tipos inflexiblemente, compilar código nativo en modo Just-In-Time (JIT), entre otras muchas ventajas. En ASP.NET es muy fácil para los programadores eliminar módulos (por ejemplo, un módulo de sesión) que la aplicación que están programando no necesita. ASP.NET ofrece unos completos servicios de almacenamiento en caché (tanto servicios integrados como API de almacenamiento en caché). ASP.NET se suministra con contadores de rendimiento que los programadores y los administradores del sistema pueden supervisar para probar nuevas aplicaciones y recopilar estadísticas de aplicaciones existentes.

Los valores de configuración de ASP.NET se guardan en archivos basados en XML, que los usuarios pueden leer y escribir. Cada una de las aplicaciones puede tener un archivo de configuración distinto y el esquema de configuración se puede extender como sea necesario. [8]

1.6.3.2 Visual Studio .NET

Visual Studio .NET es un conjunto completo de herramientas de desarrollo para la construcción de aplicaciones Web ASP, servicios Web XML, aplicaciones para escritorio y aplicaciones móviles. Visual Basic .NET, Visual C++ .NET, Visual C# .NET y Visual J# .NET utilizan el mismo entorno de desarrollo integrado (IDE), que les permite compartir herramientas y facilita la creación de soluciones en varios lenguajes. Asimismo, dichos lenguajes aprovechan las funciones de .NET Framework, que ofrece acceso a tecnologías clave para simplificar el desarrollo de aplicaciones Web ASP y servicios Web XML.

.NET Framework es un entorno multilenguaje que permite generar, implantar y ejecutar Servicios Web y aplicaciones XML.

Common Language Runtime: A pesar de su nombre, el motor de tiempo de ejecución desempeña una función tanto durante la ejecución como durante el desarrollo de los componentes. Cuando el componente se está ejecutando, el motor de tiempo de ejecución es responsable de administrar la asignación de memoria, iniciar y detener subprocesos y procesos, y hacer cumplir la directiva de seguridad, así como satisfacer las posibles dependencias del componente sobre otros componentes. Durante el desarrollo, el papel del motor de tiempo de ejecución cambia ligeramente; a causa de la gran automatización que permite (por ejemplo, en la administración de memoria), el motor simplifica el trabajo del programador, especialmente al compararlo con la situación actual de la tecnología COM. En concreto, funciones tales como la reflexión reducen de forma espectacular la cantidad de código que debe escribir el programador para convertir la lógica de empresa en componentes reutilizables.

Clases de programación unificadas: El entorno de trabajo ofrece a los programadores un conjunto unificado, orientado a objetos, jerárquico y extensible de bibliotecas de clases (API). Actualmente, los programadores de C++ utilizan las Microsoft Foundation Classes y los programadores de Java utilizan las Windows Foundation Classes. El entorno de trabajo unifica estos modelos dispares, ofreciendo a los programadores de Visual Basic y JScript la posibilidad de tener también acceso a las bibliotecas. Con la creación de un conjunto de API comunes para todos los lenguajes de programación, Common Language Runtime permite la herencia, el control de errores y la depuración entre lenguajes. Todos los lenguajes de programación, desde JScript a C++, pueden tener acceso al entorno de trabajo de forma parecida y los programadores pueden elegir libremente el lenguaje que desean utilizar.

Visual Studio presenta un nuevo diseñador de páginas Web que incluye muchas mejoras para la creación y edición de páginas Web de ASP.NET y páginas HTML. Proporciona una forma más fácil y rápida de crear páginas de formularios Web Forms que en Visual Studio .NET 2003. [10]

Visual Web Developer incluye mejoras en todas las áreas de desarrollo de sitios Web. Puede crear y mantener sitios Web en carpetas locales, como en aplicaciones de los Servicios de Internet Information Server (IIS), o a través de una conexión FTP (File

Transfer Protocol, Protocolo de transferencia de archivos). El diseñador Visual Web Developer admite todas las mejoras de ASP.NET, incluidas las casi dos docenas de nuevos controles que simplifican muchas tareas de desarrollo Web.

La vista Diseño del diseñador HTML incluye muchas mejoras que admiten las nuevas funciones de ASP.NET o facilitan el diseño WYSIWYG de páginas Web. La edición basada en tareas mediante etiquetas inteligentes le guía durante la ejecución de los procedimientos más comunes con controles, como el enlace de datos y la asignación de formato. Puede editar visualmente las nuevas páginas principales de ASP.NET. La edición de plantillas se ha mejorado para facilitar el trabajo con controles de datos, así como con nuevos controles como el control Login. Editar tablas HTML para el diseño o mostrar la información en columnas ahora es más fácil e intuitivo.

En nuestro sistema utilizamos el Microsoft Visual C# 2005. Es un lenguaje de programación diseñado para crear una amplia gama de aplicaciones que se ejecutan en .NET Framework. C# es simple, eficaz, con seguridad de tipos y orientado a objetos. Con sus diversas innovaciones, C# permite desarrollar aplicaciones rápidamente y mantiene la expresividad y elegancia de los lenguajes de tipo C.

Visual Studio admite Visual C# con un editor de código completo, plantillas de proyecto, diseñadores, asistentes para código, un depurador eficaz y fácil de usar, además de otras herramientas. La biblioteca de clases .NET Framework ofrece acceso a una amplia gama de servicios de sistema operativo y a otras clases útiles y adecuadamente diseñadas que aceleran el ciclo de desarrollo de manera significativa.

Los programadores de C# pueden incorporarse a la comunidad de desarrollo más moderna y de más rápido crecimiento, en la que pueden intercambiar código y recursos, aprovechar sus conocimientos en múltiples entornos de programación y contribuir al proceso de normalización que garantiza una participación activa y entusiasta en dicha comunidad. [9]

1.6.3.3 SQL Server 2000

SQL Server es el servidor de bases de datos de Microsoft, seguro, robusto y con las más avanzadas prestaciones: transacciones, procedimientos almacenados, triggers, etc.

SQL Server 2000 proporciona a los usuarios una excelente plataforma de base de datos optimizada para procesamiento de transacciones en línea (OLTP, *Online Transactional Processing*) a gran escala, el almacenamiento de datos y las aplicaciones de comercio electrónico. Ofrece a los usuarios un entorno de Lenguaje de marcado extensible (XML, *Extensible Markup Language*) totalmente integrado, agrega una nueva característica de minería de datos en Analysis Services y mejora la tecnología de depósito con Meta Data Services.

Analysis Services presenta la característica de minería de datos, que se puede emplear para descubrir información en cubos OLAP y bases de datos relacionales. SQL Server 2000 mejora los controles de seguridad y la conectividad del cliente, y hace posible el análisis de los datos en tiempo real. [5]

Algunas de sus características son:

- **Compatibilidad con XML:** Simplifica la integración de los sistemas de servicios de fondo y la transferencia de datos a través de servidores de seguridad mediante XML.
- **Acceso Web a los datos:** conecta las bases de datos de manera flexible, mediante el Web sin necesidad de ninguna programación adicional.
- **Alojamiento de aplicaciones:** gracias a la compatibilidad con varias instancias, SQL Server permite aprovechar totalmente sus inversiones en hardware de forma que múltiples aplicaciones se pueden ejecutar en un solo servidor o externamente.
- **Análisis del flujo de clientes:** compresión más detallada del comportamiento del cliente en línea para tomar mejores decisiones.
- **Seguridad:** las aplicaciones son seguras en cualquier entorno de red, con la seguridad basada en funciones y el cifrado de archivos y de la red.

SQL Server es una aplicación completa que realiza toda la gestión relacionada con los datos. El servidor Web sólo tiene que enviarle una cadena de caracteres (la sentencia SQL) y esperar a que le devuelvan los datos. El servidor SQL se encarga de todo. Esta división de papeles es mucho más conveniente para la publicación Web.

Puede recibir conexiones (clientes que desean manipular datos) desde cualquier ordenador conectado; escucha un puerto TCP/IP. En este aspecto es igual que un servidor Web o un servidor de correo. La diferencia es el número de puerto y, naturalmente, el protocolo con el que se comunicará con el cliente. Un navegador es un cliente para un servidor Web, pero no conoce el protocolo para hablar con un servidor de bases de datos. Por ejemplo, Access 2000 o Access XP son clientes para SQL Server. [5]

SQL Server admite la programación cliente-servidor, que se utiliza cuando queremos realizar aplicaciones que utilicen redes y que comuniquen entre sí a varios ordenadores.

SQL Server admite procedimientos guardados (stored procedures) realizados en lenguaje SQL. Se trata de procedimientos que se guardan semicompilados en el servidor y que pueden ser invocados desde el cliente. Los procedimientos guardados podrían ser la "parte servidor" de una aplicación cliente-servidor. La parte cliente podría ser una aplicación ASP o incluso una aplicación en Visual Basic o Access, que puede acceder directamente a SQL Server.

SQL Server puede manejar perfectamente bases de datos de TeraBytes con millones de registros y funciona sin problemas con miles de conexiones simultáneas a los datos. Es un servidor y está pensado para gestionar tantos clientes simultáneos como admita la potencia del hardware del equipo en el que esté instalado. [5]

1.7 Conclusiones

Actualmente el uso de las tecnologías para el desarrollo Web depende del tipo de sistema a desarrollar y de la entidad a la que está destinada. La elección de la tecnología a utilizar depende de un estudio de los requerimientos de la solución a desarrollar.

Tras el análisis realizado en este capítulo sobre el flujo de los procesos; la fundamentación de los objetivos que concebía el sistema; las necesidades de la entidad a la cual va dirigida el Sistema de Gestión a implementar, junto a las tendencias y tecnologías actuales relacionadas con la propuesta, se concluye que será desarrollado con el gestor de base de datos SQL Server 2000, la herramienta de desarrollo Visual Studio .NET 2005 con C# como lenguaje (Tecnología ASP.NET), mientras que para el análisis y diseño del sistema se utilizará la metodología RUP con UML como lenguaje de modelado.

Capítulo II: Modelo del Negocio

2.1 Introducción

Para lograr el desarrollo profesional de nuestra solución es fundamental comprender claramente el contexto en el que se desarrolla. Aprovechando la metodología RUP, se decide realizar el Modelo del Negocio, el cual está orientado a entender el funcionamiento de los procesos del negocio de la organización.

La facturación de todo software, independientemente de sus dimensiones, suele ser complicado. Como resultado, se tiende a dividirlo para una lograr una mejor comprensión y gestionar su complejidad. Tal división pudiese estar representada a través de algunos modelos, que permitan una mejor abstracción de la situación real.

Los objetivos de una modelación del negocio:

- Comprender la estructura y dinámica de la organización tratada para la implantación del sistema que brinde la solución a la problemática.
- Comprender los problemas actuales de la organización e identificar las mejoras potenciales.
- Asegurar la total comprensión de la organización por parte de los consumidores, usuarios finales y desarrolladores.
- Derivar los requerimientos del sistema que va a soportar la organización.

Con el cumplimiento de los mismos, es factible la obtención de una visión de la organización, lo cual permite definir los procesos de la misma, los roles y responsabilidades a través de los modelos de casos de uso y objetos, del negocio. [4]

2.2 Modelo del Negocio Actual

En la actualidad, la Facultad 3, cuenta con varios Vise-Decanatos. Entre ellos, el Vise-Decanato que atiende el área de Residencia. Como máximo autoridad, el Vise-Decano de Extensión Universitaria, es el encargado de llevar el control y la organización de la Residencia.

Para su logro necesita, el control de los apartamentos de la Residencia, dicho control esta basado en la situación física en la cual se encuentran los mismos así como la capacidad de explotación a la cual esta sometido, además, esporádicamente, necesita saber el inventario de cada uno de ellos; otro aspecto a controlar son los estudiantes residentes, los cuales deben responder ante una serie de actividades indispensables (las guardias, tanto estudiantil como de cuartería y la limpieza de los apartamentos).

Lo anterior esta basado en la ayuda de la Psicopedagoga e Instructoras, encargadas de interactuar con todos los medios básicos y personal, dentro del área de la Residencia.

2.3 Reglas del Negocio a considerar

Las reglas del negocio se simplifican en normas o directrices que rigen los procedimientos de una institución. Toda organización se rige por un conjunto de reglas que pueden variar, en dependencia de su usabilidad y de acuerdo al contexto actual, pero no pueden ser suprimidas pues son las que identifican al centro.

Actualmente el negocio a tratar presentan las siguientes reglas:

- Deben almacenarse todos los datos de los estudiantes y de los apartamentos.
- No pueden existir dos apartamentos con el mismo número.
- No se permite la existencia de dos residentes estudiantiles con la misma identificación (ID).
- Un residente estudiantil no puede estar registrado en más de un apartamento.
- Los resultados de las evaluaciones de los estudiantes lo informa la Psicopedagoga.

- Un residente estudiantil no puede tener dos evaluaciones el mismo día.
- Solo puede cambiar la evaluación la Psicopedagoga.
- Los apartamentos no pueden tener más de la cantidad de residentes establecido en su capacidad.
- No es permitida la convivencia de residentes estudiantiles de diferentes sexos en mismo apartamento.

2.4 Actores del Negocio

Actor del negocio es aquel 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.

Nombre del Actor	Descripción
Vise-Decano	Iniciador de los procesos para el control de la Residencia, (Control de Apartamentos y Control de Estudiantes). Gestiona toda la Información final.

2.5 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.

Nombre del Trabajador	Descripción
Instructora	Participa en el proceso de Control de Estudiantes y en el de Control de Apartamentos. Encargada de crear los informes preliminares, para la evaluación de

	los Residentes Estudiantiles y el informe general del estado de los apartamentos, junto a su inventario de medios básicos.
Psicopedagoga	Participa en el proceso de Control de Estudiantes. Encargada de crear el Informe sobre las evaluaciones de los Residentes Estudiantiles.
Estudiante	Participa en el proceso de Control de Estudiantes. Encargado de crear una opinión, acerca de su propia evaluación (como Residente Estudiantil), para lograr el desarrollo del Informe de la Psicopedagoga, acerca de las evaluaciones de los Residentes Estudiantiles.

2.6 Diagrama de Caso de Uso del Negocio

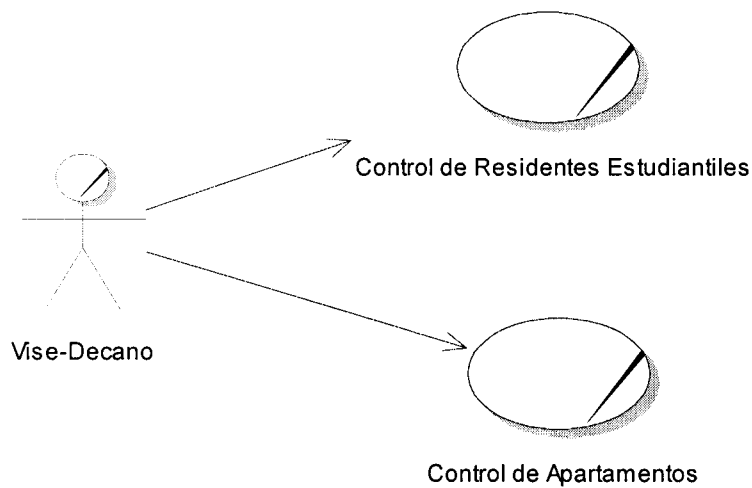


Figura 3

2.7 Casos de Uso del Negocio

La realización de un caso de uso del negocio muestra cómo colaboran los trabajadores y entidades de negocio para ejecutar el proceso. Consideramos que los diagramas de actividad y la descripción textual describe completamente el proceso de negocio.

Un diagrama de actividad describe un proceso que explora el orden de las tareas o actividades que logran los objetivos del negocio. Un grafo de actividades es como un organigrama tradicional, excepto que permite control de concurrencia además de control secuencial: una gran diferencia.

2.7.1 Caso de uso: Control Residentes Estudiantiles

Caso de Uso	Control de Residentes Estudiantiles
Actores	Vise-Decano.
Propósitos	Generar un Informe con las Evaluaciones de los Residentes Estudiantiles.
Resumen	
<p>El caso de uso se inicia cuando el Vise-Decano solicita crear un listado sobre las evaluaciones de los Residentes Estudiantiles; la Psicopedagoga pide un informe, a las Instructoras, sobre las distintas actividades realizadas por los Residentes; estas Instructoras verifican los listados de las diferentes actividades y entregan un listado de evaluaciones a la Psicopedagoga; esta última recopila toda la información y envía un listado de evaluaciones, revisado, a el Vise-Decano, el cual termina el proceso generando un informe sobre los Residentes.</p>	

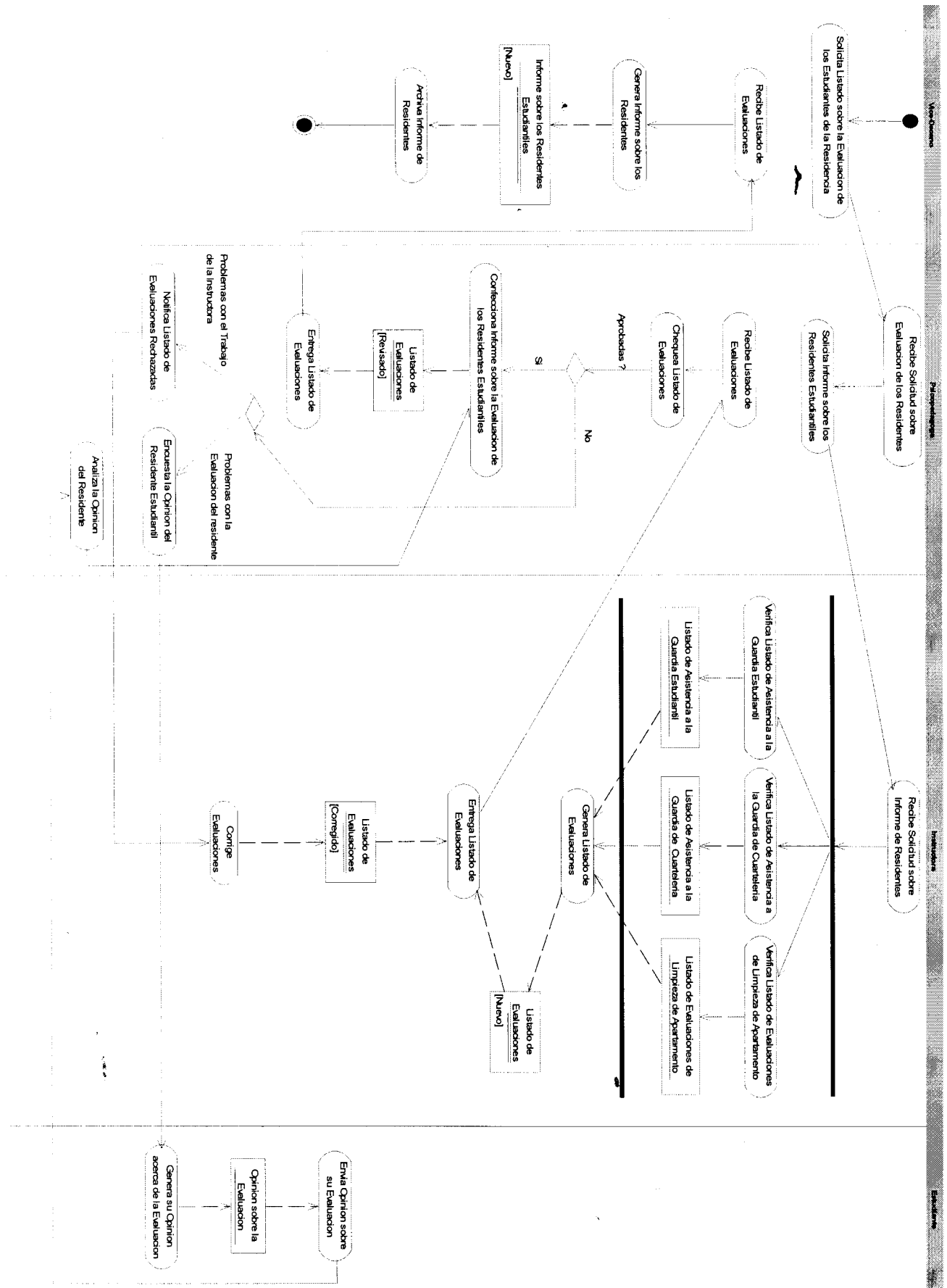


Figura 4

2.7.2 Caso de uso: Control de Apartamentos

Caso de Uso	Control de Apartamentos
Actores	Vise-Decano. }
Propósitos	Generar un Informe sobre el Estado de los Apartamentos de la Residencia Estudiantil.
Resumen	
<p>El caso de uso se inicia cuando el Vise-Decano solicita un informe, a las instructoras, sobre la situación actual de los apartamento en la Residencia Estudiantil; las Instructoras, verifican los listados que brindan la información de los apartamentos, tanto física, como de explotación, y además puede brindar el conocimiento sobre el Inventario de los mismos, con esto recopila toda la información necesaria y envía un Informe preliminar, sobre los apartamentos, al Vise-Decano, el cual chequea el mismo y confecciona un Informe final, sobre la situación actual de los mismos, para posteriormente archivarlo.</p>	

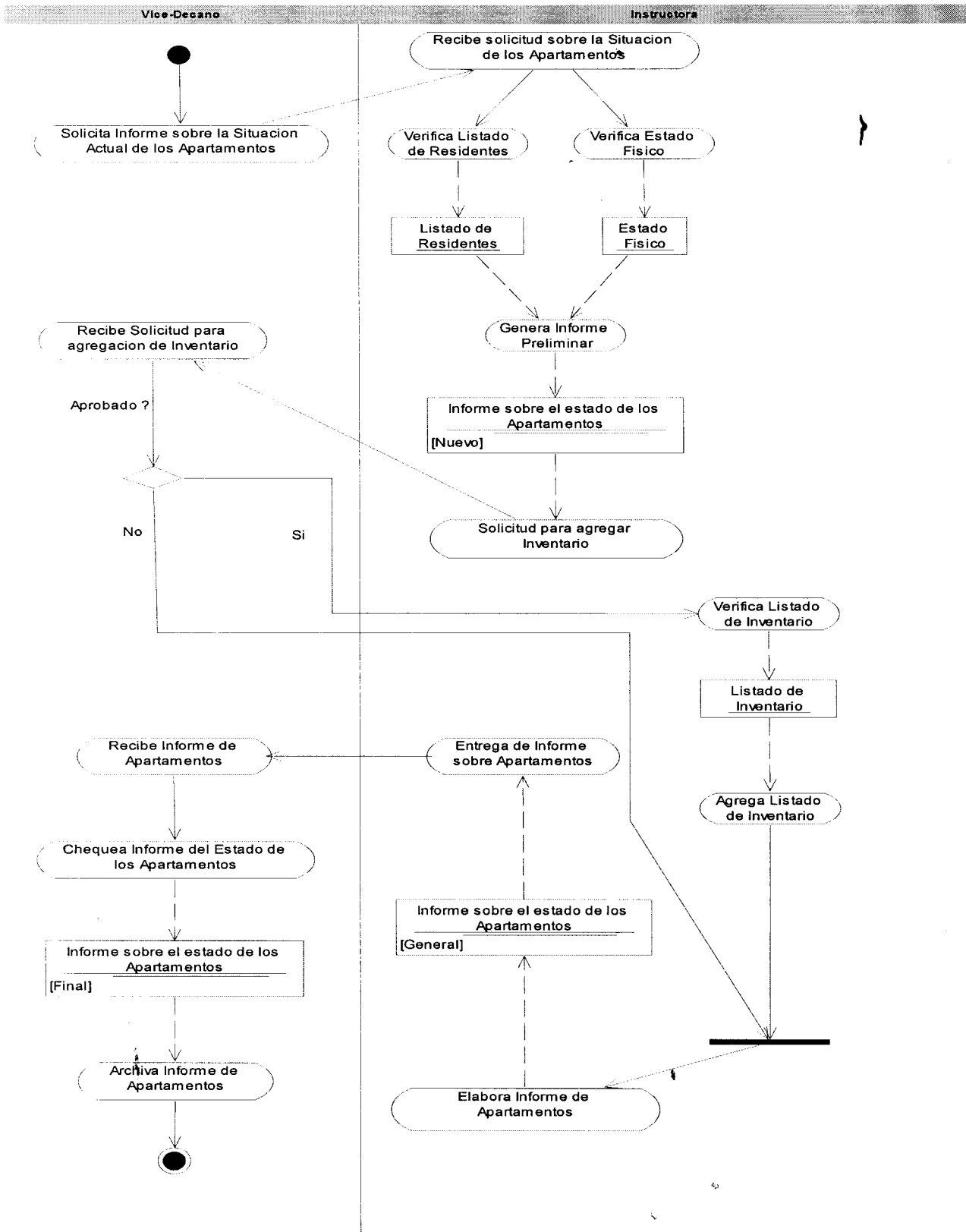


Figura 5

2.8 Modelo de Objetos

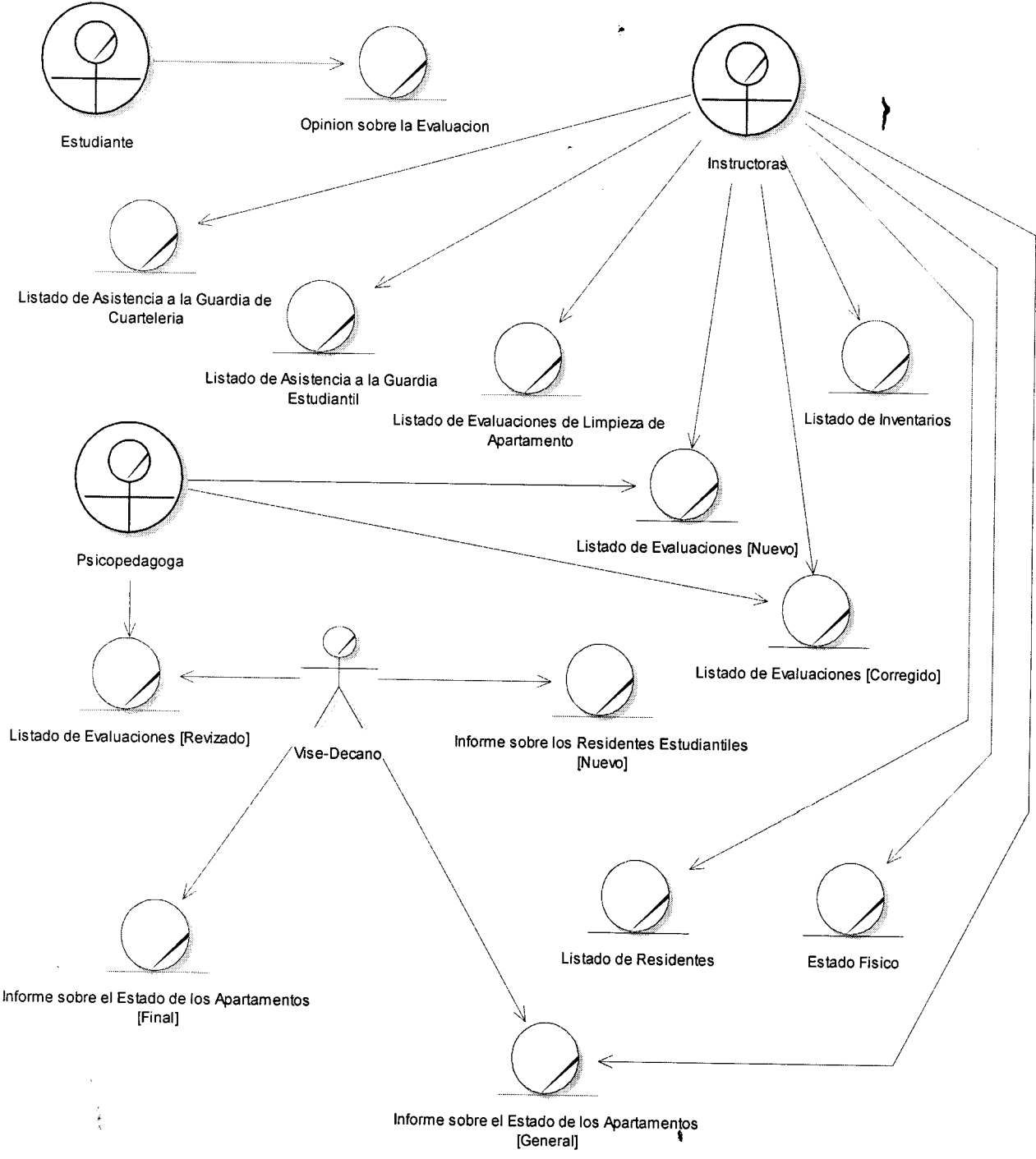


Figura 6

2.9 Conclusiones

En este capítulo se realizó un estudio detallado del negocio a tratar, aportando el entendimiento de la dinámica, estructura de la organización y del problema a resolver por el software propuesto. Lo anterior nos brindará la eficiencia necesaria para lograr un producto profesional, en el posterior desarrollo de la solución.

Como resultado del estudio surgió un actor del negocio y dos casos de uso. Además facilitó la definición para los requerimientos del sistema que brindará la solución a desarrollar.

Capítulo III: Requisitos y Descripción del Sistema Propuesto

3.1 Introducción

Este capítulo pretende recoger toda una serie de requisitos que son imprescindibles para la futura construcción de la solución a proponer como respuesta a la problemática planteada en capítulos anteriores.

Para lograr una clara comprensión de lo que se pretende con el sistema a desarrollar se utilizarán facilidades brindadas por la metodología RUP, de esta forma se han definido varios actores del sistema, se han creado varios paquetes, con sus relaciones definidas, aportando así los diagramas, con los diferentes casos de uso, necesarios para una mejor visualización de las funcionalidades que se pretenden automatizar. Lo anterior estará acompañado con la inclusión de los requisitos no funcionales del sistema, muy útiles, para una buena concepción del diseño a proponer.

3.2 Definición de Requisitos no Funcionales

Los requerimientos no funcionales son propiedades o cualidades que el sistema debe tener. Son las características que hacen al producto atractivo, usable, rápido o confiable. En muchos casos son fundamentales en el éxito del producto. Están vinculados a requerimientos funcionales, es decir una vez que se conozca lo que el sistema debe hacer podemos, determinar cómo ha de comportarse, qué cualidades debe tener o cuán rápido o grande debe ser.

Los requerimientos no funcionales forman una parte significativa de la especificación. Son importantes para que los clientes y usuarios puedan valorar las características no funcionales del producto, pues si se conoce que el mismo cumple con la toda la funcionalidad requerida, las propiedades no funcionales, pueden marcar la diferencia entre un producto y otro.

3.2.1 Apariencia o Interfaz Externa

La apariencia en todo sistema es muy importante a la hora de presentar la información al usuario final. El objetivo que se persigue con nuestro sistema es recoger en una misma interfaz una serie de servicios, lo cual conlleva al trabajo con mucha información y se hace necesario presentar la misma con un orden lógico que posibilite al usuario una navegación fácil evitando la desorientación y el cansancio.

Es una aplicación web la cual debe ser diseñada con colores adecuados, una tipografía de letra legible y de buen tamaño, las ventanas deben tener las dimensiones justas para la información que se desea exponer y la estructura del sistema como tal debe realizarse con el objetivo de hacer un buen uso del poco espacio de la pantalla del ordenador. El sistema debe cumplir todas las características citadas con anterioridad para lograr que sea una herramienta de ayuda al usuario y no otro obstáculo más en la obtención de información.

3.2.2 Usabilidad

El sistema está diseñado para ser utilizado por cualquier tipo de usuario con conocimientos básicos de computación, está pensado inicialmente orientado a la Intranet universitaria así que los principales usuarios serán estudiantes, personal docente y directivos; aunque en el futuro puede ser utilizado por otra institución. En cuanto a su aprovechamiento debe quedar claro que el sistema brindará una serie de servicios de notificación lo más flexible posible para que cada usuario los pueda personalizar y solo llegue a él lo que realmente es de su interés.

3.2.3 Rendimiento

La aplicación está concebida para una estructura cliente – servidor con una comunicación entre ellos a través de objetos remotos, por lo que debe garantizarse que el procesamiento de la información y las respuestas del sistema sean lo más rápido posible, con el objetivo de minimizar el tiempo que media entre el cambio de la información de interés para el usuario y la notificación a éste. Se sigue una lógica de

funcionamiento entre ambos para que cada cual realice las funcionalidades que mejor correspondan. Por otra parte debe manejarse la concurrencia pues habrá varios usuarios en línea o conectados además de estar controlando los distintos tipos de orígenes de información, lo cual requiere llevar a cabo una rigurosa implementación.

3.2.4 Soporte

A la hora de analizar el soporte de un sistema de software se deben valorar dos aristas fundamentales: la primera relacionada con el refinamiento del sistema y la rapidez de recuperación ante los errores y la segunda orientada a la incorporación de nuevas funcionalidades al mismo.

Nuestro sistema está diseñado para garantizar una extensión progresiva de su funcionalidad. Con esta premisa, el volumen de trabajo para los responsables en el desarrollo del sistema disminuye un poco ya que se concentran principalmente en la corrección de errores ("bug fix") y funcionalidades no integradas, logrando de esta manera el enriquecimiento del sistema.

3.2.5 Portabilidad

La portabilidad de un sistema se refiere a su capacidad de ejecutarse en distintos sistemas operativos sin necesidad de hacer grandes modificaciones en el código fuente. Las aplicaciones portables son capaces de llegar a un mayor número de personas ya que se sobrepone a las barreras creadas por los dominios de los sistemas operativos. Nuestro sistema correrá sobre Windows 2000 o superior y pudiera ejecutarse sobre Linux, ya que se encuentra implementado con tecnología .Net, que gracias a los desarrolladores del proyecto Mono ya puede ejecutarse en Sistemas Operativos Linux.

3.2.6 Seguridad

La seguridad en aplicaciones .Net es un tema del que se ha escrito bastante desde la salida de la tecnología al mercado, brindando a los desarrolladores una serie de buenas prácticas que se deben seguir a la hora de escribir aplicaciones de este tipo, las cuales serán utilizadas para programar nuestro sistema.

La entrada al sistema requiere de autenticación con el objetivo de mostrarle a cada usuario la información correspondiente y el tratamiento de los password se hace con encriptación.

Se ha de garantizar un control riguroso sobre la información teniendo en cuenta los niveles de acceso. No se permitirán accesos sin autorización al sistema, para ello se definirá una política de usuarios con roles y privilegios diferentes para que la información pueda ser consultada por las personas autorizadas.

La integridad de los datos que se almacenen en el servidor deberá ser consistente. Se validarán las entradas de datos irreales y mecanismos de vuelta atrás en procesos críticos que terminen abruptamente y produzcan estados inconsistentes de la información.

3.2.7 Confiabilidad

Con el empleo de IIS 6.0, se ofrece una mayor protección e incremento en la confiabilidad de las aplicaciones Web. De manera predeterminada, IIS ejecutará todas las aplicaciones en un proceso común o agrupado que está separado de los procesos del núcleo de IIS. Además, también puede aislar aplicaciones cuyas misiones sean críticas y que se deban ejecutarse fuera tanto de los procesos de núcleo de IIS como de los agrupados.

El sistema se mantendrá constantemente ofreciendo servicio. Está concebido para que los cambios y nuevas versiones de los servicios correspondientes a cada origen de información no requieran detener el sistema. En casos excepcionales que se necesite poner el sistema fuera de servicio se avisará con anterioridad a los usuarios.

3.2.8 Ayuda y Documentación en Línea

Por la sencillez de la aplicación no es necesaria la implementación de una ayuda para el sistema. Cualquier usuario por muy poca habilidad informática que tenga, si domina los conocimientos mínimos de navegar por una aplicación Web, los cuales se asumen conocidos, por ser este un sistema desarrollado dentro de un medio totalmente inmerso en la rama informática, como es la UCI, puede hacer uso del sistema; por demás se implementó lo más sencillo posible, brindando la claridad y viabilidad necesarias para el uso de un usuario mínimo.

3.2.9 Requerimientos de Software

- Cliente: Navegador Internet Explorer, Netscape Navigator en sus versiones 4.0 o superior.
- Servidor: Internet Information Server, SQL Server y Windows 2000 Advanced Server o Windows 2003 Server. Framework 2.0 de Microsoft.

3.2.10 Requerimientos de Hardware

El servidor donde se decida correr la aplicación servidor debe cumplir con las características enunciadas a continuación: un Pentium III o superior, con 256 MB o más de memoria RAM, conectividad. En el caso del cliente: un Pentium II o superior, con 64 MB o más de memoria y conectividad para el establecimiento de la comunicación con el servidor.

3.3 Concepción General del Sistema

Cuando cualquier usuario comienza a hacer uso del sistema, a través de la Intranet de la UCI, el mismo no necesitará autenticarse para ver la información que ya está publicada. Esta información se refiere inicialmente a través de una interfaz integrada por varias variantes de navegación que brindaran todas las funcionalidades del mismo, basadas fundamentalmente en espacios relacionados con la visualización de una gama de reportes relacionados a la actividad residencial de cada uno de sus residentes,

específicamente sus evaluaciones en las diferentes actividades: Guardia Estudiantil, Guardia de Cuartelería y Limpieza de Cuarto, pues el sistema asume que todo uso inicial es realizado por un residente estudiantil y no un directivo de la Residencia. Para la obtención final de estos reportes, antes mencionados, solo debe ingresar sus datos personales para obtener los resultados específicos. En caso de no ingresar los datos, el sistema le informará la necesidad de los mismos para poder brindar una respuesta satisfactoria.

No obstante la forma básica de respuesta del sistema, ante cualquier usuario, estos pueden pedir autenticarse, para acceder a otras funcionalidades, específicas para los diferentes roles: Vise-Decano, Psicopedagoga e Instructora. Una vez autenticados, tendrán acceso a una interfaz, específica para cada rol, la cual les facilitará algunas funcionalidades.

Tendrán en común, al igual que los usuarios comunes (residentes), la posibilidad de visualizar reportes, pero no solo de los comentados anteriormente, esta vez surgirán nuevas posibilidades de reportes específicas para cada rol. Aclarando que el rol de Vise-Decano, tendrá autorización para visualizar todos los reportes generados por el sistema; el rol Psicopedagoga, solo tendrá acceso a los reportes relacionados con los Residentes (emitidos por el rol Instructora) y con su propio trabajo (análisis de evaluaciones); de la misma forma el rol Instructora accederá a los reportes de residentes y de control de apartamentos, incluyendo los relacionados con el control de inventario de los mismos. Además estos tres roles, podrán generar reportes con la característica de poder enviarlos por correo electrónico a un destinatario especificado. Cada rol generará los reportes específicos a su trabajo:

- Psicopedagoga: Generar reportes relacionados con sus análisis sobre las evaluaciones de las Instructoras y sobre las opiniones de los residentes estudiantiles.
- Instructora: Generar reportes sobre el control de los apartamentos, su inventario y las evaluaciones sobre los residentes estudiantiles, que incluyen las actividades de guardias y limpieza de apartamentos.

- **Vise-Decano:** Acceso a generar todo tipo de reportes ofrecidos por el sistema, además de los reportes propios de su rol, como son la asignación de los residentes a las disponibilidades de la Residencia, el control de las ubicaciones y disponibilidades de los apartamentos, además del control de los datos personales de los residentes.

Acto seguido de la explicación sobre los elementos comunes a cada una de las interfaces relacionadas con los diferentes roles, detallaremos las demás funcionalidades del sistema separadas por el tipo de rol:

- **Vise-Decano:** Esta interfaz cuenta además con la posibilidad de insertar edificios, sus características y definir en ellos su cantidad de apartamentos con los detalles específicos para cada uno de estos últimos, creando toda una estructura de ubicación-disponibilidad. Incluye también, la inserción de los residentes estudiantiles y sus características, facilitando la construcción de la estructura de recursos humanos. Se le asigna la responsabilidad de administrador del sistema a este rol, por lo cual, puede acceder a opciones que le permitan gestionar los usuarios, roles y asignaciones de derechos de roles-usuarios, pertenecientes al sistema.
- **Psicopedagoga:** Interfaz que brinda la posibilidad de visualizar cada una de las evaluaciones emitidas por las Instructoras y además de las opiniones emitidas por los residentes estudiantiles. Incluye la capacidad de aceptar la inserción de análisis evaluativos sobre cada una de las evaluaciones u opiniones revisadas, para de esta forma lograr insertar toda una lista de nuevas evaluaciones, resultado de las visualizaciones previas.
- **Instructora:** Debido a la gran carga de trabajo que sustenta este rol dentro del sistema, dicha interfaz se visualizará con un número de funcionalidades mayor que las anteriormente explicadas.
 - Se brinda la posibilidad de controlar los apartamentos, lo cual incluye accesos para modificar la capacidad de residentes, de medios básicos (inventario) y estructura física, así como la de generar un calendario de limpieza para los residentes.

- Controlar las diferentes guardias, o sea definir las características de cada una para poder además generar sendos calendarios que estructuren el funcionamiento de cada una de esas guardias.
- Controlar a los residentes, apoyada en la posibilidad de hacer búsquedas de los mismos, para posterior inserción de cada una de las evaluaciones en actividades de guardias y limpieza de cuarto.

3.4 Actores del Sistema a Automatizar

Un actor representa a cualquier elemento que interactúa con el sistema, que puede: intercambiar información con él, ser un recipiente pasivo de información, representar el rol que juega una o varias personas, un equipo o un sistema automatizado. Estas operaciones pueden ser realizadas por una persona, un software, o un hardware.

A continuación se describe los actores identificados en el sistema:

Nombre del Actor	Descripción
Vise-Decano	Responsable de gestionar diferentes procesos, como los de: Control de Residentes y Seguridad. Así como generar reportes asociados a sus diferentes actividades de gestión.
Psicopedagoga	Responsable de analizar las diferentes evaluaciones emitidas, sobre los Residentes, por las Instructoras y de esta forma crear su propia evaluación. Así mismo es responsable de generar los reportes necesarios, asociados a su responsabilidad.
Instructora	Responsable mayoritaria del control de la Residencia, enfatizando este control específicamente

	sobre las disponibilidades (apartamentos), las diferentes guardias estudiantiles y los residentes. Generando de esta forma una amplia gama de reportes para mantener un mayor control sobre la Residencia.
Residente	Responsable mayoritario de consultar la mayor parte de la información del sistema así como de emitir su opinión acerca de las diferentes evaluaciones a las que pudiese estar sometido.

3.5 Modelo de Casos de Uso del Sistema

Los casos de uso forman parte de los artefactos del sistema, son fragmentos de funcionalidad que el sistema ofrece para aportar un resultado a los actores, sus atributos representan valores que una instancia de caso de uso utiliza durante su ejecución, estas instancias se consideran atómicas, es decir indivisibles. Con un caso de uso se representa cada forma en que los actores usan el sistema y se especifica una secuencia de acciones que el sistema puede llevar a cabo interactuando con sus actores.

Para una mejor organización de los elementos podemos agruparlos a través de paquetes, así se subdivide el modelo en otros más pequeños que colaboran entre sí.

La descripción de los casos de uso se hace para describir en detalle el flujo de sucesos, de una manera precisa pero a la vez fácil de leer. Se describen todas las excepciones y alternativas, incluyéndose pre y poscondiciones. Es importante definir cómo comienza, termina e interactúa cada caso de uso con sus actores.

UML introduce la noción de un paquete como el ítem universal para agrupar elementos, permitiendo a los modeladores subdividir y categorizar sistemas. Los paquetes pueden ser usados en cualquier nivel, desde el nivel más alto, donde son

usados para subdividir el sistema en dominios, hasta el nivel más bajo, donde son usados para agrupar casos de uso individuales, clases, o componentes.

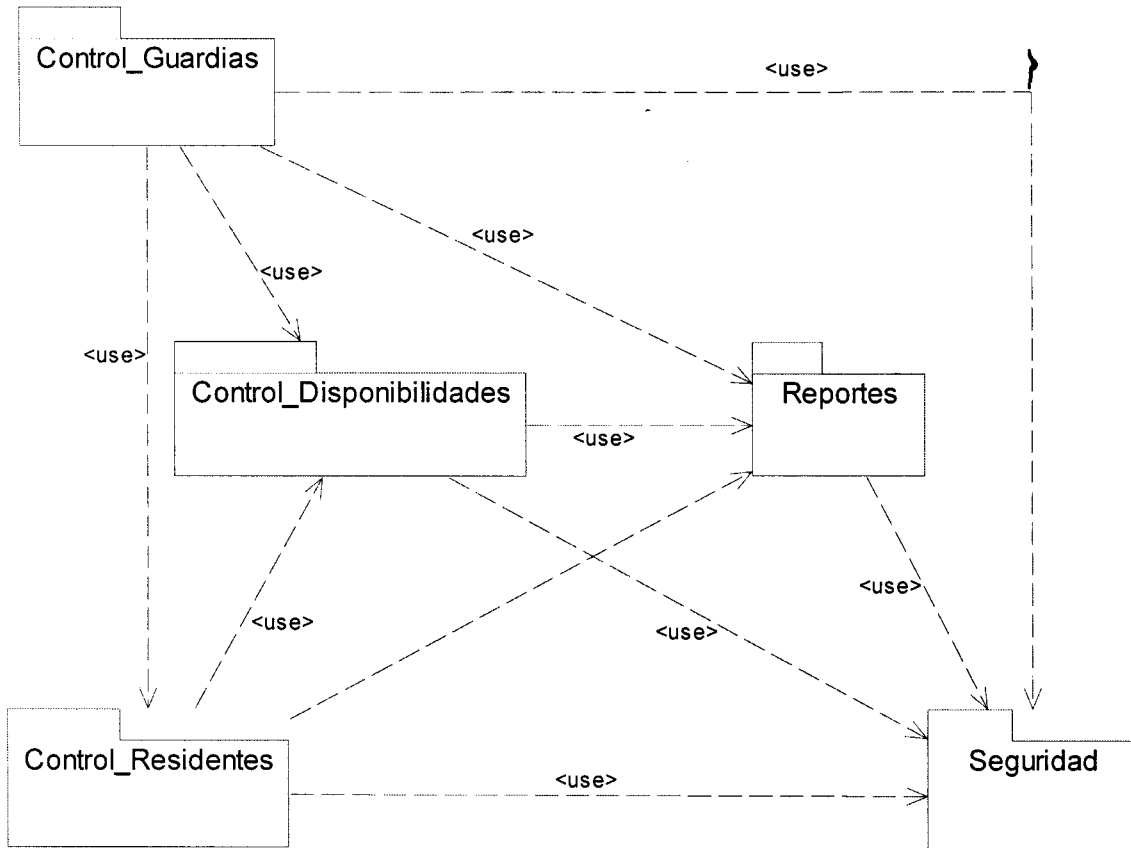


Figura 7

3.5.1 Paquete: Control de Disponibilidades

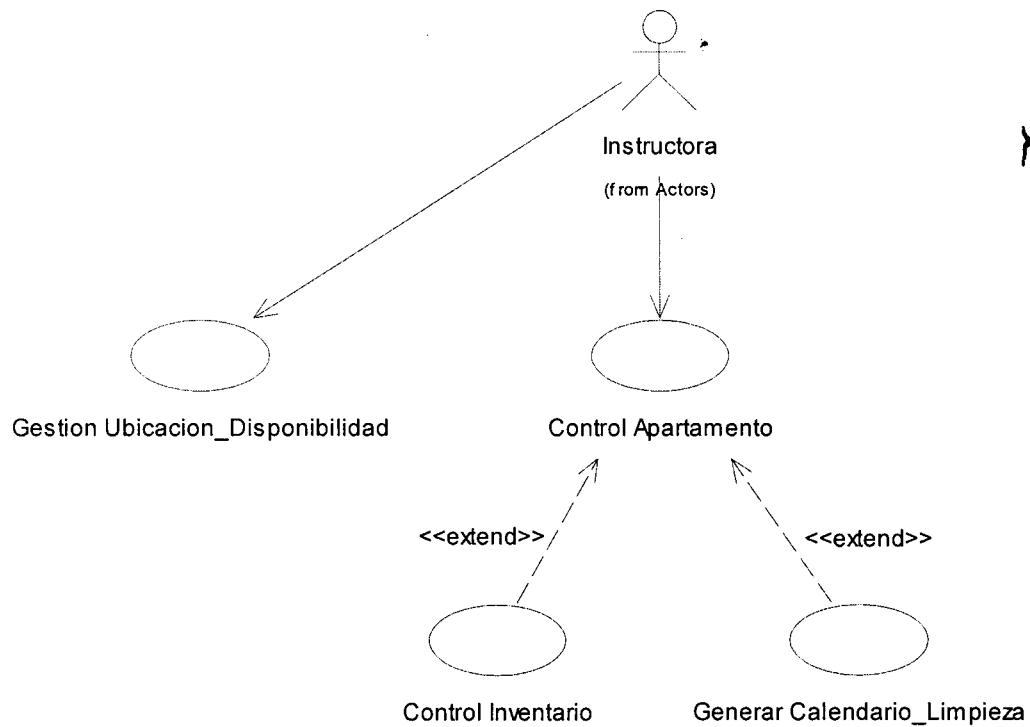


Figura 8

3.5.2 Paquete: Control de Guardias

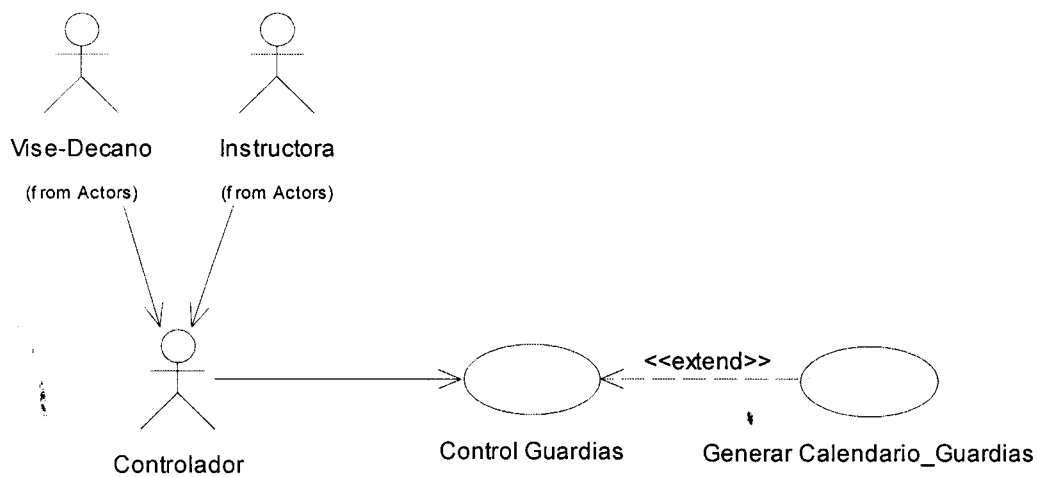


Figura 9

3.5.3 Paquete: Reportes

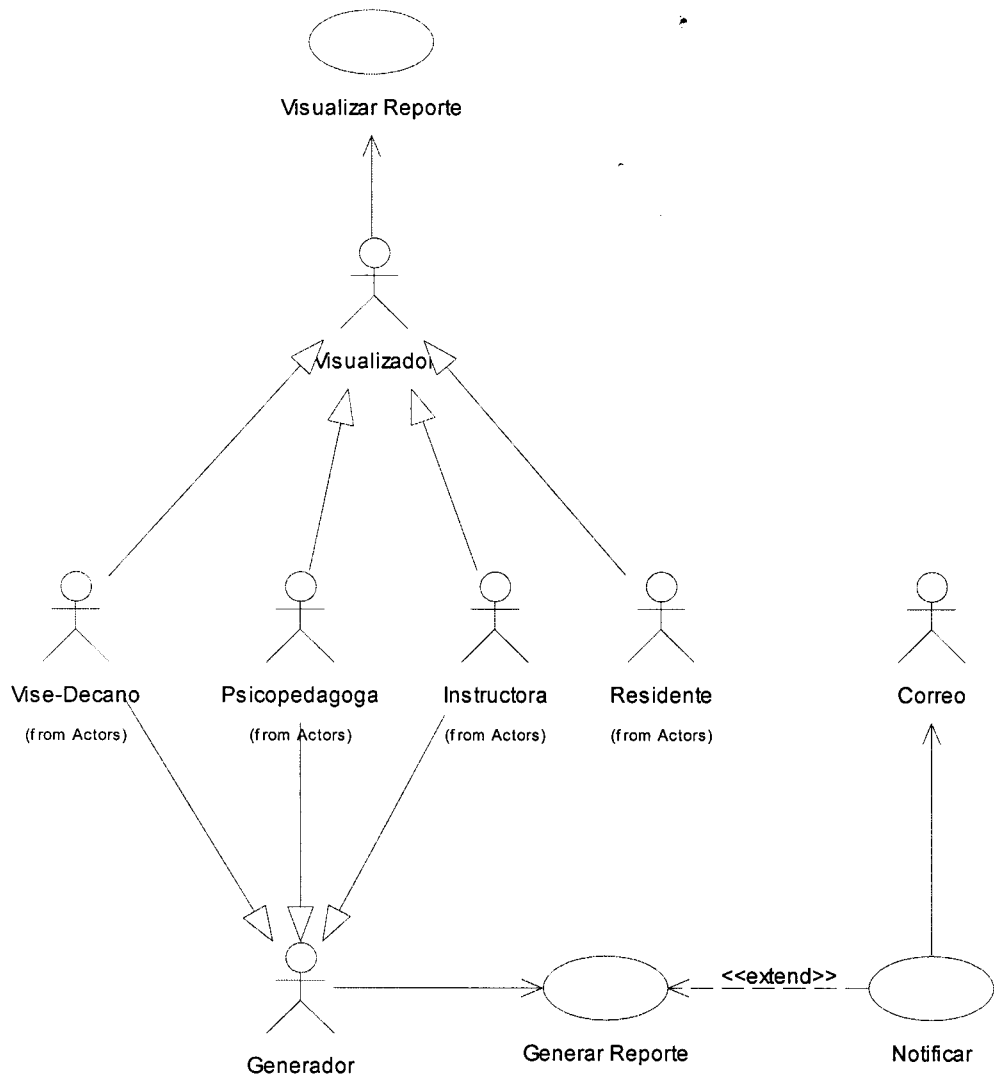


Figura 10

3.5.4 Paquete: Control de Residentes

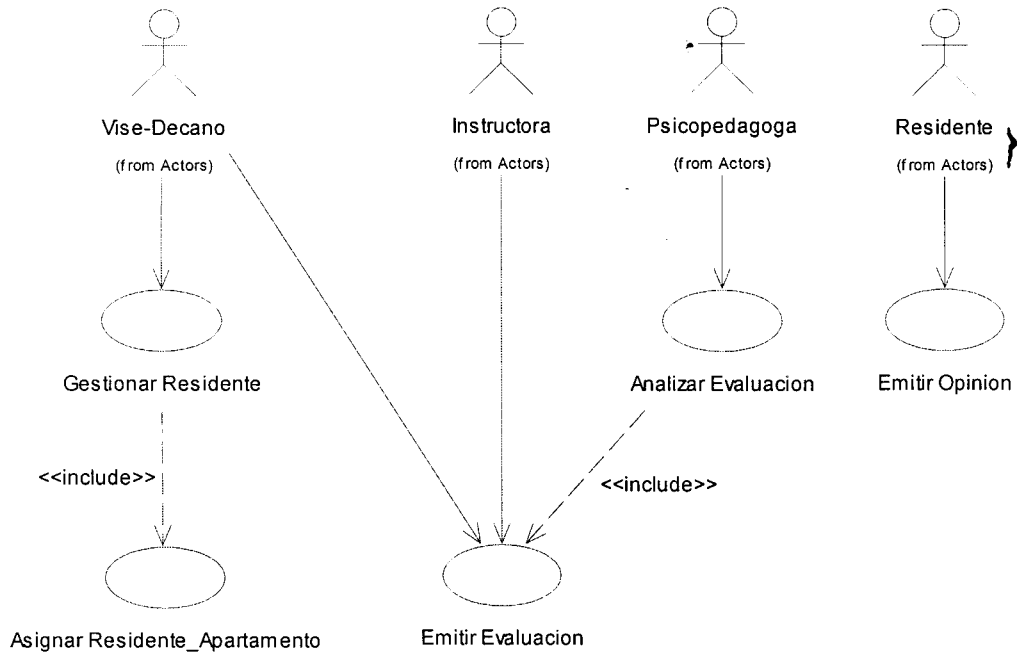


Figura 11

3.5.5 Paquete: Seguridad

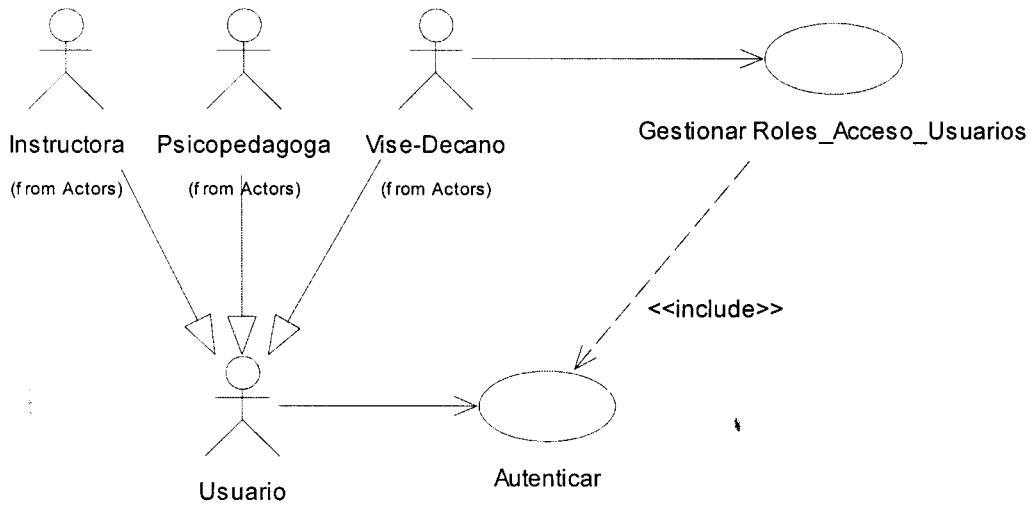


Figura 12

3.6 Descripción de los Casos de Uso del Sistema

Nombre del caso de uso	Autenticar.
Actor(es)	Vise-Decano, Psicopedagoga, Instructora.
Resumen	<p>Como característica, los tres actores inician el caso de uso como un actor genérico (Usuario).</p> <p>Comienza cuando el Usuario solicita autenticarse, el sistema le muestra una ventana para que especifique su login y su password, el usuario entra sus datos y se autentica para algunos servicios adicionales. El sistema realiza la conexión con el Servidor y chequea que las credenciales sean correctas y termina permitiendo la entrada del usuario al sistema y muestra los servicios no comunes al sistema, asociados con el tipo de rol asociado al usuario. En caso de ser incorrectas las credenciales muestra la ventana de autenticación con un mensaje de error.</p>
Precondiciones	Que el servidor se encuentre activo.
Poscondiciones	Mostrar a los diferentes tipos de usuarios autenticados, según su rol, los diferentes servicios asociados al sistema, que no forman parte del sistema para un usuario común.
Requisitos Especiales	<p>Es necesario que la conexión con el servidor se realice de la forma más eficiente posible para garantizar la velocidad del flujo de datos y poder disminuir al máximo el tiempo de respuesta al usuario.</p> <p>Que el Usuario exista en el Sistema.</p>

Nombre del caso de uso	Gestionar Roles_Acceso_Usuarios.
Actor(es)	Vise-Decano.
Resumen	<p>Comienza de dos formas:</p> <ol style="list-style-type: none"> 1. Cuando el actor solicita gestionar los roles del sistema, de esta forma logra insertar, modificar o eliminar a los mismos; asignar a cada rol los servicios a los que puede acceder dentro del sistema y los usuarios que lo integran. Para esto el sistema debe realizar la conexión con el Servidor, consultar y mostrar todos los servicios no comunes al sistema y de esos cuáles están asociados con el tipo de rol (en caso de que se este modificando al mismo). 2. Cuando el actor solicita gestionar un usuario del sistema. Cuando esto ocurre, si se desea insertar un nuevo usuario el sistema debe mostrar el listado de roles para poder asignarle uno de ellos. En caso de que se esté modificando el usuario del sistema, pues el mismo brindará un listado con los roles, para una nueva asignación.
Precondiciones	Que el servidor se encuentre activo y el usuario este autenticado como Vise-Decano.
Poscondiciones	Quedan actualizados los roles y usuarios.
Requisitos Especiales	Es necesario que la conexión con el servidor se realice de la forma más eficiente posible para garantizar la velocidad del flujo de datos y poder disminuir al máximo el tiempo de respuesta al usuario.

Nombre del caso de uso	Gestión Ubicación_Disponibilidad.
Actor(es)	Instructora.
Resumen	<p>Comienza por dos vías diferentes:</p> <ol style="list-style-type: none"> 1. Cuando el actor desea crear la estructura de un edificio con sus apartamentos dentro de la Residencia. Para esto llena un formulario con la información, referente al nuevo edificio y sus apartamentos, sobre su ubicación y disponibilidad. De esta forma el sistema agrega el edificio a la lista de disponibilidades existentes. 2. Cuando el actor desea modificar la ubicación o disponibilidad de un edificio en general ó de un apartamento específico. Para esto el sistema consulta la lista de disponibilidades y le muestra al actor todos los edificios existentes. En dependencia de las modificaciones se elimina o modifica dicho edificio o apartamento, de la lista de disponibilidades.
Precondiciones	Que el servidor se encuentre activo y el usuario este autenticado como Instructora.
Poscondiciones	Quedan almacenados y actualizados los registros de disponibilidades y ubicación.
Requisitos Especiales	Es necesario que la conexión con el servidor se realice de la forma más eficiente posible para garantizar la velocidad del flujo de datos y poder disminuir al máximo el tiempo de respuesta al usuario.

Nombre del caso de uso	Control Apartamento.
Actor(es)	Instructora.
Resumen	Comienza cuando el actor desea controlar el apartamento. Para esto el sistema debe brindar una serie de datos relacionados al apartamento, para que el actor tenga la información referente al mismo y pueda decidir si necesita generar algún reporte (Caso de uso: Generar Reportes), actualizar la información brindada acerca de la capacidad de residentes, actualizar el estado físico del inmueble, posibilidad de hacer el inventario al apartamento (Caso de uso: Control Inventario) o de gestionar el calendario de limpieza (Caso de uso: Generar Calendario_Limpieza). De esta forma el servidor responde a cada una de las acciones necesitadas por el actor y actualiza la información relacionada al apartamento.
Precondiciones	Que el servidor se encuentre activo; el usuario este autenticado como Instructora y existan los registros de ubicación y disponibilidades.
Poscondiciones	Quedan almacenados y actualizados los registros referentes a cada apartamento.
Requisitos Especiales	Es necesario que la conexión con el servidor se realice de la forma más eficiente posible para garantizar la velocidad del flujo de datos y poder disminuir al máximo el tiempo de respuesta al usuario.

Nombre del caso de uso	Control Inventario.
Actor(es)	Instructora.
Resumen	Comienza cuando el actor desea controlar el inventario de un

	<p>apartamento. Para esto el sistema debe brindar formularios que brinden la información relacionada a cada medio básico existente, brindando la posibilidad al actor de cambiar, insertar o eliminar el tipo de medio, la cantidad de los mismos y el estado físico en que se encuentra el mismo. Una vez terminado, se envían los datos al servidor y se actualiza el inventario con la posibilidad de generar un reporte de los cambios o del estado anterior (Caso de uso: Generar Reportes).</p>
Precondiciones	<p>Que el servidor se encuentre activo; el usuario este autenticado como Instructora y se conozca el apartamento a analizar.</p>
Poscondiciones	<p>Quedan almacenados y actualizados los registros referentes al inventario para el apartamento indicado.</p>
Requisitos Especiales	<p>Es necesario que la conexión con el servidor se realice de la forma más eficiente posible para garantizar la velocidad del flujo de datos y poder disminuir al máximo el tiempo de respuesta al usuario.</p>

Nombre del caso de uso	<p>Generar Calendario_Limpieza.</p>
Actor(es)	<p>Instructora.</p>
Resumen	<p>Comienza cuando el actor desea generar un calendario de limpieza. Para esto se le brinda al sistema el apartamento, para poder definir la cantidad de Residentes existentes en el mismo y asignar a cada día de la semana uno de ellos, para que realice la limpieza del apartamento. Esta información generada se archiva en el servidor y se brinda la posibilidad de hacer un reporte sobre el calendario generado (Caso de uso: Generar Reporte).</p>
Precondiciones	<p>Que el servidor se encuentre activo; el usuario este autenticado</p>

	como Instructora y se conozca la cantidad de Residentes en el apartamento a analizar.
Poscondiciones	Se crea y almacena el calendario de limpieza del apartamento analizado.
Requisitos Especiales	Es necesario que la conexión con el servidor se realice de la forma más eficiente posible para garantizar la velocidad del flujo de datos y poder disminuir al máximo el tiempo de respuesta al usuario.

Nombre del caso de uso	Control Guardias.
Actor(es)	Vise-Decano, Instructora.
Resumen	<p>Comienza cuando el actor desea controlar las guardias de la Residencia. Para esto se le solicita al sistema el tipo de guardia a visualizar o modificar. Actualmente las guardias pueden ser de dos tipos: Guardia Estudiantil y Guardia de Cuartelería.</p> <p>En caso de que el actor que inicie el CU fuese el Vise-Decano, el sistema encuesta al servidor, brindando los formularios con la información referente a la Guardia Estudiantil y la posibilidad de generar un reporte (Caso de uso: Generar Reporte) acerca de la asistencia a las mismas. En caso de cambios se envían los nuevos formularios al servidor para su actualización. Además se puede optar por la opción de visualizar o definir el calendario de cada una de las guardias (Caso de uso: Generar Calendario_Guardias).</p> <p>El mismo flujo anterior, se define si el iniciador del CU fuese la Instructora, pero esta vez enfocado solo a la Guardia de Cuartelería.</p>

Precondiciones	Que el servidor se encuentre activo; el usuario este autenticado como Vise-Decano o Instructora y se conozcan la cantidad y nombre de los todos los inquilinos de la Residencia.
Poscondiciones	Se actualizan los registros referentes a cada tipo de guardia.
Requisitos Especiales	Es necesario que la conexión con el servidor se realice de la forma más eficiente posible para garantizar la velocidad del flujo de datos y poder disminuir al máximo el tiempo de respuesta al usuario.

Nombre del caso de uso	Generar Calendario_Guardias.
Actor(es)	Vise-Decano, Instructora.
Resumen	<p>Comienza cuando el actor desea generar un calendario de guardias. Para esto el sistema brinda la lista de tipos de guardias, el actor escoge la de interés y se generan según el tipo, dos tipos de calendarios:</p> <ol style="list-style-type: none"> 1. EL sistema consulta cada edificio, verifica las disponibilidades del mismo (apartamentos habitables), crea un listado con los residentes de cada apartamento para cada uno de los pasos de escaleras, y les asigna, de forma rotativa, un día de la semana o el mes, en dependencia de la cantidad de residentes, generando así el calendario de Guardia para la Cuartelería, con la especificidad de tener en cuenta que los domingos no puede existir ninguna asignación. Esta variante se inicializa por la Instructora. 2. El sistema consulta a todos los residentes y les asigna a cada uno de forma rotativa un día del mes, creándose con estas características el calendario de la Guardia Estudiantil. Variante

<p>inicializada por el Vise-Decano.</p> <p>Toda la información generada en cada una de las variantes anteriores se archiva en el servidor y se brinda la posibilidad de hacer un reporte sobre el calendario generado (Caso de uso: Generar Reporte).</p>	
Precondiciones	Que el servidor se encuentre activo; el usuario este autenticado como Vise-Decano o Instructora, existan Residentes y que este creada la infraestructura que permita saber la ubicación y disponibilidad de los edificios con sus apartamentos.
Poscondiciones	Se crean y almacenan los calendarios de cada una de las diferentes guardias.
Requisitos Especiales	Es necesario que la conexión con el servidor se realice de la forma más eficiente posible para garantizar la velocidad del flujo de datos y poder disminuir al máximo el tiempo de respuesta al usuario.

Nombre del caso de uso	Gestionar Residente.
Actor(es)	Vise-Decano.
Resumen	Comienza cuando el actor desea crear, modificar o eliminar un residente. Para esto el sistema brinda un formulario con las características requeridas para crear toda la estructura que define a un residente típico de la Institución, como son el nombre, correo electrónico, grupo docente al cual pertenece, entre otros. De esta forma se insertan al sistema los formularios de un Residente nuevo o se actualizan los de uno existente. Además se accionan los mecanismos para asignar a cada residente una ubicación dentro de las disponibilidades de la residencia (Caso de uso: Asignar

Residente_Apartamento).	
Precondiciones	Que el servidor se encuentre activo; el usuario este autenticado como Vise-Decano.
Poscondiciones	Se crearon, actualizaron y almacenaron las informaciones relacionadas a cada residente.
Requisitos Especiales	Es necesario que la conexión con el servidor se realice de la forma más eficiente posible para garantizar la velocidad del flujo de datos y poder disminuir al máximo el tiempo de respuesta al usuario.

Nombre del caso de uso	Asignar Residente_Apartamento.
Actor(es)	Vise-Decano.
Resumen	Comienza cuando el actor desea asignar al residente a uno de los apartamentos de la residencia. Para esto el sistema chequea la lista de disponibilidades (apartamentos), verifica la capacidad de aceptar nuevos residentes y lo ubica en uno de ellos. En caso de que no se desee dar la responsabilidad de asignación al sistema, el mismo brinda la lista de apartamentos disponibles y sus capacidades para una asignación según los criterios del actor. De esta forma se inserta al sistema la ubicación de cada uno de los residentes requeridos.
Precondiciones	Que el servidor se encuentre activo; el usuario este autenticado como Vise-Decano y este creada la infraestructura de edificios y apartamentos de la residencia, para permitir la consulta de la ubicación y disponibilidad de los mismos.
Poscondiciones	Se actualizaron los formularios de capacidad en cada

	apartamento con la inserción del residente.
Requisitos Especiales	Es necesario que la conexión con el servidor se realice de la forma más eficiente posible para garantizar la velocidad del flujo de datos y poder disminuir al máximo el tiempo de respuesta al usuario.

Nombre del caso de uso	Analizar Evaluación.
Actor(es)	Psicopedagoga.
Resumen	Comienza cuando el actor desea analizar cada una de las evaluaciones de un residente. Para esto el sistema chequea el listado de evaluaciones emitida por la Instructora y brinda un formulario con los datos referentes a cada una de las evaluaciones a la que son sometidos los residentes. El actor revisa las evaluaciones o solicita chequear el listado de opiniones de los residentes para realizar la misma acción de revisión mencionada anteriormente. El sistema actualiza las listas de opiniones por revisar de los Residentes y de evaluaciones emitidas por la Instructora y archiva una lista de cambios para la nueva evaluación en caso de que considere existiesen errores en los procesos de revisión. De esta forma posteriormente emite un formulario con su propia evaluación (Caso de uso: Emitir Evaluación).
Precondiciones	Que el servidor se encuentre activo; el usuario este autenticado como Psicopedagoga y exista la lista de evaluaciones emitidas por la Instructora y la lista de opiniones de los residentes, para permitir la consulta de las mismas.
Poscondiciones	La lista de opiniones de los residentes se actualizó y la lista de evaluaciones emitidas por la instructora se chequeo y actualizó.

Requisitos Especiales	Es necesario que la conexión con el servidor se realice de la forma más eficiente posible para garantizar la velocidad del flujo de datos y poder disminuir al máximo el tiempo de respuesta al usuario.
-----------------------	--

Nombre del caso de uso	Emitir Evaluación.
Actor(es)	Instructora, Psicopedagoga, Vise-Decano.
Resumen	<p>Como característica este caso de uso es puede ser iniciado de tres formas:</p> <ol style="list-style-type: none"> 1. Comienza cuando la Instructora desea controlar a un residente y emitir una evaluación sobre el mismo. Para esto el sistema brinda un formulario de evaluaciones para cada uno de los siguientes aspectos: <ul style="list-style-type: none"> ➤ Guardia Cuartelería ➤ Limpieza Apartamento <p>El sistema chequea los listados referentes a los aspectos mencionados y brinda la información referente de cada residente, de esta forma la Instructora crea los formularios referentes a cada evaluación relacionada con los tres aspectos, para cada uno de los residentes. Así el sistema crea y archiva una lista de evaluaciones para cada uno de los Residentes.</p> 2. Comienza una vez que la Psicopedagoga haya realizado un análisis sobre la lista de evaluaciones de la Instructora (Caso de uso: Analizar Evaluación), para poder emitir una evaluación propia. Para esto la Psicopedagoga solicita al sistema la lista de cambios para la nueva evaluación, guiándose por la misma crea

	<p>una nueva lista de evaluaciones, sobre cada uno de los aspectos tratados en la primera forma de inicio de este caso de uso, para cada residente, la cual es archivada por el sistema. Con la posibilidad de generar un reporte sobre dichas evaluaciones (Caso de uso: Generar Reporte).</p> <p>3. Inicia cuando el Vise-Decano, desea evaluar a los Residentes en la Guardia Estudiantil. Para esto el actor solicita al sistema el listado de los Residentes. A cada uno de estos Residentes el sistema adjunta un formulario con las características de la evaluación. Una vez completado los formularios para cada uno de los evaluados, se archivan los cambios en el servidor. Además se brinda la posibilidad de generar un reporte sobre la actividad realizada.</p>
<p>Precondiciones</p>	<p>Que el servidor se encuentre activo; el usuario este autenticado como Instructora, Psicopedagoga o Vise-Decano. Además de existir la infraestructura que permita la consulta de los Residentes.</p>
<p>Poscondiciones</p>	<p>Vía:</p> <ol style="list-style-type: none"> 1. Se crearon, actualizaron y almacenaron las evaluaciones, relacionadas a cada residente, emitidas por la Instructora. 2. Se crearon, actualizaron y almacenaron las evaluaciones, relacionadas a cada residente, emitidas por la Psicopedagoga. 3. Se crearon, actualizaron y almacenaron las evaluaciones, relacionadas a cada residente, emitidas por el Vise-Decano.
<p>Requisitos Especiales</p>	<p>Es necesario que la conexión con el servidor se realice de la forma más eficiente posible para garantizar la velocidad del flujo de datos y poder disminuir al máximo el tiempo de respuesta al usuario.</p>

Nombre del caso de uso	Emitir Opinión.
Actor(es)	Residente.
Resumen	Comienza cuando el actor desea emitir su opinión. Para esto el sistema chequea el listado de evaluaciones emitido por la Instructora para el Residente específico que la solicita. El Residente crea un formulario con su opinión referente a cada una de las evaluaciones a la que es sometido. Este formulario es archivado por el sistema.
Precondiciones	Que el servidor se encuentre activo y exista la lista de evaluaciones emitidas por la Instructora para permitir la consulta de las mismas.
Poscondiciones	Se crea un formulario con la opinión del Residente.
Requisitos Especiales	Es necesario que la conexión con el servidor se realice de la forma más eficiente posible para garantizar la velocidad del flujo de datos y poder disminuir al máximo el tiempo de respuesta al usuario.

Nombre del caso de uso	Visualizar Reporte.
Actor(es)	Vise-Decano, Psicopedagoga, Instructora, Residente.
Resumen	<p>Este caso de uso tiene como característica una generalización de actores, pues todos inician este caso de uso como un actor comentado con el siguiente nombre: Visualizador.</p> <p>Comienza cuando el actor desea visualizar un reporte. Para esto el sistema chequea el tipo de usuario conectado y brinda un listado de todos los reportes a los cuales su rol le permite acceder.</p>

Precondiciones	Que el servidor se encuentre activo y que se encuentren definidos los roles de usuarios.
Poscondiciones	Se visualiza una lista de reportes específicos según el usuario.
Requisitos Especiales	Es necesario que la conexión con el servidor se realice de la forma más eficiente posible para garantizar la velocidad del flujo de datos y poder disminuir al máximo el tiempo de respuesta al usuario.

Nombre del caso de uso	Generar Reporte.
Actor(es)	Vise-Decano, Psicopedagoga, Instructora.
Resumen	<p>Este caso de uso tiene como característica una generalización de actores, pues todos inician este caso de uso como un actor comentado con el siguiente nombre: Generador.</p> <p>Comienza cuando el actor desea generar un reporte. Para esto el sistema chequea el tipo de usuario conectado y brinda un listado de todos los reportes a los cuales su rol permite generar. El actor selecciona el tipo de reporte a generar y brinda al sistema los datos específicos para cada tipo de reporte, de esta forma el sistema genera el reporte especificado y puede brindar la opción de notificarlo por correo (Caso de uso: Notificar).</p>
Precondiciones	Que el servidor se encuentre activo; que se encuentren definidos los roles de usuarios y se suministren los datos específicos para cada tipo de reporte.
Poscondiciones	Se genera un reporte específico según el criterio del usuario.
Requisitos Especiales	Es necesario que la conexión con el servidor se realice de la forma más eficiente posible para garantizar la velocidad del flujo de

	datos y poder disminuir al máximo el tiempo de respuesta al usuario.
--	--

Nombre del caso de uso	Notificar.
Actor(es)	Vise-Decano, Psicopedagoga, Instructora.
Resumen	<p>Este caso de uso tiene como característica una generalización de actores, pues todos inician este caso de uso como un actor comentado con el siguiente nombre: Generador.</p> <p>Comienza cuando el actor desea notificar un reporte. Para esto el sistema solicita el correo electrónico del destinatario y envía el reporte con un formato digital específico.</p>
Precondiciones	Que el servidor del sistema y el de correos se encuentren activos.
Poscondiciones	Se envió un correo electrónico al destinatario especificado.
Requisitos Especiales	Es necesario que la conexión con el servidor se realice de la forma más eficiente posible para garantizar la velocidad del flujo de datos y poder disminuir al máximo el tiempo de respuesta al usuario.

3.7 Conclusiones

La vida de un sistema informático, depende de su capacidad para brindar una respuesta a la necesidad de servicios o funcionalidades a usuarios que utilizarán el producto. De esta forma el cumplimiento de una serie de requisitos del sistema garantizará, finalmente, el éxito del mismo.

En este capítulo se desarrolla la forma en que será concebida la realización del sistema, mediante la exposición de un diagrama integrado por paquetes, agrupados por sus funcionalidades, en los cuales se abunda sobre las especificaciones sobre cada uno de los casos de uso. Estos, junto a los actores, tratan de brindar de forma explícita, un diseño a seguir para la construcción final del sistema a desarrollar.

Capítulo IV: Descripción de la Solución Propuesta

4.1 Introducción

Una vez descrita de forma detallada la propuesta del sistema para dar solución a la problemática planteada y mostrar una serie de requisitos a cumplir este capítulo se centra en las características del diseño y de implementación para la construcción del software.

Los diagramas juegan un papel fundamental como guía para el programador, en este sentido se muestran los diagramas de clases de la aplicación, separados por paquetes, lo que ofrece una visión más detallada de las estructuras a implementar para representar la dinámica del sistema. De estos diagramas se derivan el modelo lógico y físico de datos, ambos acercan al programador al nivel de almacenamiento de la información, materializándose con la creación de la base de datos. Se describe el nivel de presentación de la aplicación trazando los patrones del diseño de la misma y el modelo de despliegue para mostrar como queda distribuido finalmente el sistema.

4.2 Diagrama de Clases del Diseño

4.2.1 Paquete: Seguridad

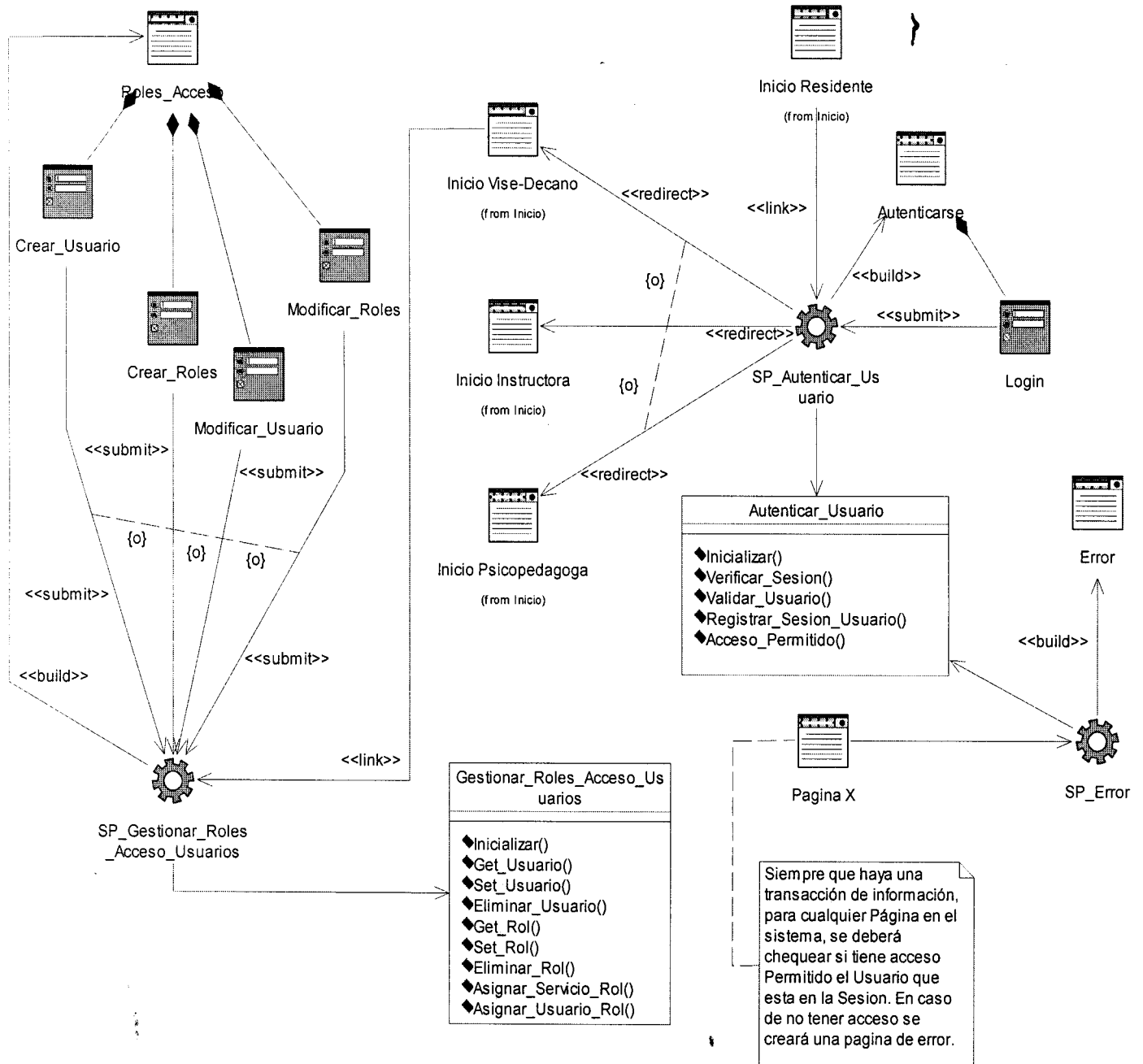


Figura 13

4.2.2 Paquete: Control de Disponibilidades

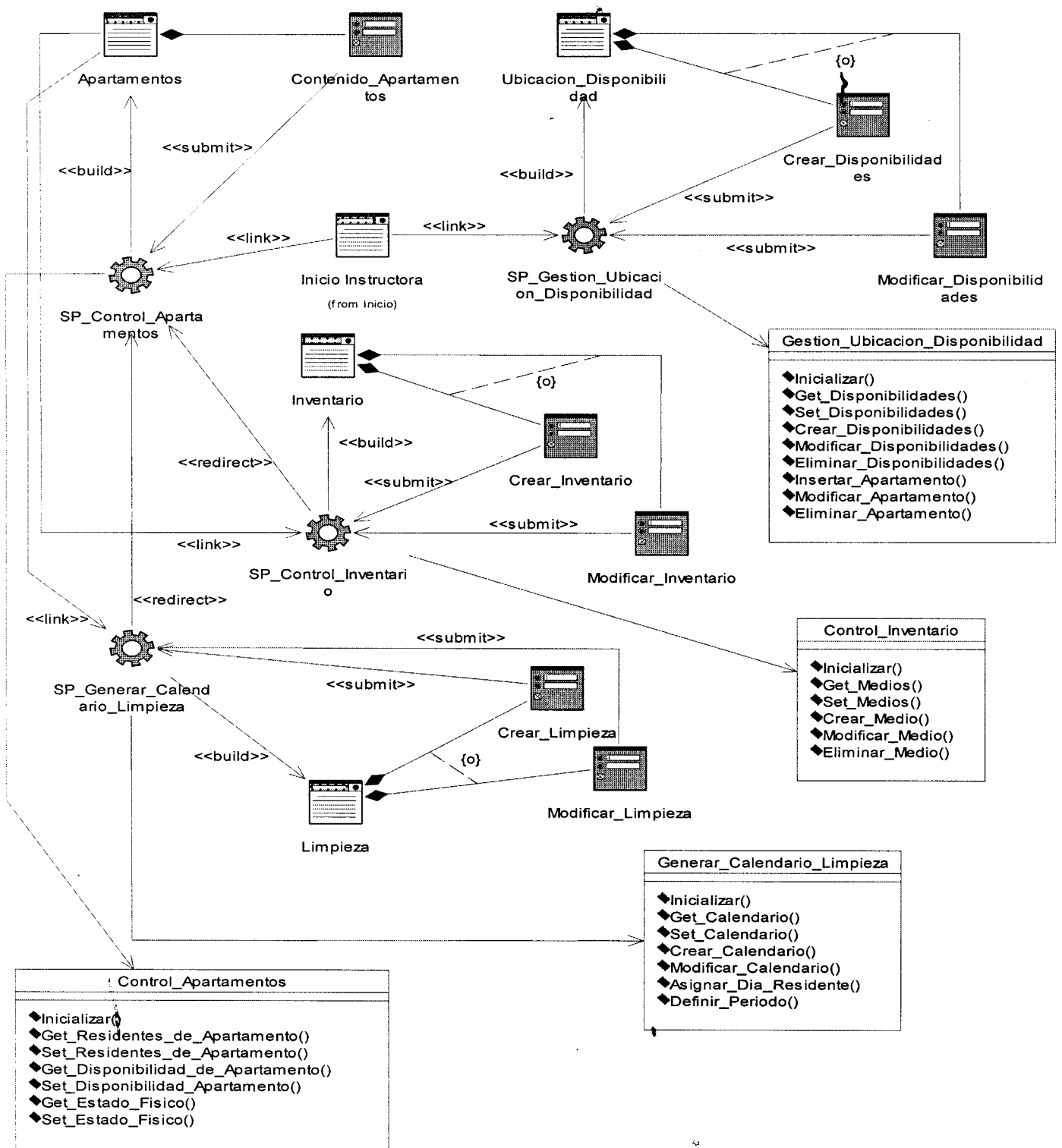


Figura 14

4.2.4 Paquete: Reportes

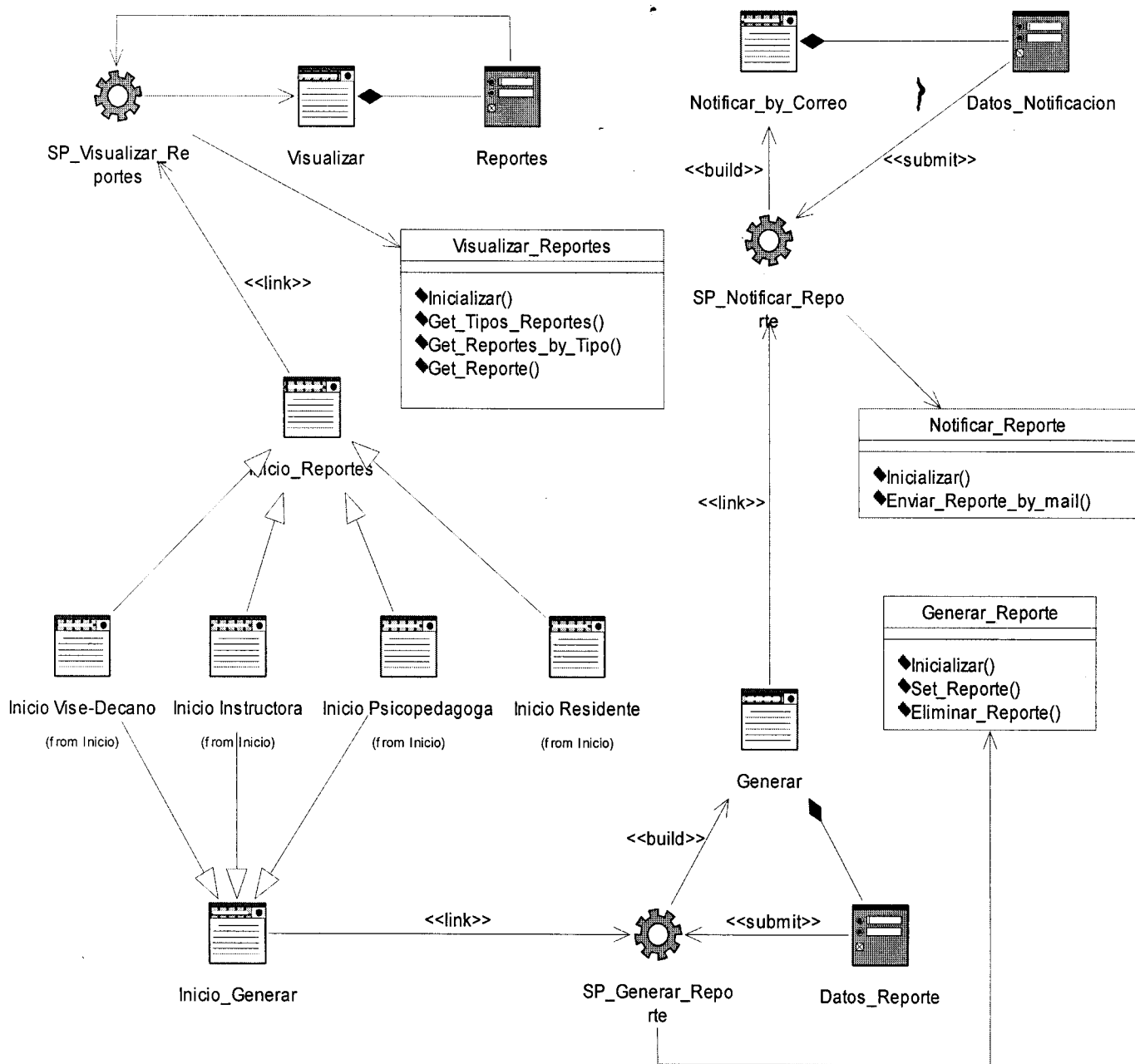


Figura 16

4.2.5 Paquete: Control de Guardias

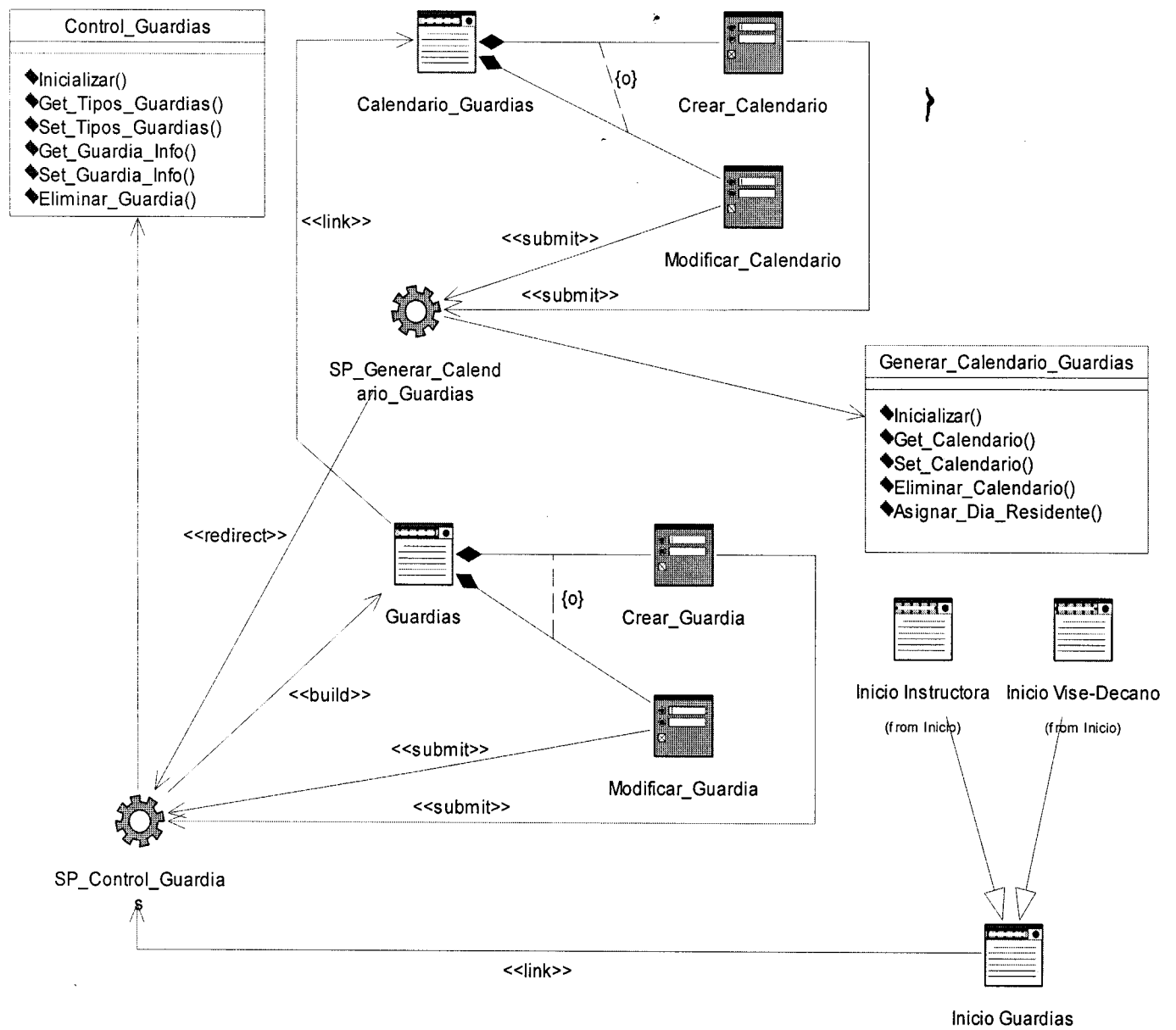


Figura 17

4.2.6 Consideraciones por Capas para la Aplicación

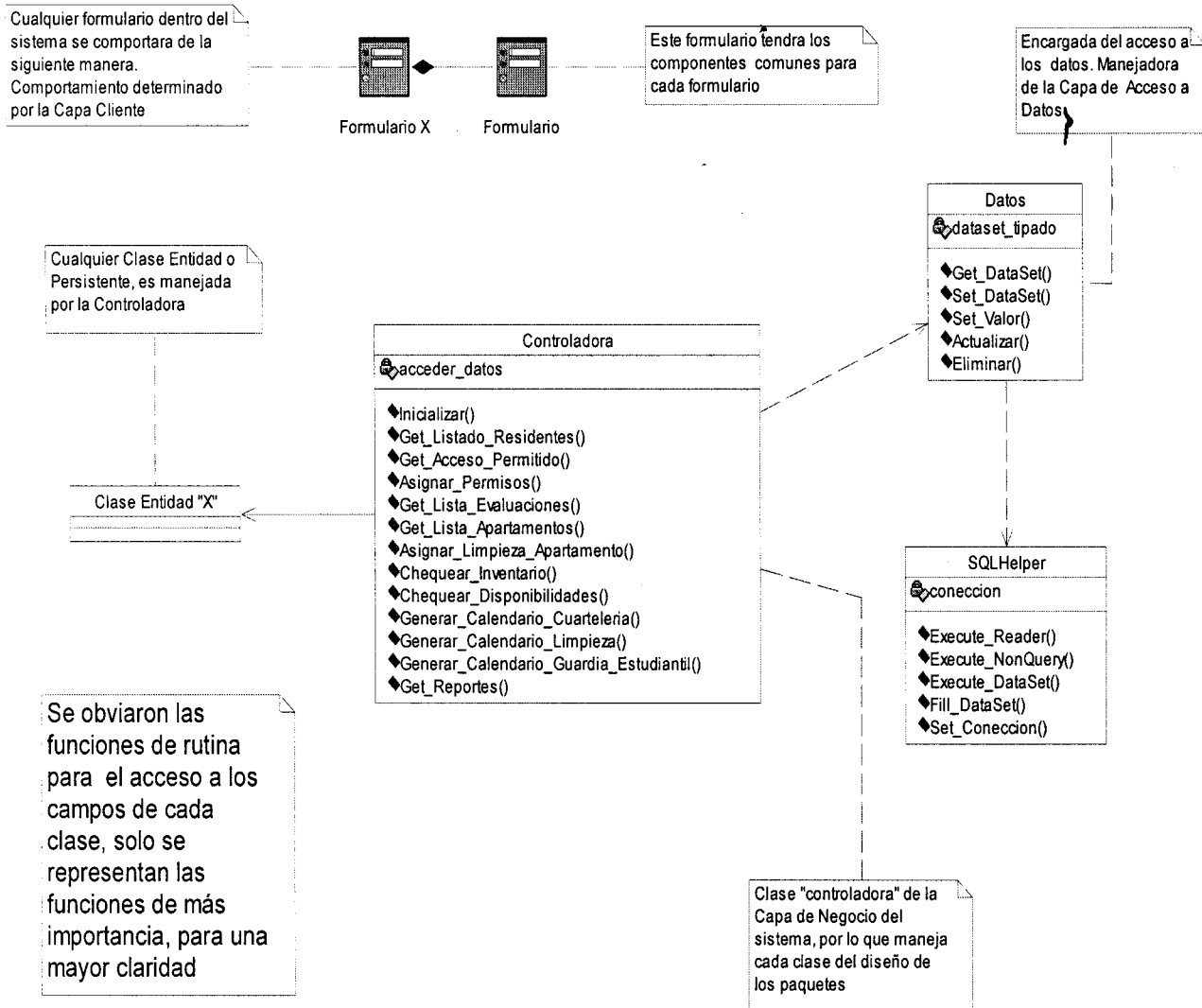


Figura 18

4.3 Principios de Diseño

4.3.1 Interfaz de Usuario

La interfaz que se brinda al usuario es uno de los elementos más importantes para el éxito final de una aplicación, máxime en los ambientes Web. Cuando se realizan diseños de este tipo es importante tener en cuenta los estándares establecidos para ello, y se debe siempre regir el trabajo a estas reglas para lograr proporcionar un producto de calidad.

Para la realización del diseño de esta aplicación se utilizó la herramienta Macromedia Dreamweaver 2004. Esta herramienta permite realizar páginas Web que pueden ser visualizadas en el Internet Explorer, NetScape Navigator y Opera. La interfaz está basada en las ventanas estándares de Windows, lo que hace posible mostrar un ambiente familiar al usuario y de fácil utilización.

Se utilizó la familia de fuentes Verdana, Arial, Helvetica, Sans-Serif de tamaño 8 y 10 puntos para diferenciar los títulos, los subtítulos, las cabeceras de tabla y el texto, logrando de esta forma uniformidad y claridad a la hora de mostrar la información, un estilo apropiado y un entorno agradable para el usuario.

Se han definido estilos propios que proporciona la reusabilidad de los mismos una vez que estén definidos, logrando una interfaz sencilla, con colores discretos, sin mucho contraste, predominando en el fondo el blanco y azul claro, forma similar los menús y las tablas; las letras en color negro, azul o verde.

En cada una de las páginas se utilizaron palabras sugerentes de acuerdo a las acciones a realizar, esto hace posible que el usuario haga un mejor uso del sistema.

Los colores que se decidieron utilizar son:

	#FFFFFF	Fondo de las páginas
	#336600	Opciones del Menú
	#003366	Títulos de Menús
	#F5FAFF	Fondo de Título en los Menús
	#C6D2E0	Color de Fila de Tabla
	#000000	Títulos de Tablas
	#7894B6	Fondo de Títulos en Tablas y pie de Tablas
	#DDEEFF	Color de los Bordos en las Tablas y Cuadros de Diálogos

SEGURIDAD Autenticarse Gestionar Roles Gestionar Usuarios	REPORTES Generar Reporte Notificar (E-mail) Visualizar Reporte	AYUDA Mapa de Navegación Ejemplos (Video)
DISPONIBILIDADES Control de Apartamento Control de Inventario Generar Calendario Limpieza Gestión Ubicación & Disponibilidad	RESIDENTES Analizar Evaluación Asignar Residencia Emitir Opinión Gestionar Residente	GUARDIAS Control de Guardias Generar Calendario Guardia

Figura 19

4.3.2 Formato de Salida de los Reportes

La aplicación desarrollada no muestra reportes, como los tradicionalmente conocidos, sino que brinda una serie de salidas en las que se visualizan de forma genérica, cada uno con las características exigidas por los usuarios: carnet de identidad, nombre del residente, correo electrónico, apartamento en el que reside, etc. De esta forma se permite una gran flexibilidad en cada salida de datos.

RESIDENTES					
No.	ID	Nombre	E-mail	Apartamento	
1	<input type="checkbox"/> 82020317445	Rolando Camacho Pupo	rcamachop@uci.cu	1152	

Figura 20

4.3.3 Ayuda

Para el sistema no se implementó una ayuda específica porque las operaciones que facilita están reflejadas de manera bastante clara y de fácil entendimiento, teniendo en cuenta que va a ser utilizado por usuarios que no necesariamente tienen conocimientos amplios dentro de la rama informática, por lo que no será difícil para ellos el desenvolvimiento en la aplicación. El usuario tendrá en cada momento las orientaciones necesarias para usar correctamente el sistema. La interfaz facilita la

navegación y muestra los datos de forma adecuada para el buen entendimiento de todas las personas que puedan utilizarlo. No obstante se cuenta con la presencia de un mapa de navegación además de la posibilidad de visualizar algunos tutoriales, que ejemplifican mediante ejemplos, las diferentes posibilidades del sistema para cada tipo de usuario.

4.3.4 Tratamiento de Excepciones

Para garantizar la estabilidad y confiabilidad de un sistema es imprescindible llevar a cabo un detallado tratamiento de excepciones y realizar validaciones siempre que se introduzcan datos al sistema, de esta forma se impide que el mismo pueda presentar comportamientos inesperados o falle al tratar de procesar datos incorrectos.

En el caso del sistema en cuestión se realizan validaciones tanto en el Cliente como en el Servidor, en ambos casos hay personas que gestionan información en el sistema, ya sea adicionando, modificando o eliminando datos.

El diseño de la interfaz en gran medida guía a los usuarios a no cometer errores a la hora de introducir datos en el sistema, sin embargo no se descarta la posibilidad de que surjan errores en el trabajo con la aplicación, de esta forma se mostrarán alertas en caso de alguna excepción. Se le permite al usuario reafirmar la operación que va a realizar siempre que se trate de eliminar o modificar cualquier parte de la información.

La estructura Cliente – Servidor provoca que en ocasiones ocurran excepciones debido a problemas de comunicación, que el Cliente haga algún pedido al Servidor y este no se encuentre activo en ese momento.

4.4 Diseño de la Base de Datos

4.4.1 Modelo Lógico de Datos

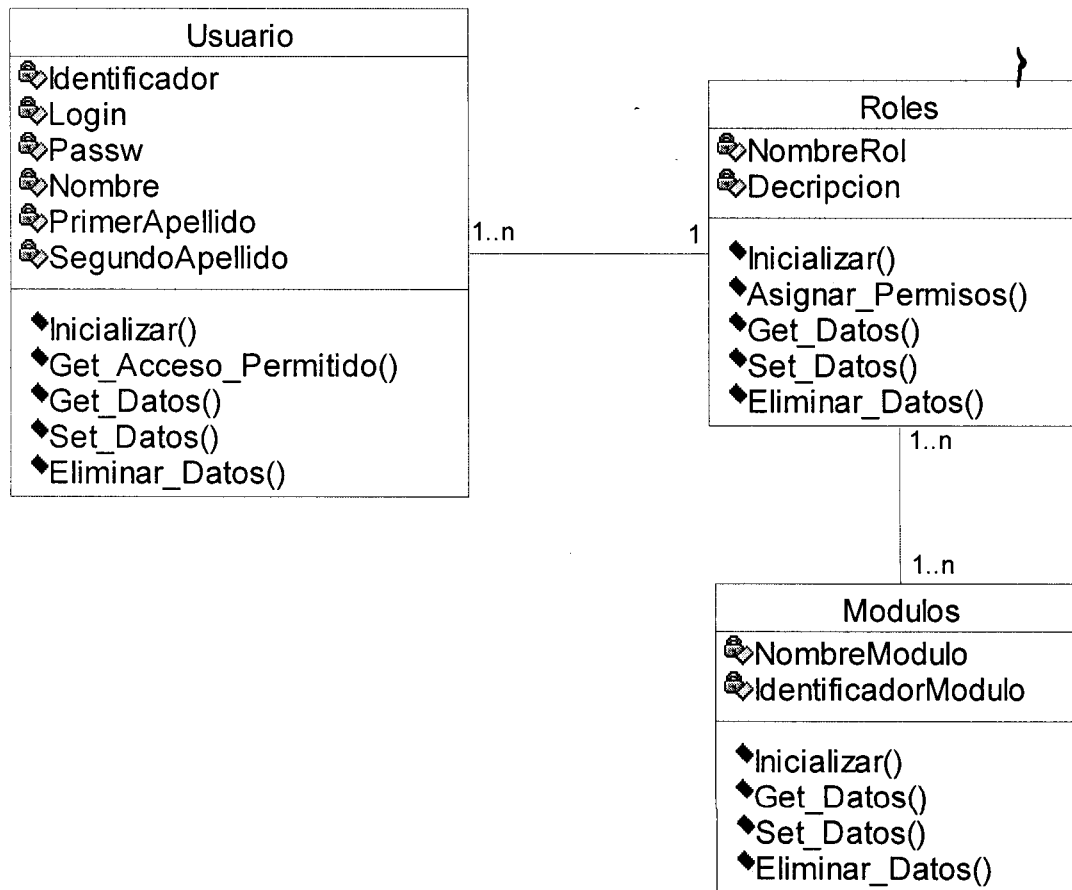
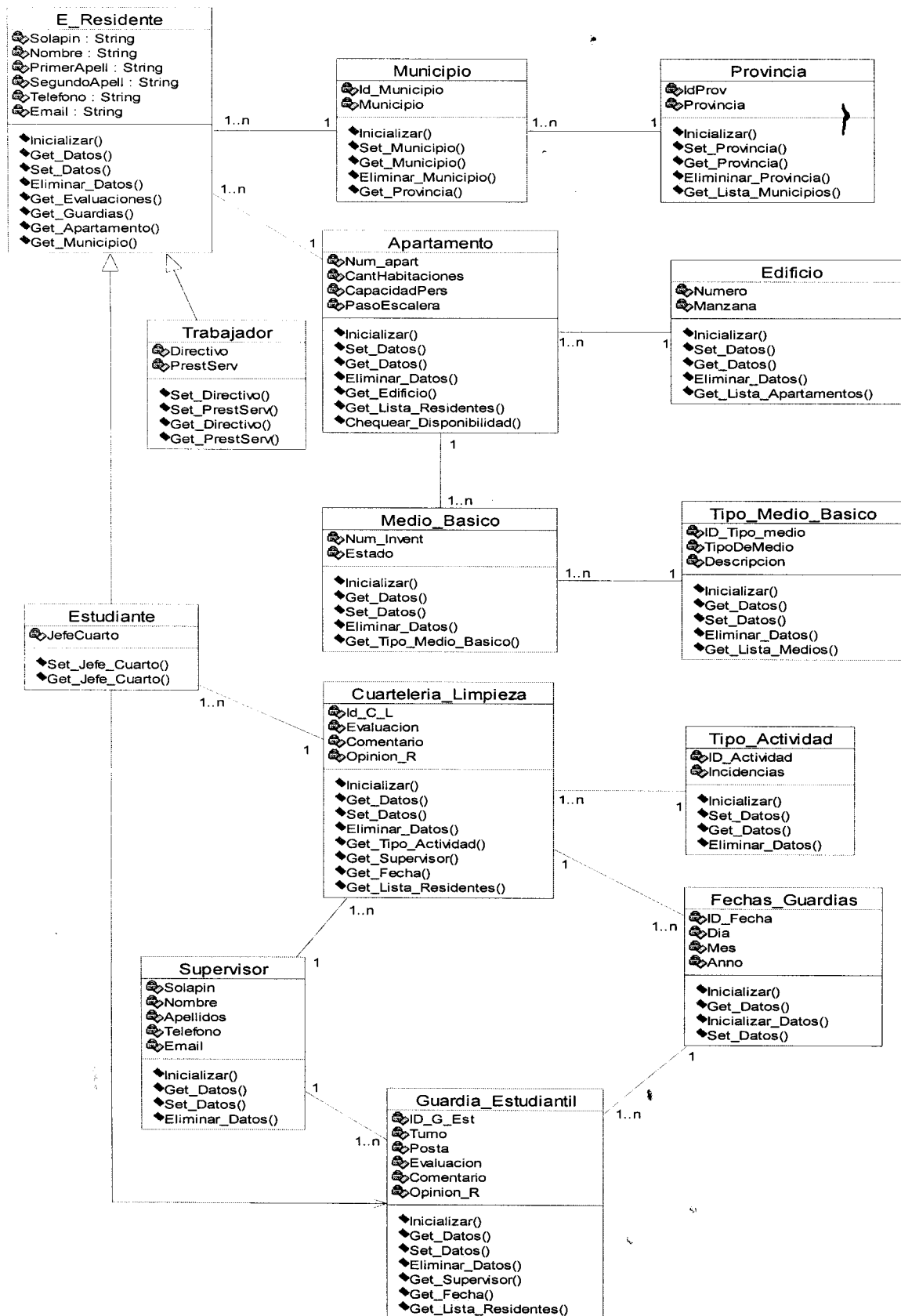


Figura 21



4.4.2 Modelo Físico de Datos

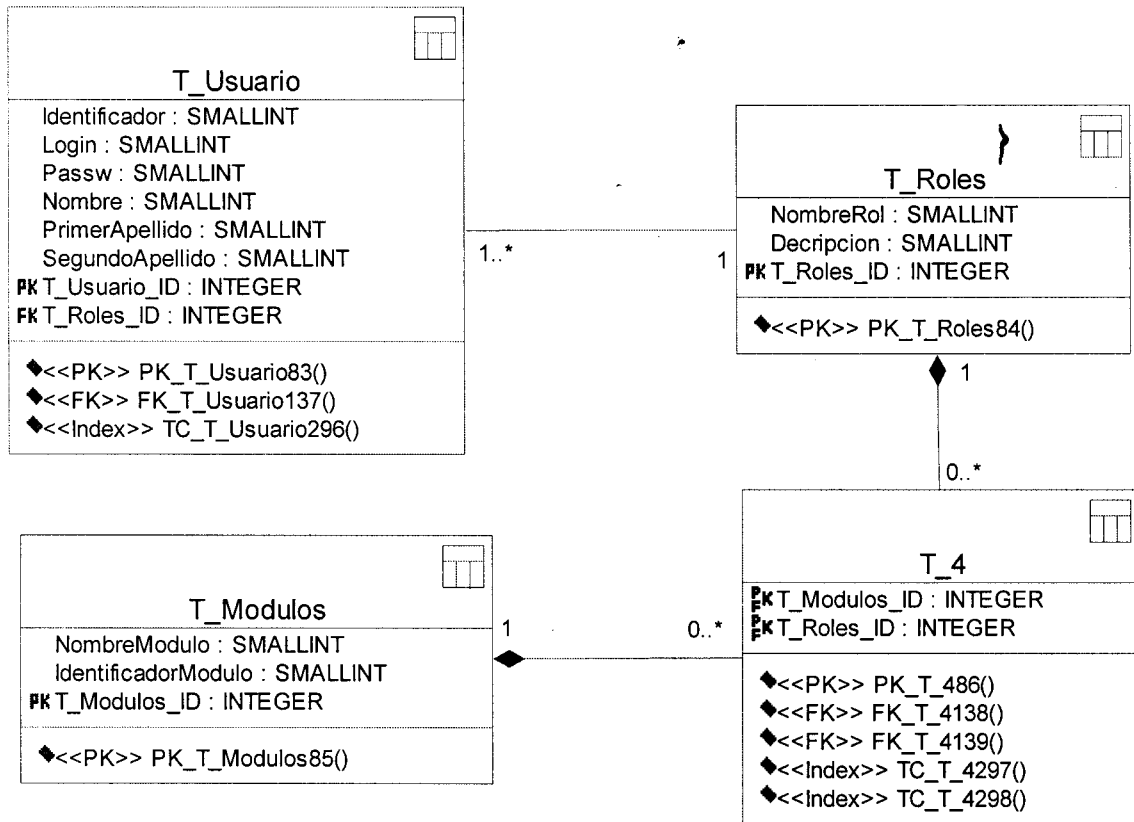


Figura 22



4.5 Modelo de Despliegue

4.5.1 Modelo de Capas Empleado

- Capa de Interfaz: Contiene todo lo relacionado con la presentación de la información. En ella se generan, en dependencia de los pedidos realizados por los usuarios, los eventos que desencadenan los procesos que ocurren en el sistema. Incluye las reglas establecidas en cuanto a la interfaz del sistema, estilos, colores, fuente de letras, etc.
- Capa de Reglas del Negocio: Contiene todas las clases y métodos que rigen el comportamiento del sistema. Es la que posibilita la interacción entre la capa de acceso a datos y la de presentación.
- Capa de Acceso a Datos: Contiene todo lo relacionado con el almacenamiento y las operaciones básicas sobre los datos. En el caso del Servidor es una BD en SQL Server que además de las tablas incluye procedimientos almacenados, diagrama de relaciones. Contiene los métodos necesarios para interactuar con la base de datos, es la que hace posible el intercambio con la capa de reglas del negocio.

4.5.2 Diagrama de Despliegue

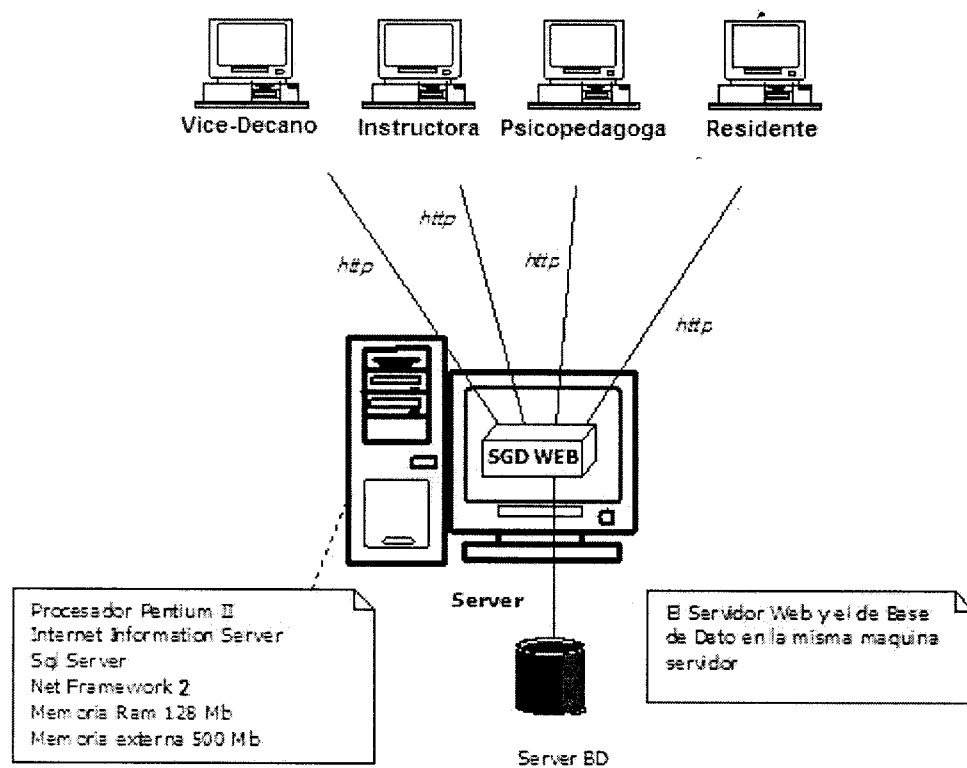


Figura 23

4.6 Conclusiones

La fase de diseño trata de conservar la estructura del sistema propuesta en la etapa de análisis y sirve de esquema para la implementación, la cual tiene como propósito desarrollar la arquitectura y el sistema como un todo, basándose en un enfoque iterativo e incremental.

La arquitectura multicapas, sin lugar a dudas, plantea una serie de ventajas:

- Se gana en organización, lo que permite agrupar métodos por su funcionalidad.
- La migración de código, pues puede ser utilizado en otra plataforma
- Reusabilidad de código, muy acorde con las dimensiones de los proyectos actuales.

Además la división por capas brinda un nivel de abstracción apropiado donde una capa no tiene que ver con las características de implementación de la otra, además las capas se ajustan al trabajo en equipo pudiendo ser desarrollada por personas distintas, ofreciendo mayor funcionalidad e independencia a la aplicación.

La elaboración del diagrama de clases constituye uno de los pasos más importantes en el desarrollo del proyecto por sus consecuencias directas en el funcionamiento de la aplicación, quedan descritos además, los estándares de diseño, la distribución de los principales nodos que conforman el sistema y el establecimiento de las conexiones entre ellos, modelados en el diagrama de despliegue.

Capítulo V: Estudio de Factibilidad

5.1 Introducción

La planificación de proyectos es un proceso indispensable antes de comenzar a desarrollar el mismo. Estimar los resultados del proyecto y los valores de costo, tiempo y recursos requeridos constituyen el punto en que radica su importancia, pues se puede determinar con anterioridad si el proyecto a realizar que se propone, es costeable, con el objetivo de evitar las pérdidas de recursos que afectan la economía de la entidad o empresa donde se lleva a cabo.

Los estudios de factibilidad de los trabajos de investigación constituyen la base de las decisiones que se tomen para su desarrollo, por lo que deben ser lo suficientemente precisos para evitar errores que tienen un alto costo social directo, en cuanto a los medios materiales y humanos que impliquen, así como por la pérdida de tiempo en la utilización de métodos menos prácticos.

Para realizar el estudio de factibilidad y el plan de un proyecto se tiene en cuenta un conjunto de restricciones:

- Organizativas: estructuras, procesos y personas.
- Económica: costos y beneficios.
- Técnica: habilidades, experiencia y recursos.
- Tiempo: fechas de cumplimiento.

5.2 Planificación basada en Casos de Uso

La especificación de los requerimientos mediante Casos de Uso ha probado ser uno de los métodos más efectivos para capturar la funcionalidad de un sistema. Este hecho se puede apreciar en algunas metodologías actuales ampliamente difundidas, como el Proceso Unificado de Rational, en las cuales se propone especificar la funcionalidad de los sistemas mediante la utilización de Casos de Uso.

El método de Casos de Uso permite documentar los requerimientos de un sistema en términos de Actores y Casos de Uso. Un Actor típicamente representa a un usuario humano o a otro sistema que interactúa con el sistema bajo análisis. Un Caso de Uso representa un gránulo funcional del sistema bajo análisis, relatado como una secuencia de acciones que uno o más actores llevan a cabo en el sistema para obtener un resultado de valor significativo. [11]

5.2.1 Calcular los Puntos de Casos de Uso (PCU)

5.2.1.1 Calcular el Factor de Peso de los Actores sin Ajustar (FPA)

Nombre del Actor	Complejidad
Vise-Decano	Persona que interactúa con el sistema mediante una interfaz gráfica. Complejidad 3.
Psicopedagoga	Persona que interactúa con el sistema mediante una interfaz gráfica. Complejidad 2.
Instructora	Persona que interactúa con el sistema mediante una interfaz gráfica. Complejidad 3.
Residente	Persona que interactúa con el sistema mediante una interfaz gráfica. Complejidad 2.

Tipo de Actor	Factor	# Actores	Resultado
Simple	1	0	0
Promedio	2	2	4
Complejo	3	2	6
Total (FPA)			10

5.2.1.2 Calcular el Factor de Peso de los CU (FPCU)

Casos de Uso	Peso
Gestión Ubicación Disponibilidad }	10
Control Apartamentos	10
Control Inventario	5
Generar Calendario Limpieza	5
Control de Guardias	5
Generar Calendario Guardias	10
Visualizar Reporte	5
Generar Reporte	5
Notificar	5
Gestionar Residente	5
Asignar Residente Apartamento	5
Emitir Evaluación	5
Analizar Evaluación	5
Emitir Opinión	5
Gestionar Roles Acceso Usuarios	10
Autenticar	5

Tipo de CU	Descripción	Factor	# CU	Resultado
Simple	1-3 transacciones	5	12	60
Promedio	4-7 transacciones	10	4	40
Complejo	8 o más transacciones	15	0	0
Total (FPCU)				100

Calculando los Puntos de Casos de Uso sin ajustar (PCU):

$$PCU = FPA + FPCU$$

$$PCU = 10 + 100 = 110$$

5.2.2 Calcular Puntos de CU ajustados (PCUA)

Una vez calculados los PCU (sin ajustar) , estos se deben ajustar teniendo en cuenta un grupo de factores técnicos y ambientales.

$$PCUA = PCU * FCT * FA$$

FCT = Factor de Complejidad Técnica.

FA = Factor de Ambiente.

5.2.2.1 Calcular Factor de Complejidad Técnica (FCT)

El Factor de Complejidad Técnica (FCT) Se estima mediante la cuantificación del peso de un grupo de factores que determinan la complejidad técnica del software. A cada factor se le asigna un valor de 0 a 5 de acuerdo con la relevancia del mismo.

Factor	Descripción	Peso	Valor	Factor	Comentario
T1	Sistema Distribuido	2	4	8	Es una aplicación cliente-servidor.

T2	Tiempo de respuesta	1	5	5	Su objetivo fundamental es agilizar todos los procesos de gestión de contenidos.
T3	Eficiencia por el usuario	1	5	5	Garantiza que se gestionen los contenidos eficientemente.
T4	Procesamiento interno complejo	1	2	2	No hay cálculos complejos.
T5	Reusabilidad	1	4	4	Es totalmente reusable.
T6	Facilidad de instalación	0.5	2	1	Solo montar el servidor web y de base de datos.
T7	Facilidad de uso	0.5	4	2	Fácil de usar.
T8	Portabilidad	2	5	10	Garantiza la portabilidad hacia varios sistemas.
T9	Facilidad de cambio	1	2	2	No requiere un gran costo de mantenimiento.
T10	Concurrencia	1	4	4	Garantiza la concurrencia de usuarios.
T11	Objetivos especiales de seguridad	1	3	3	Seguridad normal.
T12	Acceso directo a terceras partes	1	0	0	Los usuarios que navegan por la Web no tienen acceso directo.
T13	Facilidades especiales de	1	1	1	Pocos usuarios internos, sistema fácil de usar.

	entrenamiento a usuarios finales				
Total (TFactor)					47

$$FCT = 0.6 + 0.01 * TFactor = 0.6 + 0.01 * 47 = 0.6 + 0.47$$

$$FCT = 1.07$$

5.2.2.2 Calcular Factor Ambiente (FA)

Se consideran las habilidades, entrenamientos y experiencias del grupo de desarrollo, se estima de forma similar al FCT.

Factor	Descripción	Peso	Valor	Factor	Comentario
E1	Familiaridad con el modelo de proyecto utilizado	1.5	3	4.5	No existía una alta familiaridad con el modelo de proyecto utilizado.
E2	Experiencia en la aplicación	0.5	2	1	Experiencia baja en cuanto al lenguaje de programación utilizado.
E3	Experiencia en Orientada a Objetos (OO).	1	4	4	Existe una experiencia anterior en la programación OO.
E4	Capacidad del analista líder	0.5	3	1.5	El analista líder tiene experiencias anteriores de otros proyectos

					semejantes.
E5	Motivación	1	5	5	Alta motivación
E6	Estabilidad de los requerimientos	2	4	8	Existen posibilidades de cambios futuros.}
E7	Personal media jornada	-1	4	-4	No siempre se trabaja media jornada.
E8	Dificultad en el lenguaje de programación	-1	3	-3	El lenguaje no tiene una alta complejidad.
Total (EFactor)					17

$$FA = 1.4 + (-0.03 * EFactor) = 1.4 + (-0.03 * 17) = 1.4 - 0.51$$

$$FA = 0.89$$

Con los Puntos de Casos de Uso PCU (sin ajustar) y los Factores de Complejidad Técnica (FCT) y de Ambiente (FA) calculados:

Puntos de Casos de Uso ajustados (PCUA)

$$PCUA = PCU * TCF * FA = 110 * 1.07 * 0.89$$

$$PCUA = 104.753$$

5.2.3 Calcular Esfuerzo (E)

Convertir los Puntos de Casos de Uso Ajustados a Esfuerzo de desarrollo.

Esfuerzo (E)

$$E = PCUA * FC$$

FC: El valor de FC como media se adopta sea 20 horas-hombre (H/H), pero puede oscilar entre 15 y 30 H/H en dependencia de los FA, de esta forma surge un refinamiento que propone una granularidad algo más fina, según el siguiente criterio:

- Se contabilizan cuántos factores de los que afectan al Factor de ambiente están por debajo del valor medio (3), para los factores E1 a E6.
- Se contabilizan cuántos factores de los que afectan al Factor de ambiente están por encima del valor medio (3), para los factores E7 y E8.
- Si el total es 2 o menos, se utiliza el factor de conversión 20 horas-hombre/Punto de Casos de Uso, es decir, un Punto de Caso de Uso toma 20 horas-hombre.
- Si el total es 3 o 4, se utiliza el factor de conversión 28 horas-hombre/Punto de Casos de Uso, es decir, un Punto de Caso de Uso toma 28 horas-hombre.
- Si el total es mayor o igual que 5, se recomienda efectuar cambios en el proyecto, ya que se considera que el riesgo de fracaso del mismo es demasiado alto. [11]

De acorde al texto anterior, se le aplican las normas sugeridas a los FA tratados en el documento, donde se obtiene un FC = 20 H/H. Se concluye:

$$E = 104.753 * 20 = 2095.06$$

Esfuerzo total (E_{total})

$$E_{total} = E / 0.4 = 2095.06 / 0.4$$

$$E_{total} = 5237.65$$

5.2.4 Calcular Tiempo de Desarrollo (TDES)

$$TDes = E_{total} / CH_{total}$$

TDes: Tiempo de Desarrollo.

CH_{total}: Cantidad de Hombres

$$TDes = 5237.65 / 1 = 5237.65$$

Tras los cálculos anteriores, el tiempo estimado para el desarrollo del sistema, por una sola persona, es de 5237.65 horas.

5.2.5 Calcular el Costo Total (C_{total})

$$C_{total} = E_{total} * CHH$$

C_{total} : Costo Total

CHH: Costo por Hombre Horas

$$CHH = K * THP$$

K: Coeficiente que tiene en cuenta los costos indirectos (1.5 y 2.0).

THP: Tarifa Horaria Promedio

Adjudicando \$225.00, mínimo en la escala, de salario a cada trabajador del proyecto con una cantidad de horas de trabajo de 160 en el mes, se desprende:

$$THP = 225/160$$

$$THP = 1.41$$

$$C_{total} = E_{total} * K * THP = 5237.65 * 2 * 1.41$$

$$C_{total} = 14770.173$$

El costo total del Proyecto es de \$14 770.17

5.3 Beneficios de la Utilización del Sistema

5.3.1 Beneficios Tangibles

En el mercado internacional existen soluciones que brindan algunas de las funcionalidades que se requieren, pero no engloban todas las que se necesitan ni se adaptan a las especificidades de la Facultad 3, además de que son comerciales, lo que

implicaría un gasto monetario que superaría el costo de la realización del sistema según los cálculos de este capítulo.

El sistema tiene un costo de realización de \$14770.17 en moneda nacional, \$590.80 CUC, si se compara esta cifra con el costo ofrecido por empresas que radican en el país (Ver Anexo #3) es tangible la diferencia, pues se ahorran \$1409.20 CUC (opciones en el exterior serían más costosas). Además considerable disminución de material de oficina, reduce los costos, al automatizarse una serie de operaciones que son realizadas por los trabajadores de forma manual.

5.3.2 Beneficios Intangibles

La realización de este proyecto resulta muy factible a la Facultad 3. Constituirá una herramienta de trabajo diario, brindando facilidades como ahorro de tiempo y recursos. Con la utilización de la tecnología .NET las páginas se visualizan más rápido y se agiliza la gestión de contenidos. El sistema propuesto está concebido para que sea portable y genérico, de modo que servirá a las diferentes facultades que deseen adaptarlo a sus necesidades.

El hecho de tener la información almacenada en una base de datos hace posible que sea más confiable y segura. Se mejora la calidad y rapidez de los servicios en cuanto a tiempo de respuesta, procesamiento y organización de la información. Además beneficiará a los usuarios finales, pues los contenidos que se exponen se harán más visibles y comprensibles, habrá una contribución al uso de las nuevas tecnologías y se elevará el número de visitantes al futuro Portal de la Facultad 3, asegurando la futura fidelidad de estos a los nuevos servicios.

La puesta en funcionamiento de este sistema representa un aporte al desarrollo del plan de informatización de la sociedad y el empleo eficiente de las TIC, lo que contribuye con la batalla de ideas que se lleva a cabo en nuestro país.

5.4 Análisis de Costo – Beneficio

Tomando como base el estudio de factibilidad realizado y el conjunto de beneficios tangibles e intangibles que brinda el sistema propuesto, se determina que el mismo es factible y que su empleo contribuye en gran medida al desarrollo satisfactorio de la Facultad 3.

El costo estimado es bajo en comparación con los del resto de las soluciones existentes en el mercado nacional e internacional, a pesar de no incluir los costos adicionales que conllevarían la completa realización y posterior implantación y mantenimiento del sistema.

5.5 Conclusiones

El estudio de factibilidad de un proyecto de desarrollo de software constituye un factor importante y una herramienta clave en el análisis de la estimación de los resultados del proyecto y de los valores de tiempo, costo y recursos requeridos. Este contribuye a estimar el alcance que tendrá el producto en desarrollo.

El sistema propuesto se considera económicamente factible debido a todos los beneficios que reporta y al impacto social que conlleva. Lo anteriormente expuesto permite concluir que el producto puede ser implantado en la entidad prevista.

Conclusiones

Los sistemas de gestión han ido en crecimiento junto con el desarrollo de la informática y las telecomunicaciones, es inevitable aceptar su utilidad y llevarla a cabo en cualquier entidad que lo requiera

Con la propuesta de este Sistema de Gestión se logró una nueva forma de gestionar la información, acerca a sus usuarios la información que les interesa y brinda la posibilidad de notificar sobre algún reporte o cambio de evaluación dentro del sistema. Con esto garantiza que sus usuarios mantengan un nivel de actualización elevado y obtengan la información que necesitan de forma directa, sin tener que realizar búsquedas que en la mayoría de los casos resultan engorrosas debido a los grandes volúmenes de información que existen.

Se implementó una estructura Cliente – Servidor que permite realizar en el Servidor toda la gestión de la información de cada uno de los usuarios que utilizan el sistema.

Se realizó un estudio de las principales herramientas y tecnologías actuales relacionadas con el producto a construir, que sirvió para fundamentar que el sistema tendrá una interfaz Web, programado en la plataforma .NET con lenguaje de programación C#, utilizando como gestor de base de datos SQL Server 2000 e Internet Information Server como servidor Web para publicarlo.

El análisis y diseño del sistema se llevó a cabo utilizando la metodología RUP, con el lenguaje de modelado UML, representando las características de la aplicación mediante una serie de diagramas que conforman esta metodología.

Además se realizó un estudio de factibilidad el cual dio como resultado la factibilidad del sistema, con un costo de realización muy inferior al equivalente de productos similares en el mercado. Incluso ofrece una serie de beneficios tanto tangibles como no tangibles para la entidad en la que se desarrolla.

Como resultado final del trabajo realizado se puede concluir que se obtuvo un sistema dinámico, altamente flexible y extensible con el cual se da cumplimiento a los objetivos trazados y se solucionan los problemas presentados.

Recomendaciones

Tras haber cumplidos los objetivos trazados con la realización del Sistema de Gestión, se recomienda:

- Que el sistema una vez cumplido el periodo de prueba se proponga su implantación en las demás facultades de la UCI.
- Que se le incorporen nuevas funcionalidades para globalizar y centralizar todas las tareas y servicios de la Institución Residencial.
- Que se continúe el trabajo en función de los reportes que se pueden obtener sobre la gestión de la Residencia en general.
- Apoyándose en la reusabilidad del código que brinda el sistema, se migre el mismo hacia Open Source, específicamente a la variante que brinda MONO, para que pueda ser explotado sobre esas plataformas.
- Seguir desarrollando la interfaz para responder a los diferentes requerimientos que puedan surgir a raíz de las necesidades crecientes de los usuarios.

Referencias Bibliográficas

- [1] Aja Quiroga, Lourdes. *Gestión de información, gestión del conocimiento y gestión de la calidad en las organizaciones*. Acimed, Vol. 10, Junio 2002. }
- [2] Anónimo. *Software de Gestión Comercial (1ra parte)*. Datahouse Company, Argentina. <http://www.datahousecompany.com.ar/comercial1.html> (4/3/2006).
- [3] Bustelo Ruesta, C. *Tendencias en la Gestión de la Información, la documentación y el conocimiento en las organizaciones*. El Profesional de la Información, Vol. 10, No.12, Diciembre 2001.
- [4] Cruz, M. L. G. *Sistema para la Gestión de Acuerdos para Directivos Universitarios*. Habana, CUJAE. 2004.
- [5] Errazuris, B. *Características SQL Server 2000*. 2000.
- [6] González, D. A. H. & M. S. M. A. Ampuero RUP. *El Proceso Unificado de Desarrollo de Software*. 2003.
- [7] Marqués Andrés, M. *Apuntes de Ficheros y Base de Datos*. Universitat Jaume I, 2001.
- [8] Microsoft Corporation, MSDN en Español, ASP.NET. 2005.
http://www.microsoft.com/spanish/msdn/centro_recursos/asp_net/default.aspx
- [9] Microsoft Corporation, MSDN en Español. 2005.
<http://www.microsoft.com/spanish/msdn/vstudio/default.asp>
- [10] Microsoft Corporation, MSDN en Español, C#. 2005.
<http://www.microsoft.com/spanish/msdn/vcsharp/productinfo/vcsharp03/default.asp>
- [11] Peralta, M. *Estimación del Esfuerzo Basada en Casos de Uso*. Centro de Ingeniería del Software e Ingeniería del Conocimiento (CAPIS), Instituto Tecnológico de Buenos Aires.
- [12] Pereda, H.F. *Enfoque hacia la Gestión*. 2002.
- [13] *Rational Unified Process Proceso Hecho Práctico*. 2000.

Referencias Bibliográficas

[14] Sáenz, H. *Comparación entre Sistemas de Gestión de Base de Datos (sgbd) bajo licenciamiento libre y comercial*. Trabajo de Grado para optar por el título de Ingeniero en Sistemas, Universidad Católica de Colombia, Facultad Ingeniería de Sistemas, 2004.

[15] *Servidor de Base de Datos*. 2000. Disponible en:
<http://www.arsys.es/soporte/programacion/sql.htm>

[16] *Tutorial de UML*. 2001. Disponible en:
http://www.solocursos.net/tutorial_de_uml-slccurso1026071.htm

[17] *UML*. 2000. Disponible en:
http://www.itera.com.mx/itera/productos/uml_centro_recursos.asp

[18] Vaca, F.S. *Responsabilidad de la Gestión*. 2001.

Bibliografía

Meyer Bertrand, *Construcción de software orientado a objetos*, Prentice Hall, Segunda edición. 1997. }

Glosario de Términos

C: Es un lenguaje de programación estructurado, de propósito general, que ha sido usado extensamente en sistemas operativos y aplicaciones y ha tenido un amplio seguimiento en la comunidad académica.

C++: Es un lenguaje de programación orientado a objetos, visto para muchos como el mejor lenguaje para hacer aplicaciones de gran alcance.

COM+: Es una extensión de *Component Object Model* (COM). Estrategia de Microsoft para desarrollar programas utilizando segmentos de programas ya construidos. COM + una arquitectura orientada a objetos y un conjunto de servicios de sistema operativos.

FTP: Es uno de los diversos protocolos de la red Internet, concretamente significa *File Transfer Protocol* (Protocolo de Transferencia de Archivos) y es el ideal para transferir datos por la red.

HTTP: Es el protocolo de la Web (WWW), usado en cada transacción. Las letras significan *Hyper Text Transfer Protocol*, es decir, protocolo de transferencia de hipertexto. El hipertexto es el contenido de las páginas web, y el protocolo de transferencia es el sistema mediante el cual se envían las peticiones de acceder a una página web, y la respuesta de esa web, remitiendo la información que se verá en pantalla.

Java: Es una plataforma de software desarrollada por Sun Microsystems, de tal manera que los programas creados en ella puedan ejecutarse sin cambios en diferentes tipos de arquitecturas y dispositivos computacionales.

JScript.NET: Es un lenguaje script orientado a objetos que combina un conjunto de características existentes en el JScript clásico con los mejores aspectos de los lenguajes basados en clases.

RAM: *Random-Access Memory* (memoria de acceso aleatorio). Se trata de una memoria volátil, es decir, pierde su contenido al desconectar la energía eléctrica. Se

utilizan normalmente como memorias temporales para almacenar resultados intermedios y datos similares no permanentes.

Servicios Web XML: Permiten el intercambio de datos y la llamada remota de lógica de aplicaciones utilizando la mensajería XML para mover datos a través de servidores de seguridad y entre sistemas heterogéneos.

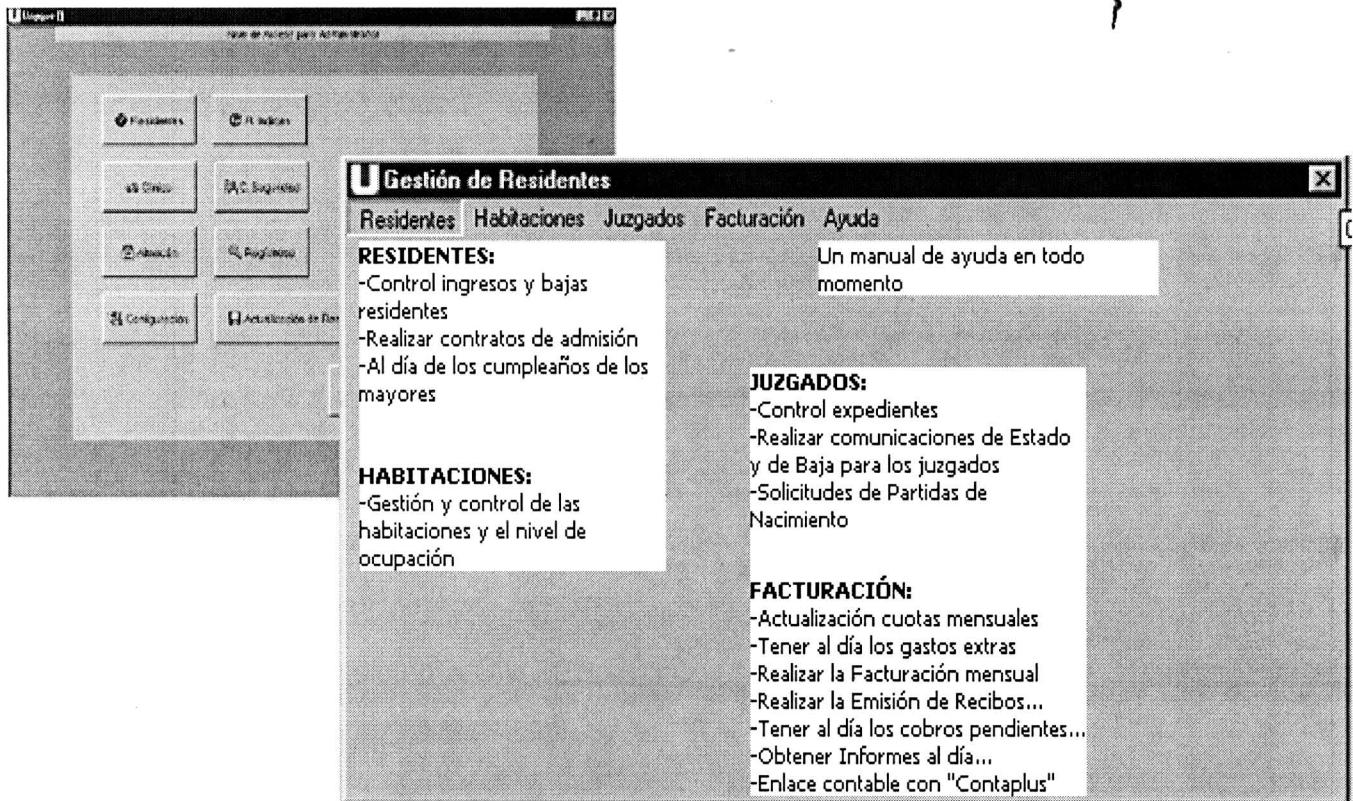
SGBD: *Sistemas Gestores de Bases de Datos* son un tipo de software muy específico, dedicado a servir de interfaz entre las bases de datos y las aplicaciones que la utilizan.

TCP/IP: Conjunto básico de protocolos de comunicación de redes, popularizado por Internet, que permiten la transmisión de información en redes de computadoras. El nombre TCP/IP proviene de dos protocolos importantes de la familia, el *Transmission Control Protocol (TCP)* y el *Internet Protocol (IP)*.

Terabytes: Es una unidad de medida en informática y su símbolo es el *TB*. Es equivalente a 2^{40} bytes.

Anexos

Anexo #1: Interfaz del sistema UNIGES.



Anexo #2: Interfaz del sistema RESIGES.

Residentes

Temporal Nº habitación / Tipo

Nº Reg. Ext. habit. Cod. Reg.

Nombre y apellidos Fecha nacimiento Edad Fecha ingreso

DNI Sexo Estado Civil Situación actual

Nº Reg. Social Fecha Reg. Social

Valido Asistido Semi-asistido

Seguridad Social Adelitas Aosa Sanitas Otro

Compañía Seguros Póliza fallecimiento Teléfono póliza

Familiar 1 **Familiar 2**

Nombre Nombre Dirección Código postal de inscripción C.P.

ID Teléfono familia Teléfono 1 Teléfono 2 Teléfono 3 Teléfono 4 Teléfono 5

Población Provincia Observaciones

Ordenar por: Nombre, Nº reg., Habitación

Total residentes

Plazas libres

Manualidad Historia Social Documentos Imprimir contrato

GUARDAR SIN GUARDAR

Anexo #3: Tarifas de Precios ofrecidas en el Portal Cubano CUBASI.

Ofertas	Sitios promocionales estáticos (solo presencia en Internet)	Sitios corporativos dinámicos
Diseño básico hasta 9 páginas	380.00 USD	860.00 USD
Diseño básico hasta 15 páginas	760.00 USD	1 300.00 USD
Diseño plus hasta 30 páginas	1 100.00 USD	2 000.00 USD
Diseño plus hasta 50 páginas	1 500.00 USD	2 700.00 USD
Por más de 50 páginas el precio se acuerda puntualmente		
-Anteproyecto: estructura de la página y diseño gráfico (dos propuestas) y su presentación (pago por adelantado) 10 % de la suma total del proyecto. -Mantenimiento posventa (opcional) 20% del precio total por año.		