



TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE
INGENIERÍA INFORMÁTICA

Autor: ■■■■

Yusniel Ávila Malagón

Tutor: ■■■■

Lic. Niurvis Legrá Pérez

Ciudad de la Habana, Abril 2006

DECLARACIÓN DE AUTORÍA

Por este medio declaro que Yusniel Avila Malagón es el único autor de este trabajo y autorizo a la Universidad de las Ciencias Informáticas (UCI) para que haga el uso que se estime pertinente con este trabajo.

Para que así conste, firmo la presente a los 5 días del mes de Abril del 2006.

Firma del Autor

Firma del Tutor

OPINIÓN DEL USUARIO DEL TRABAJO DE DIPLOMA.

El Trabajo de Diploma titulado "Gestión de Alojamiento", fue desarrollado en la Universidad de las Ciencias Informáticas. Esta entidad considera que, en correspondencia con los objetivos trazados, el trabajo realizado le satisface:

Totalmente _____

Parcialmente en un _____ %

Los resultados de este Trabajo de Diploma le reportan a esta entidad los beneficios siguientes:

Y para que así conste, se firma la presente a los ____ días del mes de _____ del año _____

Representante de la entidad

Cargo

Firma

Cuño

OPINIÓN DEL TUTOR DEL TRABAJO DE DIPLOMA

Título: Gestión de Alojamiento.

Autor: Yusniel Avila Malagon

El tutor del presente Trabajo de Diploma considera que durante su ejecución el estudiante mostró las cualidades que a continuación se detallan.

<Aquí el tutor debe expresar cualitativamente su opinión y medir (usando la escala: muy alta, alta, adecuada) entre otras las cualidades siguientes:

- Independencia
 - Originalidad
 - Creatividad
 - Laboriosidad
 - Responsabilidad >

< Además, debe evaluar la calidad científico-técnica del trabajo realizado (resultados y documento) y expresar su opinión sobre el valor de los resultados obtenidos (aplicación y beneficios) >

Por todo lo anteriormente expresado considero que el estudiante está apto para ejercer como Ingeniero Informático; y propongo que se le otorgue al Trabajo de Diploma la calificación de <nota>. <Además, si considera que los resultados poseen valor para ser publicados, debe expresarlo también>

Firma

Fecha

El mundo necesita:

-Hombres que sean sinceros y honrados en lo más íntimo de sus almas

-Hombres que no teman dar al pecado el nombre que le corresponde

-Hombres que no se vendan ni se compren

-Hombres que se mantengan de parte de la justicia aunque se desplomen los cielos.

ELENA DE WHITE

AGRADECIMIENTOS

A Miguel Arcángel Gonzalez Estévez por confiar en mí.

A Fabio y Matilde Montes de Oca, por ser como mis padres aquí en la universidad.

A Luis Lamela Fung por

A mi tutora Niurvis, por aguantarme tanto y por no perder la paciencia.

Al piquete (Yadiel, El chiqui, El rubio, El mulo, El taba, Ronaldo, El yerry, Ipi) en fin a todos, gracias por compartir tantos momentos maravillosos dentro de este proyecto futuro.

A Hardys Perez Bermudes por ser mi amigo.

A mis compañeros de aula Yulier Casas Estrada (casas), Yidier Romero (el ruso), Miguel Angel Martinez, Liudmila, Edgar, Ailec, Phill, A los clones, en fin a todos.

A mi socito Sergito Layet.

A Daniel, El flaco, Robe.

A Yanet Hernandez Mesa por ser una amiga de verdad.

A todo el personal de residencia.

A mi tía María Elena, Rosy y Laury.

A mis abuelos y en especial a mi abuela Albina.

A mis tíos, en especial a Ibrí y Manolo.

A mis Hermanos, que sigan el camino trazado y que lo recorran mucho mejor que yo.

A Esperanza y Jesús, que los quiero de verdad.

En fin a todos ellos muchas GRACIAS.

A mis dos tesoros
Reina Malagón Gonzalez y Ana Laura Perez Gonzalez (Madres)
A mi Timotea por ser alguien especial en mi vida.

Y en especial a mi papá, todo mi éxito se lo dedico a él...

RESUMEN

Este trabajo tiene como propósito la descripción de una propuesta de sistema informático que automatice los procesos de alojamiento en la Universidad de las Ciencias Informáticas (UCI). La universidad cuenta con un gran número de personas que se alojan en sus edificaciones. Es de interés de la Dirección de Informatización, que el registro de la ubicación o distribución de estas personas en dichas instalaciones se realice de forma automatizada, de manera tal que se garantice mayor eficiencia y control.

El Sistema de Gestión de Alojamiento es una aplicación web que controla esta actividad y cumple además con las siguientes funcionalidades: gestionar permutas, eliminar personas, crear y destruir estructuras (Residencias, Manzanas, Edificios, Apartamentos). Además, modificar las características de los apartamentos y un sistema de reportes que le permite a los diferentes tipos de directivos de la residencia, obtener una información precisa acerca de las instalaciones y sus ocupantes.

En este documento se recogen los resultados del trabajo de investigación realizado. Se identifican y describen los procesos que conforman la gestión de alojamiento, se describen sistemas informáticos similares que se han desarrollado en otros centros, y se dan argumentos para demostrar que la situación problemática requiere de un sistema nuevo, ya que ninguna otra solución satisface todas sus necesidades. Posteriormente, se hace un análisis comparativo acerca de las tecnologías existentes y se seleccionan las más apropiadas. Finalmente se muestran los resultados del diseño de la propuesta del sistema.

INTRODUCCIÓN	2
FUNDAMENTACIÓN DEL TEMA	5
1.1 INTRODUCCIÓN	5
1.2 GESTIÓN	5
1.3 GESTIÓN DE ALOJAMIENTO	5
1.4 LA DIRECCIÓN DE RESIDENCIA DE LA UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS.	5
1.5 DESCRIPCIÓN ACTUAL DE CÓMO SE LLEVA EL NEGOCIO	6
1.5.1 <i>Reglas del Negocio</i>	7
1.6 SISTEMAS AUTOMATIZADOS EXISTENTES	7
1.7 PROPUESTA DE SOLUCIÓN: SISTEMA DE GESTIÓN DE ALOJAMIENTO.	9
1.8 CONCLUSIONES	9
TENDENCIAS Y TECNOLOGÍAS ACTUALES A CONSIDERAR	10
2.1 INTRODUCCIÓN	10
2.2 APLICACIONES WEB	10
2.3 LA TECNOLOGÍA .NET	11
2.3.1 <i>Arquitectura Framework.NET</i>	12
2.4 LENGUAJES DE PROGRAMACIÓN PARA LA WEB	12
2.5 ASP.NET	15
2.6 ARQUITECTURA DE SOFTWARE	18
2.7 ARQUITECTURA EN 3 CAPAS	18
2.8 WEB SERVICES	19
2.9 SISTEMA GESTOR DE BASES DE DATOS (SGBD)	20
2.10 FUNDAMENTACIÓN DE LA METODOLOGÍA A UTILIZAR	24
2.11 RUP (RATIONAL UNIFIED PROCESS)	24
2.12 UML (UNIFIED MODELING LANGUAGE)	25
2.13 OTRAS HERRAMIENTAS UTILIZADAS	26
2.14 PROPUESTA DE TECNOLOGÍAS	27
2.15 CONCLUSIONES	27
DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN	28
3.1 INTRODUCCIÓN	28
3.2 MODELO DEL NEGOCIO	28
3.2.1 <i>Actores del negocio</i>	28
3.2.2 <i>Trabajadores del negocio</i>	29
3.3 DESCRIPCIÓN DEL PROCESO DEL NEGOCIO	29
3.3.1 <i>Diagrama del Caso de Uso del Negocio</i>	30
3.3.2 <i>Expansión del Caso de Uso</i>	30

3.4 MODELO DEL SISTEMA	32
3.4.1 <i>Requisitos Funcionales</i>	33
3.4.2 <i>Requisitos no Funcionales</i>	36
3.4.4 <i>Casos de Uso del Sistema</i>	38
3.4.5 <i>Diagrama de Casos de Uso del Sistema</i>	42
3.4.6 <i>Descripción de los Casos de Uso del Sistema</i>	43
3.5 CONCLUSIONES	58
CONSTRUCCIÓN DE LA PROPUESTA DE SOLUCIÓN	59
4.1 INTRODUCCIÓN	59
4.2 DIAGRAMAS DE CLASES	59
4.3 DIAGRAMA DE CLASES DEL DISEÑO	59
4.3.1 <i>Diagrama de Clases Diseño Web: Distribuir Manualmente</i>	60
4.3.2 <i>Diagrama de Clases Diseño Web: Gestionar Permuta</i>	60
4.3.3 <i>Diagrama de Clases Diseño Web: Eliminar Individuo</i>	61
4.3.4 <i>Diagrama de Clases Diseño Web: Mostrar Características de un Individuo</i>	61
4.3.5 <i>Diagrama de Clases Diseño Web: Modificar Características de un Apartamento</i>	62
4.3.6 <i>Diagrama de Clases Diseño Web: Adicionar Estructura (Residencia)</i>	62
4.3.7 <i>Diagrama de Clases Diseño Web: Adicionar Estructura (Manzana)</i>	63
4.3.8 <i>Diagrama de Clases Diseño Web: Adicionar Estructura (Edificio)</i>	63
4.3.9 <i>Diagrama de Clases Diseño Web: Adicionar Estructura (Apartamento)</i>	64
4.3.10 <i>Diagrama de Clases Diseño Web: Eliminar Estructura</i>	64
4.3.11 <i>Diagrama de Clases Diseño Web: Gestionar Reporte</i>	65
4.3.12 <i>Diagrama de Clases Diseño Web: Mostrar Plazas Libres por Edificio</i>	65
4.4 DISEÑO DE LA BASE DE DATOS	66
4.4.1 <i>Diagrama de Clases Persistentes</i>	66
4.4.2 <i>Modelo de Datos</i>	67
4.4.3 <i>Descripción de las Tablas de la Base de Datos</i>	68
4.5 PRINCIPIO DE DISEÑO DE INTERFAZ	72
4.5.1 <i>Estándares de la Interfaz de la Aplicación</i>	72
4.6 FORMATO DE LOS REPORTES	73
4.7 TRATAMIENTO DE EXCEPCIONES	74
4.8 ESTÁNDARES DE CODIFICACIÓN	74
4.9 MODELO DE DESPLIEGUE	75
4.10 CONCLUSIONES	76
CONCLUSIONES GENERALES	77
RECOMENDACIONES	78
BIBLIOGRAFÍA	79
GLOSARIO DE TÉRMINOS	81

INTRODUCCIÓN

La Universidad de Ciencias Informáticas (UCI) se encuentra inmersa en un gran proceso de informatización, en el cual se llevan a cabo un conjunto de proyectos con el fin de automatizar los principales procesos que en ella se desarrollan.

La UCI cuenta con un gran número de instalaciones, la mayoría de ellas está dedicada al alojamiento de los estudiantes y profesores que en ella llevan una vida interna. Como es una universidad en construcción, con frecuencia se crean nuevas instalaciones destinadas al alojamiento de personas. La dirección de la universidad ha acordado distribuir las edificaciones por residencias, donde cada residencia está distribuida por manzanas y a su vez cada manzana por edificios. Todo esto para facilitar la ubicación de las distintas facultades en el área de alojamiento correspondiente y para lograr un mayor control de estas, se han creado varias direcciones de residencia. La Dirección de Informatización ha delimitado su meta en la automatización a corto plazo del grueso de los procesos que aquí tienen lugar.

La gestión del alojamiento en las universidades es el objeto de estudio del presente Trabajo de Diploma y, más específicamente, el campo de los sistemas de gestión de alojamiento.

De ahí que la Dirección de Informatización se propusiera la elaboración de un Sistema para la Gestión del Alojamiento en la Universidad, que automatizara el proceso y así satisfacer las necesidades de los trabajadores de esta área.

Es importante aclarar que el proceso para el alojamiento en la UCI presenta características especiales que exigirán de cualquier sistema informático, un alto nivel de flexibilidad que logre asimilar lo mejor posible los constantes cambios que en la misma se producen, siempre en aras del mejorar el proceso de organización.

Actualmente distribuir todas estas personas en las diferentes instalaciones que están destinadas para este fin, es una tarea que se lleva a cabo manualmente, ya que no existe un sistema que gestione toda esta información. La ubicación en residencia de los estudiantes, profesores, o visitantes, debe ser registrada y el control de estas distribuciones como las nuevas edificaciones que se construyen de la forma en que se lleva, trae consigo desorganización e ineficiencia, más aún cuando la matrícula y la cantidad de edificaciones crecen de forma gradual cuando comienza un nuevo curso.

Constituyen estas las razones que dan origen al siguiente problema:

¿Cómo facilitar la gestión del alojamiento, de forma tal que toda la información se trabaje y se conserve de manera organizada, en la Dirección de Residencia de la UCI?

Planteándose para este el siguiente objetivo general:

Desarrollar un Sistema Automatizado que permita a los trabajadores de las direcciones de residencia gestionar toda la actividad de alojamiento, elevando así el control de la información y el nivel de informatización de la universidad.

Y los objetivos específicos:

- Posibilitar la gestión de las estructuras que conforman el área de alojamiento de la universidad.
- Permitir la realización de movimientos de personas en las estructuras del área de alojamiento.
- Crear una interfaz Web para que todos los usuarios de la universidad puedan acceder a la misma.

Para la implementación de la aplicación se utilizó ASP.NET. Se utiliza la metodología RUP para el desarrollo del sistema y UML como lenguaje de modelación para describir su funcionamiento.

Este trabajo se encuentra estructurado de la siguiente forma:

Capítulo I. En este capítulo se explica cómo se ha venido tratando el problema de la ubicación en residencia desde el primer año de vida de la universidad, mencionando los principales aspectos que generaron la idea de crear esta herramienta para eliminar estos problemas planteados.

Capítulo 2. Contiene un estudio valorativo de las tecnologías, técnicas, tendencias, metodologías y software usados en la actualidad y específicamente los utilizados para dar solución al problema.

Capítulo 3. Describe el negocio a través de un Modelo de Negocio, se hace el análisis del sistema a desarrollar. Se definen las funcionalidades del sistema y se describen detalladamente, utilizando herramientas de modelación y los principales procesos del mismo.

Capítulo 4. Aborda aspectos relacionados con la construcción de la solución propuesta, se modelan los diagramas de clases de diseño, se plantea el modelo de datos, y se especifican los principios para el diseño gráfico y la implementación.

FUNDAMENTACIÓN DEL TEMA

1.1 Introducción

En el presente capítulo se describen los procesos actuales de la Dirección de Residencia de la UCI y se identifican los principales problemas que motivan esta investigación. Se analizan varios sistemas que se encuentra en explotación en otros lugares y que pudieran constituir una posible variante de solución, justificándose su no conveniencia. Se plantea la propuesta de solución y finalmente, se fundamentan los objetivos propuestos.

1.2 Gestión

Su significado viene de la palabra, préstamo del latín *gestio*, *gestionis* 'gestión de llevar a cabo algo' derivado del supino de *gerere*, 'ejecutar' de la familia etimológica de *gesto*.

Conjunto de operaciones que se realizan para administrar y dirigir un negocio, acción o trámite que se ejecutan para conseguir o resolver algo. [4]

1.3 Gestión de Alojamiento

Las instalaciones dedicadas al hospedaje de personas tienen determinado criterio de selección, pero en general es que existan capacidades libres para brindar este servicio, luego, atendiendo a estos parámetros determinados por ellos, como capacidad, tiempo de estancia, facilidad de pago, entre otras, el cliente tiene la posibilidad de acceder a este servicio.

1.4 La Dirección de Residencia de la Universidad de las Ciencias Informáticas.

El área destinada para el alojamiento tiene un Director de Residencia (DR) y está distribuida por residencias, y cada residencia está distribuida por manzanas y a su vez cada manzana por distintos edificios, todo esto para facilitar la ubicación de las distintas facultades en la residencia correspondiente. Cada residencia tiene un Jefe de Residencia y a su vez estos

tienen personal encargado para la labor en la misma (Instructoras) y que están a cargo de la atención al personal en residencia y de reportar cualquier tipo de anomalía que exista.

Para la ubicación en general de los estudiantes en residencia, se tiene en cuenta una serie de datos, como son facultad, grupo y sexo. Para el caso de los profesores, se tiene en cuenta la categoría docente y el sexo y sólo se permitirán los apartamentos mixtos (matrimonios), con previa autorización del Jefe de Residencia.

Los procesos que tienen lugar en las direcciones de residencia y que están dentro del campo de acción de este trabajo son los siguientes, ubicar personas, la búsqueda de información de cualquier persona alojada en la residencia, apertura o cierre de cualquier estructura (manzana, edificio, apartamento) dentro de esta, la realización de permutas, dar baja a personas de la residencia y la emisión de reportes.

1.5 Descripción actual de cómo se lleva el negocio

La Dirección de Informatización tiene el listado del nuevo ingreso para su ubicación en la Residencia Universitaria, entonces pide a la Dirección de Residencia un listado de los edificios que tiene disponible así como el estado de cada apartamento y si está habitable, su capacidad. La Dirección de Residencia le pide a cada Jefe de Residencia este listado, estos a su vez llaman a sus Instructoras para que realicen esta labor, después de un trabajo bien detallado, envían este listado a los Jefes de Residencia y estos a la Dirección de Residencia, finalmente llegan a manos de la Dirección de Informatización. Con el listado del personal a ubicar y el listado de las capacidades habitables por edificio, comienza el proceso de ubicación, como no existe ningún sistema que realice todo este proceso, la operación se lleva a cabo de forma manual.

1.5.1 Reglas del Negocio

Se identificaron las siguientes reglas que debe seguir la aplicación que se desarrolle, a fin de respetar y garantizar las restricciones que existen en el negocio:

- 1 Cualquier persona de la universidad puede obtener información acerca de las personas que se alojan en la residencia universitaria.
- 2 El Director de Residencia es el único que puede hacer todos los movimientos que tienen lugar durante la gestión de alojamiento en cualquier residencia. Los Jefes de cada Residencia sólo tendrán derecho administrativo en sus residencias, no podrán efectuar ninguna acción en otra residencia.
- 3 Las instructoras pueden modificar las características actuales de los apartamentos en su residencia, confeccionar reportes y manejar información sobre las plazas libres por edificio.
- 4 Sólo el Jefe de Residencia puede abrir o cerrar una residencia, manzana, edificio o apartamento.
- 5 Solo el Jefe de Residencia puede realizar permutas.
- 6 Solo el Jefe de Residencia puede sacar a una persona de residencia, ya sea porque cause baja u otro motivo.
- 7 Un estudiante de un sexo x no puede convivir en un edificio donde los habitantes sean del sexo opuesto.
- 8 Un profesor de un sexo x no debe convivir en un apartamento donde los habitantes sean del sexo opuesto, salvo en caso de matrimonio.
- 9 No se pueden efectuar permutas entre profesores y estudiantes.

1.6 Sistemas automatizados existentes

A continuación se hace un estudio sobre algunos sistemas existentes en el ámbito nacional e internacional para la gestión del alojamiento.

Sistema del Colegio Mayor Universitario de Alicante, software dedicado al alojamiento de estudiantes en la residencia estudiantil. Brinda una facilidad de pago acorde con el tipo de habitación que se elija. Existen habitaciones con

cierto nivel de comodidad y también existen otras habitaciones con menos nivel para aquellos estudiantes que no tengan facilidad de pago.

La principal característica de este software es el comercio, ya que es un requisito indispensable el pago del servicio en la residencia, brindando facilidad de pago en tarjetas de créditos o al contado en la residencia estudiantil. [5]

Sistema de la Universidad de Cantabria, software dedicado al alojamiento de los estudiantes. Brinda la opción de buscar los pisos de alquiler para los estudiantes, teniendo en cuenta las siguientes características, localidad, tipo de habitación, sexo y renta (la facilidad de pago). En este sistema se puede reservar la habitación por el tiempo que dure el curso universitario atendiendo a la facilidad de pago. [6]

En Cuba existen varios sistemas de Gestión de Alojamiento, en su mayoría dirigidos a la reservación de habitaciones en hoteles y con características similares a las que se describen a continuación.

Sistema de Alojamiento Hotel Meliá Varadero, software que ofrece facilidad de pago en cualquier tipo de moneda. Brinda la opción de escoger el tipo de habitación que se desea reservar, la fecha y la cantidad de personas que lo harán. Posibilita ver si esa habitación está disponible para el día que se desea reservar, disminuyendo así los trámites a realizar. Da la facilidad de buscar capacidades dentro del mismo hotel o en otros hoteles del complejo turístico.

Una vez realizada la solicitud, se le envía un correo electrónico al e-mail especificado y no será válida la reserva de alojamiento hasta que no se confirme la solicitud vía e-mail. [7]

Como puede apreciarse, la totalidad de estos softwares no se ajustan a las necesidades y peculiaridades de la residencia en la UCI. Están muy lejos de las políticas que siguen las universidades cubanas para el alojamiento de las personas en sus residencias.

1.7 Propuesta de solución: Sistema de Gestión de Alojamiento.

Teniendo en cuenta la situación generada por el problema planteado anteriormente, se propone la elaboración de un sistema informático que brinde soporte a los procesos de las direcciones de residencia descritos anteriormente, teniendo estos procesos como sus principales funcionalidades.

Además debe incluir una serie de funcionalidades de administración del sistema, que permitan gestionar toda la información con la que este trabaja (estructura organizativa, permisos de usuario, roles en el sistema, información sobre las personas que están alojadas en la residencia, etc.), y que brinden la mayor flexibilidad posible, con vistas a su adaptación a cualquier tipo de situación que pueda presentarse.

El Sistema de Gestión de Alojamiento, para todas sus acciones consulta la información contenida en la base de datos central que es la que proporciona los datos de las personas que se van a alojar o están alojadas.

Además de las funcionalidades que soportan todo el flujo de procesos que ocurren en las direcciones de residencia en cuanto al alojamiento de personas, la solución propuesta brinda la posibilidad de adaptarse a cualquier disposición diferente que exista en cuanto a quienes pueden ser las personas que se alojen en la universidad, lo cual resulta actualmente todo un proceso engorroso.

1.8 Conclusiones

En este capítulo se ha descrito el objeto de estudio de este trabajo, determinándose los problemas que presenta. Se ha demostrado la necesidad de cambio y se ha planteado una propuesta de solución.

TENDENCIAS Y TECNOLOGÍAS ACTUALES A CONSIDERAR

2.1 Introducción

Para el desarrollo del sistema se realizó un estudio sobre las posibles herramientas a utilizar en su construcción. Teniendo en cuenta las tendencias actuales y las novedades en este campo.

En el presente capítulo se hace un análisis de las tecnologías y tendencias que existen en la actualidad a nivel mundial y que pudieran ser útiles en el desarrollo de la propuesta de solución. Se tienen en cuenta los servidores Web, los lenguajes de programación para la Web, los Sistemas Gestores de Bases de Datos mayormente utilizados a escala internacional, las distintas metodologías de desarrollo de software así como el lenguaje de marcado extensible (XML).

Finalmente, se seleccionan las más apropiadas tecnologías, teniendo en cuenta que las que se utilicen deben garantizar el cumplimiento de los intereses de los usuarios y de la universidad en general.

2.2 Aplicaciones Web

La plataforma Web tiene una arquitectura Cliente/Servidor. Funciona en modo “desconectado”, lo que significa que un usuario, usando un navegador o cliente, hace a través de la red, una petición de una página Web a un servidor Web, el servidor recibe dicha petición, la procesa y le envía la respuesta al cliente (en formato HTML, también a través de la red), este la recibe y se desconecta.

Se puede apreciar que los componentes de la arquitectura Web son: el servidor Web, la red física que permite la comunicación y un navegador o cliente.

Existen casos que esta arquitectura es un poco más compleja, o sea, incluye un nuevo elemento: una aplicación que se ejecuta en el servidor. Este tipo de arquitectura permite manejar lógica de negocio a través de una “aplicación Web”, que es como se denomina a este tipo de sistemas Web. La aplicación que se ejecuta en el servidor se encarga de controlar el estado del negocio y de gestionar los datos almacenados con ayuda de algún Sistema Gestor de Bases de Datos (SGBD).

En la actualidad se ha generalizado el uso de aplicaciones Web dada las grandes posibilidades que brindan, y dado que los clientes sólo necesitan un navegador, capaz de interpretar código con formato HTML, para hacer uso de ellas: no tienen que instalar ningún componente de software adicional. Por tanto se propone que el sistema a desarrollar consista en una aplicación Web.

2.3 La tecnología .NET

Microsoft.NET es el conjunto de nuevas tecnologías en las que Microsoft ha estado trabajando durante los últimos años, con el objetivo de obtener una plataforma sencilla y potente para distribuir el software en forma de servicios, de forma tal que puedan ser suministrados remotamente, comunicándose y combinándose unos con otros totalmente independiente de la plataforma, lenguaje de programación y modelo de componentes con los que hayan sido desarrollados.

.NET ofrece un entorno de desarrollo de aplicaciones llamado Visual Studio .NET que consta de varios lenguajes de programación como Visual Basic .NET, Visual C#, Visual FoxPro y Visual C++ .NET. Estos lenguajes combinan las características de los lenguajes existentes con nuevas posibilidades para proporcionar un potente sistema de desarrollo. A continuación, se detallan algunas de las características de la Arquitectura .NET.

2.3.1 Arquitectura Framework.NET

La arquitectura .NET (.NET Framework) es el modelo de programación de la plataforma .NET para construir y ejecutar los servicios .NET. El objetivo de esta arquitectura es la de reducir la complejidad en el desarrollo de este tipo de aplicaciones, permitiendo a los desarrolladores centrarse en escribir la lógica específica del servicio a desarrollar.

2.4 Lenguajes de programación para la Web

- C#

C# es un lenguaje de programación poderoso, diseñado para escribir aplicaciones empresariales. Es una evolución de los lenguajes C y C++. Utiliza muchas de las características de C++ en las áreas de instrucciones, expresiones y operadores.

Microsoft C#

Desde su introducción en febrero de 2001, numerosos desarrolladores han comenzado a crear software utilizando el lenguaje de programación C#. Incluso en Microsoft, C# se ha utilizado para crear varias aplicaciones comerciales, entre las que se incluyen .NET Framework, las propiedades Web de MSN y el SDK de Tablet PC. Como tal, C# ha demostrado ser un lenguaje adecuado para la creación de software comercial de alta calidad.

Gran parte de las características del lenguaje C# se crearon con cuatro objetivos de diseño diferentes en mente:

Disponer de un sistema de tipos unificado y simplificar el modo en que el lenguaje utiliza los tipos de valor y referencia.

Utilizar un diseño basado en componentes, establecido a través de características como comentarios XML, atributos, propiedades, eventos y delegados.

Disponer de espacio práctico para los desarrolladores establecido a través de las capacidades únicas del lenguaje C#, entre las que se incluye la manipulación segura del puntero y la comprobación de desbordamiento, entre otras.

Utilizar construcciones de lenguaje pragmáticas, como las instrucciones `foreach` y `using`, que aumentan la productividad del desarrollador.

Características potenciales futuras

Sin lugar a dudas, las innovaciones futuras relativas a C# se basarán en un sistema de tipos unificado, un desarrollo basado en componentes, espacio en cabeza para los desarrolladores y construcciones de lenguaje pragmáticas. El lenguaje proporciona la capacidad de generar componentes de sistema duraderos en virtud de las siguientes características:

- ✓ Total compatibilidad entre COM y plataforma para integración de código existente.
- ✓ Gran robustez, gracias a la recolección de elementos no utilizados (liberación de memoria) y a la seguridad en el tratamiento de tipos.
- ✓ Seguridad implementada por medio de mecanismos de confianza intrínsecos del código.
- ✓ Plena compatibilidad con conceptos de meta datos extensibles.

Además, es posible interactuar con otros lenguajes, entre plataformas distintas, y con datos heredados, en virtud de las siguientes características:

- Plena interoperabilidad por medio de los servicios de COM+ 1.0 y .NET Framework con un acceso limitado basado en bibliotecas.
- Compatibilidad con XML para interacción con componentes basados en tecnología Web.
- Capacidad de control de versiones para facilitar la administración y la implementación.

- **PHP**

Su interpretación y ejecución se da en el servidor en el cual se encuentra almacenada la página y el cliente solo recibe el resultado de la ejecución. Cuando el cliente hace una petición al servidor para que le envíe una página Web, enriquecida con código PHP, el servidor interpretará las instrucciones mezcladas en el cuerpo de la página y las sustituirá con el resultado de la ejecución antes de enviar el resultado a la computadora del cliente. Además es posible utilizarlo para generar archivos, Flash o JPG, entre otros.

Permite la conexión a numerosas bases de datos de forma nativa tales como MySQL, Postgres, Oracle, ODBC, IBM DB2, Microsoft SQL Server y SQLite, lo cual permite la creación de Aplicaciones Web muy robustas.

PHP tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos tales como UNIX, Linux, Windows y Mac OS X, y puede interactuar con los servidores de Web más populares.

- **PERL**

Perl (Practical Extraction and Report Language) es un lenguaje de programación desarrollado por Larry Wall (lwall at netlabs.com) inspirado en otras herramientas de UNIX como son: sed, grep, awk, c-shell, para la administración de tareas propias de sistemas UNIX.

No establece ninguna filosofía de programación concreta. No se puede decir que sea orientado a objetos, modular o estructurado aunque soporta directamente todos estos paradigmas; su punto fuerte son las labores de procesamiento de textos y archivos.

No es ni un compilador ni un intérprete, está en un punto intermedio, cuando mandamos a ejecutar un programa en Perl, se compila el código fuente a un código intermedio en memoria que se optimiza como si se fuera a elaborar un programa ejecutable pero es ejecutado por un motor, como si se tratase de un intérprete.

Lenguaje de programación basado en scripts portable a casi cualquier plataforma. Es muy utilizado para escribir CGIs. Uno de sus elementos más potentes son las expresiones regulares, que a partir de su versión en Perl han sido adoptadas por otros lenguajes y plataformas como .NET o Javascript.

Lenguaje optimizado para el escaneo de texto arbitrario de ficheros. Es también un buen lenguaje para tareas de administración de sistemas. Es un lenguaje con intención de ser práctico en lugar de bonito. Satisface las tres virtudes del programador: flojera, impaciencia y petulancia.

2.5 ASP.NET

Es un marco de trabajo de programación generado en Common Language Runtime que puede utilizarse en un servidor para generar eficaces aplicaciones Web. Ofrece varias ventajas importantes acerca de los modelos de programación Web anteriores:

Mejor rendimiento. Puede aprovechar las ventajas del enlace anticipado, la compilación just-in-time, la optimización nativa y los servicios de caché desde el primer momento. Esto supone un incremento espectacular del rendimiento antes de siquiera escribir una línea de código.

Compatibilidad con herramientas de primer nivel. El marco de trabajo de ASP.NET se complementa con un diseñador y una caja de herramientas muy completos en el entorno integrado de programación (Integrated Development Environment, IDE) de Visual Studio.

Eficacia y flexibilidad. La biblioteca de clases de .NET Framework, la Mensajería y las soluciones de Acceso a datos, se encuentran accesibles desde el Web de manera uniforme. ASP.Net es también independiente del lenguaje, por lo que puede elegir el que mejor se adapte a la aplicación o dividirla en varios lenguajes.

Simplicidad. Facilita la realización de tareas comunes, desde el sencillo envío de formularios y la autenticación del cliente hasta la implementación y la configuración de sitios.

Facilidad de uso. Emplea un sistema de configuración jerárquico, basado en texto, que simplifica la aplicación de la configuración al entorno del servidor y las aplicaciones Web. No se requiere el reinicio del servidor, ni siquiera para implementar o reemplazar el código compilado en ejecución.

Escalabilidad y disponibilidad. El motor de tiempo de ejecución de ASP.NET controla y administra los procesos de cerca, por lo que si uno no se comporta adecuadamente (filtraciones, bloqueos), se puede crear un proceso nuevo en su lugar, lo que ayuda a mantener la aplicación disponible constantemente para controlar solicitudes.

Posibilidad de personalización y extensibilidad. Permite insertar código en el nivel adecuado, siendo posible extender o reemplazar cualquier

subcomponente del motor de tiempo de ejecución de ASP.NET con su propio componente escrito personalizado.

Seguridad. Con la autenticación de Windows integrada y la configuración por aplicación, se puede tener la completa seguridad de que las aplicaciones están a salvo.

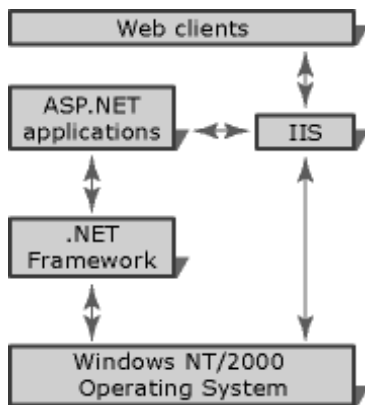


Figura1. Arquitectura ASP.NET

Con la aparición del comercio electrónico se ha producido un incremento de la complejidad del desarrollo de este tipo de sistemas, lo cual supone un conjunto de desafíos para los desarrolladores, de los cuales pudiera citarse:

Implementación de interfaces Web enriquecidas.

Separación del cliente y el servidor.

Ejecución sin control de estado.

Capacidades del cliente desconocidas.

Complicaciones en el acceso a datos.

Complicaciones con la escalabilidad.

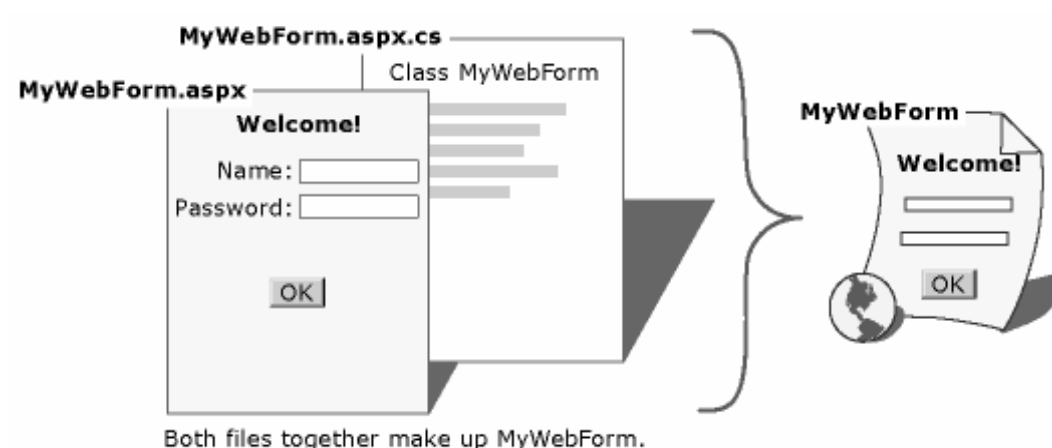


Figura2. WebForms

La plataforma ASP.NET asume estos retos proporcionando a los desarrolladores las siguientes características:

Modelo de objetos intuitivo y consistente: El marco de trabajo de las páginas ASP.NET presenta un modelo de objetos que permite concebir a los formularios como unidades, no como piezas separadas en el cliente y en el servidor. Con este modelo, se programan las páginas en una forma más intuitiva que en las aplicaciones Web tradicionales, incluyendo la capacidad de establecer propiedades para los elementos del formulario y responder a eventos. Por otro lado, los controles del servidor de ASP.NET son una abstracción del contenido físico de una página HTML y de la interacción directa entre el navegador y el servidor. En sentido general, se pueden utilizar los controles del servidor de la misma forma en que pudiera trabajarse con los controles en aplicaciones clientes sin tener que pensar en como crear el HTML para presentar y procesar los controles y su contenido.

Modelo de programación dirigido por eventos. Las paginas WebForms traen a las aplicaciones Web el familiar modelo de escribir manipuladores para eventos que ocurran tanto en el cliente como en el servidor. El marco de trabajo de ASP.NET abstrae este modelo de tal forma que el mecanismo subyacente de captura del evento en el cliente, su transmisión al servidor y la llamada a método apropiado es automática y transparente para el programador. El resultado es una clara y fácil estructura de código que soporta desarrollo dirigido por eventos.

Administración de estado intuitiva. El marco de trabajo de ASP.NET automáticamente manipula la tarea de mantener el estado de la información específica de la aplicación. Esta es llevada a cabo sin un uso intensivo de los recursos del servidor y puede ser implementada con o sin el envío de cookies al navegador.

Aplicaciones independientes del navegador. ASP.NET permite la creación de toda la lógica de la aplicación en el servidor, eliminado la necesidad de producir código para diferentes navegadores. Sin embargo, este aún permite que automáticamente se tome ventaja de características específicas de los diferentes navegadores mediante la escritura de código del lado del cliente para mejorar el rendimiento.

2.6 Arquitectura de Software

La Arquitectura de Software es, a grandes rasgos, una vista del sistema que incluye los componentes principales del mismo, la conducta de esos componentes según se la percibe desde el resto del sistema y las formas en que los componentes interactúan y se coordinan para alcanzar la misión del sistema. La vista arquitectónica es una vista abstracta, aportando el más alto nivel de comprensión y la supresión o diferimiento del detalle inherente a la mayor parte de las abstracciones. [8]

2.7 Arquitectura en 3 capas

En las aplicaciones diseñadas usando un modelo de tres capas, el sistema es dividido en datos, negocio y presentación. La idea de esta arquitectura está basada principalmente en la capacidad de estabilidad que nos ofrece. Esta disposición favorece al mantenimiento de la aplicación, a la vez que reduce el tiempo necesario para lanzar nuevas versiones del producto. [11]

1.- Capa de presentación: Es la que ve el usuario, presenta el sistema al usuario, le comunica la información y captura la información del usuario, dando un mínimo de proceso (realiza un filtrado previo para comprobar que no hay errores de formato). Esta capa se comunica únicamente con la capa de negocio.

2.- Capa de negocio: Es donde residen los programas que se ejecutan, recibiendo las peticiones del usuario y enviando las respuestas tras el proceso. Se denomina capa de negocio (e incluso de lógica del negocio) pues es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos para almacenar o recuperar datos de él.

3.- Capa de datos: Es donde residen los datos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos,

reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

Ventajas

Desarrollos paralelos (en cada capa)

Aplicaciones más robustas debido al encapsulamiento

Mantenimiento y soporte más sencillo (es más sencillo cambiar un componente que modificar una aplicación monolítica)

Mayor flexibilidad (se pueden añadir nuevos módulos para dotar al sistema de nueva funcionalidad)

Alta escalabilidad. La principal ventaja de una aplicación distribuida bien diseñada es su buen escalado, es decir, que puede manejar muchas peticiones con el mismo rendimiento simplemente añadiendo más hardware. El crecimiento es casi lineal y no es necesario añadir más código para conseguir esta escalabilidad.

2.8 Web services

Los servicios Web son la revolución informática de la nueva generación de aplicaciones que trabajan colaborativamente en las cuales el software esta distribuido en diferentes servidores.

Los servicios Web XML permiten que las aplicaciones compartan información y que además invoquen funciones de otras aplicaciones independientemente de cómo se hayan creado las aplicaciones, cuál sea el sistema operativo o la plataforma en que se ejecutan y cuáles los dispositivos utilizados para obtener acceso a ellas. Aunque los servicios Web XML son independientes entre sí, pueden vincularse y formar un grupo de colaboración para realizar una tarea determinada.

Los servicios XML Web Services son los elementos fundamentales en la evolución hacia la computación distribuida a través de Internet. Se están convirtiendo en la plataforma de integración de aplicaciones gracias a los estándares abiertos y al énfasis en la comunicación y colaboración entre personas y aplicaciones. Las aplicaciones se crean utilizando los servicios XML

Web Services múltiples de origen distinto que funcionan conjuntamente, sin importar su ubicación o la forma en que se implementaron.

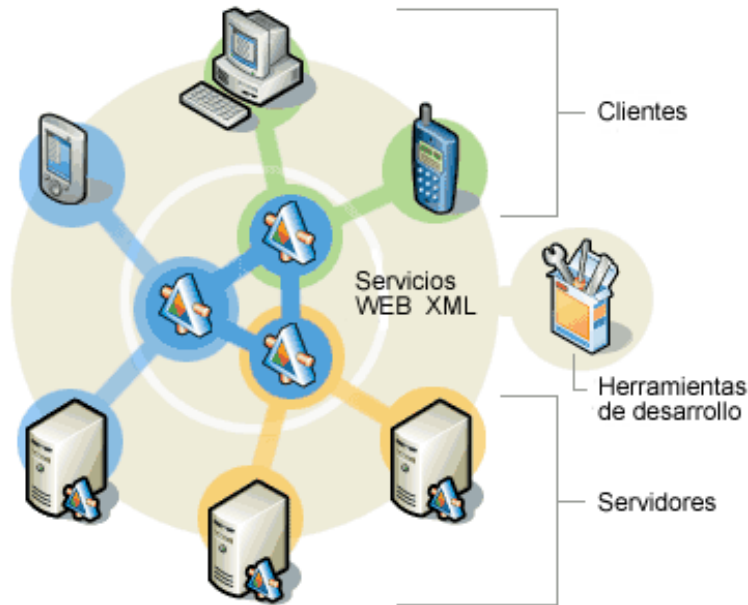


Fig. 3 Los servicios Web pueden conectar a diversos tipos de aplicaciones.

En la UCI todas las aplicaciones que se utilizan en la intranet ofrecen o “consumen” servicios Web de otras, es decir, existe una interrelación entre los sistemas de la red para lograr la reutilización de funcionalidad de estos.

2.9 Sistema Gestor de Bases de Datos (SGBD)

Un Sistema Gestor de Bases de Datos (SGBD) puede definirse como un paquete generalizado de software, que se ejecuta en un sistema computacional anfitrión, centralizando los accesos a los datos y actuando de interfaz entre los datos físicos y el usuario. Las principales funciones que debe cumplir un SGBD se relacionan con la creación y mantenimiento de la base de datos, el control de accesos, la manipulación de datos de acuerdo con las necesidades del usuario, el cumplimiento de las normas de tratamiento de datos, evitar redundancias e inconsistencias y mantener la integridad. Los SGBD permiten al programador convencional ahorrarse horas de trabajo dedicadas a la seguridad, gestión de los datos, chequeo de errores, etc. [9]

Entre los SGBD comúnmente utilizados en el mundo tenemos Oracle, MySQL, Microsoft SQL Server, PostgreSQL, Internase, entre otros. Todos estos presentan un enfoque relacional con un buen basamento matemático centrado en el Álgebra Relacional.

- **PostgreSQL**

PostgreSQL es un servidor de base de datos relacional libre, liberado bajo la licencia BSD. Es una alternativa a otros sistemas de bases de datos de código abierto (como MySQL, Firebird y MaxDB), así como sistemas propietarios como Oracle o DB2.

Algunas de sus principales características son:

Claves ajenas también denominadas Llaves ajenas o Llaves Foráneas (foreign keys)

Disparadores (triggers)

Vistas

Integridad transaccional

Acceso concurrente multiversión (no se bloquean las tablas, ni siquiera las filas, cuando un proceso escribe)

Capacidad de albergar programas en el servidor en varios lenguajes.

Herencia de tablas

Tipos de datos y operaciones geométricas

- **Oracle**

Oracle es un sistema de administración de base de datos (o RDBMS por el acrónimo en inglés de Relational Data Base Management System), fabricado por Oracle Corporation. Se considera a Oracle como uno de los sistemas de bases de datos más completos, destacando su:

Soporte de transacciones.

Estabilidad.

Escalabilidad.

Es multiplataforma.

Su mayor defecto es su enorme precio, que es de varios miles de euros (según versiones y licencias). Otro aspecto que ha sido criticado por algunos especialistas es la seguridad de la plataforma, y las políticas de suministro de parches de seguridad, modificadas a comienzos de 2005 y que incrementan el nivel de exposición de los usuarios. En los parches de actualización provistos durante el primer semestre de 2005 fueron corregidas 22 vulnerabilidades públicamente conocidas, algunas de ellas con una antigüedad de más de 2 años.

Aunque su dominio en el mercado de servidores empresariales ha sido casi total hasta hace poco, recientemente sufre la competencia del Microsoft SQL Server de Microsoft y de la oferta de otros RDBMS con licencia libre como PostgreSQL, MySql o Firebird. Las últimas versiones de Oracle han sido certificadas para poder trabajar bajo **Linux**.

- **Microsoft SQL Server**

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

Entre sus características figuran:

Soporte de transacciones.

Gran estabilidad.

Gran seguridad.

Escalabilidad.

Soporta procedimientos almacenados.

Incluye también un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.

Permite trabajar en modo cliente-servidor donde la información y datos se alojan en el servidor y las terminales o clientes de la red sólo accedan a la información.

Además permite administrar información de otros servidores de datos

Este sistema incluye una versión reducida, llamada MSDE con el mismo motor de base de datos pero orientado a proyectos más pequeños. Microsoft SQL Server constituye la alternativa de Microsoft a otros potentes sistemas gestores de bases de datos como son Oracle o Sybase.

Es común desarrollar completos proyectos complementando Microsoft SQL Server y Microsoft Access a través de los llamados ADP (Access Data Project). De esta forma se completa una potente base de datos (Microsoft SQL Server) con un entorno de desarrollo cómodo y de alto rendimiento (VBA Access) a través de la implementación de aplicaciones de dos capas mediante el uso de formularios Windows.

Para el desarrollo de aplicaciones más complejas (tres o más capas), Microsoft SQL Server incluye interfaces de acceso para la mayoría de las plataformas de desarrollo, incluyendo .NET.

Microsoft SQL Server, al contrario de su más cercana competencia, no es multiplataforma, ya que sólo está disponible en Sistemas Operativos de Microsoft.

Selección del Sistema SGBD

Como SGBD se seleccionó el Microsoft SQL Server 2000 ya que es un programa completo de base de datos y análisis. Tiene capacidad para consultar la base de datos mediante un explorador y compatibilidad con el Lenguaje de marcado extensible (XML, *Extensible Markup Language*), es un gestor de bases de datos totalmente habilitado para Web. Además, ostenta marcas de referencia en cuanto a escalabilidad y confiabilidad, que son críticas para el éxito de bases de datos de gran tamaño. El SQL Server permite lograr una gran velocidad en el procesamiento de transacciones, y agilidad en todas sus operaciones.

Teniendo en cuenta de que la infraestructura de las aplicaciones que existen en la Universidad están en su gran mayoría soportadas por este gestor, se incluyó como requerimiento especial por parte del cliente principal del sistema que se utilizara el SQL Server 2000 como SGBD.

2.10 Fundamentación de la metodología a utilizar

La calidad en el desarrollo y mantenimiento del software se ha convertido hoy en día en uno de los principales objetivos estratégicos de las organizaciones, debido a que cada vez más, los procesos principales dependen de los sistemas informáticos para su buen funcionamiento. En los últimos años se han publicado diversos estudios y estándares en los que se exponen los principios que se deben seguir para la mejora de los procesos de software.

Para controlar, y planificar la propuesta que presenta este trabajo, se decidió utilizar como metodología el Proceso Unificado de Modelado (RUP), por sus características y las facilidades que aporta a todo el proceso. Y teniendo en cuenta de que viene acompañado de una herramienta muy buena que soporta cada uno de los procesos que necesitamos.

2.11 RUP (Rational Unified Process)

El Proceso Unificado de Modelado (RUP) es el producto final de tres décadas de desarrollo y uso práctico. Su desarrollo sigue un camino desde 1967 con la Metodología Ericsson (Ericsson Approach), una aproximación de desarrollo basada en componentes, que introdujo el concepto de caso de uso; pasando por el proceso Objectory de Rational (publicado en 1997) hasta el Proceso Unificado de Rational

El Proceso Unificado es una propuesta de proceso para el desarrollo de software orientado a objeto que utiliza *Unified Model Language (UML)* para describir todo el proceso. Está basado en componentes, lo cual quiere decir que el sistema software en construcción está formado por componentes software interconectados a través de interfaces bien definidas

Sus características principales son

- Guiado/Manejado por casos de uso.
- Centrado en arquitectura.
- Iterativo e Incremental.
- Desarrollo basado en componentes.
- Utilización de un único lenguaje de modelación.
- Proceso Integrado.

2.12 UML (Unified Modeling Language)

El desarrollo del Unified Modeling Lenguaje, (UML) empezó en octubre de 1994, cuando Grady Booch y Jim Rumbaugh en la Rational Software Corp. empezaron a trabajar para unificar el Booch y la OMT. Un proyecto versión 0.8 del Método Unificado (UML), como se llamó desde un comienzo, salió al público en octubre de 1995. En el otoño de 1995, Ivar Jacobson se unió a la compañía y unió su esfuerzo al nuevo modelo, uniendo el OOSE al UML.

En resumen UML es el resultado de la experiencia sumada, anotaciones, y conceptos. Ya que todos las metodologías bases han tenido una aplicación extensa en el campo del la POO se han desarrollado en la práctica, tienen su historia, y han sido aplicados en una gran variedad de industrias y problemas por lo que pueden ser clasificadas como muy maduras. *UML no es una salida revolucionaria de Booch, OMT, y OOSE, sino una evolución y síntesis de estos tres.*

El UML es un lenguaje gráfico para visualizar, especificar, construir y documentar los artefactos de un sistema con gran cantidad de software. UML proporciona una forma estándar de escribir los planos de un sistema, cubriendo tanto las cosas conceptuales, tales como procesos del negocio y funciones del sistema, como las cosas concretas, tales como las clases escritas en un lenguaje de programación específico, esquemas de bases de datos y componentes software reutilizables.[7]

2.13 Otras Herramientas Utilizadas

Como vamos a utilizar una aplicación Web para confeccionar la propuesta de este trabajo, se hace necesario tener en cuenta la utilización de un editor de páginas Web.

- **Macromedia Dreamweaver MX 2004**

Para estas funciones la elección no ha sido muy difícil, ya que la herramienta de creación de sitios Web más utilizada en la actualidad es el Macromedia Dreamweaver MX 2004. Con esa herramienta podremos desarrollar cualquier sitio Web personal con características de sitio profesional y utilizar casi todos los recursos de la Web, así como realizar aplicaciones que se ejecuten en servidor y vinculaciones dinámicas de datos, logrando un diseño fácil y óptimo.

- **Rational Rose**

Es una herramienta para “modelado visual”, que forma parte de un conjunto más amplio de herramientas que juntas cubren todo el ciclo de vida del desarrollo de software. Permite completar una gran parte de las disciplinas (flujos fundamentales) del proceso unificado de Rational (RUP) e incluye un conjunto de herramientas de ingeniería inversa y generación de código que allanan el camino hasta el producto final.

2.14 Propuesta de tecnologías

Tomando como base los elementos antes expuestos de algunas de las herramientas y tecnologías disponibles actualmente, utilizaremos la plataforma .NET de Microsoft (C# y ASP .NET) para el desarrollo de una aplicación que permita aprovechar al máximo sus ventajas.

Se utilizó SQL Server como Gestor de Base de Datos por su fortaleza y capacidad para grandes volúmenes de información.

La implantación en la UCI de tecnología de código abierto eventualmente, obligará a las aplicaciones existentes en la intranet construidas en .NET a migrar hacia plataformas como Mono, la versión de código abierto de .Net. La solución aquí planteada puede ser migrada a esta plataforma de una manera segura, ya que no se han utilizado ninguna característica no estándar de la misma.

2.15 Conclusiones

En este capítulo profundizamos en el conocimiento de algunos conceptos necesarios para la comprensión de este trabajo. Además realizamos un análisis completo de las tecnologías que serán utilizadas a lo largo del desarrollo del sistema propuesto, y se fundamentaron las elecciones del lenguaje, el sistema gestor de bases de datos, y la metodología a utilizar. Una vez conocidas las herramientas óptimas, y los conceptos a utilizar, podemos empezar a desarrollar la propuesta de sistema.

DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN

3.1 Introducción

En el presente capítulo se hace la descripción de la propuesta que trae este trabajo, para ello se describen los procesos del negocio que tiene que ver con el objeto de estudio. Es importante conocer cómo funcionan estos procesos. Se enumeran los requisitos funcionales y no funcionales que debe tener el sistema que proponemos, lo que permite hacer una concepción general del sistema, e identificar mediante un Diagrama de Caso de Uso, las relaciones de los actores que interactúan con el sistema, y las secuencias de acciones con las que interactúan.

3.2 Modelo del negocio

3.2.1 Actores del negocio

Tabla 1. Actores del negocio

ACTOR DEL NEGOCIO	DESCRIPCION
Dirección de Informatización	Entidad que se ocupa de automatizar todos los procesos en la Universidad

3.2.2 Trabajadores del negocio

Tabla 2.Trabajadores del negocio

TRABAJADORES DEL NEGOCIO	DESCRIPCION
Director de Residencia	Persona que tiene a su cargo la máxima dirección de la Residencia Universitaria en su totalidad.
Jefe de Residencia	Persona responsable de una dirección de residencia. Controla el funcionamiento en la residencia que administra. Orienta y dirige a las instructoras que trabajan en la residencia.
Instructora	Responsable del funcionamiento de la vida interna de cada edificio, además de ser como educadoras para nuestros estudiantes y guías para nuestros profesores.

3.3 Descripción del proceso del negocio

Caso de Uso del Negocio

El caso de uso Distribuir personal en Residencia se inicia cuando la Dirección de Informatización tiene los listados de los estudiantes y los listados de los profesores para su ubicación en la Universidad, pide a la Dirección de Residencia un listado de las capacidades habitables que existen en cada edificio.

La residencia está distribuida por tres áreas, estas áreas a su vez están distribuidas por manzanas, cada residencia tiene un Jefe de Residencia e Instructoras encargadas del funcionamiento de la misma. Cada Jefe de Residencia pide a sus Instructoras los apartamentos habitables por edificios, estas después de realizar este trabajo, confeccionan el listado de las capacidades por apartamentos, el cual es enviado al Jefe de Residencia, quien al recibirlo lo envía a la Dirección de Informatización.

Cuando la Dirección de Informatización tiene los listados de los estudiantes y el listado de profesores a ubicar en Residencia y las capacidades habitables por edificio, comienza la ubicación manualmente teniendo en cuenta los distintos parámetros predefinidos, como son Facultad, Año, Sexo y Grupo.

3.3.1 Diagrama del Caso de Uso del Negocio

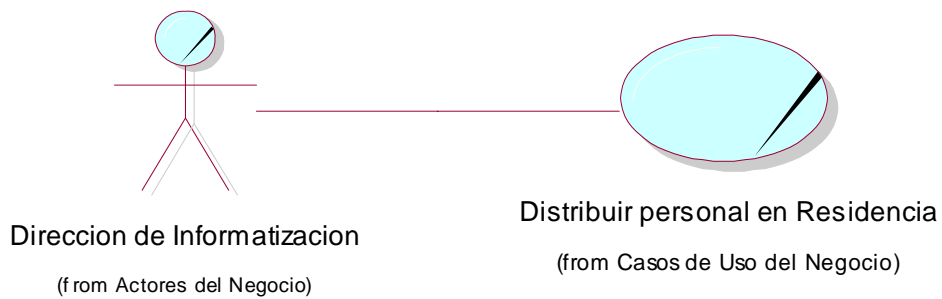


Fig. 4 Diagrama de Caso de Uso del Negocio

3.3.2 Expansión del Caso de Uso

Diagrama de actividades

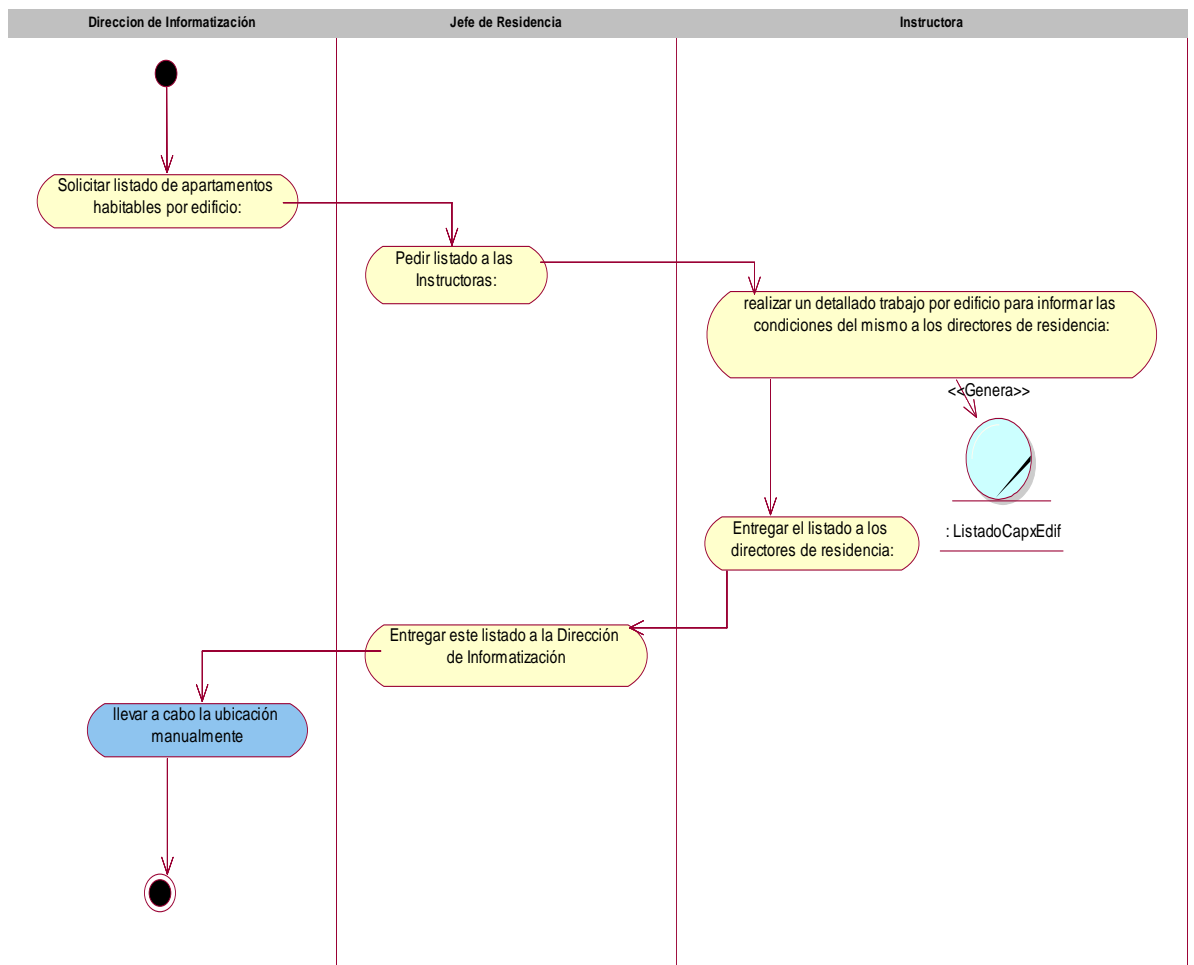


Fig. 5 Diagrama de Actividades

Modelo de Objetos

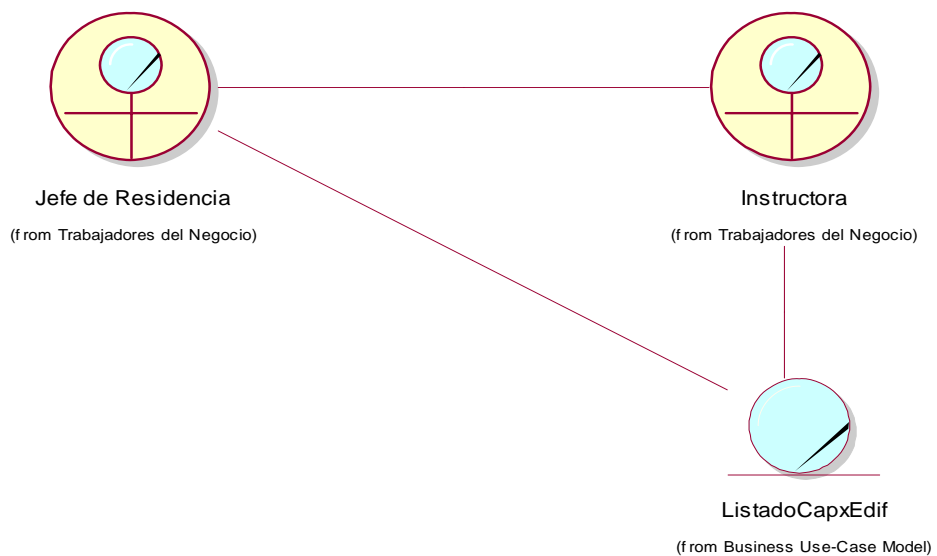


Fig. 6 Modelo de Objetos

3.4 Modelo del sistema

Para cumplimentar los objetivos propuestos al inicio de este trabajo, y teniendo en cuenta todos los requerimientos planteados, se definen varios roles para diferenciar el nivel de acceso al sistema, Administradores, Usuarios Avanzados, Usuarios Simples.

El rol de Administrador, los usuarios que tengan este rol podrán realizar la distribución de la matrícula en la Residencia y hacer todos los cambios pertinentes en la misma, como Gestionar Permuta, Crear o Destruir Estructuras, Manzanas o Residencias Eliminar o Ubicar Manualmente un estudiante o un profesor en Residencia. El rol de Administrador lo tendrá el Director de Residencia y los Jefes de Residencia de cada área en la que está dividida la misma. El sistema le otorga todos los permisos al Director de Residencia, este podrán realizar todas las acciones en cualquier residencia. Los Jefes de Residencia podrán realizar las mismas acciones que el DR pero sólo en la residencia que dirigen. Al introducir su clave, el sistema cargará sólo aquellas áreas permitidas para este usuario.

El Usuario Avanzado sólo tendrá permiso para efectuar pequeños cambios, como el de Gestionar Reportes, Modificar Características de un Apartamento y Mostrar Plazas Libres por Edificio. Al introducir su clave en la aplicación, esta cargará sólo dichas opciones. En este rol se encontrarán las Instructoras de los edificios.

En el rol Usuarios Simples están todos aquellos usuarios que accedan a la aplicación dentro de la Universidad, pero solo tendrán permiso para buscar información acerca de personas que ya se encuentran alojadas, no para modificar.

3.4.1 Requisitos Funcionales

R1. Autenticar usuario

1.1 Usuario UCI

1.2 Clave UCI

1.3 Si el usuario que intenta conectarse es un usuario del dominio uci.cu, cargar sólo aquellas opciones que le están permitidas de acuerdo al rol que tenga asociado en la aplicación.

R2 Realizar distribución manual de estudiantes y profesores.

2.1 Para el caso de los estudiantes:

2.1.1 Permitir que el usuario realice una búsqueda de la persona a ubicar atendiendo a varios criterios (nombre, usuario, carné de identidad).

2.1.2 Brindar la posibilidad de seleccionar el estudiante a ubicar

2.1.3 Mostrar los apartamentos disponibles en la residencia

2.1.4 Seleccionar el apartamento donde se ubicará el estudiante

2.1.5 Mostrar un mensaje de confirmación o error en dependencia de si pudo realizar la acción.

2.2 Para el caso de los profesores:

2.2.1 Permitir que el usuario realice una búsqueda de la persona a ubicar atendiendo a varios criterios (nombre, usuario, carné de identidad).

2.2.2 Brindar la posibilidad de seleccionar el profesor a ubicar

2.2.3 Mostrar los apartamentos de profesores disponibles en la residencia

2.2.4 Seleccionar el apartamento donde se ubicará el profesor

2.2.5 Mostrar un mensaje de confirmación o error en dependencia de si pudo realizar la acción.

R3 Mostrar listado de estudiantes por apartamento.

3.1 Mostrar, dado el número del apartamento, un listado con los nombres de los estudiantes que lo habitan y sus datos (Primer Nombre, Segundo Nombre, Primer Apellido, Segundo Apellido, Sexo, Tipo de Persona).

R4 Mostrar apartamentos con plazas libres dentro de un edificio.

4.1 Mostrar dado un edificio, un listado con los apartamentos que no están ocupados totalmente así como la cantidad de capacidades libres por apartamento.

R5 Mostrar listado de profesores por apartamento.

5.1 Mostrar dado el número del apartamento, un listado con los nombres de los profesores que lo habitan y sus datos correspondientes (Primer Nombre, Segundo Nombre, Primer Apellido, Segundo Apellido, Sexo, Tipo de Persona).

R6 Mostrar características de un estudiante.

6.1 Permitir que el usuario realice una búsqueda del estudiante atendiendo a varios criterios (nombre, usuario, carné de identidad).

6.2 Mostrar dado un estudiante a buscar, Apartamento, Primer Nombre, Segundo Nombre, Primer Apellido, Segundo Apellido, Sexo, Tipo de Persona.

R7 Mostrar características de un profesor.

7.1 Permitir que el usuario realice una búsqueda del profesor atendiendo a varios criterios (nombre, usuario, carné de identidad).

7.2 Mostrar dado un profesor a buscar, Apartamento, Primer Nombre, Segundo Nombre, Primer Apellido, Segundo Apellido, Sexo, Tipo de Persona.

R8 Eliminar un estudiante.

8.1 Permitir que el usuario que tenga privilegios de administración en el sistema, elimine a un estudiante de un apartamento liberando una capacidad en el mismo.

R9 Eliminar un profesor.

9.1 Permitir que el usuario con privilegios de administración elimine a un profesor de un apartamento liberando una capacidad en el mismo.

R10 Mostrar características de un edificio.

10.1 Mostrar capacidad total, cantidad de habitantes, cantidad de apartamentos, cantidad de capacidades libres.

R11 Modificar características de un apartamento.

11.1 Mostrar número del apartamento, capacidad total, cantidad de habitantes, si está habitable, teléfono.

11.2 Brindar la opción de modificar las características del apartamento.

R12 Adicionar nueva manzana

12.1 Crear una nueva manzana y adicionarla al listado de manzanas registradas

R13 Adicionar un nuevo edificio

13.1 Crear un nuevo edificio y adicionarlo al listado de edificios registrados

R14 Gestionar Permutas

14.1 Efectuar las permutas de profesores o estudiantes, para los estudiantes tener en cuenta sexo y facultad y para los profesores, sexo.

R15 Adicionar Residencia

15.1 Crear una residencia y adicionarla al listado de residencias registradas

R16 Gestionar Reportes.

16.1 Brindar la posibilidad de confeccionar reportes con los siguientes datos:

16.1.1 Reporte general, mostrar por residencia todas las manzanas y de cada una de ellas los edificios que la componen con sus respectivos apartamentos. De cada apartamento, mostrar número, capacidad, sexo, teléfono y si está habitable o no.

16.1.2 Reporte de residencia, mostrar todas las residencias registradas y de cada una de ellas la cantidad de personas de sexo femenino y masculino, capacidades libres y las capacidades ocupadas.

16.1.3 Reporte de manzana, mostrar todas las manzanas de una residencia, y de cada manzana su número, la cantidad de personas de sexo femenino y masculino, capacidades libres y las capacidades ocupadas.

16.1.4 Reporte de edificio, mostrar todos los edificios de una residencia, y de cada edificio su número, la cantidad de personas de sexo femenino y masculino, capacidades libres y las capacidades ocupadas.

16.1.5 Reporte de personas, mostrar listado de personas por edificio y apartamento.

16.2 Salvar los reportes en un documento.

R17 Adicionar un Apartamento

17.1 15.1 Crear un apartamento y adicionarlo al listado de apartamentos registrados de un edificio.

R18 Eliminar Estructura

18.1 Eliminar Residencia, Manzana, Edificio o Apartamento y eliminar del listado de estructuras registradas, la seleccionada para eliminar.

R19 Distribuir Automáticamente

19.1 Distribuir automáticamente los estudiantes y profesores en las instalaciones de residencia, atendiendo a los principios trazados por la Dirección de Residencia.

3.4.2 Requisitos no Funcionales

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

Apariencia o interfaz externa:

- Interfaz amigable, lo más homogénea posible.
- Diseño sencillo, permitiendo la utilización del sistema sin mucho entrenamiento.

Soporte

- Garantía de instalación y prueba del sistema, además de un breve entrenamiento a los futuros usuarios.

Seguridad

1. Existencia de distintos roles que establezcan las acciones que pueden realizar los usuarios.
2. Permanencia de las acciones realizadas por cada usuario en un historial.
3. Utilización de transacciones para el trabajo con los datos, para garantizar la integridad de los mismos.

Confiabilidad

- Garantía de un tratamiento adecuado de las excepciones y validación de las entradas del usuario.

3.4.3 Actores del sistema

Tabla 3: Actores del Sistema

ACTORES DEL SISTEMA	DESCRIPCIÓN
Director de Residencia	Tiene todos los privilegios administrativos del sistema. Puede realizar cualquier operación.
Jefe de Residencia	Puede realizar cualquier operación, pero sólo sobre la residencia que dirige.
Instructora	Puede Gestionar Reportes, Modificar Características de un Apartamento y Mostrar Plazas Libres por Edificio. Todo esto en la residencia a la que pertenece.
Usuario Simple	Usuario del dominio uci. Sólo puede buscar información.
Atomizador	Sistema encargado de la ubicación automática de los estudiantes y profesores que ingresan cada año a la universidad.

3.4.4 Casos de Uso del Sistema

Tabla 4: Caso de Uso 1 del Sistema

CU-1	Autenticar usuario
Actor(es)	Usuario(Todo aquel que pertenezca al dominio)
Resumen	Los usuarios introducen sus credenciales del dominio para que el sistema las verifique y estos puedan hacer todas las funcionalidades que puedan según su rol
Referencia	R1

Tabla 5: Caso de Uso 2 del Sistema

CU-2	Distribuir Manualmente
Actor(es)	Director de Residencia, Jefe de Residencia
Resumen	Dado un criterio de búsqueda el sistema muestra el o las personas asociadas al criterio de búsqueda introducido, el sistema le brinda la posibilidad de seleccionar el apartamento donde desea ubicar a la persona buscada y ubicarla
Referencia	R1, R2

Tabla 6: Caso de Uso 3 del Sistema

CU-3	Mostrar Listado de personas por edificio
Actor(es)	Director de Residencia, Jefes de Residencia o Instructora
Resumen	Se necesita saber la cantidad de personas que habitan un edificio, escogen el edificio y entonces el sistema devuelve una lista con la cantidad de personas que habitan el edificio con los datos de cada uno.
Referencia	R1, R3, R5

Tabla 7: Caso de Uso 4 del Sistema

CU-4	Mostrar Plazas Libres por Edificio
Actor(es)	Director de Residencia, Jefes de Residencia o Instructora
Resumen	Se desea ver las plazas libres por edificio. El sistema muestra solo la residencia a la que esta asociada el rol del usuario, entonces muestra las plazas libres por edificio de esa residencia.
Referencia	R1, R10

Tabla 8: Caso de Uso 5 del Sistema

CU-5	Mostrar Características de un Individuo
Actor(es)	Director de Residencia, Jefe de Residencia, Instructora o Usuario Simple
Resumen	Al seleccionar la opción buscar, el sistema brinda la opción de seleccionar la búsqueda por persona o por apartamento, después

	de seleccionar el tipo de búsqueda el sistema muestra las características de la persona o de las personas seleccionadas
Referencia	R1, R6, R7

Tabla 9: Caso de Uso 6 del Sistema

CU-6	Eliminar Individuo
Actor(es)	Director de Residencia o Jefe de Residencia
Resumen	Para Eliminar Individuo, el sistema brinda la opción de buscar el nombre del individuo a eliminar, después es borrado de la base de datos de residencia
Referencia	R1, R8, R9

Tabla 10: Caso de Uso 7 del Sistema

CU-7	Modificar Características de un Apartamento
Actor(es)	Director de Residencia, Jefe de Residencia o Instructora
Resumen	Al seleccionar la opción Modificar, el sistema carga todas las residencias de la universidad, asociado a ellas todas las manzanas de cada una, con todos los edificios de cada manzana. Se selecciona el apartamento que desea modificar y se modifica
Referencia	R1, R11

Tabla 11: Caso de Uso 8 del Sistema

CU-8	Adicionar Estructura
Actor(es)	Director de Residencia, Jefe de Residencia
Resumen	Para Adicionar Estructura (Residencia, Manzana, Edificio, Apartamento) depende de la opción que seleccione, el sistema le brindará las opciones pertinentes para adicionarla
Referencia	R1, R12, R13, R15, R17

Tabla 12: Caso de Uso 9 del Sistema

CU-9	Eliminar Estructura
Actor(es)	Director de Residencia o Jefe de Residencia
Resumen	Al escoger la opción Eliminar Estructura, el sistema brinda la opción de seleccionar la estructura a eliminar, se selecciona la estructura y el sistema la elimina de la base de datos Residencia
Referencia	R1, R18

Tabla 13: Caso de Uso 10 del Sistema

CU-10	Gestionar Permuta
Actor(es)	Director de Residencia o Jefe de Residencia
Resumen	Al seleccionar la opción Permutar, el sistema carga una serie de opciones como, seleccionar residencia, seleccionar edificio y seleccionar el apartamento de las personas que serán permutadas, después, marcar los que se permutarán y se efectúa la permuta
Referencia	R1, R14

Tabla 14: Caso de Uso 11 del Sistema

CU-11	Gestionar Reporte
Actor(es)	Director de Residencia, Jefe de Residencia o Instructora
Resumen	Al seleccionar la opción Reporte, el sistema carga las opciones asociadas a esta acción, y genera reportes del tipo (Por Residencia, Por Manzana, Por Edificio, Reporte General, Listado de Personas), este sistema de reportes permite al usuario salvar todos los datos en un documento.
Referencia	R1, R16

3.4.5 Diagrama de Casos de Uso del Sistema

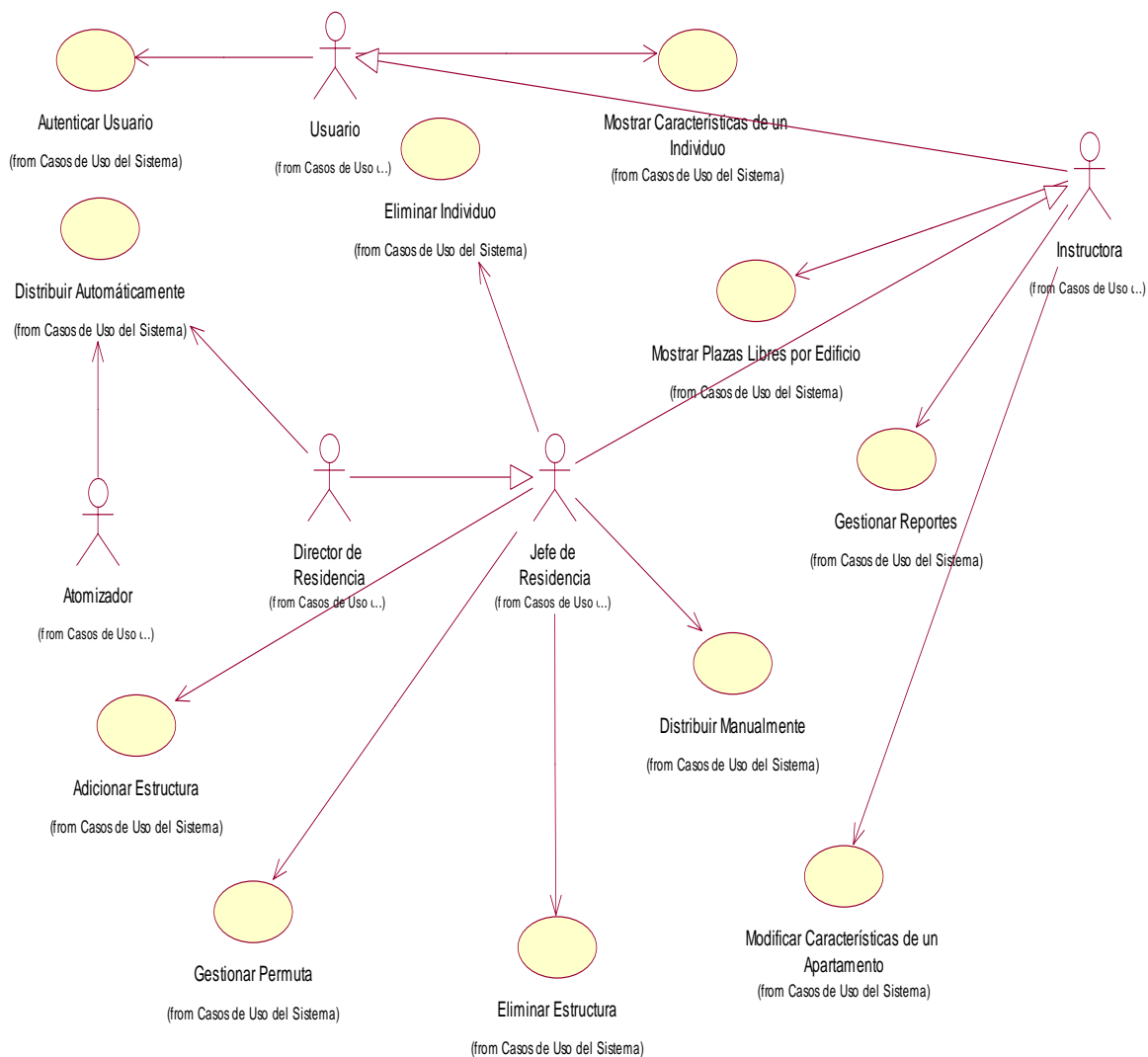


Fig. 7 Diagrama de Casos de Uso del Sistema

3.4.6 Descripción de los Casos de Uso del Sistema

Tabla 15: Expansión del Caso de Uso Autenticar Usuario

Caso de Uso	Autenticar Usuario	
Actor(es):	Usuario	
Propósito	Brindar una forma diferenciada de entrar al sistema, dependiendo del tipo de usuario, el sistema cargará sólo las opciones permitidas para este usuario	
Resumen	Un usuario del dominio se conecta al sistema e introduce su identificador de usuario y su clave. El sistema analiza los datos introducidos y en dependencia del rol del usuario, muestra las funcionalidades a las que tienen derecho.	
Referencia	R1	
Precondiciones	Están creados grupos para los diferentes roles que jugarán los usuarios.	
Poscondiciones	El usuario tiene acceso a las funcionalidades del sistema que le son permitidas.	
Flujo Normal de Eventos		
	Acción del Actor	Respuesta del Sistema
	1. El caso de uso se inicia cuando un usuario del dominio se conecta al sistema	2. El sistema muestra una ventana donde el usuario debe introducir, identificador de dominio y su clave.
	3. El usuario entra su identificador de dominio y su clave	4. El sistema valida los datos introducidos por el usuario.
		5. El sistema muestra sólo aquellas funcionalidades a las que tiene acceso el usuario

Tabla 16: Expansión del Caso de Uso Distribuir Manualmente

Caso de Uso	Distribuir Manualmente	
Actor(es):	Director de Residencia, Jefe de Residencia	
Propósito	Permitir la ubicación de aquellas personas que por una causa u otra no han sido ubicadas	
Resumen	Dado un criterio de búsqueda, el sistema muestra la o las personas asociadas al criterio de búsqueda introducido, el sistema le brinda la posibilidad al (Director de Residencia o Jefe de Residencia) de seleccionar el apartamento donde desea ubicar a la persona buscada y ubicarla	
Referencia	R1, R2	
Precondiciones	<ul style="list-style-type: none"> • Existe al menos un estudiante o un profesor sin ubicación • Existe al menos una capacidad vacía en la residencia 	
Poscondiciones	El estudiante (es) o profesor (es) queda (n) ubicado (s)	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El caso de uso se inicia cuando el Director de Residencia o Jefe de Residencia selecciona la opción distribuir manualmente	2. El sistema pide al Director de Residencia o Jefe de Residencia entrar un criterio de búsqueda, este puede ser, por nombre, por apellido, por alias, por número de carné de identidad, por número de solapín	
3. El Director de Residencia o Jefe de Residencia entra el criterio de búsqueda	4. El sistema muestra la o las personas asociadas al criterio introducido, mostrando (Foto, Nombre, Primer Apellido, Segundo Apellido)	
5. El Director de Residencia o Jefe de Residencia selecciona la persona a ubicar	6. El sistema muestra los apartamentos con capacidades disponibles para la ubicación	

7. El Director de Residencia o Jefe de Residencia le asigna el apartamento a la persona seleccionada	El sistema muestra un mensaje de confirmación de la acción ejecutada, actualizando la base de datos con la nueva ubicación

Tabla 17: Expansión del Caso de Uso Mostrar Listado de personas por edificio

Caso de Uso	Mostrar Listado de personas por edificio	
Actor(es):	Director de Residencia, Jefe de Residencia, Instructora	
Propósito	Brindar la posibilidad de ver las capacidades ocupadas por apartamentos en los edificios y los datos asociados a los ocupantes	
Resumen	Se necesita saber la cantidad de personas que habitan un edificio, escogen el edificio y entonces el sistema devuelve una lista con la cantidad de personas que habitan el edificio con los datos de cada uno.	
Referencia	R1, R3, R5	
Precondiciones		
Poscondiciones	Se muestra el listado de personas por edificios con sus datos.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El caso de uso se inicia cuando el Director de Residencia, Jefes de Residencia o Instructora selecciona la opción de mostrar listado de persona por edificio	2. El sistema carga los edificios que solo el usuario conectado puede ver, atendiendo a su rol asociado	
3. El Director de Residencia, Jefe de Residencia o Instructora selecciona el edificio para que sean mostrados sus integrantes con sus respectivos datos	4. El sistema muestra un listado de los habitantes por apartamentos con sus datos asociados (Apartamento, Primer Nombre, Segundo Nombre, Primer Apellido, Segundo Apellido,	

	Sexo, Tipo de Persona)

Tabla 18: Expansión del Caso de Uso Mostrar Plazas Libres por Edificio

Caso de Uso:	Mostrar Plazas Libres por Edificio	
Actor(es):	Director de Residencia, Jefe de Residencia, Instructora	
Propósito	Brindar información de todas las plazas libres que existen por edificio	
Resumen	Se desea ver las plazas libres por edificio. El sistema muestra solo la residencia a la que esta asociada el rol del usuario, entonces muestra las plazas libres por edificio de esa residencia	
Referencia	R1, R10	
Precondiciones		
Poscondiciones	Se muestran las plazas libres por edificio	
Flujo Normal de Eventos		
	Acción del Actor	Respuesta del Sistema
	1. El caso de uso se inicia cuando el Director de Residencia, Jefe de Residencia o Instructora selecciona la opción de mostrar plazas libres por edificio	2. El sistema muestra solo la residencia con los edificios correspondientes al Director de Residencia, Jefe de Residencia o Instructora
	3. El Director de Residencia, Jefe de Residencia o Instructora selecciona la estructura (Residencia, Manzana, Edificio) en la que desea ver las plazas libres.	3. El sistema muestra el o los edificios con las capacidades libres y que cantidad son por apartamentos.

Tabla 19: Expansión del Caso de Uso Mostrar Características de un Individuo

Caso de Uso:	Mostrar Características de un Individuo
Actor(es):	Director de Residencia, Jefe de Residencia, Instructora, Usuario Simple

Propósito	Brindar información general de la persona a buscar	
Resumen	Al seleccionar la opción buscar, el sistema brinda la opción de seleccionar la búsqueda por persona o por apartamento, después de seleccionar el tipo de búsqueda el sistema muestra las características de la persona o de las personas seleccionadas	
Referencia	R1, R6, R7	
Precondiciones		
Poscondiciones	Se muestran los datos de la persona seleccionada	
Flujo Normal de Eventos		
	Acción del Actor	Respuesta del Sistema
	1. El caso de uso se inicia cuando el Director de Residencia, Jefe de Residencia, Instructora o Usuario Simple selecciona la opción Buscar	2. El sistema pide al Director de Residencia, Jefe de Residencia, Instructora o Usuario Simple introducir un criterio de búsqueda, por persona o por apartamento
	3. El usuario introduce el tipo de búsqueda	4. El sistema muestra las características de lo seleccionado, si es persona, muestra los datos pertinentes del mismo (Apartamento, Primer Nombre, Segundo Nombre, Primer Apellido, Segundo Apellido, Sexo, Tipo de Persona), si es apartamento, muestra los integrantes de dicho apartamento con sus respectivos datos asociados.

Tabla 20: Expansión del Caso de Uso Eliminar Individuo

Caso de Uso:	Eliminar Individuo
Actor(es):	Director de Residencia, Jefe de Residencia
Propósito	Eliminar aquellas personas que por un motivo u otro

	causen baja de la universidad	
Resumen	Para Eliminar Individuo, el sistema brinda la opción de buscar el nombre del individuo a eliminar, después es borrado de la base de datos de residencia	
Referencia	R1, R8, R9	
Precondiciones	Existe el individuo a eliminar	
Poscondiciones	El individuo es eliminado de la BD de residencia	
Flujo Normal de Eventos		
	Acción del Actor	Respuesta del Sistema
	1. El caso de uso se inicia cuando el Director de Residencia o Jefe de Residencia selecciona la opción Eliminar Individuo	2. El sistema brinda la opción de buscar el individuo a eliminar, puede buscarlo por nombre, por número de carné de identidad, por solapín
	3. El usuario selecciona el individuo a eliminar	4. El sistema elimina de la base de datos Residencia, todo lo referente al individuo, como su Ubicación, Edificio, Número del Apartamento, Teléfono.

Tabla 21: Expansión del Caso de Uso Modificar Características de un Apartamento

Caso de Uso:	Modificar Características de un Apartamento
Actores:	Director de Residencia, Jefe de Residencia, Instructora
Propósito	Modificar los datos actuales de un apartamento
Resumen	Al seleccionar la opción Modificar, el sistema carga todas las residencias de la universidad, asociado a ellas todas las manzanas de cada una, con todos los edificios de cada manzana. Se selecciona el apartamento que desea modificar y se modifica
Referencia	R1, R11
Precondiciones	
Poscondiciones	Las características del apartamento seleccionado son

	modificadas
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El caso de uso se inicia cuando el Director de Residencia, Jefe de Residencia o Instructora selecciona la opción Modificar	2. El sistema muestra las residencias, las manzanas, los edificios con sus respectivos apartamentos.
3. El Director de Residencia, Jefe de Residencia, Instructora selecciona la residencia, manzana y edificio donde se encuentra el apartamento a modificar	4. El sistema muestra las características actuales del apartamento y la opción de modificarlas
5. El usuario modifica la información actual del apartamento	6. El sistema introduce en la base de datos los cambios realizados por el usuario conectado

Tabla 22: Expansión del Caso de Uso Adicionar Estructura

Caso de Uso:	Adicionar Estructura
Actores:	Director de Residencia, Jefe de Residencia
Propósito	Registrar en la base de datos de residencia las nuevas estructuras que se crean en nuestra universidad
Resumen	Para Adicionar Estructura (Residencia, Manzana, Edificio, Apartamento) depende de la opción que seleccione, el sistema le brindará las opciones pertinentes para adicionarla
Referencia	R1, R12, R13, R15, R17
Precondiciones	Que no exista la estructura a Adicionar
Poscondiciones	La nueva estructura es adicionada
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema

<p>1. En caso de uso se inicia cuando el Director de Residencia o Jefe de Residencia desea Adicionar una Estructura</p>	<p>2. El sistema brinda la opción al usuario de escoger el tipo de acción que quiere realizar.</p> <ul style="list-style-type: none"> a. Adicionar una Residencia b. Adicionar una Manzana c. Adicionar un Edificio d. Adicionar un Apartamento
<p>3. El Director de Residencia o Jefe de Residencia selecciona la acción a realizar. Si selecciona la acción a, ver Sección Adicionar Residencia. Si selecciona la acción b, ver Sección Adicionar Manzana. Si selecciona la acción c, ver Sección Adicionar Edificio. Si selecciona la acción d, ver Sección Adicionar Apartamento.</p>	
<p>Sección Adicionar Residencia</p>	
<p>4. El Director de Residencia o Jefe de Residencia selecciona la opción Adicionar una Residencia</p>	<p>5. El sistema brinda la opción al Director de Residencia o Jefe de Residencia de introducir el número de la Residencia que será adicionada</p>
<p>6. El Director de Residencia o Jefe de Residencia introduce el número de la Residencia</p>	<p>7. El sistema comprueba si el número que entró el usuario ya pertenece a una residencia ya creada, si ya existe , lanza un mensaje (Ya esta creada esa residencia), sino, adiciona la nueva residencia</p>
<p>Sección Adicionar Manzana</p>	
<p>4. El Director de Residencia o Jefe de Residencia selecciona la opción de</p>	<p>5. El sistema brinda la opción de especificar primero en que residencia</p>

Adicionar Manzana	creará la nueva manzana
6. El Director de Residencia o Jefe de Residencia selecciona la residencia	7. Después de seleccionada la residencia, el sistema brinda la opción de introducir el número de la manzana que será adicionada
8. El Director de Residencia o Jefe de Residencia introduce el número de la manzana que será adicionada	9. El sistema comprueba si el número que entró el usuario ya pertenece a una manzana ya creada, si ya está creada, lanza un mensaje de error, si no, se adiciona la manzana especificada
Sección Adicionar Edificio	
4. El Director de Residencia o Jefe de Residencia selecciona la opción de Adicionar Edificio	5. El sistema pide al Director de Residencia o Jefe de Residencia que especifique la residencia y la manzana donde se adicionará el nuevo edificio
6. Después de seleccionar la residencia y la manzana donde se adicionará el nuevo edificio, introduce el número del mismo	7. Si ya existe ese edificio, el sistema lanza un mensaje (Ya esta creado ese edificio), sino, es adicionado en la residencia y la manzana especificada, el nuevo edificio.
Sección Adicionar Apartamento	
4. El Director de Residencia o Jefe de Residencia selecciona la opción de Adicionar Apartamento	5. El sistema pide al usuario conectado que especifique la residencia y el edificio donde será adicionado el nuevo Apartamento
6. El Director de Residencia o Jefe de Residencia selecciona la residencia y el edificio donde desea adicionar el nuevo Apartamento	7. El sistema pide al Director de Residencia o Jefe de Residencia que introduzca el número del apartamento, capacidad, si es habitable, las observaciones del mismo, el teléfono y a que facultad

	pertenece. Además le brinda la opción de Auto generar.
8. El Director de Residencia o Jefe de Residencia introduce los datos pedidos por el sistema o selecciona la opción de Auto generar. Si selecciona la opción de auto generar ver Sección Auto generar.	9. Si el número del apartamento ya existe el sistema lanza una ventana con un mensaje de error, de lo contrario introduce en la base de datos la información que el Director de Residencia o Jefe de Residencia introduce.
Sección Auto generar	
<i>Esta opción está diseñada para edificios que tengan la misma cantidad de apartamentos por escaleras.</i>	
9. El Director de Residencia o Jefe de Residencia selecciona la opción de Auto generar	10. El sistema pide al Director de Residencia o Jefe de Residencia que introduzca el número de escaleras y cantidad de apartamentos por escaleras
11. El Director de Residencia o Jefe de Residencia introduce los datos pedidos por el sistema	12. El sistema auto genera los apartamentos.

Tabla 23: Expansión del Caso de Uso Eliminar Estructura

Caso de Uso:	Eliminar Estructura
Actor(es):	Director de Residencia, Jefe de Residencia
Propósito	Eliminar Estructuras que ya no sean de utilidad para la universidad
Resumen	Al escoger la opción Eliminar Estructura, el sistema brinda la opción de seleccionar la estructura a eliminar, se selecciona la estructura y el sistema la elimina de la base de datos Residencia
Referencia	R1, R18

Precondiciones	Existe la estructura a eliminar	
Poscondiciones	La estructura seleccionada es eliminada	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El caso de uso se inicia cuando el Director de Residencia o Jefe de Residencia selecciona la opción Eliminar Estructura	2. El sistema brinda la opción de seleccionar la estructura a eliminar	
3. El Director de Residencia Jefe de Residencia selecciona la estructura Residencia, Manzana, Edificio o Apartamento a eliminar	4. El sistema elimina la estructura seleccionada por el usuario y actualiza la base de datos	

Tabla 24: Expansión del Caso de Uso Gestionar Permuta

Caso de Uso:	Gestionar Permuta	
Actor(es):	Director de Residencia, Jefe de Residencia	
Propósito	Intercambiar personas que por distintos motivos necesiten de otra ubicación. Siempre y cuando sea aprobado por la máxima dirección en residencia.	
Resumen	Al seleccionar la opción Permutar, el sistema carga una serie de opciones como, seleccionar residencia, seleccionar edificio y seleccionar apartamento de las personas que serán permutadas, después marcar los que se permutarán y efectúa la permuta	
Referencia	R1, R14	
Precondiciones	Las personas a permutar deben ser del mismo sexo	
Poscondiciones		
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El caso de uso se inicia cuando el Director de Residencia o Jefe de Residencia selecciona la opción	2. El sistema carga opciones para especificar (Residencia, Edificio, Apartamento) donde se encuentra	

Permutar	ubicada actualmente la persona o las personas a permutar
3. El Director de Residencia o Jefe de Residencia selecciona (Residencia, Edificio, Apartamento) de la persona o de las personas que serán permutadas	4. El sistema muestra los integrantes del apartamento seleccionado con su foto y sus datos pertinentes (Apartamento, Primer Nombre, Segundo Nombre, Primer Apellido, Segundo Apellido, Sexo, Tipo de Persona)
5. El Director de Residencia o Jefe de Residencia selecciona (Residencia, Edificio, Apartamento) hacia donde se efectuará la permuta	6. El sistema muestra los integrantes del apartamento seleccionado con su foto y datos pertinentes
7. El Director de Residencia o Jefe de Residencia selecciona el o los integrantes de un apartamento que serán permutados por los seleccionados del otro apartamento	8. El sistema efectúa la permuta intercambiando los integrantes marcados, actualizando la base de datos

Tabla 25: Expansión del Caso de Uso Gestionar Reporte

Caso de Uso:	Gestionar Reporte
Actor(es):	Director de Residencia, Jefe de Residencia, Instructora
Propósito	Generar reportes para un mejor trabajo en Residencia
Resumen	Al seleccionar la opción Reporte, el sistema carga las opciones asociadas a esta acción, y genera reportes del tipo (Por Residencia, Por Manzana, Por Edificio, Reporte General, Listado de Personas), este sistema de reportes permite al usuario salvar todos los datos en un documento
Referencia	R1, R16
Precondiciones	
Poscondiciones	Se genera el reporte solicitado y se salva.

Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El caso de uso se inicia cuando usuario Director de Residencia, Jefe de Residencia o Instructora selecciona la opción Reporte	2. El sistema carga una serie de opciones a escoger por el Director de Residencia, Jefe de Residencia o Instructora, a Reporte por Residencia b Reporte por Manzana c Reporte por Edificio d Listado de Personas e Reporte General
3. El Director de Residencia, Jefe de Residencia o Instructora selecciona la residencia donde se hará el Reporte	
4. El Director de Residencia, Jefe de Residencia o Instructora selecciona el reporte a generar. Si selecciona el reporte a, ver Sección Reporte por Residencia. Si selecciona el reporte b, ver Sección Reporte por Manzana. Si selecciona el reporte c, ver Sección Reporte por Edificio. Si selecciona el reporte d, ver Sección Listado de Personas. Si selecciona el reporte e, ver Sección Reporte General.	5. El sistema brinda la posibilidad al Director de Residencia, Jefe de Residencia o Instructora de guardar los datos del Reporte
6. El Director de Residencia, Jefe de Residencia o Instructora puede seleccionar o no la opción de guardar	7. El sistema guarda el reporte en un documento si el Director de Residencia, Jefe de Residencia o

los datos del reporte.	Instructora lo desea.
Sección Reporte por Residencia	
4.1 El Director de Residencia, Jefe de Residencia o Instructora selecciona la opción Reporte por residencia.	4.2 El sistema muestra todas las residencias registradas y de cada una de ellas, la cantidad de personas de sexo femenino y masculino, capacidades libres y las capacidades ocupadas.
Sección Reporte por Manzana	
4.1. El Director de Residencia, Jefe de Residencia o Instructora selecciona la opción de Reporte por Manzana	4.2. El sistema muestra todas las manzanas de una residencia, y de cada manzana su número, la cantidad de personas de sexo femenino y masculino, capacidades libres y las capacidades ocupadas.
Sección Reporte por Edificio	
4.1. El Director de Residencia, Jefe de Residencia o Instructora selecciona la opción Reporte por Edificio	4.2 El sistema muestra todos los edificios de la residencia, y de cada edificio su número, la cantidad de personas de sexo femenino y masculino, capacidades libres y las capacidades ocupadas.
Sección Listado de Personas	
4.1. El Director de Residencia, Jefe de Residencia o Instructora selecciona la opción Listado de Personas	4.2 El sistema muestra una lista de personas por edificio de la residencia seleccionada, con su número de Apartamento, Primer Nombre, Segundo Nombre, Primer Apellido, Segundo Apellido, Sexo, Tipo de Persona
Sección Reporte General	
4.1. El Director de Residencia, Jefe de Residencia o Instructora selecciona la	4.2. El sistema muestra una lista con las manzanas de la residencia

opción Reporte General	seleccionada, la cantidad total de personas que tiene cada manzana, los edificios de la manzana con la cantidad total de personas que habitan el edificio, la cantidad de apartamentos por edificio con (Número de apartamento, Capacidad, Sexo, Teléfono, Si está Habitable)

Tabla 26: Expansión del Caso de Uso Distribuir Automáticamente

Caso de Uso	Distribuir Automáticamente	
Actores:	Director de Residencia, Atomizador	
Propósito	Ubicación de los estudiantes y profesores en las instalaciones de la Universidad	
Resumen	El caso de uso se inicia cuando el Director de Residencia selecciona la opción Distribución Automática. El sistema le brinda la posibilidad de conectarse con la aplicación creada para la distribución automática, que con solo ejecutarla, esta realiza la distribución automática de los estudiantes y profesores.	
Referencia	R1, R19	
Precondiciones	El sistema Atomizador tiene el listado de estudiantes y profesores a ubicar	
Poscondiciones	Quedan ubicados en sus apartamentos los estudiantes y profesores	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El caso de uso inicia cuando El Director de Residencia selecciona la opción de distribuir automáticamente el personal en residencia	2. El sistema carga la aplicación Atomizador que se encarga de la distribución automática	
3. El Director de Residencia ejecuta la		

aplicación Atomizador	
4. La aplicación Atomizador distribuye automáticamente las personas en residencia, en las capacidades disponibles.	

3.5 Conclusiones

En este capítulo se hizo una descripción de la propuesta de solución a través de la modelación del negocio: la identificación de los actores, trabajadores y los casos de uso correspondientes; el planteamiento de los requisitos funcionales y no funcionales de la aplicación que se va a desarrollar y la modelación de la misma en términos de casos de uso de sistema. A partir de este punto se puede comenzar a construir el sistema.

CONSTRUCCIÓN DE LA PROPUESTA DE SOLUCIÓN

4.1 Introducción

En este capítulo se realiza el diseño de la propuesta de solución, modelándose los artefactos que contribuyen al manejo de las complicaciones que pueden implicar la construcción de aplicaciones Web. Se plantean los estándares de codificación que se van a utilizar y finalmente, se realiza el modelo de despliegue y el modelo de implementación para una mejor descripción de la solución.

4.2 Diagramas de Clases

El Diagrama de Clase es el diagrama principal de diseño y análisis para un sistema. En él, la estructura de clases del sistema se especifica, con relaciones entre clases y estructuras de herencia. Durante el análisis del sistema, el diagrama se desarrolla buscando una solución ideal. Durante el diseño, se usa el mismo diagrama, y se modifica para satisfacer los detalles de las implementaciones. [10]

4.3 Diagrama de Clases del Diseño

Un Diagrama de Clases de Diseño muestra la especificación para las clases software de una aplicación. Incluye la siguiente información:

- Clases, asociaciones y atributos.
- Interfaces, con sus operaciones y constantes.
- Métodos.
- Navegabilidad.
- Dependencias.

A diferencia del Modelo Conceptual, un Diagrama de Clases de Diseño muestra definiciones de entidades software más que conceptos del mundo real (Referencia).

4.3.1 Diagrama de Clases Diseño Web: Distribuir Manualmente

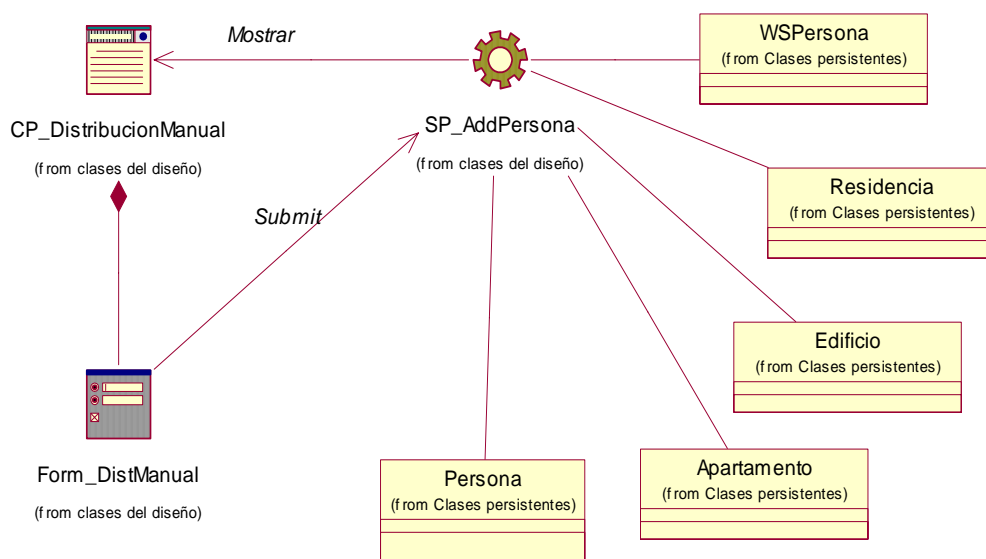


Fig. 8 Diagrama de Clases de Diseño (Distribuir Manualmente)

4.3.2 Diagrama de Clases Diseño Web: Gestionar Permuta

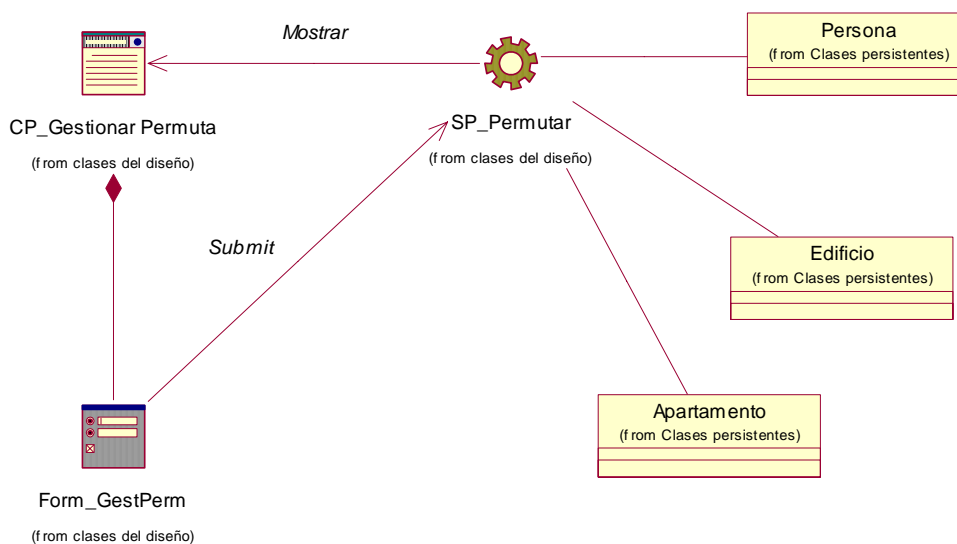


Fig.9 Diagrama de Clases de Diseño (Gestionar Permuta)

4.3.3 Diagrama de Clases Diseño Web: Eliminar Individuo

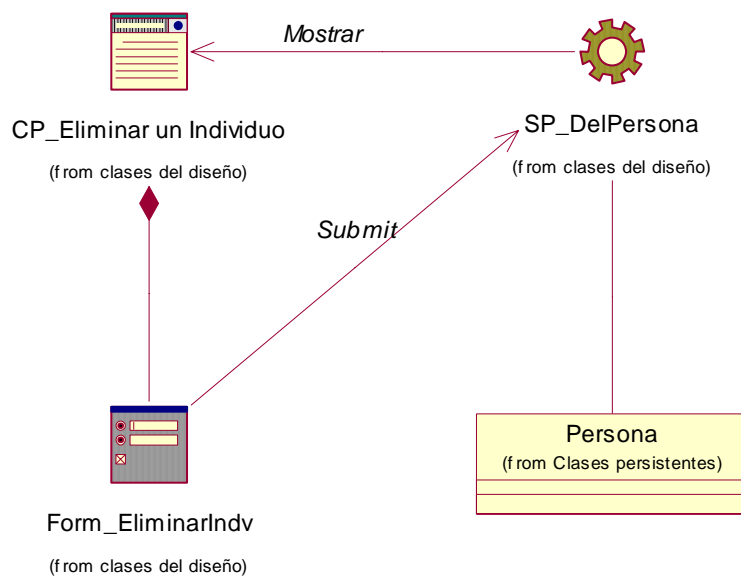


Fig.10 Diagrama de Clases de Diseño (Eliminar Individuo)

4.3.4 Diagrama de Clases Diseño Web: Mostrar Características de un Individuo

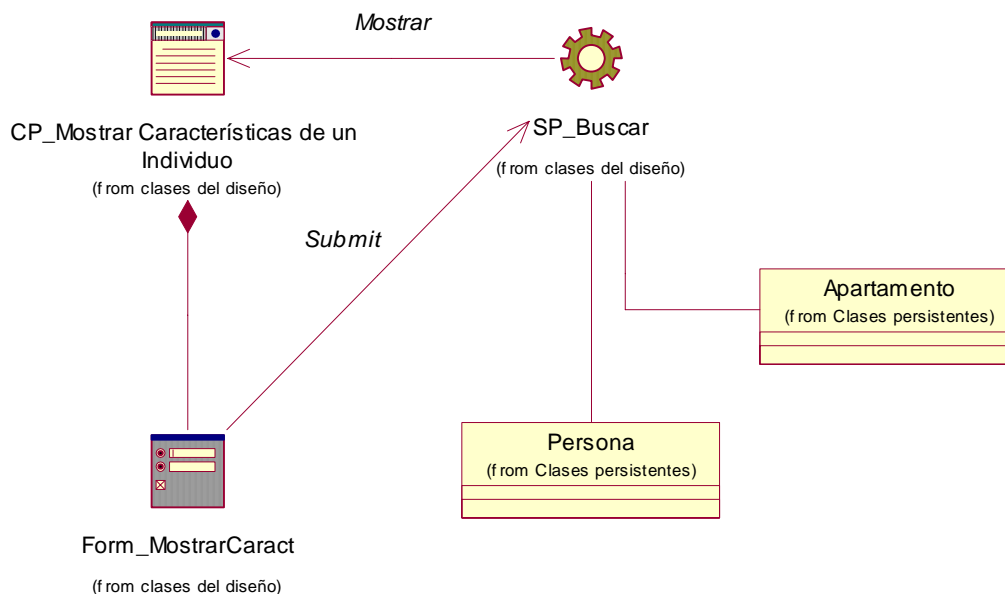


Fig.11 Diagrama de Clases de Diseño (Mostrar Características de un Individuo)

4.3.5 Diagrama de Clases Diseño Web: Modificar Características de un Apartamento

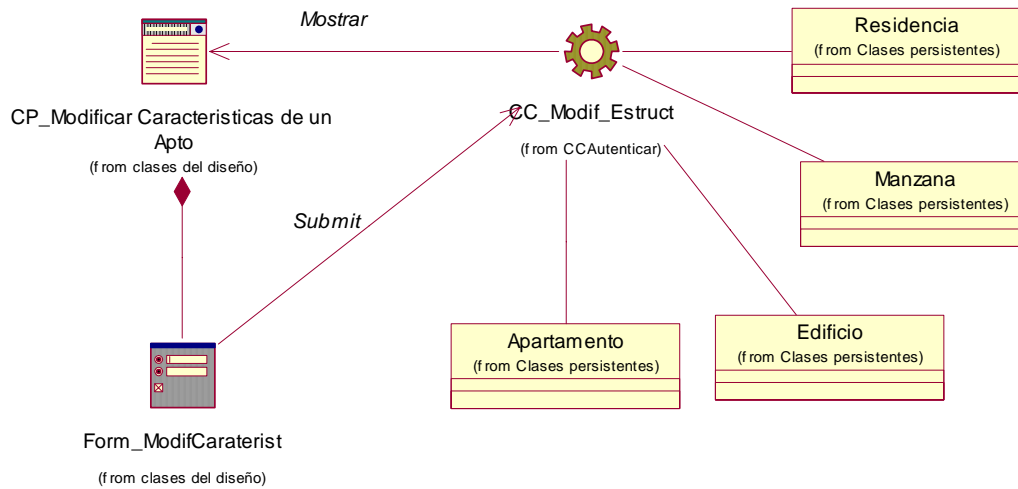


Fig.12 Diagrama de Clases de Diseño (Modificar Características de un Apartamento)

4.3.6 Diagrama de Clases Diseño Web: Adicionar Estructura (Residencia)

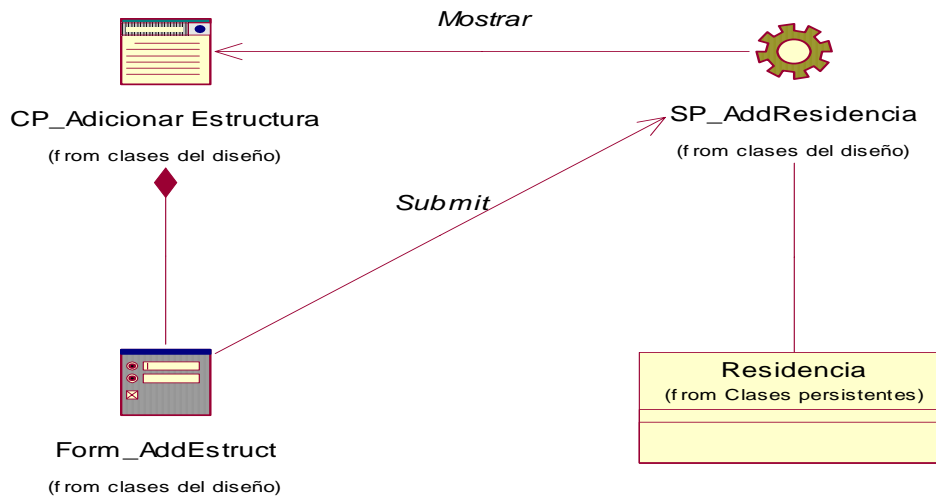


Fig.13 Diagrama de Clases de Diseño (Adicionar Residencia)

4.3.7 Diagrama de Clases Diseño Web: Adicionar Estructura (Manzana)

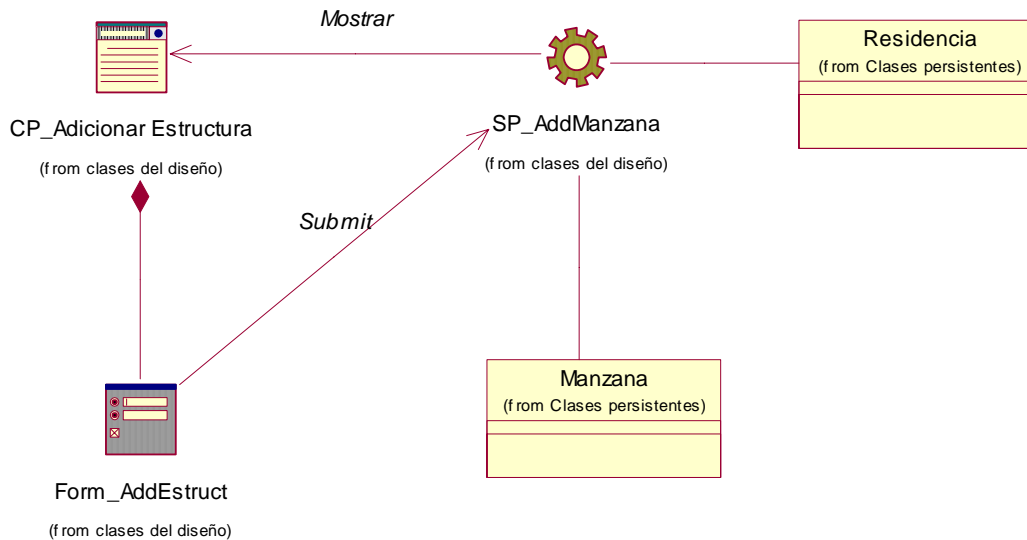


Fig.14 Diagrama de Clases de Diseño (Adicionar Manzana)

4.3.8 Diagrama de Clases Diseño Web: Adicionar Estructura (Edificio)

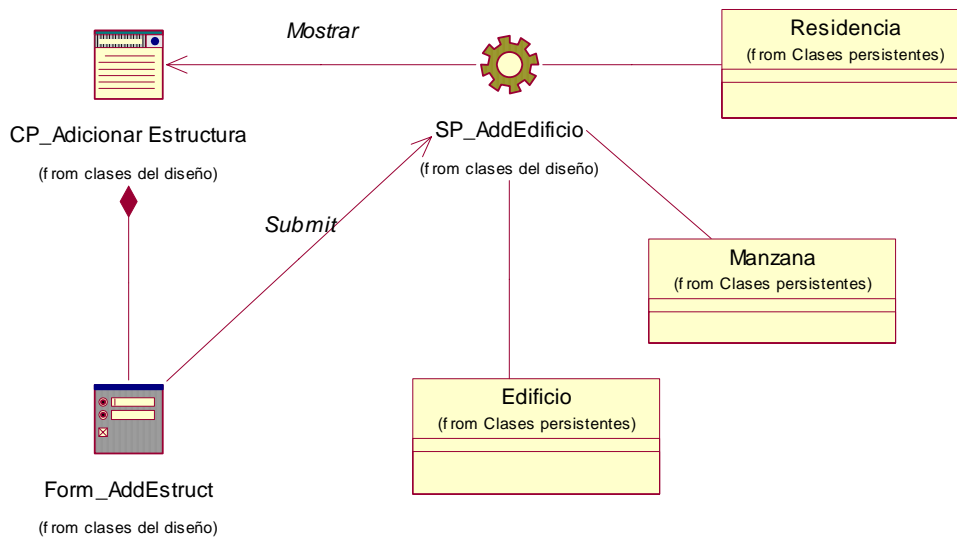


Fig.15 Diagrama de Clases de Diseño (Adicionar Edificio)

4.3.9 Diagrama de Clases Diseño Web: Adicionar Estructura (Apartamento)

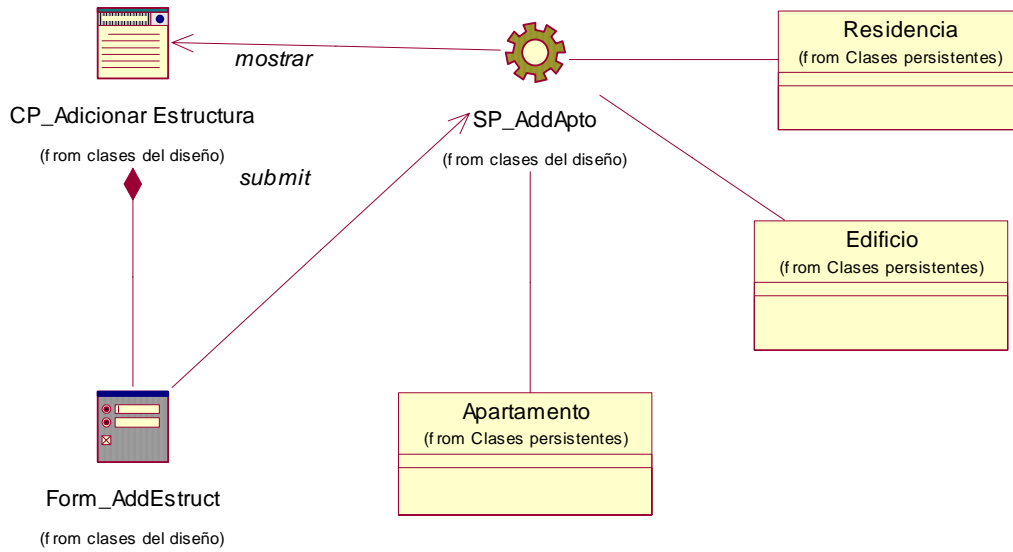


Fig.16 Diagrama de Clases de Diseño (Adicionar Apartamento)

4.3.10 Diagrama de Clases Diseño Web: Eliminar Estructura

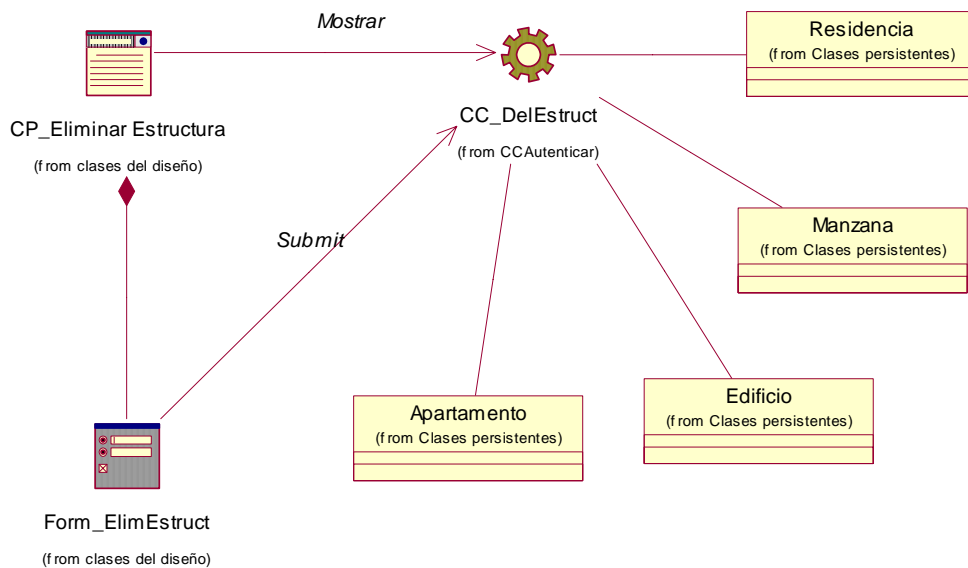


Fig.17 Diagrama de Clases de Diseño (Eliminar Estructura)

4.3.11 Diagrama de Clases Diseño Web: Gestionar Reporte

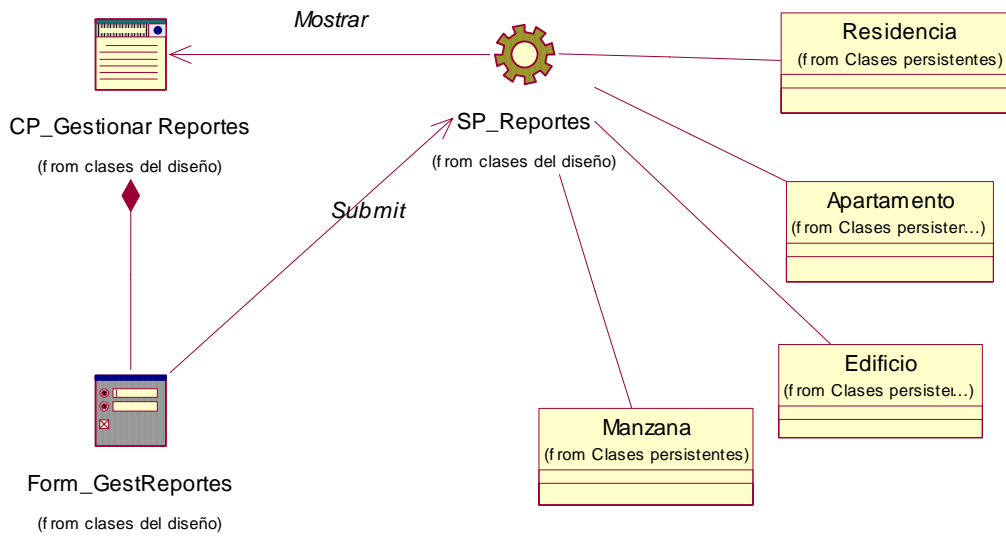


Fig.18 Diagrama de Clases de Diseño (Gestionar Reportes)

4.3.12 Diagrama de Clases Diseño Web: Mostrar Plazas Libres por Edificio

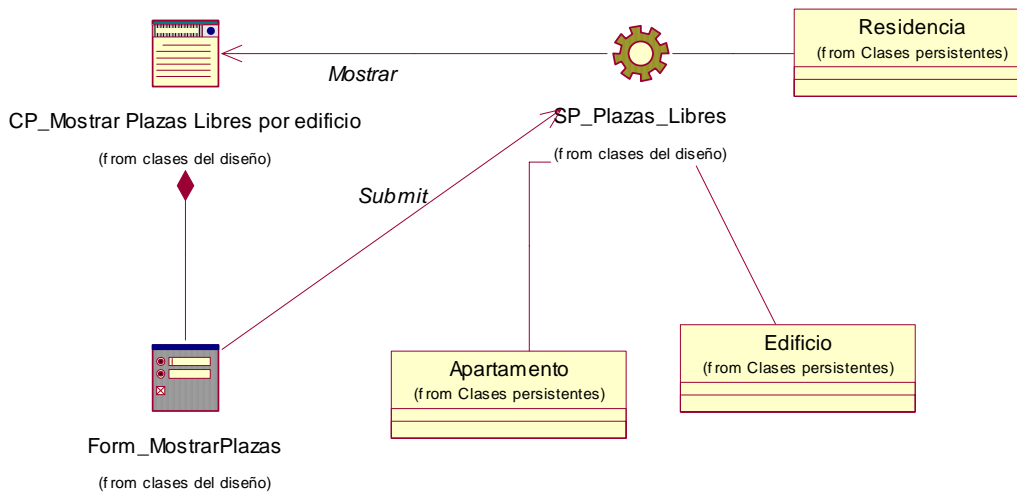
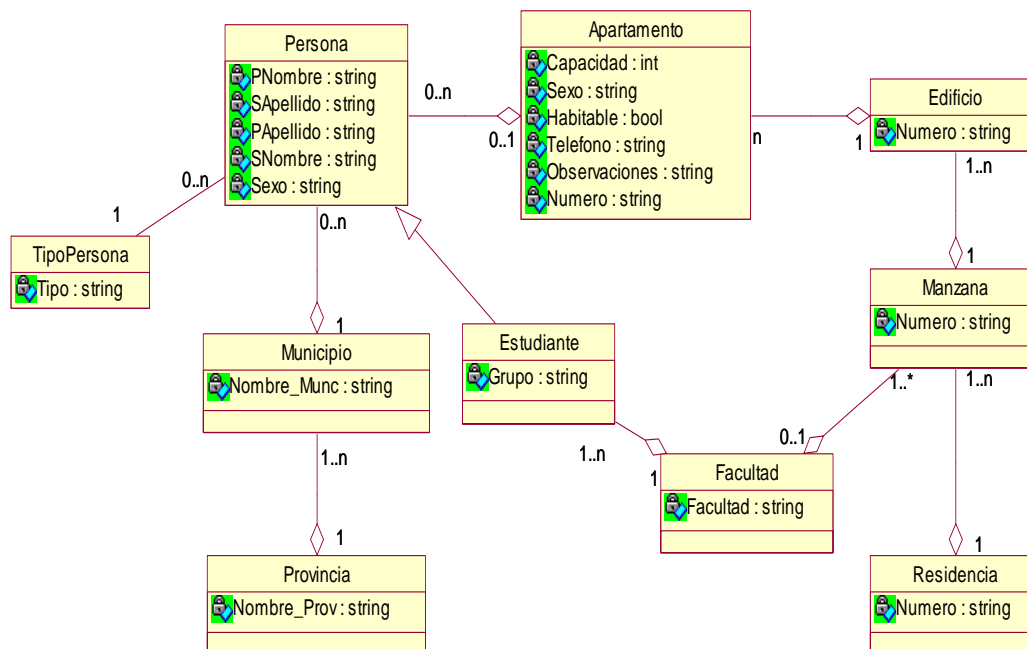


Fig.19 Diagrama de Clases de Diseño (Mostrar Plazas Libres por Edificio)

4.4 Diseño de la Base de Datos

4.4.1 Diagrama de Clases Persistentes



4.4.2 Modelo de Datos

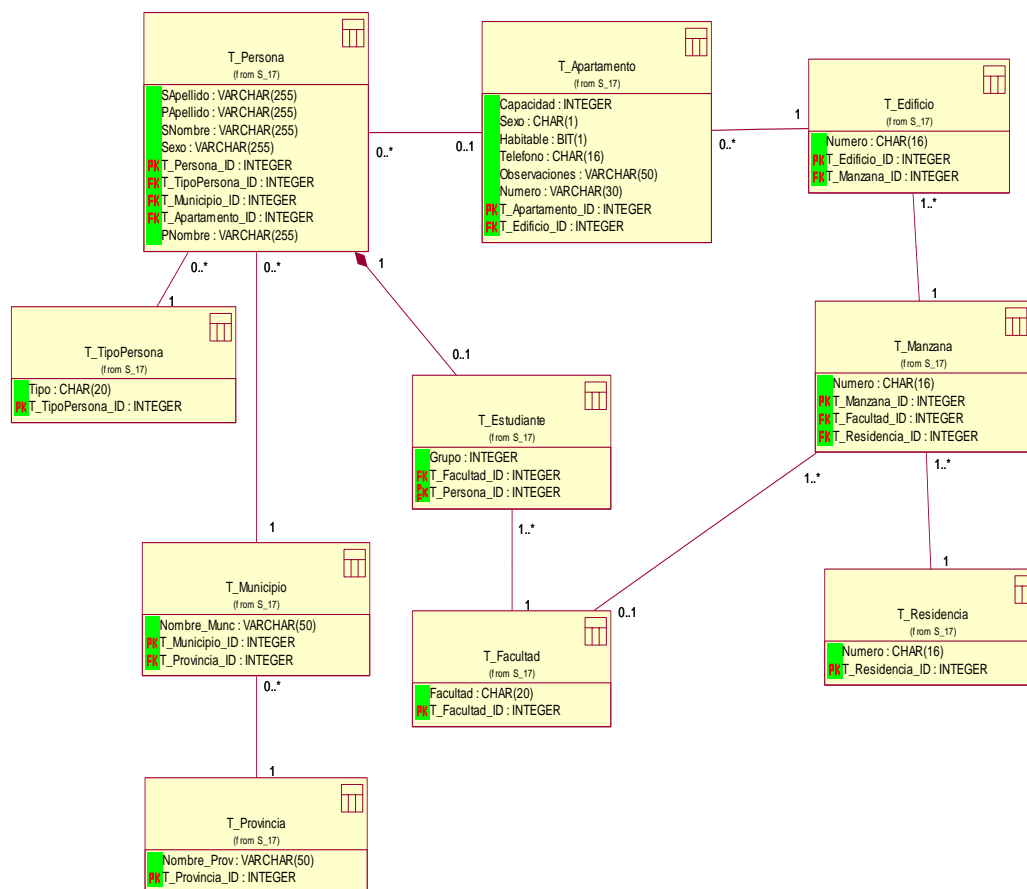


Fig.21 Modelo de datos

4.4.3 Descripción de las Tablas de la Base de Datos

Tabla. 27 Tabla de la Base de Datos, Apartamento

Nombre: Apartamento		
Descripción: Contiene todos los datos asociados a los apartamentos de la residencia.		
Atributo	Tipo	Descripción
IdApartamento	numeric	Contiene el id correspondiente al apartamento
IdEdificio	numeric	Contiene el id correspondiente al edificio
Capacidad	int	Contiene la capacidad total del apartamento
Sexo	char	Contiene el tipo de sexo de las personas del apartamento
Habitable	bit	Si está en (1) es habitable si está en (0) no es habitable
Teléfono	char	Contiene el número de teléfono del apartamento
Observaciones	varchar	Contiene información del estado del apartamento
Número	char	Contiene el número que corresponde a ese apartamento

Tabla. 28 Tabla de la Base de Datos, Persona

Nombre: Persona		
Descripción: Contiene datos necesarios de cada persona.		
Atributo	Tipo	Descripción
idPersona	numeric	Contiene el id correspondiente a cada persona
idTipoPersona	numeric	Contiene el id correspondiente al tipo de persona
IdProvincia	int	Contiene el id correspondiente a la provincia
idResidencia	numeric	Contiene el id correspondiente a cada residencia
PNombre	nvarchar	Contiene el primer nombre de la persona

SNombre	nvarchar	Contiene el segundo nombre de la persona
PApellido	nvarchar	Contiene el primer apellido de la persona
SApellido	nvarchar	Contiene el segundo apellido de la persona
Sexo	char	Contiene el sexo correspondiente a la persona

Tabla. 29 Tabla de la Base de Datos, Edificio

Nombre: Edificio		
Descripción: Contiene información de los edificios de la residencia.		
Atributo	Tipo	Descripción
idEdificio	numeric	Contiene el id correspondiente a cada edificio
idResidencia	numeric	Contiene el id correspondiente a cada residencia
Número	char	Contiene el número que tiene cada edificio

Tabla. 30 Tabla de la Base de Datos, Facultad

Nombre: Facultad		
Descripción: Contiene datos correspondiente a las facultades de la universidad.		
Atributo	Tipo	Descripción
idFacultad	int	Contiene el id correspondiente a la facultad
Número	char	Contiene el número correspondiente a la facultad

Tabla. 31 Tabla de la Base de Datos, Manzana

Nombre: Manzana		
Descripción: Contiene datos correspondientes a las manzanas de la residencia.		
Atributo	Tipo	Descripción
idManzana	numeric	Contiene el id correspondiente a cada

		manzana
idFacultad	char	Contiene el id correspondiente a la facultad
idResidencia	numeric	Contiene el id correspondiente a cada residencia
Número	char	Contiene el número de cada manzana

Tabla. 32 Tabla de la Base de Datos, Municipio

Nombre: Municipio		
Descripción: Contiene datos correspondientes a los municipios.		
Atributo	Tipo	Descripción
Id_Municipio	int	Contiene el id correspondiente a cada municipio
Id_Provincia	int	Contiene el id correspondiente a cada provincia
Nombre_Mun	varchar	Contiene el nombre correspondiente a cada municipio

Tabla. 33 Tabla de la Base de Datos, Provincia

Nombre: Provincia		
Descripción: Contiene datos correspondientes a una provincia		
Atributo	Tipo	Descripción
Id_Provincia	int	Contiene el id correspondiente a la provincia
Nombre_Prov	varchar	Contiene el nombre correspondiente a la provincia

Tabla. 34 Tabla de la Base de Datos, Residencia

Nombre: Residencia		
Descripción: Contiene datos correspondientes a las residencias de la universidad		
Atributo	Tipo	Descripción
idResidencia	numeric	Contiene el id correspondiente a cada residencia
Número	char	Contiene el número correspondiente a cada residencia

Tabla. 35 Tabla de la Base de Datos, TipoPersona

Nombre: TipoPersona		
Descripción: Contiene datos correspondientes al tipo de persona		
Atributo	Tipo	Descripción
idTipoPersona	numeric	Contiene el id correspondiente a el tipo de persona
Tipo	char	Contiene el nombre del los diferentes tipos de personas que existen

Tabla. 36 Tabla de la Base de Datos, Estudiante

Nombre: Estudiante		
Descripción: Contiene datos correspondiente al estudiante.		
Atributo	Tipo	Descripción
idPersona	nvarchar	Contiene el id correspondiente a cada persona
idFacultad	int	Contiene el id correspondiente a la facultad
Número	char	Contiene el número correspondiente a la facultad

4.5 Principio de Diseño de Interfaz

Se considera que una interfaz tiene éxito cuando la media de las personas puede interaccionar por sentido común, sobre algo que ha definido otra persona. [12]

El diseño de la interfaz es uno de los puntos fundamentales a tratar a la hora de la presentación de la aplicación, teniendo en cuenta que es lo que ve el usuario y por lo tanto, debe ser lo más amigable y comprensible posible. Una aplicación con una interfaz bien diseñada debe tener, además de un buen diseño gráfico, una buena navegabilidad, usabilidad y distribución de los contenidos.

Para el diseño de la interfaz de usuario de este sistema se han seguido los siguientes principios:

1. Mostrar al usuario solamente aquellas opciones a las cuales tiene derecho acceder, dado el rol que juegue en la aplicación.
2. Cualquier usuario podrá navegar sin ningún problema, sin tener en cuenta su nivel informático.

4.5.1 Estándares de la Interfaz de la Aplicación

El objetivo es construir un sistema fácil y seguro, estableciendo unos requisitos mínimos de fabricación, eliminando inconsistencias y variaciones innecesarias en la interfaz.

Para lograr un diseño consistente de la interfaz de la aplicación, se respetó en todas las páginas el esquema Cabecera-Navegador-Contenido. La cabecera contiene el nombre de la aplicación en la esquina superior izquierda. En el navegador, se incluyen los enlaces a las distintas secciones. En el contenido se muestran todas las funciones y pasos que debe seguir el usuario para realizar la acción deseada.



Fig. 22 Esquema de la Página.

4.6 Formato de los Reportes

El sistema brinda la opción al usuario de generar reportes, estos reportes contienen diferente información, depende solo de cómo el usuario lo desee y el sistema los genera. Dando la posibilidad de salvar dicha información.

Un ejemplo



Fig. 23 Vista de Reportes

4.7 Tratamiento de Excepciones

El tratamiento de errores posibilita el buen funcionamiento de una aplicación dándole una mejor apariencia ante los clientes. Para prevenir errores por parte del usuario, sólo se le brindan las opciones mínimas necesarias, a la hora de efectuar cualquier operación, por ejemplo, se deshabilitan determinados botones, si el usuario no tiene acceso a las páginas correspondientes.

Mediante la validación en el lado del cliente, se garantiza que los datos suministrados por los usuarios, se almacenen íntegros y no existan inconsistencias. Para esto, se verifican los campos obligatorios, y se revisa el tipo de datos, mostrándose en caso de algún error, mensajes de alerta.

Otro tipo de error que puede ocurrir son los que no pueden ser detectados en la parte del cliente, pues ocurren internamente en la aplicación. En este caso se muestra un mensaje en la página indicando el error detectado, dichos mensajes son bastante específicos y entendibles para los usuarios.

4.8 Estándares de Codificación

Un estándar de codificación completo, comprende todos los aspectos de la generación de código. Si bien los programadores deben implementar un estándar de forma prudente, éste debe tender siempre a lo práctico. Un código fuente completo debe reflejar un estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez. Al comenzar un proyecto de software, establezca un estándar de codificación para asegurarse de que todos los programadores del proyecto trabajen de forma coordinada. Cuando el proyecto de software incorpore código fuente previo, o bien cuando realice el mantenimiento de un sistema de software creado anteriormente, el estándar de codificación debería establecer cómo operar con la base de código existente. [12]

El estilo de código utilizado fue:

**/ para cualquier comentario*/*

Nombre de las clases comienzan con mayúscula, en caso de que sea compuesto por dos palabras, la primera letra de la segunda palabra también comienza con mayúscula

Atributos de las clases comienzan con mayúscula

Métodos de las clases comienzan con mayúscula

Indentación a 1 espacio

4.9 Modelo de Despliegue

El modelo de despliegue constituye la vista física de la aplicación. La Vista de Despliegue se construye para ilustrar la distribución física del sistema a lo largo de varios nodos de procesamiento, también conocida como Modelo de Despliegue. Esta vista incluye, además de la distribución del procesamiento, la distribución física de los procesos y los hilos.

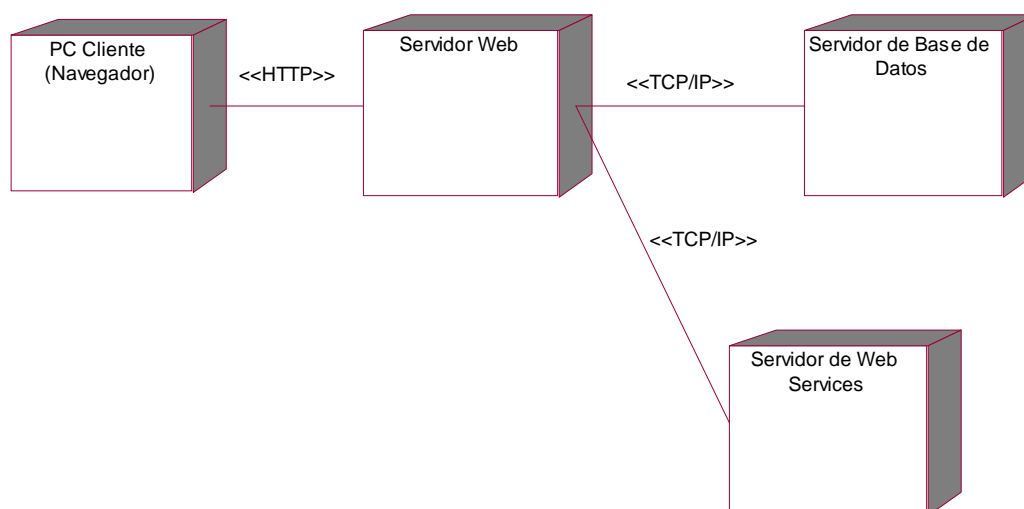


Fig.24 Modelo de Despliegue

4.10 Conclusiones

En este capítulo se ha llevado a cabo la descripción de las clases y demás elementos necesarios para la implementación. Se obtuvo el diagrama de clases web del sistema. Se definieron las clases persistentes y a partir de esto, se construyó el modelo de datos. Se expusieron las pautas seguidas para el diseño de la interfaz, y se explicó cómo está estructurada la aplicación físicamente, mediante el modelo de despliegue.

CONCLUSIONES GENERALES

Con el presente Trabajo de Diploma se provee a la UCI de un Sistema de Alojamiento automatizado para la ubicación del personal en la residencia, permitiendo un control efectivo que anteriormente se realizaba de forma manual y contribuyendo así a elevar el nivel de informatización de la universidad.

Con el Sistema para la Gestión de Alojamiento se garantiza el manejo de la información de la ubicación de estudiantes y profesores; se hace posible la gestión de las estructuras que conforman el área de alojamiento de la universidad; la realización de movimientos de personas en la residencia y se crea una interfaz Web para que todos los usuarios de la universidad puedan acceder a la misma. Se logró, además, agilizar y facilitar el trabajo a los compañeros que laboran en las direcciones de residencias de la universidad, que actualmente no cuentan con una herramienta que les brinde la posibilidad de gestionar las estructuras y personas alojadas en ellas de manera flexible.

RECOMENDACIONES

Una de las principales recomendaciones que se hacen, es migrar todo el sistema hacia software libre, ya que la Dirección de Informatización está inmersa en migrar todas sus aplicaciones hacia esta plataforma.

Seguir trabajando en perfeccionar la ayuda de la aplicación, ya que pueden existir muchos usuarios que no tengan mucho nivel en el uso de las tecnologías y se encuentren desorientados en determinados momentos.

Agregar el sistema de Gestión de Alojamiento como un módulo al Sistema General de la Residencia, que estará formado por, Gestión de Aseo, Gestión de Lavandería, Control de Inventarios, Hotel UCI.(Todo esto está en proceso de estudio e investigación)

Bibliografía

1. Jacobson, Ivar; Booch, Grady y Rumbaugh, James. *El Proceso Unificado de Desarrollo de Software*. Editorial Félix Varela, La Habana, 2004.
2. Larman, Craig. UML y Patrones. *Introducción al análisis y diseño orientado a objetos*. Editorial Félix Varela. La Habana, 2004.
3. Introduction to Web Forms Pages”, MSDN, Abril, 2003
4. Gestión, Significado. Disponible en <http://www.diccionarios.com>
5. Sistemas Automatizados, *Información y Reservas de Plazas*, Disponible en <http://www.colegiomayoralicante.com>
6. Sistemas Autorizados, *Sistema de la Universidad de Cantabria*, Disponible en <http://www.ceuc.unican.es/sie/bolsapisos.asp>
7. Sistemas Autorizados, Reserva de Hoteles en Cuba. Disponible en http://www.cuba.cu/reserva/hoteles/varadero/hoteles_varadero.php
8. Sistema Gestor de Base de Datos, Definición. Disponible en <http://es.wikipedia.org>
9. Arquitectura de Software, Definición. Disponible en http://www.microsoft.com/spanish/msdn/arquitectura/roadmap_arq/intro.asp
10. Diagrama de Clases, Definición. Disponible en <http://es.tldp.org/Tutoriales/doc-modelado-sistemas-UML/multiple-html/x219.html>
11. Arquitectura en Capas, Disponible en <http://es.wikipedia.org/wiki/Arquitectura>
12. Diseño web. Elementos de interfaz. Disponible en <http://www.agapea.com/Diseno-Web-Elementos-de-interfaz-n10363i.htm>
13. Estándares de Codificación, Disponible en <http://msdn.microsoft.com/library/spa/default.asp?url=/library/SPA/vsent7/html/vxconCodingStandardsCodeReviews.asp>
14. Teleclases de Ingeniería de Software, Disponible en. <http://internos.uci.cu/Teleclases>
15. Cómo programar en .NET, Disponible en <http://www.elguille.info/NET/default.aspx>

GLOSARIO DE TÉRMINOS

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

1. **Alojamiento:** Acción que realiza la Dirección de Residencia para otorgar a una persona dentro de la universidad la posibilidad de tener un lugar donde vivir
2. **Dirección de Informatización:** Es el departamento encargado de automatizar todos los procesos dentro de la universidad.
3. **Estructura:** Se denominan estructuras al grupo de (Residencia, Manzana, Edificio, Apartamento)
4. **Manzana:** Área, pero mucho más pequeña que la residencia, donde podemos encontrar, uno o varios edificios. Es válido destacar que la residencia está formada por muchas manzanas.
5. **Residencia:** Áreas en las que se encuentran varios edificios de la universidad, destinados al Alojamiento de personas
6. **Permuta:** Acción que se realiza a través de personas y apartamentos, aplicado a la universidad, una o varias personas pueden dejar de vivir en un apartamento y comenzar a convivir en otro.