

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Facultad 3



**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN
CIENCIAS INFORMÁTICAS**

**Título: Informatización de la Residencia Universitaria. Rol Analista del
Módulo Gestión de Alojamiento.**

**Autores: Aneyvis Hernández China.
Yuneisi Díaz Rodríguez.**

Tutor: Dr. Pascual Verdecia Vicet.

La Habana, junio 2007

*La ciencia es el alma de la prosperidad de las naciones y la
fuente de vida de todo progreso.*

Louis Pasteur

DECLARACIÓN DE AUTORÍA

Por este medio declaramos que Yuneisi Diaz Rodriguez y Aneyvis Hernández Chinae somos las únicas autoras de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas (UCI) para que haga el uso que estime pertinente con este trabajo.

Para que así conste firmamos la presente a los ____ días del mes de Junio del 2007.

Firma del Autor

Firma del Autor

Firma del Tutor

A mami y papi....

A mis abuelos.....

De Ane

A mis padres....

De Yuni

Agradecimientos

Queremos agradecer a nuestra Revolución y a nuestro comandante por permitirnos estudiar en esta Universidad, y formarnos como profesionales. A nuestro tutor que nos acogió sin peros, por dedicarnos tanto tiempo, por su paciencia y su dedicación. (Bartolo, ja ja).

Al personal de Residencia que estuvo frente al proyecto, en especial a Wualdo por todo el tiempo que nos dedicó, a Janny por su apoyo y la información brindada.

De Aneyvis.

A mis padres, Estrella y Osbel, por ser tan especiales, confiar en mí, apoyarme siempre y estar a mi lado en todo momento. Los quiero mucho.

A mi abuela María por ser mi otra mamá, por su paciencia y por quererme tanto.

A mi abuelo Pedro, por todo su cariño, por estar ahí fuerte siempre.

A mis abuelos que no están hoy pero me dieron todo su amor cuando los tuve cerca.

A mi hermano por ser tan bueno y comprensible conmigo, por quererme tanto.

A toda mi familia por educarme y apoyarme siempre.

A todas mis amistades por permitirme compartir tantos momentos maravillosos estos 5 años.

A Osmany por su apoyo permanente, por su amor, su comprensión, que tanto me ayudaron a levantar el ánimo.

A Julio por ser mi amigo de siempre, por estar en los momentos difíciles para resolverlos y darme su hombro como apoyo.

A la familia de mi novio por todo su apoyo y cariño.

No pueden faltar estos nombres que tanto quiero y nunca voy a olvidar, Yela Yari, Dailin, Yuni, gracias por esa amistad de verdad.

A todos los que de una forma u otra han contribuido a mí formación como persona y profesional.

De Yuneisi

A mi mamita y mi papito por apoyarme, comprenderme y guiarme siempre, por amarme y dedicarme sus vidas. Gracias por ser especiales.

A mi abuelita Pucha por estar siempre conmigo, educarme y quererme mucho.

A toda mi familia por ser única, porque siempre están ahí cuando los necesito.

A mi tío Felo y familia por acogerme estos cinco años en su casa como una hija más.

A todas las amistades que en estos cinco años me han permitido compartir momentos inolvidables, en especial a mis amigas Ane, Yary, Yela, Daylin, gracias por estar siempre conmigo en las buenas y malas, siempre las voy a querer.

A Didier por todo el apoyo, cariño y la alegría que me brinda.

A Julio por ser un gran amigo, por escucharme y guiarme en momentos difíciles, por aguantar mis malacrianzas con el inglés, por quererme y respetarme siempre.

A mis amigas de infancia por apoyarme siempre y confiar en mí, en especial a mi hermanita Marileysi.

A Osmany por ser un buen profesor, por ser mi amigo, por ayudarnos con la tesis, por ser mucho Osmany.

A todos los que de una forma u otra han contribuido a mi formación como persona y como profesional.

Resumen

La Universidad de Las Ciencias Informáticas está desarrollando una serie de proyectos productivos, dentro de estos se encuentran los encargados de automatizar los principales procesos que se desarrollan dentro de la misma, uno de estos proyectos es, Informatización de la Residencia, encargado de hacer un sistema que garantice la automatización de todo el trabajo que en ella se realiza. Uno de los módulos por los que está conformado este proyecto es, Gestión de Alojamiento, encargado de controlar principalmente el registro de la ubicación o distribución del personal interno en las instalaciones existentes en la universidad, garantizando mayor eficiencia.

Para desarrollar este módulo se hace necesario un entendimiento entre clientes y desarrolladores, este trabajo está basado en el desempeño del rol de analista, el cual es el encargado de lograr dicho entendimiento. El mismo tiene entre sus objetivos, realizar una correcta captura de requisitos y generar los artefactos correspondientes que garanticen la futura implementación del sistema. Se utilizó como metodología de desarrollo RUP, UML como lenguaje de modelación y la herramienta Rational Rose Enterprise Edition. Los resultados obtenidos de la captura de requisitos son avalados por los clientes, con total satisfacción y en algunos artefactos con métricas de calidad.

Índice

INTRODUCCIÓN.....	1
CAPÍTULO #1: FUNDAMENTACIÓN TEÓRICA.....	6
1.1 INTRODUCCIÓN	6
1.2 FUNDAMENTACIÓN DEL TEMA.....	6
1.2.1 Gestión.....	6
1.2.2 Gestión de Alojamiento.....	6
1.2.3 Sistemas de Gestión de Alojamiento:.....	6
1.2.4 Sistemas automatizados existentes.....	7
1.3 TENDENCIAS Y TECNOLOGÍAS ACTUALES.....	10
1.3.1 Metodologías más importantes y usadas en el mundo.....	10
1.3.2 Metodología escogida para desarrollar el trabajo (RUP).....	17
1.3.3 Lenguaje de Modelado a utilizar.....	18
1.3.4 Herramientas.....	19
1.4 PATRONES DE CASOS DE USO.....	21
1.5 ROL ANALISTA	24
1.6 INGENIERÍA DE REQUISITOS.....	25
1.6.1 Características de los requisitos.....	26
1.6.2 Pasos a seguir en la ingeniería de requisitos:.....	27
1.6.3 Técnicas utilizadas en la ingeniería de requisitos.....	28
CONCLUSIONES.....	30
CAPÍTULO # 2 MODELO DEL NEGOCIO Y REQUISITOS.....	31
2.1 INTRODUCCIÓN	31
2.2 ESTADO ACTUAL DEL NEGOCIO.....	31
2.3 REGLAS DEL NEGOCIO.....	32
2.4 ACTORES DEL NEGOCIO.....	32
2.5 TRABAJADORES DEL NEGOCIO	33
2.6 DIAGRAMA DE CASOS DE USO DEL NEGOCIOS (CUN)	34
2.7 DESCRIPCIÓN CASOS DE USO DEL NEGOCIO (CUN).....	34
2.8 DIAGRAMAS DE ACTIVIDADES.....	39
2.9 MODELO DE OBJETO.....	41
2.10 REQUISITOS CAPTURADOS.....	41
2.10.1 Requisitos Funcionales	42
2.10.2 Requisitos no Funcionales.....	45
CONCLUSIONES:.....	46
CAPÍTULO # 3 PROPUESTA DE SOLUCIÓN Y ANÁLISIS DE LOS RESULTADOS.....	47
3.1 INTRODUCCIÓN	47
3.2 DESCRIPCIÓN DEL SISTEMA	47
3.3 ACTORES DEL SISTEMA.....	48

3.4 PATRONES DE CASOS DE USO APLICADOS EN ESTE TRABAJO.....	49
3.5 CASOS DE USOS DEL SISTEMA.....	50
3.6 DIAGRAMA DE CASOS DE USO DEL SISTEMA (DCUS)	51
3.7 ESPECIFICACIÓN DE LOS CASOS DE USO.	52
Autenticar Usuario.....	52
Gestionar roles _ usuarios.....	53
CRUD Estructura	56
CRUD Apto	61
CRUD Criterios de ubicación para trabajadores.....	64
Gestionar alojamiento manual.....	67
Gestionar alojamiento automático	72
Baja automática de estudiantes	78
Gestionar baja manual de trabajadores.....	80
Registrar ubicaciones en un historial.....	82
Visualizar Reportes	83
Buscar.....	92
3.8 ANÁLISIS DE LOS RESULTADOS	95
CONCLUSIONES	101
CONCLUSIONES GENERALES	102
RECOMENDACIONES.....	103
REFERENCIAS BIBLIOGRÁFICAS.....	104
ANEXOS	106
GLOSARIO DE TÉRMINOS	112

Introducción.

La tecnología avanza a pasos inmensos cada día logrando tan alto grado de desarrollo que hoy está en todas partes y, si no en todas, en la mayoría de las actividades del ser humano: en la educación, la industria, el comercio, las finanzas, la investigación, etc. La informática avanza con un acelerado ritmo de desarrollo, por lo que hoy, conocer la tecnología y utilizarla ya no constituye ningún privilegio, por el contrario, es una necesidad. El uso de la tecnología es un factor determinante en los niveles de eficiencia y competitividad tanto a nivel empresarial como personal.

Cuba ha estado enmarcada en estos últimos años en la estrategia de lograr la informatización de la sociedad, proceso de utilización ordenada y masiva de las tecnologías de la Información y las Comunicaciones, satisfaciendo las necesidades en las esferas de la sociedad, para lograr más eficiencia y eficacia en todos los procesos, aumentando de este modo la competitividad y creando una cultura digital en la población, como característica imprescindible en la formación del hombre nuevo, y de esta forma acercar más la sociedad hacia el objetivo de lograr un desarrollo sostenible.

Las aplicaciones informáticas en Cuba han estado dirigidas en primer término al sector productivo, la salud, actividad pública y los sistemas de educación nacional. La Industria cubana del software está llamada a convertirse en una significativa fuente de ingresos para el país, como resultado del correcto aprovechamiento de las ventajas del alto capital humano disponible. Se hace un gran esfuerzo para potenciar esta industria, la creación de La Universidad de las Ciencias Informática (UCI) ha entrado a jugar un papel importante en el desarrollo de la misma, y en la materialización de los proyectos asociados al programa cubano de informatización. Siendo partícipe de muchos eventos relacionados con la informatización e investigación, y en algunos de los casos la protagonista, como por ejemplo UCIENCIA, donde participan personas de todo el país.

Esto es posible gracias a que La Universidad esté concebida como un centro de docencia, producción e investigación, lo que ha permitido desarrollar un conjunto de proyectos, mediante los cuales, se están automatizando los principales procesos que en ella se desarrollan. Aquí están involucradas la mayoría de las áreas de la institución, incluyendo la residencia, la cual abarca gran extensión, existe una determinada cantidad de edificios de residencia, pues hoy la universidad cuenta con más de 11 mil personas internas incluyendo trabajadores y estudiantes, esta cifra hace que se necesite un gran volumen de trabajo para el control de todo lo relacionado con esta área.

La dirección de la universidad ha establecido la ubicación de sus estudiantes en la beca por direcciones de residencia, cada una de las cuales abarca de 3 a 4 facultades situadas geográficamente por manzanas, algunas compartidas, con un número determinado de edificios, exceptuando la dirección de atención a profesores, en la cual deben estar ubicados todos los profesores, cuadros y trabajadores internos del centro.

Con frecuencia, y como parte de la ejecución del proceso inversionista, se edifican nuevos inmuebles que originan la ampliación gradual de las capacidades de alojamiento y con ello cambios en la zonificación y ubicación de los residentes.

La distribución de todas las personas en las diferentes instalaciones que están destinadas al alojamiento se ha hecho de diferentes formas en casi todos los cursos, pero por lo general siempre se realiza de forma manual por los vicedecanos de residencia, los técnicos generales o psicopedagogas de las respectivas facultades, con los listados de los grupos en base a la habitabilidad de los apartamentos, la zonificación aprobada y los requerimientos establecidos de antemano.

Actualmente las mudanzas masivas, traslados entre apartamentos, la clasificación de los inmuebles en habitables o no, entre otras muchas operaciones, se realiza de forma manual y los datos son archivados en papel o en sistemas no profesionales por diversas personas, resultando sumamente engorrosa la gestión.

Todo lo relacionado con la residencia está en igualdad de condiciones, por lo que la dirección de informatización propuso la elaboración de un sistema que automatizara todo el proceso que se lleva a cabo en la residencia, en aras de mejorar el proceso de organización y satisfacer las necesidades de los trabajadores de la misma. Este consta con 6 módulos: Medios básicos, Personal, Aseo y lavandería, Trabajo educativo, Reporte de indisciplinas y Alojamiento, todos distribuidos en un grupo de proyecto de la facultad #3. Cada módulo tienen asociado un equipo de trabajo conformado por roles.

Uno de los roles más importantes dentro del proceso, es el rol de analista el cual se encarga de lograr el entendimiento entre los clientes y desarrolladores, tiene que analizar el problema y describirlo con el propósito de que se le de solución mediante un sistema informático. En este trabajo se desarrolla el rol de analista en el módulo Gestión de Alojamiento, partiendo de la no existencia de un acuerdo común entre los clientes y desarrolladores en lo que el software debe hacer, la no delimitación de las fronteras del negocio, la falta de un mejor entendimiento de los requisitos que debe cumplir el mismo por parte de los

desarrolladores, existe la necesidad de definir los prototipos de interfaces no funcionales enfocadas a necesidades y objetivos del usuario.

Después de hacer un análisis de la problemática existente se plantea como:

Problema científico:

Se dificulta el establecimiento y mantenimiento de un acuerdo entre clientes y equipo de desarrollo respecto a los requisitos que debe satisfacer el sistema que automatizará el proceso de alojamiento en la residencia UCI.

Objeto de estudio:

Ingeniería de Requisitos

Objetivo general:

Obtener los artefactos correspondientes al rol de analista, a partir de una correcta captura de requisitos, que garantice un entendimiento común entre clientes y desarrolladores.

Campo de acción:

Gestión y desarrollo de requisitos para el módulo Gestión de Alojamiento.

Hipótesis:

Si se realiza una correcta captura de requisitos y se generan los correspondientes artefactos, entonces se logrará un entendimiento entre clientes y desarrolladores que facilite el desarrollo del Módulo de Gestión de Alojamiento.

Tareas específicas:

- Analizar sistemas que permitan informatizar procesos similares en el mundo y en Cuba.
- Estudiar referencias bibliográficas relacionadas con el tema a desarrollar.
- Seleccionar la metodología, herramienta y lenguaje de modelado a utilizar.
- Definir los artefactos que se obtendrán como resultado de la metodología empleada.
- Identificar actores del negocio.
- Identificar trabajadores del negocio.
- Obtener Casos de Uso del negocio (CUN), y su respectivo diagrama.
- Obtener Diagramas de Actividades de los CUN.
- Obtener Modelo de Objeto.
- Obtener Reglas del Negocio.
- Aplicar técnicas para la captura de requisitos.

- Obtener Requisitos funcionales y no funcionales.
- Obtener Casos de Uso del Sistema.
- Aplicar patrones de casos de uso.
- Realizar la descripción textual de los casos de uso del sistema.
- Diseñar prototipo de interfaz no funcional.
- Aplicar métricas para medir la calidad de los resultados obtenidos.

Metodologías de la investigación:

De las metodologías de investigación existentes, en este trabajo se hace referencia a dos de ellas, las cuales fueron utilizadas en la investigación realizada.

Métodos teóricos:

- Método analítico – sintético.
- Histórico-lógico
- Modelación

Métodos empíricos.

- Entrevista.

Dentro de los métodos teóricos se utilizó:

El Método analítico – sintético con el objetivo de buscar la esencia de los fenómenos, los rasgos que lo caracterizan y los distinguen, analizar teorías y documentos, permitiendo la extracción de los elementos más importantes que se relacionan con el objeto de estudio.

El Histórico-lógico que permitió estudiar de forma analítica la trayectoria histórica real de los fenómenos, su evolución y desarrollo, con el objetivo de constatar teóricamente como han evolucionado fenómenos en períodos de tiempo, en toda su trayectoria o en un fragmento temporal de la lógica de su desarrollo.

Y uno de los más utilizados y más importantes es el de Modelación, pues este permite la creación de modelos, propuestas, alternativas y estrategias. El modelo es una reproducción simplificada de la realidad, cumple una función heurística, que permite descubrir y estudiar nuevas relaciones y cualidades del objeto de estudio. La modelación es el proceso mediante el cual creamos modelos con vistas a investigar la realidad, presenta determinada correspondencia con el objeto del conocimiento y en algunas etapas el

modelo está en condiciones de sustituir el objeto que se estudia. La modelación de un proceso permite predecir la respuesta de dicho proceso a variaciones de algunos de sus parámetros, sin tener que ejecutar el proceso en la realidad. (CICE 2006)

Dentro de los métodos empíricos el más utilizado es el de las Entrevistas pues se realizaron una serie de conversaciones planificadas con los clientes para obtener información necesaria y valiosa en el desarrollo de la investigación.

Este trabajo se encuentra estructurado de la siguiente forma:

Capítulo 1: En este capítulo se explica que son los sistemas de gestión de alojamiento, se hace un estudio de las metodologías, herramientas, lenguaje de modelado, patrones de casos de uso y técnicas para la captura de requisitos.

Capítulo 2: Se describe el negocio a modelar, explicando sus reglas, se exponen los diagramas de actividades de cada caso de uso, los requisitos funcionales y no funcionales, explicando las técnicas empleadas para la captura de los mismo.

Capítulo 3: Describe la propuesta de solución, mediante el diagrama de los casos de uso del sistema, se hace la descripción textual de los casos de uso, se explica la justificación de los patrones de casos de uso empleados, y se hace un análisis de los resultados obtenidos.

Capítulo #1: Fundamentación Teórica

1.1 Introducción

En el presente capítulo se explica el tema a desarrollar, así como las tendencias y tecnologías actuales a considerar en el trabajo, se hace un estudio de las metodologías de desarrollo más usadas, se argumenta el por qué de la escogida para este trabajo, se explica el lenguaje de modelado a emplear y las herramientas que se utilizan, así como la descripción de los patrones de casos de usos existentes, se exponen las funciones que desempeña el rol de analista en el desarrollo de un sistema y su importancia, así como una explicación de la ingeniería de requisitos como aspecto fundamental en este rol y de las técnicas para la elicitación de los mismos.

1.2 Fundamentación del tema.

1.2.1 Gestión.

El significado de la palabra gestión, viene de la acción de llevar a cabo algo. Es la acción o trámite que junto con otros, se lleva a cabo para conseguir o resolver una cosa. Conjunto de operaciones que se realizan para dirigir y administrar un negocio. (*Diccionarios online*).

1.2.2 Gestión de Alojamiento.

En el marco de este trabajo, consiste en la acción de dar hospedaje a determinadas personas por determinadas razones. Existen diferentes tipos de instalaciones dedicadas a dar alojamiento, cada instalación cuenta con sus criterios de alojamiento y realiza esta gestión de forma diferente. Unas de las principales tareas que se llevan a cabo en este proceso son, llevar el control de todo el personal que se tiene alojado, las capacidades con que cuenta la instalación, el tiempo de estancia de cada persona, entre otras, que varían en dependencia del tipo de instalación.

1.2.3 Sistemas de Gestión de Alojamiento:

Los Sistemas dedicados a la gestión de alojamiento no son más que la forma de automatizar todas las funciones que se realizan en este proceso, es decir, es una aplicación informática que automatiza todo el proceso de alojar a una persona o conjunto de personas en una institución dedicada a este fin, es la forma

de tener informatizado todo lo que se hace de forma manual y complejo en muchos de los casos. Estos sistemas brindan determinadas funciones que favorecen todo el trabajo.

1.2.4 Sistemas automatizados existentes.

A continuación se hace un estudio de diferentes sistemas de gestión de alojamiento que existen, tanto internacionales como nacionales.

Sistemas Internacionales:



Newhotel: Software Hotelero Integrado.

Es un software de gestión hotelera de los más sofisticado, seguro y completo ofrecido por la actual tecnología informática, es la última generación de una familia de software para hotelería con veinte años en el mercado. Tiene más de mil instalaciones en Hoteles de 25 países del Mundo con las tecnologías líderes de Microsoft Windows, Base de Datos Oracle.

Tiene flexibilidad y posibilidades de configuración lo que le permiten adaptarse a las reales necesidades de cualquier Hotel y satisfacer sus exactos requerimientos fiscales y los objetivos de gestión y control. Ofrece rapidez de respuesta en todas las operaciones, agilidad en los check-in y check-out, histórico de huéspedes. Le permite hacer la gestión de Cuentas de No-Alojados o Eventos de la manera más directa. Dispone de cientos de reportes, estadísticas y pronósticos de formatos configurables. Otras importantes funcionalidades integradas en NewHotel son:

- La organización y control de Ama de Llaves.

- Mantenimiento.

- Avisos para los Huéspedes.

- Control de Presupuesto del Hotel.

- Módulos de Reservas por Internet.

Las funciones de recepción y atención al huésped permiten un tratamiento rápido y eficiente, pues todas las informaciones sobre cada huésped, sus preferencias, estancias anteriores y precios aparecen de forma instantánea. Las operaciones de check-in y de check-out, así como consulta de cuentas, transferencia de cargos, son muy rápidas. Una de las funcionalidades permite dar de alta un huésped que se envíe para otra unidad y elegir si la facturación se hace desde el Hotel de origen. (*Newhotel*).



Adagio: Sistema de Gestión Hotelera.

Adagio es la solución en software de gestión hotelera, para controlar y tener conocimiento de cómo se encuentra el hotel en cuanto a: ocupación, cajas, reservas, entre otros tantos datos posibles para consultar. Con él se gestiona el hotel de una forma rápida y eficiente. Su desarrollo robusto lo hace muy sólido y a pruebas de muchos fallos, manteniendo la copia de seguridad al día, puede solucionar errores y restablecer el sistema en tiempo. Adagio posee dos versiones actualmente, Adagio C/S y Adagio LT. (*Adagio. Sistema de Gestión Hotelera.*)

Adagio C/S

El Sistema de Gestión Adagio C/S, está orientado a grandes hoteles, de grandes necesidades de control y es multiusuario. Está orientado a un control riguroso de todos los movimientos dentro del hotel, y posee sistemas para encontrar movimientos extraños con gran facilidad. Tiene como función de ingreso del pasajero, cobro de factura, facturación mostrador, check-IN y check-OUT de los pasajeros. Gestiona las reservas, genera el estado de ocupación futuro y el ingreso de clientes a la agenda. Posee opciones de exportación de datos a MS Excel y a HTML para trasportar datos sin ningún inconveniente. (*Adagio C/S*)

Adagio LT

El Sistema de gestión Adagio LT es para el control de pequeños establecimientos hoteleros de hasta dos estrellas. Está diseñado para que el control del establecimiento sea mucho más simple y rápido cuando se necesite información de ocupación, nivel de reservas, generar estadísticas, control de gastos, posee una planilla de estado de ocupación, que con su simple estructura podrá ver como se encuentra el establecimiento. Aquí usted podrá hacer las reservaciones de sus habitaciones, realizar el check-IN de las mismas. Modificar reservas ya ingresada, eliminar y buscar reservas. No es un sistema multiusuario y no maneja permisos de usuario, pues fue desarrollado para hoteles pequeños. (*Adagio LT*)

Sistemas Nacionales:

La mayoría de los Hoteles en Cuba tienen el mismo sistema de Alojamiento Online, todos dan primeramente a sus clientes información detallada del centro en particular, en la que incluyen, la categoría de instalación, dirección y localización, el beneficio de acceder a esta área, los términos y condiciones,

las políticas de pago, las facilidades que ofrece y ofertas especiales, entre otras, con el objetivo de que el cliente pueda elegir bien su destino y no se arrepienta luego de haber reservado.

El software, que permite al cliente reservar online una vez decidido, brinda la posibilidad de elegir la moneda de pago, seleccionar la fecha de entrada y de salida, comprobar las fechas disponibles de reservas, para ahórrales tiempo, seleccionar el horario de entrada al hotel el día que comienza la reservación, reservar la cantidad de habitaciones que desee y del tipo que más le acomode, una vez seleccionada esta te muestra los servicios de habitación que se brindan en la misma, en algunas instalaciones se permite elegir el tipo de cama. El sistema solicita que se le especifique la cantidad de adultos que se hospedarán en la habitación y la cantidad de niños, entre otros datos.

Una vez llenado estos datos, el sistema muestra el precio de cada habitación por noche y el monto total de la reservación, si el cliente desea continuar con la reservación, introduce los datos personales y los envía. El sistema le muestra un resume de la reserva con todos sus datos y le envía al correo del cliente, un número de confirmación de reserva, porque el sistema no registra la reserva hasta que no es confirmada por el cliente nuevamente y se verifica la valides de la tarjeta de crédito.

El sistema brinda además las opciones de modificar la reserva o cancelarla. Si el cliente cancela su reservación con 72 horas o más de antelación a la fecha de entrada al hotel, le rembolsan el monto total pagado menos una pequeña suma por gastos administrativos. Si en cambio el cliente cancela su reservación con menos de 72 horas de antelación a la fecha de entrada al hotel, le rembolsan el monto total pagado, menos el valor de la primera noche de estancia prevista. Este sistema de reservas requiere para ser utilizado en la PC cliente, Internet Explorer, Netscape versión 4.1 o superior con la opción javascript. *(Reserva en Línea de Hoteles en Cuba)*

(Ver anexos 1, 2, 3 y 4)

Después de hacer un análisis de estos sistemas de gestión de alojamiento, se llegó a la conclusión de que no sirven como ejemplo, ni se pueden tomar como guía para desarrollar este trabajo, pues son de gestión hotelera, que no cumplen con las condiciones que requiere el sistema a desarrollar. El propósito principal de estos sistemas es el comercio y tienen como requisito fundamental, el cobro de los servicios que brindan.

Sistema en la Universidad de las Ciencias informática:

En la universidad el autor Ávila Malagón, desarrolló su trabajo de diploma sobre la problemática existente en Residencia, el cual tenía como objetivo desarrollar un sistema automatizado que gestionara todas las actividades de alojamiento. Esto surge como un proyecto de la facultad 1 llegando a implementar una aplicación. En el curso 2005-2006 la aplicación se comenzó a utilizar en la ubicación inicial, pero no cumplió con todas las expectativas esperadas, pues no recogía todos los requisitos que se necesitaban, para darle cumplimiento a todas las funcionalidades deseadas por la dirección de la residencia, esto, unido a la no actualización sistemática de la misma, provocó que al finalizar el período académico la información que brindaba era irreal y resultó inoperable para el actual.

De aquí que la Dirección de Informatización decidiera establecer en la facultad #3 un proyecto productivo que diera solución a esta problemática existente en la gestión de alojamiento y en otros procesos desarrollados en la misma área de la residencia, con el objetivo de obtener una nueva aplicación multiplataforma que brinde plena satisfacción en su uso y que cumpla las políticas de La Universidad y el país sobre el proceso migratorio hacia software libre.

1.3 Tendencias y Tecnologías actuales.

1.3.1 Metodologías más importantes y usadas en el mundo

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos software, en la que se van indicando paso a paso todas las actividades a realizar para lograr informatizar el proceso deseado, indicando qué personas deben participar en el desarrollo de las actividades y qué papel deben desempeñar. Además detallan la información que se debe producir como resultado de una actividad y la información necesaria para comenzarla.

El conjunto de técnicas indica cómo debe ser realizada una actividad determinada, identificada en la metodología. Combina el empleo de unos modelos o representaciones gráficas con el empleo de unos procedimientos detallados.

En el mundo existen diversas y variadas de estas metodologías, pero no todas se integran a las necesidades requeridas para el buen desarrollo de un software, por lo que la metodología a utilizar debe ser exquisitamente escogida y adaptada de acuerdo a las peculiaridades del proyecto en particular, para

tratar de minimizar al máximos, los riesgos a los que se está expenso inevitablemente, a la hora de desarrollar un software, para obtener los resultados esperados con el menor costo y esfuerzo posible.

En estos últimos años se han desarrollado dos corrientes en lo referente a las metodologías de desarrollo, las llamadas Metodologías Tradicionales o Métodos Pesados y las Metodologías Ágiles o los Métodos Ligeros. Mientras lo Métodos Pesados intentan conseguir el objetivo común, por medio de orden y documentación, los Métodos Ligeros tratan de mejorar la calidad del software por medio de una comunicación directa e inmediata entre las personas que intervienen en el proceso.(MOLPECERES 2003)

Programación Extrema (Extreme Programming, XP):

Se clasifica como Metodología Ágil y es una de las más exitosas y popular en la actualidad, utilizadas para proyectos de corto plazo, equipo pequeño y cuyo plazo de entrega es en tiempo record. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto.

Características de XP:

Pruebas Unitarias: se basa en las pruebas realizadas a los principales procesos, de tal manera que adelantan algo hacia el futuro, se pueden hacer pruebas de las fallas que pudieran ocurrir. Es como adelantarse a obtener los posibles errores.

Refabricación: se basa en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.

Programación en pares: una particularidad de esta metodología es que propone la programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento. (SANCHEZ 2004)

XP propone:

- Empezar en pequeño y añadir funcionalidad con retroalimentación continua.
- Que el manejo del cambio se convierta en parte sustantiva del proceso.
- Que el costo del cambio no dependa de la fase o etapa.
- No introducir funcionalidades antes que sean necesarias.
- Que el cliente o el usuario se convierte en miembro del equipo.(SANCHEZ 2004)

Lo fundamental en este tipo de metodología es:

- La comunicación, entre los usuarios y los desarrolladores.
- La simplicidad, al desarrollar y codificar los módulos del sistema.
- La retroalimentación, concreta y frecuente del equipo de desarrollo, el cliente y los usuarios finales. (SANCHEZ 2004)

Roles en XP:

- **Programador:** Escribe las pruebas unitarias y produce el código del sistema.
- **Cliente:** Escribe las historias de usuario y las pruebas funcionales para validar su implementación. Además, asigna la prioridad a las historias de usuario y decide cuáles se implementan en cada iteración centrándose en aportar mayor valor al negocio.
- **Encargado de pruebas:** Ayuda al cliente a escribir las pruebas funcionales. Ejecuta las pruebas regularmente, difunde los resultados en el equipo y es responsable de las herramientas de soporte para pruebas.
- **Encargado de seguimiento:** Proporciona realimentación al equipo. Verifica el grado de acierto entre las estimaciones realizadas y el tiempo real dedicado, para mejorar futuras estimaciones. Realiza el seguimiento del progreso de cada iteración.
- **Entrenador:** Es responsable del proceso global. Debe proveer guías al equipo de forma que se apliquen las prácticas XP y se siga el proceso correctamente.
- **Consultor:** Es un miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto, en el que puedan surgir problemas.
- **Gestor:** Es el vínculo entre clientes y programadores, ayuda a que el equipo trabaje efectivamente creando las condiciones adecuadas. Su labor esencial es de coordinación. (PATRICIO LETELIER TORRES 2003)

Pasos del ciclo de desarrollo de XP (a grandes rasgos):

1. El cliente define el valor de negocio a implementar.
2. El programador estima el esfuerzo necesario para su implementación.
3. El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo.
4. El programador construye ese valor de negocio.

5. Vuelve al paso 1. (PATRICIO LETELIER TORRES 2003)

El ciclo de vida ideal de XP consiste de seis fases:

- Exploración
- Planificación de la Entrega
- Iteraciones
- Producción
- Mantenimiento
- Muerte del Proyecto.(PATRICIO LETELIER TORRES 2003)

Pero, justamente en su agilidad, XP olvida puntos claves de control (como la descripción total de las acciones desde el punto de vista del usuario o caso de uso) en un proceso de desarrollo y cae en el terreno de las suposiciones.

Marco de Soluciones Microsoft (Microsoft Solution Framework, MSF):

Es un compendio de las mejores prácticas en cuanto a administración de proyectos se refiere, es una metodología flexible e interrelacionada con una serie de conceptos, modelos y prácticas de uso, que controlan la planificación, el desarrollo, la gestión de proyectos tecnológicos y se centra en los modelos de proceso y de equipo dejando en un segundo plano las elecciones tecnológicas. MSF reconoce que la tecnología no es la única pieza clave para implantar soluciones exitosas. Para lograr el éxito, es necesario incorporar estrategias de manejo de riesgos y administración de recursos, así como una definición clara de roles y responsabilidades del equipo.(CONSULTORES 2006)

Características de MSF:

Adaptable: es parecido a un compás, usado en cualquier parte como un mapa, del cual su uso es limitado a un específico lugar.

Escalable: puede organizar equipos tan pequeños entre 3 o 4 personas, así como también, proyectos que requieren 50 personas o más.

Flexible: es utilizada en el ambiente de desarrollo de cualquier cliente.

Tecnología Agnóstica: porque puede ser usada para desarrollar soluciones basadas sobre cualquier tecnología.(SANCHEZ 2004)

Fases de MSF

Análisis y Diseño

Desarrollo
Implantación
Estabilización

Modelos encargados de planificar las diferentes partes implicadas en el desarrollo de un proyecto:

Modelo de Arquitectura del Proyecto.
Modelo de Equipo.
Modelo de Proceso.
Modelo de Gestión del Riesgo.
Modelo de Diseño de Proceso.
Modelo de Aplicación.

Roles de MSF

Los equipos organizados bajo MSF son pequeños y multidisciplinarios, en los cuales los miembros comparten responsabilidades y balancean las destrezas del equipo para mantenerse enfocados en el proyecto que están desarrollando. Comparten una visión común del proyecto y se enfocan en implementar la solución, con altos estándares de calidad y deseos de aprender.

El modelo de equipos de MSF tiene seis roles que corresponden a las metas principales de un proyecto y son responsables por las mismas. Cada rol puede estar compuesto por una o más personas, la estructura circular del modelo, con óvalos del mismo tamaño para todos los roles, muestra que no es un modelo jerárquico y que cada todos los roles son igualmente importantes en su aporte al proyecto. Aunque los roles pueden tener diferentes niveles de actividad durante las diversas etapas del proyecto, ninguno puede ser omitido.(CONSULTORES 2006)

Desarrollo Guiado por la Funcionalidad (Feature Driven Development, FDD.)

Se puede considerar a medio camino entre RUP y XP, pero si hay que distinguir entre pesado y ligero, se puede decir que es más similar a este último. Está pensado para proyectos con tiempo de desarrollo relativamente cortos (menos de un año). Se basa en un proceso iterativo con iteraciones cortas (2 semanas) que producen un software funcional que el cliente y la dirección de la empresa pueden ver y

monitorizar. Las iteraciones se deciden en base a funcionalidades (de ahí el nombre del proceso), que son pequeñas partes del software con significado para el cliente.(REYNOSO 2004)

FDD se divide en 5 fases:

1. Desarrollo de un modelo general
2. Construcción de la lista de funcionalidades
3. Plan de releases en base a las funcionalidades a implementar
4. Diseñar en base a las funcionalidades
5. Implementar en base a las funcionalidades.

Roles de FDD

Hay tres categorías de rol en FDD:

Roles claves: Los seis roles claves de un proyecto son:

1. Administrador del proyecto: Tiene la última palabra en materia de visión, cronograma y asignación del personal.
2. Arquitecto jefe: Puede dividirse en arquitecto de dominio y arquitecto técnico.
3. Manager de desarrollo: Puede combinarse con arquitecto jefe o manager de proyecto.
4. Programador jefe: Participa en el análisis del requerimiento y selecciona rasgos del conjunto a desarrollar en la siguiente iteración.
5. Propietarios de clases: Trabajan bajo la guía del programador jefe en diseño, codificación, prueba y documentación repartidos por rasgos.
6. Experto de dominio: Puede ser un cliente, patrocinador, analista de negocios o una mezcla de todo eso.

Roles de soporte: Los cinco roles de soporte comprenden:

1. Administrador de entrega: Controla el progreso del proceso revisando los reportes del programador jefe y manteniendo reuniones breves con él, reporta al manager del proyecto.
2. Abogado de lenguaje: Conoce a la perfección el lenguaje y la tecnología.
3. Ingeniero de construcción: Se encarga del control de versiones de los builds y publica la documentación.
4. Herramientita: Construye herramientas o mantiene bases de datos y sitios Web.
5. Administrador del sistema: Controla el ambiente de trabajo o productiza el sistema cuando se lo entrega.

Roles adicionales: Los tres roles adicionales son:

1. Los de verificadores
2. Encargados del despliegue
3. Escritores técnicos.

(REYNOSO 2004), (CALABRIA 2003)

A modo de conclusiones podemos decir que FDD es un proceso que ayuda al equipo a producir resultados periódicos y tangibles, que utiliza pequeños bloques que contienen la funcionalidad del sistema, llamados features y organiza los bloques que están relacionados entre sí, en una lista llamada feature set. Hace énfasis en la obtención de resultados cada dos semanas e incluye estrategias de planificación que hacen que las funcionalidades puedan desarrollarse en dichos lapsos.

Proceso Unificado de Desarrollo (RUP).

RUP es el resultado de varios años de desarrollo y uso práctico en el que se han unificado técnicas de desarrollo. La versión que se ha estandarizado vio la luz en 1998 y se conoció en sus inicios como Proceso Unificado de Rational 5.0; de ahí las siglas con las que se identifica a este proceso de desarrollo. (*Conferencia de Ingeniería de Software .Introducción a la Ingeniería de Software 2005-2006*)

Sus características fundamentales son:

- Guiado y Manejado por casos de uso.
- Centrado en arquitectura.
- Iterativo e Incremental.

Flujos de trabajo en RUP

Modelamiento del negocio: Describe los procesos de negocio, identificando quiénes participan y las actividades que requieren automatización.

Requerimientos: Define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.

Análisis y diseño: Describe cómo el sistema será realizado a partir de la funcionalidad prevista y las restricciones impuestas, indica con precisión lo que se debe programar.

Implementación: Define cómo se organizan las clases y objetos en componentes, cuáles nodos se utilizarán y la ubicación en ellos de los componentes y la estructura de capas de la aplicación.

Prueba: Busca los defectos a lo largo del ciclo de vida.

Instalación: Produce release del producto y realiza actividades (empaquete, instalación, asistencia a usuarios, etc.) para entregar el software a los usuarios finales.

Administración del proyecto: Involucra actividades con las que se busca producir un producto que satisfaga las necesidades de los clientes.

Administración de configuración y cambios: Describe cómo controlar los elementos producidos por todos los integrantes del equipo de proyecto en cuanto a: utilización/actualización concurrente de elementos, control de versiones, etc.

Ambiente: Contiene actividades que describen los procesos y herramientas que soportarán el equipo de trabajo del proyecto; así como el procedimiento para implementar el proceso en una organización. (*Conferencia de Ingeniería de Software .Introducción a la Ingeniería de Software 2005-2006*)

RUP se divide en 4 Fases:

Inicio: Se describe el negocio y se delimita el proyecto describiendo sus alcances con la identificación de los casos de uso del sistema, en otras palabras el objetivo en esta etapa es determinar la visión del proyecto, es la puesta en marcha.

Elaboración: Se define la arquitectura del sistema y se obtiene una aplicación ejecutable que responde a los casos de uso que la comprometen. A pesar de que se desarrolla a profundidad una parte del sistema, las decisiones sobre la arquitectura se hacen sobre la base de la comprensión del sistema completo y los requerimientos (funcionales y no funcionales) identificados de acuerdo al alcance definido.

Construcción: Se obtiene un producto listo para su utilización que está documentado y tiene un manual de usuario. Se obtiene 1 o varios release del producto que han pasado las pruebas. Se ponen estos release a consideración de un subconjunto de usuarios.

Transición: El release ya está listo para su instalación en las condiciones reales. Puede implicar reparación de errores.(SANCHEZ 2004)

Grupos de Roles en RUP

Analistas, Desarrolladores, Administradores, Soporte, Roles Adicionales

1.3.2 Metodología escogida para desarrollar el trabajo (RUP).

Para desarrollar este trabajo se decidió utilizar una de las metodologías pesada, pues están basadas en normas provenientes de estándares seguidos por el entorno de desarrollo y proceso mucho más

controlado, con numerosas políticas y normas, mientras que las ágiles están basadas en heurísticas provenientes de prácticas de producción de código y procesos menos controlados y con pocos principios.

La metodología escogida, RUP, es una de las más generales de las existentes actualmente, ya que está pensada para adaptarse a cualquier proyecto, principalmente con grupos de producción grandes, donde el ciclo de vida está dirigido por casos de uso que reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos, los casos de uso guían a partir de aquí, todo el proceso de desarrollo, donde la arquitectura del software es esencial y muestra una visión común del sistema, en la que el equipo de proyecto y los usuarios, deben estar de acuerdo, la arquitectura muestra la visión común del sistema y se expresa mediante modelos.

RUP divide el proceso de desarrollo, en ciclos de iteración, teniendo un producto final, al concluir cada ciclo y en cada uno hace exigencia del uso de artefactos, para lograr los hitos propuestos en cada una de las fases, con el propósito de obtener un incremento en el proceso de desarrollo, siendo por este motivo una de las metodologías más importante para alcanzar un grado de certificación en el desarrollo del software. (*Conferencia de Ingeniería de Software .Introducción a la Ingeniería de Software 2005-2006*)

En RUP el cliente interactúa con el equipo de desarrollo mediante reuniones y existe entre ellos un contrato prefijado, pero no forma parte del equipo de desarrollo como en las metodologías Ágiles.

1.3.3 Lenguaje de Modelado a utilizar.

Lenguaje Unificado de Modelado (**UML**, *Unified Modeling Language*), es un lenguaje de propósito general, que pueden usar todos los modeladores, es uno de los más conocidos y utilizado en la actualidad. UML es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software (RUMBAUGH 2000). UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables.

UML no pretende ser un método de desarrollo, no incluye un proceso de desarrollo paso a paso, pero sí incluye todos los conceptos que se consideran necesarios para utilizar un proceso moderno iterativo, basado en construir una sólida arquitectura para resolver requisitos dirigidos por casos de uso.

Se utiliza para definir un sistema de software, para detallar los artefactos en el sistema y para documentar y construir, es el lenguaje en el que está descrito el modelo. Se puede aplicar en una gran

variedad de formas para soportar una metodología de desarrollo de software (tal como el Proceso Unificado de Rational), pero no especifica en sí mismo qué metodología o proceso usar.

UML se puede usar para modelar distintos tipos de sistemas: sistemas de software, sistemas de hardware, y organizaciones del mundo real. UML ofrece diferentes tipos de diagramas. (IVAR JACOBSON)

Diagramas de estructura estática

- Diagrama de clases
- Diagrama de objetos
- Diagrama de casos de uso

Diagramas de comportamiento

- Diagramas de interacción (secuencia y colaboración)
- Diagrama de estados
- Diagrama de actividad

Diagramas de implementación

- Diagrama de componentes
- Diagrama de despliegue

1.3.4 Herramientas

En la actualidad existen diversas y variadas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo y mantenimiento del software, reduciendo el coste de las mismas en términos de tiempo y de dinero. Estas aplicaciones son conocidas como Herramientas CASE (Ingeniería de Software Asistida por Ordenador, por sus siglas en ingles: **Computer Aided Software Engineering**) y ayudan en el desarrollo del software en tareas como: las fases de planificación, análisis de requisitos y estrategia del desarrollo, usando, entre otros diagramas UML, en el proceso de realizar un diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores, entre otras.

Las herramientas escogidas para el desarrollo de este trabajo fueron:

Rational Rose

Es una herramienta para el modelado de sistemas complejos y de tiempo real que unifica todos sus equipos de desarrollo a través del modelamiento, el cual está basado en UML. Rational Rose permite visualizar, entender, y refinar los requerimientos y arquitectura antes de enfrentar el código, evitando esfuerzos desperdiciados en el ciclo de desarrollo. Usar una sola herramienta de modelado a través del ciclo de vida de desarrollo, permite asegurar que se está construyendo el sistema correcto.

Para los analistas ofrece la capacidad de modelar y visualizar sus procesos de negocios y destacar oportunidades para aumentar la eficiencia, mejora la comunicación entre los clientes y los desarrolladores. Al modelar los casos de uso en Rose permite que la solución creada se haga con el usuario en mente.

Esta herramienta unifica a los analistas de negocios, sistemas y datos al permitirles crear y administrar modelos en una herramienta con un solo lenguaje de modelado.(INDUDATA)

Esta herramienta, se escogió para el desarrollo de este trabajo, por todas las facilidades que ofrece y por ser la más conocida y usada en la UCI.

Microsoft office Visio 2003.

La misma es una solución de diseño de dibujos y creación de diagramas que facilita transformar conceptos tecnológicos y empresariales en diagramas visuales, los cuales hacen que se organicen y entiendan mucho mejor las ideas que se quieren transmitir. Permite crear, con facilidad y sin mucha preparación los siguientes diagramas:

Diagrama Web, Diagrama de Bloques, de Base de datos, de Reuniones creativas, de Proceso empresarial, diagramas de flujo, de programación de proyectos, líneas de tiempo, entre otros más. Permite compartir dichos diagramas, revisarlos e incluir en ellos comentarios, además de guardarlos, importarlos y exportarlos en diferentes formatos.

Visio permite documentar las bases de datos de Microsoft SQL Server y Microsoft Access y diseñar las estructuras de bases de datos propuestas utilizando diagramas de relaciones entre entidades. Tiene incluido una plantilla de interfaz de usuario Microsoft Windows XP. (*Visio 2003*)

Esta herramienta es utilizada en este trabajo, como una herramienta de apoyo por las posibilidades de diseño que brinda, para modelar los prototipos de interfaz no funcionales del sistema que se propone.

1.4 Patrones de casos de uso.

“Cada patrón describe un problema que ocurre una y otra vez en nuestro entorno, para describir después el núcleo de la solución a ese problema, de tal manera, que esa solución pueda ser usada más de un millón de veces sin hacerlo siquiera dos veces de la misma forma”. (Conferencia de Ingeniería de Software. Patrones de diseño 2005-2006)

Christopher Alexander

Los patrones permiten y han permitido en diferentes áreas del conocimiento humano rehusar la esencia de la solución de un problema al enfrentar nuevos problemas similares. Es así que los patrones constituyen una especie de mecanismo de registro y concentración de experticia. En la práctica se hace difícil, en algunas ocasiones, escoger qué patrón resulta más apropiado para resolver el problema. Existen diferentes patrones de casos de usos, algunos de estos son:

Concreta Extensión o Inclusión. Este patrón está dividido en concreta extensión o concreta inclusión

- **Extensión:** Consiste en dos casos de uso y una relación extendida entre ellos. Puede ser instalado en si mismo, así como extendido en el caso de uso base. El referente puede ser concreto o abstracto. Este patrón se aplica cuando un flujo puede extender el flujo de otro caso de uso así como ser realizado en si mismo.
- **Inclusión:** En este Patrón, se incluye una relación del caso de uso base al caso de uso de inclusión. El último puede ser instalado en si mismo. El caso de uso base puede ser concreto o abstracto. (GUNNAR ÖVERGAARD 2004)

Caso de uso grande: Estructura un caso de uso que comprende un número grande de acciones. Este se divide en Sucesión larga y Caminos múltiples.

- **Sucesión Larga:** Contiene un caso de uso que consiste en una sucesión muy larga de acciones que siempre serán realizadas como una unidad. Este Patrón debe usarse cuando al general el caso de uso grande producirían dos casos de uso donde uno de ellos siempre será realizado inmediatamente después del otro; por lo menos uno de ellos no modelará un uso completo del sistema. El inconveniente es que la descripción del caso del uso se pone muy larga, requiriendo estructuración del texto.
- **Caminos Múltiples:** Modela casos de uso múltiples, donde cada uno de ellos modela una alternativa del uso del sistema. Este Patrón se aplica cuando la usabilidad a ser modelada consiste en flujos alternativos múltiples. (GUNNAR ÖVERGAARD 2004)

CRUD: crear (create), leer (read), actualizar (update), borrar (delete).

Este patrón consiste en crear, leer, actualizar, y borrar segmentos de información en un solo caso de uso que forma una unidad conceptual. Puede usarse completo o parcial.

- **Completo:** Consiste en un caso de uso, llamado CRUD o (Administrador de Información), que modela todos los funcionamientos diferentes que pueden realizarse en un segmento de información de un cierto tipo, como crear, leer, actualizar, y borrar.
- **Parcial:** Modela una de las alternativas del caso de uso como un caso de uso separado. Este patrón es preferible cuando una de las alternativas del caso de uso es más significativa, más larga, o mucho más compleja que las otras alternativas. (GUNNAR ÖVERGAARD 2004)

Sistema de capas. Estructura el modelo del caso de uso para que cada uno se defina dentro de una capa. Está dividido en los siguientes patrones:

- **Reutilización:** Este patrón contiene dos casos de uso y dos paquetes con una relación de importación entre los paquetes. El caso de uso definido en la capa superior incluye una relación al caso de uso que se define en la capa más baja y se importa en la capa superior. El caso de uso definido en la capa más baja es a menudo, pero no siempre, abstracto. Es apropiado usarlo cuando el caso de uso empieza en la capa superior, pero usará un servicio definido en la capa más baja. No es conveniente cuando el caso del caso de uso empieza en la capa más baja.
- **Adición:** En este patrón, el caso de uso definido en la capa superior extiende el caso de uso definido en la capa más baja. Aquí el caso de uso empieza en la capa más baja, pero los servicios definidos en la capa superior se inserta en él. El caso de uso en la capa superior es normalmente el abstracto. Es aplicable cuando el caso del caso de uso empieza en la capa más baja, pero no debe usarse si empieza en la capa superior.
- **Especialización:** En este patrón el caso de uso definido en la capa superior es una especialización del caso de uso definida en la capa más baja; es decir, el anterior tiene una generalización al último. Es aplicable cuando el caso de uso superior es del mismo tipo que el caso de uso más bajo. (GUNNAR ÖVERGAARD 2004)

Actores múltiples: Está dividido en Roles distintos y Rol común.

- **Roles distinto:** El patrón de los Roles distintos consiste en un caso de uso y por lo menos dos actores. Es usado cuando los dos actores juegan roles diferentes hacia el caso de uso; es decir, ellos actúan diferentemente con el caso de uso.

- **Rol común:** Los dos actores juegan el mismo papel hacia el caso del uso. Este papel es representado por otro actor, heredado por los actores que comparten este papel. (GUNNAR ÖVERGAARD 2004)

Servicio Opcional: Se divide en Adición, Independiente y Especialización

- **Adición:** Contiene dos casos de uso y una relación extendida. El primer caso de uso modela la parte importante en el sistema. El segundo caso de uso modela una suma al primer caso de uso que puede agregarse al sistema. Solo la parte adicional se expresa en el segundo caso de uso, el cual es abstracto; es decir, no se realizara en si mismo.
- **Especialización:** En este patrón, el caso de uso importante se especializa en un caso de uso incluyendo la parte opcional. Se usa cuando el caso de uso importante incluye una forma simple de la conducta opcional. El caso de uso hijo especializará esta parte simple en la conducta opcional, más avanzada.
- **Independiente:** Modela los casos de uso opcionales separados sin las relaciones con los casos de usos importantes. Se aplica cuando el servicio opcional es independiente de los servicios importantes.(GUNNAR ÖVERGAARD 2004)

Vista Ortogonal: Proporciona vistas diferentes de los flujos de un sistema que son percibidas indistintamente por diversos stakeholders. Están la vista ortogonal de especialización y la de descripción.

- **Especialización:** Este patrón consiste en un caso de uso y una colección de especializaciones de ese caso de uso. El anterior (el caso de uso padre) modela lo que realmente se realiza dentro del sistema, considerando que cada uno de los otros casos de uso (cada uno de los casos de uso hijos) modela un uso específico del caso de uso padre cuando es percibido por el usuario.
- **Descripción:** En algunas situaciones, un caso de uso incluye mucho más comportamiento que el que los usuarios del sistema perciben. En esta alternativa, se modela el caso de uso como uno solo, con el comportamiento extra descrito en secciones separadas de la descripción del caso de uso. Esta alternativa es preferible usarla cuando los stakeholders tienen percepciones diferentes de un caso de uso; algunos stakeholders pueden, no tener certeza de la magnitud del comportamiento del caso de uso. (GUNNAR ÖVERGAARD 2004)

Reglas del negocio: Son un compendio de información proveniente de políticas que deben cumplirse o condiciones que deben satisfacerse, en el negocio. Este patrón se divide en Definición Estática y Modificación Dinámica.

- **Definición Estática:** Este patrón es aplicado al modelado de todos los casos de uso que son afectados por las reglas del negocio en la organización. Es apropiado cuando no es necesario cambios en las reglas del negocio mientras el sistema está en uso.
- **Modificación Dinámica:** Este patrón contiene un caso de uso llamado Regla Administrativa, el cual crea, actualiza y borra reglas del negocio. Es usado cuando la colección de reglas debe ser modificada dinámicamente; es decir modificadas mientras el sistema está levantado y corriendo.(GUNNAR ÖVERGAARD 2004)

1.5 Rol Analista

¿Qué es un rol?

Un rol es una definición abstracta de un conjunto de actividades realizadas y de artefactos obtenidos. Los roles son realizados típicamente por un individuo, o un conjunto de individuos, trabajando juntos en equipo. Un miembro del equipo de proyecto cumple normalmente muchos roles. Los roles no son individuos; en lugar de ello, describen cómo los individuos se comportan en el negocio y qué responsabilidades tienen estos individuos.(GUERRERO)

El rol de Analista:

Un analista, en la disciplina de la ingeniería del software, es aquel individuo que tiene como cometido analizar un problema y describirlo con el propósito de ser solucionado mediante un sistema informático. Es imprescindible en cualquier organización, debido al abanico de destrezas que éste posee y los beneficios que le produce. No sólo se encarga de estudiar la organización y desarrollar un sistema automatizado, la labor del analista es mas que eso, es también la de asesorar, supervisar, recomendar y modificar procesos internos y algunas veces de modificar la estructura misma de la empresa, con el propósito de lograr los objetivos que se proponen.

El Analista de Sistema surge de la necesidad de recopilar, desglosar, catalogar y analizar información necesaria de una empresa para poder proponer nuevos métodos, mejores o modificar los actuales para que así aumente el desempeño de los departamentos dentro de la organización. Un analista se vale de la información de entrada, los procesos modificadores y la información de salida, para así definir los procesos intermedios y poder entender con claridad a la organización. Todos estos flujos y procesos son

estudiados sistemáticamente para poder determinar si son los adecuados, si se deben mejorar o si deben ser reemplazados por otros más idóneos.

Las cualidades que se esperan de un analista son esencialmente la capacidad de abstracción y de análisis, facilidad de comunicación que le permita relacionarse en forma significativa con diferentes personas diariamente, que tenga conocimientos básicos de usabilidad, pues cualquier sistema que no esté al servicio de los usuarios o diseñado pensado en el, no tiene mucho sentido.

Actividades que realiza:

Detalla la especificación de una parte de la organización describiendo el flujo de trabajo de uno o varios casos de uso del negocio. Establece qué actores del negocio y casos de uso del negocio existen y como trabajan entre ellos.

Especifica los trabajadores del negocio y las entidades necesarias para realizar un caso de uso del negocio y distribuye el comportamiento del caso de uso del negocio a éstos.

Define las responsabilidades, las operaciones, las cualidades, y las relaciones de uno o varios trabajadores del negocio y entidades de negocio.

Conduce y coordina los requerimientos y los Casos de Uso, modelando y especificando las funcionalidad del sistema y delimitando el sistema; por ejemplo, estableciendo qué actores y casos de uso existen y como interactúan.

Conduce y coordina los prototipos y el diseño de la interfaz usuario, esto lo hace construyendo prototipos de Interfaces de usuarios. (Ayuda extendida del Rational Rose)

1.6 Ingeniería de Requisitos.

Desde el inicio del desarrollo de sistemas ha estado presente el gran problema de la identificación de los requisitos, debido a que no es un proceso que pueda ser determinado matemáticamente, en el cual los datos son extraídos de las personas interesadas, lo que hace que estos puedan variar en dependencia de la persona a la que se consulte. Todo esto hace que la ingeniería de requisitos se considere como una etapa clave en el desarrollo de software. Es el proceso de definir los servicios que el cliente requiere del sistema y las restricciones bajo las cuales este opera y es desarrollado. Su importancia es considerada esencial, pues los errores más comunes y más costosos de reparar, así como los que más tiempo consumen se deben a una inadecuada ingeniería de requisitos. Mediante el tratamiento de requisitos se

especifican los servicios que debe proporcionar el sistema, consiste en un proceso iterativo y cooperativo de análisis del problema. Todo este proceso de recopilar información, definir las necesidades del sistema son procesos complejos, pues hay que identificar los requisitos que el sistema debe cumplir para satisfacer las necesidades de los usuarios finales y de los clientes, además para que el sistema sea considerado un software de calidad.(MARÍA JOSEFA ESCALONA 2002), (PRESSMAN 2005)

1.6.1 Características de los requisitos.

Los requisitos se dividen en requisitos funcionales y requisitos no funcionales. Los requisitos funcionales definen las funciones que el sistema será capaz de realizar, describen las transformaciones que el sistema realiza sobre las entradas para producir salidas. Los requisitos no funcionales tienen que ver con características que de una u otra forma puedan limitar el sistema, como por ejemplo, el rendimiento (en tiempo y espacio), interfaces de usuario, fiabilidad (robustez del sistema, disponibilidad de equipo), mantenimiento, seguridad, portabilidad, estándares, entre otras. (MARÍA JOSEFA ESCALONA 2002; TORRES)

Las características de un requisito son sus propiedades principales. Un conjunto de requerimientos en estado de madurez, deben presentar una serie de características tanto individualmente como en grupo. Algunas de estas características son:

- Necesario: Si su omisión provoca una deficiencia en el sistema a construir, y además su capacidad, características físicas o factor de calidad no pueden ser reemplazados por otras capacidades del producto o del proceso.
- Conciso: Si es fácil de leer y entender. Su redacción debe ser simple y clara para aquellos que vayan a consultarlo en un futuro.
- Completo: Si no necesita ampliar detalles en su redacción, es decir, si se proporciona la información suficiente para su comprensión.
- Consistente: Si no es contradictorio con otro requerimiento.
- No ambiguo: Cuando tiene una sola interpretación. El lenguaje usado en su definición, no debe causar confusiones al lector.
- Verificable: Un requerimiento es verificable cuando puede ser cuantificado de manera que permita hacer uso de los siguientes métodos de verificación: inspección, análisis, demostración o pruebas. (PRESSMAN 2005), (TORRES)

1.6.2 Pasos a seguir en la ingeniería de requisitos:

La ingeniería de requisitos facilita el mecanismo apropiado para comprender lo que quiere el cliente, analizando necesidades, confirmando su viabilidad, negociando una solución razonable. Este proceso puede ser descrito en pasos diferentes:

Identificación de Requisitos, Análisis de Requisitos y Negociación, Especificación de Requisitos, Validación de Requisitos y Gestión de Requisitos.

Identificación o Elicitación de requisitos:

Es una de las primeras actividades a realizarse en la Ingeniería de requisitos, y una de las más difíciles, parece bastante simple preguntar al cliente, a los usuarios y a los que están involucrados en los objetivos del sistema, como los sistemas se ajustan a las necesidades del negocio y como este va a ser utilizado en el día a día, pero al final esto que parece simple es complicado. Se ha identificado una serie de problemas en la elicitación de requisitos:

- **Problema de Alcance:** No está bien definido el límite del sistema que ha sido aportado por los clientes, lo cual puede confundir más que dar un aporte de claridad a los objetivos del sistema.
- **Problemas de comprensión:** Los clientes no están completamente seguros de lo que necesitan, tienen una pobre comprensión de las capacidades y limitaciones de su entorno de computación, no existe un total entendimiento del dominio del problema, hay dificultades para comunicar las necesidades. Omiten información por considerar que es obvia, especifican requisitos ambiguos o pocos estables.
- **Problemas de volatilidad:** Los requisitos no permanecen estáticos, sino que pueden variar durante el desarrollo del sistema. Los usuarios durante el período que se tarda en desarrollar el sistema pueden cambiar políticas o necesidades de la organización.(PRESSMAN 2005)

Análisis y negociación de requisitos:

La diversa gama de fuentes de las cuales provienen los requisitos, hace necesario un análisis de los mismos antes de definir si son adecuados para el cliente. Los requisitos se agrupan por categorías y se organizan en subconjuntos, se estudia cada requisito en relación con el resto, se examinan los requisitos en su consistencia, completitud y ambigüedad. Se califican en base a las necesidades de los clientes. Todo esto se hace con el fin de identificar la importancia que tiene un requerimiento en términos de

implementación. La prioridad de cada requisito dependerá de las necesidades que tenga el negocio. En esta actividad se pueden presentar una serie de conflictos con el cliente, que se deben resolver a través de un proceso de negociación, incrementando la comunicación con el mismo.(PRESSMAN 2005)

Especificación de Requisitos:

En esta actividad se genera un documento, que contiene una descripción completa de las necesidades y funcionalidades del sistema que será desarrollado, describe el alcance del sistema y la forma en como hará sus funciones. Es el producto final sobre los requisitos obtenidos, describe la información que entra y sale del sistema.

Validación de Requisitos:

La validación es la actividad de la que permite demostrar que los requerimientos definidos en el sistema son los que realmente quiere el cliente; además revisa que no se haya omitido ninguno, que no sean ambiguos, inconsistentes o redundantes. Esta actividad revisa el cumplimiento de las características de la especificación de requisitos. La validación de requisitos es importante pues, de ella depende que no existan elevados costos de mantenimiento para el software desarrollado.

Gestión de Requisitos:

La gestión de requisitos es un conjunto de actividades que ayudan al equipo de trabajo a identificar, controlar, seguir los requisitos y los cambios en cualquier momento. Los cambios en los requisitos involucran modificaciones que pueden tener impacto en otros requisitos, por lo que es necesario que la administración de cambios involucre actividades como: guardar historial de cada requerimiento, identificar dependencias entre ellos, y mantener un control de versiones.(PRESSMAN 2005)

1.6.3 Técnicas utilizadas en la ingeniería de requisitos.

Existen varias técnicas que se han desarrollado para darle cumplimiento a las actividades que se desarrollan dentro de la ingeniería de requisitos, las cuales permiten hacer este proceso de una forma más eficiente y segura. Algunas de estas técnicas son:

Entrevistas:

Resultan una técnica muy aceptada, y su uso está ampliamente extendido. Las entrevistas le permiten al analista tomar conocimiento del problema y comprender los objetivos de la solución buscada. Mediante esta técnica el equipo de trabajo se acerca al problema de una forma natural.

Existen diferentes tipos de entrevistas:

- **Entrevistas de Cuestionarios:** Este tipo de entrevistas recomienda que se genere un cuestionario de preguntas relacionadas con varios aspectos de un sistema, el cual será aplicado al cliente para comenzar la captura de requisitos. Las preguntas que deben realizarse en esta técnica, deben ser preguntas de alto nivel y abstractas que pueden realizarse al inicio del proyecto para obtener información sobre aspectos globales del problema del usuario y soluciones potenciales.
- **Entrevistas abiertas:** Las entrevistas abiertas son muy usadas y evita muchos problemas de los métodos por cuestionarios. Es decir, el entrevistador formula una pregunta y luego le permite al entrevistado responder como lo desee. El entrevistador puede pedir mas detalles pero no fija los términos de la entrevista. Esto parece más benigno que los cuestionarios, pero el problema de ver si la pregunta tiene respuesta o si la respuesta es parte del repertorio normal de discurso del entrevistado, permanece.
- **Entrevistas en grupos de desarrollo:** Este tipo de entrevistas recomienda formar grupos específicos con el personal del cliente. Estos grupos tendrán en común algún área de trabajo o especialidad. EL objetivo es poder contar con los expertos en cierta área de la empresa para poder llegar en conjunto a la especificación de requisitos.
- **Discusiones:** Este tipo de entrevistas pretende que el entrevistador sostenga una discusión con el Cliente sobre su problemática para tratar de determinar en conjunto los requisitos del sistema.

Análisis de Protocolo:

Esta técnica parte de la idea de que el cliente cuenta con un modelo mental preexistente del sistema deseado y en base a este modelo ya existente se puede analizar y obtener los requisitos del sistema. Es una técnica muy poco utilizada debido a que los Clientes rara vez poseen una idea clara de lo que desean en su sistema.

Lluvia de ideas:

Esta técnica es también denominada **Tormenta de ideas**, es una herramienta de trabajo grupal, reuniones en grupo que facilita el surgimiento de nuevas ideas sobre un tema o problema determinado. Es una técnica de grupo para generar ideas originales en un ambiente relajado, donde los participantes muestran sus ideas de forma libre. Las ideas de una persona, serán asociadas de manera distinta por cada miembro, y hará que aparezcan otras por contacto. Entre los principios de la lluvia de ideas están: no

realizar críticas, hasta que no agoten las ideas, se ha de crear una atmósfera de trabajo en la que nadie se sienta amenazado. Ayuda a desarrollar ideas unificadas basadas en la experiencia de un experto.

Casos de Uso:

Es una técnica bastante utilizada, en la cual se captura cada una de las funciones del sistema y en base a cada una de ellas especifica los requisitos del mismo. Cada caso de uso proporciona uno o más escenarios que indican cómo debería interactuar el sistema con el usuario. Normalmente, en los casos de usos se evita el empleo de jergas técnicas, prefiriendo en su lugar un lenguaje más cercano al usuario final.

Introspección:

Esta técnica recomienda que el ingeniero de requisitos se ponga en el lugar del cliente y trate, con algunos conocimientos del estado del negocio actual, de imaginar como desearía el cliente el Sistema y en base a estas suposiciones comenzar a recomendarle las funcionalidades que debería presentar el sistema. (MARÍA JOSEFA ESCALONA 2002), (TORRES), (GOTTHERLF PABLO 2003)

Conclusiones

Con el estudio realizado en este capítulo, se pudo comprender de una forma más detallada el tema a desarrollar, permitió escoger la metodología de desarrollo que mejor se ajusta para realizar este trabajo, así como la herramienta y el lenguaje de modelado a utilizar. Mediante el análisis de las diferentes técnicas existentes, para la captura de requisitos, se reconocieron aquellas que se podían emplear para lograr un mayor entendimiento con los clientes, de la misma forma ocurrió con el estudio de los diferentes patrones de casos de uso, permitiendo conocer su funcionalidad, para poder escoger posteriormente aquellos que cumplen con las necesidades del sistema.

Capítulo # 2 Modelo del Negocio y Requisitos.

2.1 Introducción

En este capítulo se describen los procesos del negocio, con el propósito de entender la estructura y la dinámica de la organización en la cual se va a implantar el sistema, comprender los problemas actuales de la organización e identificar las mejoras potenciales y derivar los requisitos que deberá satisfacer el sistema propuesto.

Para lograr esos propósitos, se redactan las reglas del negocio, se identifican los actores, trabajadores y procesos del negocio y se obtienen los siguientes artefactos: Diagrama de Casos de Uso del negocio (CUN), Modelo de objeto, Especificación de los CUN y Diagramas de actividad de los CUN. Además se especifican los Requerimientos Funcionales (RF) y no funcionales (RNF) que deberá cumplir el sistema.

2.2 Estado actual del negocio

El proceso de distribuir personal en la residencia, es decir, el alojamiento de todos los estudiantes y trabajadores internos de la universidad, constituye una necesidad al inicio de cada curso escolar. Para esta ubicación se tienen en cuenta una serie de parámetros como son, residencia, facultad, grupo y sexo en el caso de los estudiantes, y para los profesores y trabajadores se tiene en cuenta la categoría docente, cargo que ocupa, años de trabajos, el sexo (aunque pueden existir apartamentos mixtos en el caso de matrimonios u otras excepciones), entre otros.

Uno de los primeros pasos en este proceso es obtener la zonificación que establece La Dirección de residencia, que consiste en distribuir todos los edificios entre las 4 residencias y dentro de las residencias en las diferentes facultades. Luego en cada facultad, con el listado de todos los estudiantes y los edificios disponibles con que cuenta, el vicedecano de residencia, con la ayuda de instructoras o técnicos generales hacen la distribución de todos los estudiantes de forma manual, tratando de ubicarlos de acuerdo a los grupos docentes o grupos de proyectos. En la residencia de profesores, es responsabilidad de la directora de la residencia llevar a cabo esta tarea, con el listado obtenido por Capital Humano y teniendo en cuenta los criterios ubicación, que determina una comisión de la universidad. En esta área, el proceso es de forma más flexible, se permite muchas veces, ubicarlos por afinidad, tratando de lograr una mejor convivencia.

Dentro de este proceso se encuentra la realización de permutas o cambio de ubicación, esto se hace cuando es necesario mover a un becario de apartamento o de edificio por razones autorizadas.

Posibles procesos a automatizar:

Alojar todo el personal en residencia.

Cambiar alojamiento.

Verificar capacidades disponibles.

Verificar criterios de ubicación de trabajadores internos.

2.3 Reglas del negocio

Las siguientes reglas del negocio describen políticas que deben cumplirse o condiciones que deben satisfacerse, por lo que regulan algún aspecto del negocio.

1. No pueden haber dos residencias con el mismo número.
2. No pueden haber dos manzanas con el mismo número.
3. No pueden haber dos edificios con el mismo número.
4. No pueden haber dos apartamentos con el mismo número.
5. No puede haber dos residentes con la misma identificación.
6. Cada facultad estará asignada a una dirección de residencia.
7. Un becario no puede estar registrado en más de un apartamento.
8. Un apartamento no puede tener más de la cantidad de becarios establecida como capacidad.
9. Estudiantes y trabajadores no pueden convivir en un mismo apartamento.
10. No pueden convivir estudiantes de ambos sexos (M, F) en un mismo apartamento.
11. No pueden haber trabajadores de ambos sexos (M, F) en un mismo apartamento, salvo excepciones o matrimonio.
12. No se pueden efectuar permutas entre trabajadores y estudiantes.
13. No se pueden efectuar permutas entre becarios de diferente sexo.
14. Debe existir un solo Director por residencia.

2.4 Actores del negocio

Un actor del negocio es cualquier individuo, grupo, entidad, organización, máquina o sistema de información externos; con los que el negocio interactúa y lo que se modela como actor, es el rol que se

juega cuando se interactúa con el negocio, un actor del negocio representa un tipo particular de usuario del negocio más que un usuario físico, ya que varios usuarios físicos pueden realizar el mismo papel en relación al negocio, o sea, ser instancias de un mismo actor. (*Conferencia de Ingeniería de Software. Fase de Inicio. Modelo del Negocio. 2005-2006*)

Tabla 1. Actores del negocio

ACTOR DEL NEGOCIO	DESCRIPCION
Interesados	Persona que realiza la solicitud de cambio de ubicación de un becario, puede ser, desde un estudiante o profesor hasta un directivo de residencia, entiéndase, Instructora, Psicopedagoga, Técnico General, Director de Residencia, Decano o Vicedecano.
Vicerrector de Residencia	Persona que tiene a su cargo la máxima dirección de la residencia universitaria en su totalidad.

2.5 Trabajadores del negocio

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

Tabla 2. Trabajadores del negocio

TRABAJADORES DEL NEGOCIO	DESCRIPCION
Director de Residencia	Persona que tiene a su cargo la dirección de una de las residencias.
Vicedecano de Residencia	Persona que atiende a los becarios de una facultad en representación del decano.

2.6 Diagrama de Casos de Uso del Negocios (CUN)

El diagrama que se muestra a continuación representa gráficamente los procesos del negocio y su interacción con los actores del mismo.

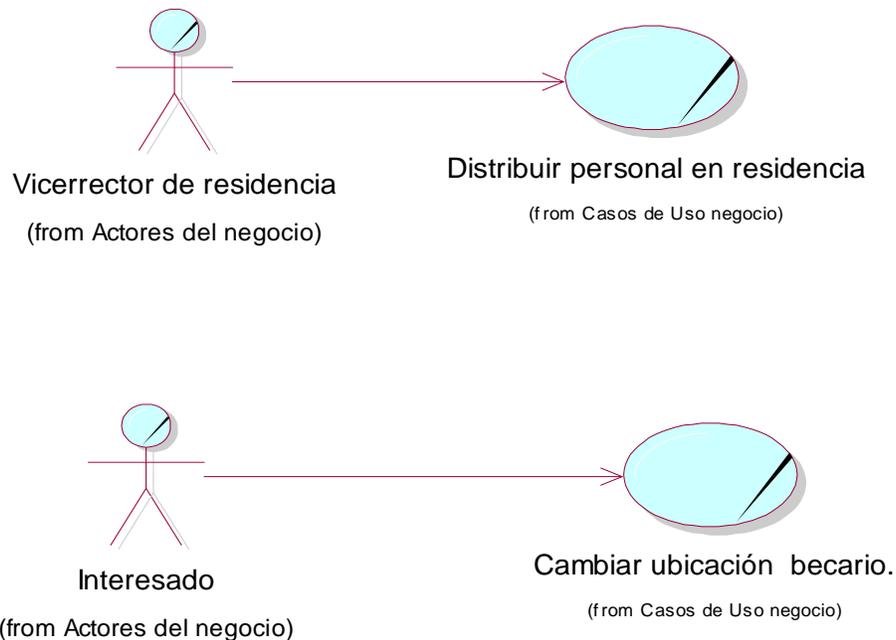


Figura 1 Diagrama de CUN

2.7 Descripción Casos de Uso del Negocio (CUN).

La realización de un CUN muestra cómo colaboran los trabajadores y entidades del negocio para ejecutar el proceso. Cada realización se puede documentar utilizando las descripciones textuales, para describir el proceso de negocio y dar información necesaria para los flujos de trabajo que se ejecutan posteriormente.

Tabla 3. Descripción del CU Distribuir personal en residencia.

Caso de Uso	Distribuir personal en residencia	
Actores	Vicerrector de residencia.	
Trabajadores	Vicedecano de residencia, Director de Residencia	
Resumen	El CUN se inicia cuando el Vicerrector de Residencia solicita que se le de alojamiento a todos los estudiantes y trabajadores en la residencia, el Vicedecano y el Director de la residencia de profesores atienden la solicitud y llevan a cabo el proceso, teniendo como resultado un listado con la ubicación de cada becario y uno con las capacidades que le quedaron disponibles.	
Precondiciones	Que este hecha la zonificación	
Flujo Normal de Eventos		
Sección "Principal"		
Acción del Actor	Respuesta del Sistema	
1 El Vicerrector solicita que se le de alojamiento a todos los estudiantes y trabajadores internos.	1.1 El Vicedecano recibe la solicitud y atiende la parte de los estudiantes. 1.1.1 El Vicedecano puntualiza la zonificación correspondiente a su facultad. 1.1.2 El Vicedecano verifica si son suficientes las capacidades de los edificios correspondientes a su facultad de acuerdo a la matrícula de la misma. 1.1.3 El Vicedecano procede a la ubicación manual de todos los estudiantes atendiendo a los criterios de ubicación 1.1.4 El Vicedecano crea un listado con la ubicación de cada estudiante y otro con las capacidades que le quedaron disponible	
	1.2 El Director de la Residencia de trabajadores también recibe la solicitud.	

	<p>1.2.1 El Director de la Residencia verifica:</p> <ul style="list-style-type: none"> - El listado de los trabajadores con sus datos y categoría. - Los criterios de ubicación. - Las capacidades, dada la zonificación <p>1.2.2 En coordinación con estas verificaciones procede a la ubicación manual de todos los trabajadores internos.</p> <p>1.2.3 El Director de la Residencia crea un listado con la ubicación de cada trabajador y otro con las capacidades que le quedaron disponible.</p>
Flujo alterno de Eventos	
Acción del Actor	Respuesta del Sistema
	<p>1.1.2a De no ser suficiente las capacidades disponibles en la facultad, se coordina con la dirección de residencia la ubicación de estos estudiantes en capacidades disponibles de otras facultades, dentro de la propia dirección de residencia.</p>

Tabla 4. Descripción del CU Cambiar ubicación de un becario.

Caso de Uso:	Cambiar ubicación de un becario.
Actores:	Interesado
Trabajadores:	Director de Residencia, Vicedecano de Residencia.
Resumen:	El CUN se inicia cuando el Interesado solicita el cambio de ubicación de un becario y presenta los motivos por lo cual lo solicita, los mismos son analizado y de acuerdo a su objetividad y a las capacidades disponibles, la solicitud del mismo es atendida o rechazada.
Precondiciones:	Que posea una ubicación previa.

Flujo Normal de Eventos	
Sección “Principal”	
Acción del Actor	Respuesta del Negocio
<p>1. El interesado solicita el cambio de ubicación de un becario y presenta los motivos.</p>	<p>1.1. El Director de Residencia recibe la solicitud y los motivos de la misma.</p> <p>1.2. El Director de Residencia verifica si el becario a cambiar es estudiante o trabajador.</p> <p>1.3. Si es estudiante el Director de Residencia verifican si es interno (dentro de la propia facultad o residencia) o externo (entre residencias) el cambio.</p> <p>1.4. Si es interno, el Vicedecano analiza los motivos de la solicitud.</p> <p>1.5. Si son objetivos los motivos de la solicitud, el Vicedecano verifica las capacidades disponibles en dependencia del sexo.</p> <p>1.6. Si existen capacidades disponibles el Vicedecano ejecuta el cambio.</p> <p>1.7. El Vicedecano informa al solicitante la nueva ubicación del becario.</p> <p>1.8. El Vicedecano actualiza el listado de las ubicaciones.</p>
Flujos Alternos	
Acción del Actor	Respuesta del Negocio
	<p>1.3a Si es trabajador, el Director de Residencia analiza si son objetivos los motivos de la solicitud de cambio.</p>

	<p>1.4a Si son objetivos los motivos, el Director de Residencia verifica si existen capacidades disponibles.</p> <p>15a Si existen capacidades disponibles el Director de Residencia realiza el cambio.</p> <p>1.6a El Director de Residencia informa al solicitante la nueva ubicación del becario.</p> <p>1.7a El Director de Residencia actualiza el listado de las ubicaciones.</p>
Flujos Alternos	
Acción del Actor	Respuesta del Negocio
	<p>1.4b Si la solicitud para el estudiante es externa el Director de Residencia es quien verifica los motivos y realiza además las acciones 1.5, la 1.6, la 1.7 y la 1.8</p>
Flujos Alternos	
Acción del Actor	Respuesta del Negocio
	<p>1.5b Si no son objetivos los motivos de la solicitud, el cambio es rechazado.</p> <p><i>Nota:</i> Esto se cumple para estudiantes y trabajadores</p>
Flujos Alternos	
Acción del Actor	Respuesta del Negocio
	<p>1.6b Si no existen capacidades disponibles, el cambio es rechazado.</p> <p><i>Nota:</i> Esto se cumple para estudiantes y trabajadores</p>
Poscondiciones	

2.8 Diagramas de Actividades

Los casos de uso del negocio consisten en secuencias de actividades que, en conjunto, producen algo para el actor del negocio. El proceso consiste de un flujo básico de una o más alternativas de flujos. La estructura del flujo se describe gráficamente con la ayuda de un diagrama de actividad y describe un proceso que explora el orden de las tareas o actividades que logran los objetivos del negocio.

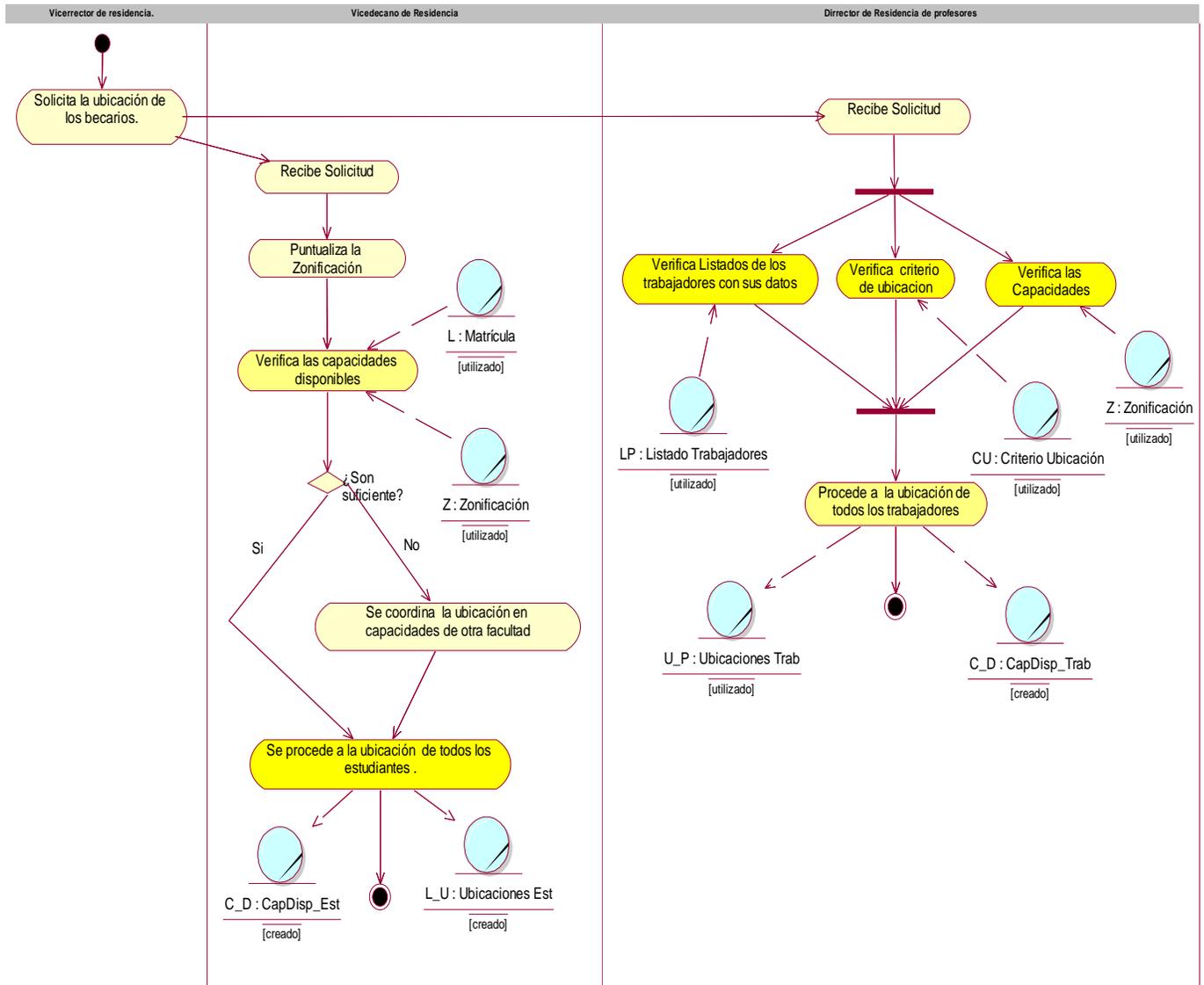


Figura 2. Diagrama de actividad del CUN Distribuir personal en residencia.

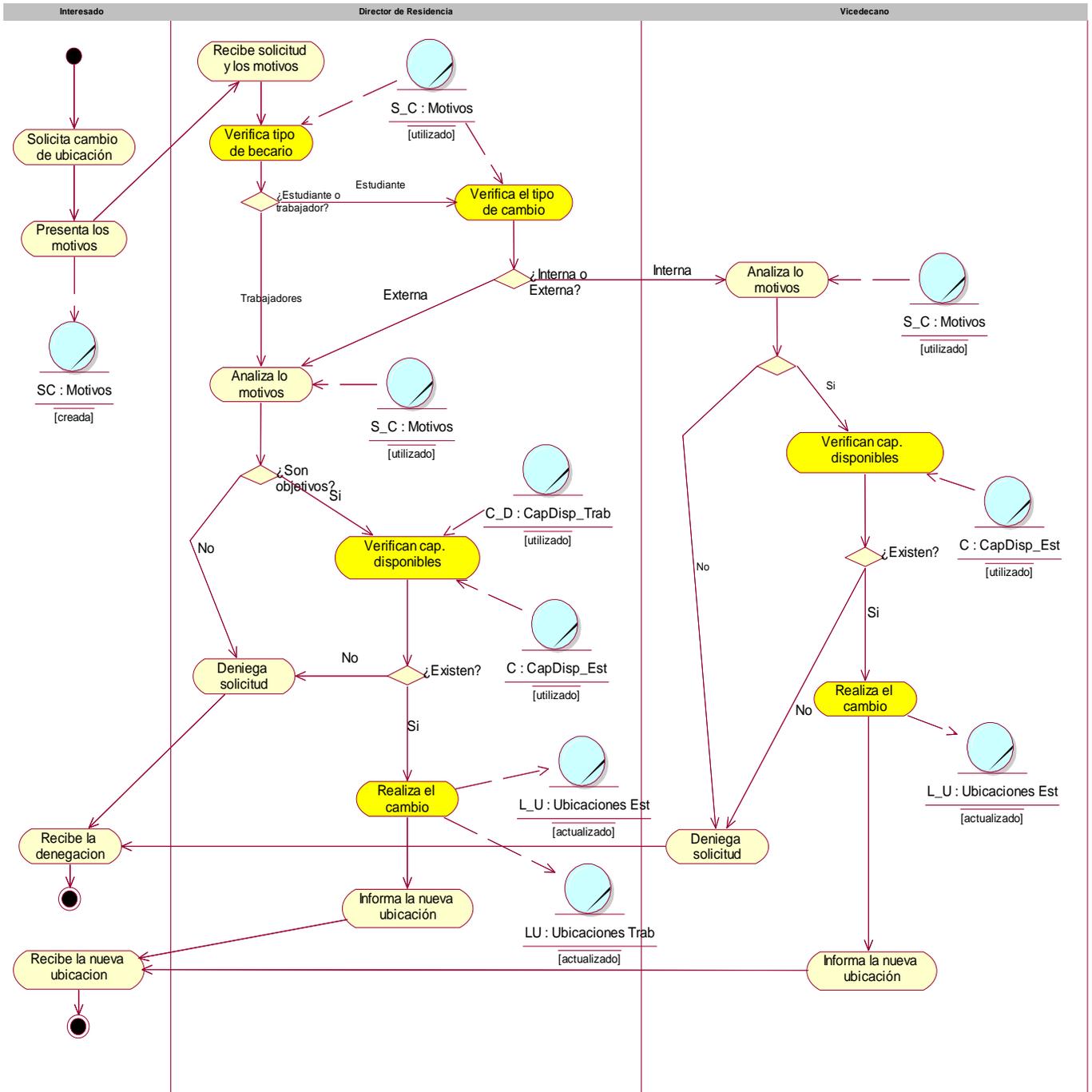


Figura 3. Diagrama de actividad del CUN Cambiar ubicación.

2.9 Modelo de objeto

El siguiente modelo de objeto describe cómo colaboran los trabajadores y las entidades del negocio dentro del flujo de trabajo del proceso de negocio.

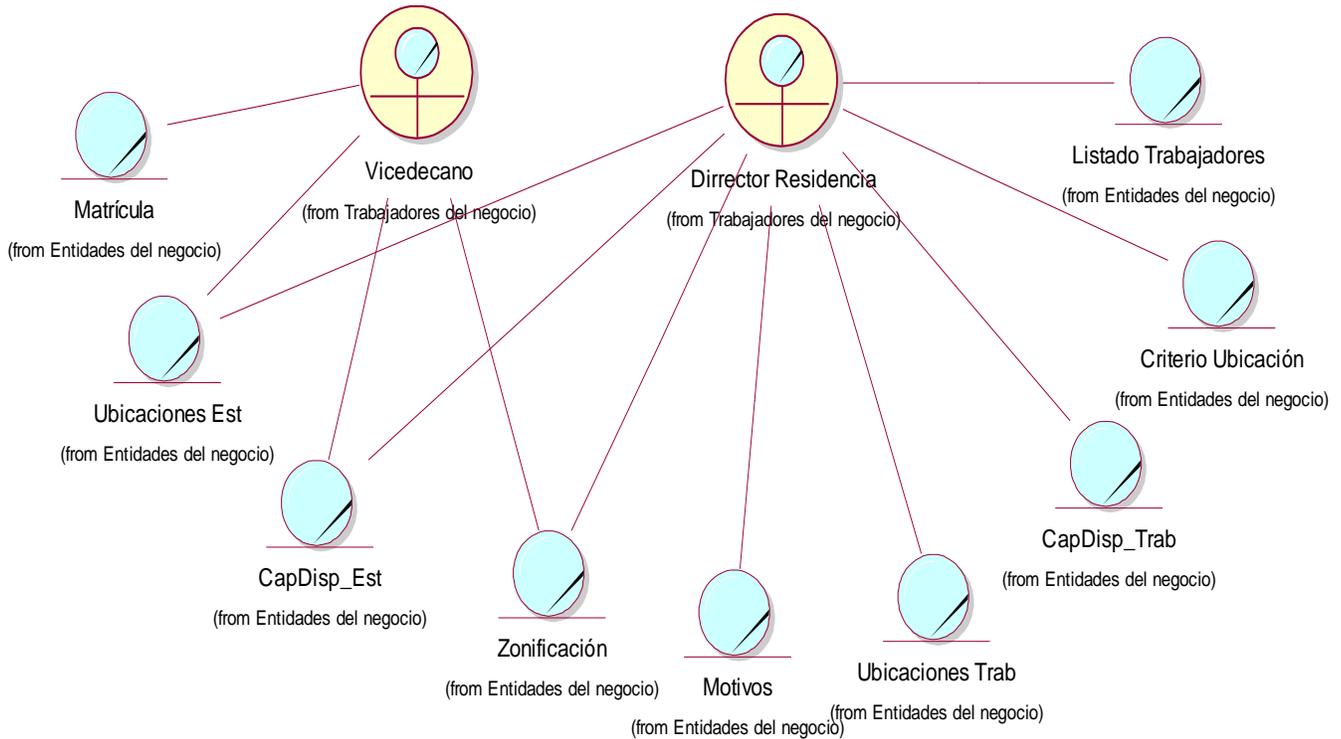


Figura 4 Modelo de Objeto

2.10 Requisitos capturados.

De las técnicas para la elicitación de requisitos explicadas en el capítulo 1, se usaron en este trabajo la lluvia de ideas y las entrevistas y dentro de ellas las entrevistas abiertas y las discusiones.

Entrevistas:

Abiertas: Se le hicieron entrevistas a los clientes, mediante preguntas que se formularon en la conversación que se sostuvo, el cliente pudo responder abiertamente, se realizaron varias entrevistas para lograr entender lo que realmente se necesitaba, llegando a un acuerdo entre clientes y analistas.

Discusiones: Por el nombre de esta técnica parece que no es muy apropiada usarla, pero el nombre no significa que se llegue a una pelea entre clientes y analistas sino que lo que se discuten son las ideas para entender la problemática, con esta técnica se entendieron una buena parte de los requisitos del sistema.

Lluvia de ideas: Para llevar a cabo esta técnica se realizaron reuniones de grupo con los clientes, se debatieron las ideas de todos los participantes en un ambiente relajado. Cada miembro de la reunión interpretaba las ideas de otro, aportándole más, o explicando el por que no era correcta, desde su punto de vista. Mediante esta técnica se logró que todos opinaran y que se desarrollaran ideas.

2.10.1 Requisitos Funcionales

Los requerimientos funcionales definen las funciones que el sistema será capaz de realizar. Describen las transformaciones que el sistema realiza sobre las entradas para producir salidas.

R1. Autenticar usuario.

R2. Gestionar roles.

2.1 Adicionar usuarios y asignarle un rol de los que existirán y determinar sobre que Residencia o Facultad ejercerá esos privilegios.

2.2 Modificar a los usuarios del sistema el rol que tiene asignados o el área sobre la cual puede ejercer.

2.3 Eliminar usuarios

R3. Gestionar Residencia

3.1 Adicionar residencia.

3.2 Modificar los datos de la residencia

3.3 Eliminar residencia.

R4. Gestionar Edificio

4.1 Adicionar edificios a una de las residencias existentes, especificándole datos como el número, la cantidad de pisos, de escaleras, de apartamentos por escalera, facultad a la que pertenece y la manzana en la que se encuentra.

4.2 Modificar a un edificio existente algunos de sus datos.

4.3 Eliminar un edificio existente de la base de datos.

R5. Gestionar Apartamento.

5.1 Adicionar un Apartamento a un edificio determinado y especificarle datos como la cantidad de cuartos que tiene, cantidad máxima de personas que se pueden alojar en el mismo, si es habitable o no, y de no serlo una nota de por qué.

5.2 Modificar a un Apartamento existente cualquiera de sus datos.

5.3 Eliminar Apartamento existente de la BD.

R6. Gestionar Alojamiento Manual.

Dar o modificar ubicación, a todos los becarios de la universidad de forma manual en el sistema.

R7. Gestionar Alojamiento Automático.

Debe permitir alojar a todos los estudiantes y trabajadores internos de forma automática, dado algunos criterios de ubicación.

R8. Gestionar Permutas

8.1 Gestionar Permutas dobles

Debe permitir intercambiar el alojamiento de becarios de dos apartamentos, siempre y cuando sean del mismo sexo o de la misma categoría, entiéndase, ambos estudiantes o ambos profesores o trabajadores.

8.2 Gestionar Permutas triples.

Debe permitir realizar un intercambio triple de alojamiento, bajo las mismas condiciones que la permuta doble.

8.3 Gestionar permuta por desperfecto.

Debe permitir realizar un cambio de ubicación de un apartamento completo hacia otras capacidades disponibles.

R9. Gestionar Baja automáticamente de estudiantes

El sistema debe dar automáticamente baja de la BD de beca a todos los estudiantes que se le den de baja en akademos, notificar al administrador del sistema de la acción y actualizar su capacidad de alojamiento como disponible.

R10. Gestionar Baja manual de trabajadores.

Debe permitir eliminar un trabajador de la BD de beca, actualizando automáticamente su alojamiento como disponible.

R11. Gestionar criterios de ubicación de trabajadores.

Debe permitir crear criterios de ubicación para todos los trabajadores internos y basados en las categorías, definir las condiciones de convivencias que deben tener. Modificar dichos criterios o eliminarlos.

R12. Gestionar traslado de facultad.

Debe permitir conocer los estudiantes que se trasladen de facultad, para darle de baja en la facultad que se encontraba y facilitarle un nuevo alojamiento en la otra facultad.

R13. Obtener historial.

Visualizar e imprimir un historial mostrando de un estudiante todas las ubicaciones que a tenido en un periodo de tiempo determinado, y dado un apartamento todos los estudiantes que han vivido ahí, ambos con fecha de alta y baja.

R14. Obtener Reportes estadísticos.

Todos estos reportes pueden ser visualizados e impresos.

14.1. Obtener Reporte general

14.2 Obtener Reporte de residencia

14.3 Obtener Reporte por facultad.

14.4 Obtener Reporte de edificio

14.5 Obtener Reporte de personas.

R15. Obtener reporte de las bajas y licencias universitarias y traslado de facultad.

Se pueden visualizar e imprimir estos reportes.

R16. Obtener reportes Estadísticos de las bajas, licencias o traslado de facultad.

Se puede visualizar e imprimir estos reportes.

R17. Obtener Listas.

17.1 Visualizar Listas de estudiantes

17.2 Imprimir listas de estudiantes.

R18. Buscar

Permite buscar en el sistema una persona, un usuario del mismo, un edificio o un apartamento y mostrarlo con sus datos.

2.10.2 Requisitos no Funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. En muchos casos los requerimientos no funcionales son fundamentales en el éxito del producto. Normalmente están vinculados a requerimientos funcionales, una vez que se conozca lo que el sistema debe hacer se puede determinar cómo ha de comportarse, qué cualidades debe tener o cuán rápido o grande debe ser.

Este módulo, junto con el resto de los módulos de la aplicación de residencia, formará parte de una aplicación Web, que estará instalada en un servidor del nodo central. A la misma tendrán acceso los usuarios utilizando un navegador Web.

Requerimientos de Software

- Se necesita tener un servidor Zope. Debido a que este puede ejecutarse en los sistemas operativos más difundidas, no se especifica un sistema operativo obligatorio para el mismo (Linux, Windows NT/2000/XP, Solaris, FreeBSD, NetBSD, OpenBSD, y Mac OS X, entre otros.)
- Se necesita además un servidor de Base de Datos que sea software libre, recomendado PostgreSQL,
- Para el cliente, se requiere para la utilización del sistema, disponer de un navegador Web, puede ser el Internet Explorer, Mozilla Firefox o cualquier otro navegador que soporte javascript y CSS 2.0 (hoja de estilo en cascada) o superior.

Restricciones en el diseño y la implementación

- El sistema se implementará en Zope/Plone, utilizando, Python como lenguaje de programación.
- El servidor Web Zope se encontrará ubicado detrás de un servidor Apache, el cual garantizará las conexiones seguras.
- El sistema utilizará un gestor de Base de Datos PostgreSQL.

Requerimientos de Hardware

- El sistema va a estar ubicado en un servidor Web, y la dirección de informatización debe garantizar la capacidad de cómputo necesaria para que la aplicación se ejecute sin problemas.
- Todos los usuarios deben contar con acceso a la red local de la Universidad, para poder acceder al servidor.

Requerimientos de apariencia o interfaz externa

- La Interfaz será amigable, lo más homogénea posible.
- Diseño sencillo, permitiendo una buena utilización del sistema.

Requerimientos de Seguridad

- Se establecerá la existencia de distintos roles que establezcan las acciones que pueden realizar cada usuario, previendo que la información sea manejada solamente por personas autorizadas, garantizando la integridad y confiabilidad de la misma.
- El sistema debe comunicarse usando un protocolo seguro (https), durante la autenticación del usuario, para evitar que las contraseñas viajen por la red en texto plano.

Requerimientos de Soporte

- Garantía de instalación, prueba y mantenimiento del sistema, además de un breve entrenamiento a los usuarios finales.
- Debe entregarse un manual de usuario, que sirva de tutorial a los clientes para un mejor entendimiento del sistema.

Conclusiones:

La correcta descripción del estado actual del negocio permitió, identificar los principales procesos del mismo, describir detalladamente los correspondientes casos de uso y modelar los respectivos diagramas de actividades y el modelo de objeto.

Los requisitos capturados a partir del empleo de las técnicas mencionadas, proporcionan la información necesaria para el desarrollo del modelado del sistema.

Capítulo # 3 Propuesta de Solución y Análisis de los Resultados.

3.1 Introducción

Una vez especificados todos los requisitos y logrado un entendimiento con los clientes, de lo que se quiere hacer, en el presente capítulo se da paso a describir la propuesta de solución. Para hacer entendible la forma en que está concebido el sistema, se modela el diagrama de Casos de usos del Sistema (CUS), se describen de forma detallada cada uno de estos y se confeccionan los prototipos de interfaces no funcionales. Para analizar los resultados obtenidos en el trabajo, se aplican métricas de calidad y se comprueba el grado de satisfacción de los clientes.

3.2 Descripción del sistema

El sistema está concebido para que solo tengan acceso a él, personal autorizado, mediante el usuario UCI, al autenticarse, se cargarán las funcionalidades a las que tenga acceso según su rol, se tienen en cuenta tres niveles de acceso:

Usuario: El personal que desempeñe este rol solo tendrá acceso a realizar búsquedas y a visualizar una serie de reportes generados por el sistema. Los actores del sistema que tomarán este rol, por lo general son, las instructoras y las psicopedagogas.

Usuario avanzado: El personal con este rol tendrá acceso a las mismas funcionalidades que el Usuario y además se le adicionan nuevas funcionalidades como: Gestionar alojamiento manual, Gestionar alojamiento automático, Gestionar baja manual de profesores, pero, solo en su Dirección de Residencia, o sea, un Usuario avanzado de la Residencia #1 no podrá realizar estas operaciones con entidades de otra residencia. Los actores del sistema que desempeñarán este rol, por lo general son, el Vicedecano de Residencia y el Técnico General.

Administrador: El personal con este rol heredará todas las funcionalidades del Usuario y del Usuario avanzado y se le adicionan otras responsabilidades como la de crear las estructuras de las residencias, la de gestionar roles y gestionar criterios de ubicación.

Existirá dos tipos de administradores un Administrador General que tendrá acceso a todas las funcionalidades del sistema sin restricciones de área, y otro Administrador de Residencia que tendrá acceso a todas las funcionalidades pero en una sola dirección de residencia.

Los actores del sistema que desempeñarán este rol por lo general son, los Directores de Residencia y el Vicerrector de Residencia.

3.3 Actores del sistema

Tabla 5 Actores del Sistema

ACTORES DEL SISTEMA	DESCRIPCION
Instructora	Responsable del funcionamiento de la vida interna de cada edificio.
Técnico General	Se encarga de la atención del área administrativa en una facultad.
Vicedecano de Residencia	Atiende a los becarios de una facultad en representación del decano.
Director de Residencia	Tiene a su cargo la dirección de una de las Residencia Universitaria.
Vicerrector de Residencia	Tiene a su cargo la dirección de la Residencia Universitaria en su totalidad.
Akademos	Sistema externo que nos brinda servicios Web.

3.4 Patrones de Casos de Uso aplicados en este trabajo.

En el desarrollo de este trabajo se utilizaron varios Patrones de CU con el objetivo de rehusar soluciones factibles, anteriormente probadas para problemas reiterados, conocidos y estudiados, facilitando así el entendimiento del modelo de CUS propuesto.

Entre estos tenemos los patrones de:

Concreta Extensión o Inclusión

Inclusión: En el Modelo del Sistema existen varios casos de usos bases, que tienen una relación incluye, con el caso de uso de inclusión Registrar Ubicación en un historial, pues el comportamiento definido para este, se inserta explícitamente dentro del comportamiento definido, para los casos de usos bases, Gestiona alojamiento automático, Gestionar alojamiento manual, Gestionar bajas manuales de trabajadores y Gestionar bajas automáticas de estudiantes. Se hace esta partición para que el comportamiento separado del caso de uso incluido pueda reutilizarse por los cuatro casos de uso base.

Caso de uso grande: Fue utilizado para estructurar caso de usos que presentaban un gran número de acciones pues la usabilidad consiste en múltiples flujos alternativos. De aquí que se usara más específicamente dentro de este patrón, la parte de **Caminos Múltiples**. Uno de los casos de uso modelado bajo este patrón es Visualizar Reportes.

CRUD Completo: En la elicitación de requisitos, se obtuvo como una de las necesidades, brindar funcionalidades que permitieran a los usuarios poder adicionar, leer, modificar y eliminar informaciones y datos determinados, de aquí, que se emplee el patrón Administrador de información o CRUD como es más conocido, para modelar casos de casos de uso, que agruparan estas funcionalidades y darles un nombre sugerente para los clientes y futuros desarrolladores, quedando de la siguiente forma: CRUD Criterios de ubicación, CRUD Estructura, CRUD Apartamento.

3.5 Casos de Usos del Sistema

1. Autenticar usuario. (R1)
2. Gestionar roles. (R2)
3. CRUD Estructura. (R3, R4)
4. CRUD Apto. (R5)
5. CRUD Criterios de ubicación de profesores. (R11)
6. Gestionar Alojamiento Manual (R6, R8)
7. Gestionar Alojamiento Automático (R7)
8. Gestionar Bajas automática de estudiantes. (R9, R12)
9. Gestionar Bajas manual de trabajadores (R10)
10. Registrar ubicaciones en un historial. (R13)
11. Visualizar Reportes. (R13, R14,R15, R16,R17)
12. Buscar. (R18)

3.6 Diagrama de Casos de Uso del Sistema (DCUS)

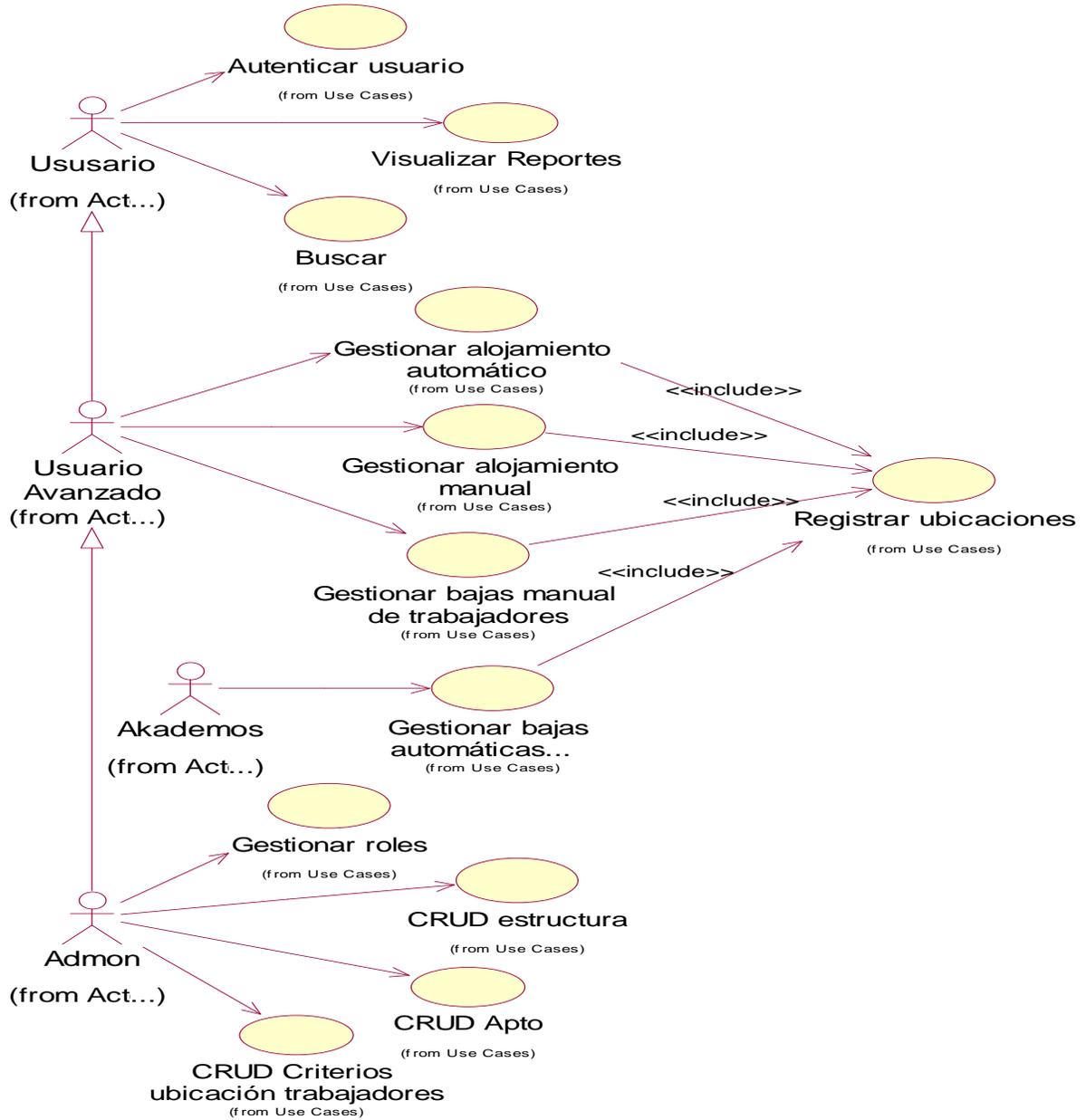
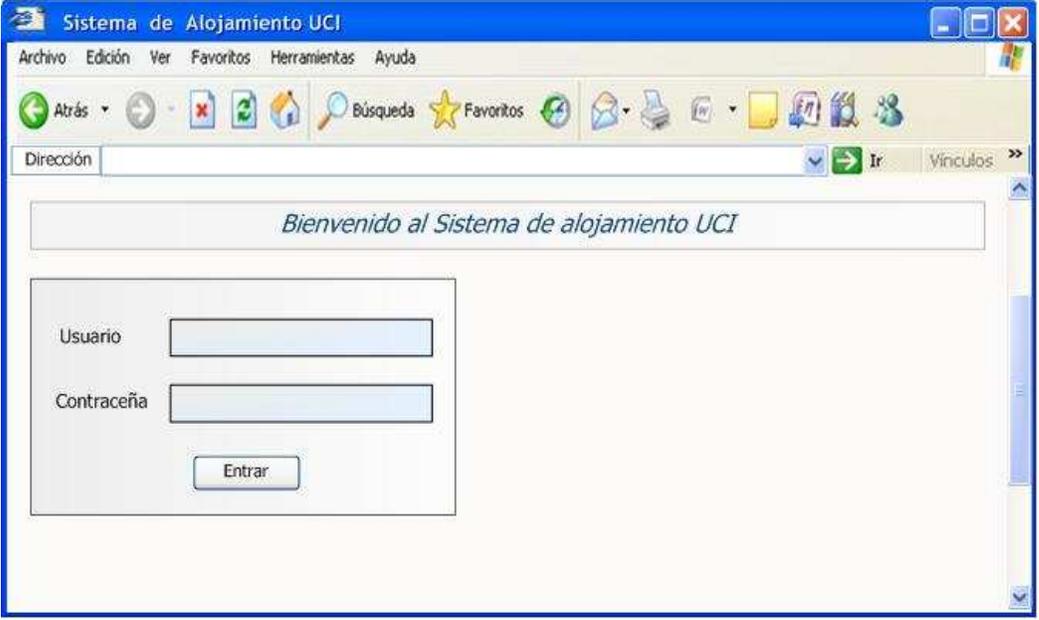


Figura 5 Diagrama de CUS

3.7 Especificación de los Casos de Uso.

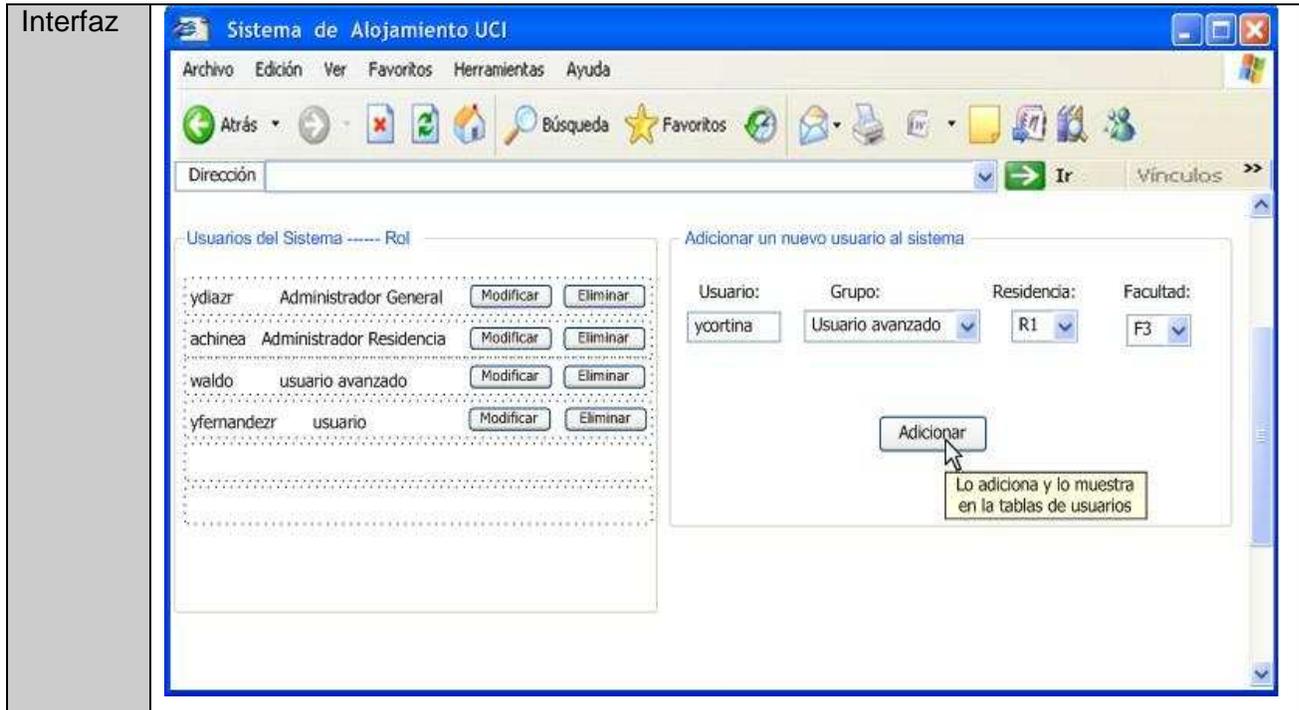
Tabla 6. Especificación del CU Autenticar usuario

Caso de Uso	Autenticar usuario.
Actores	Usuario
Propósito	Que solo tenga acceso al sistema el personal autorizado.
Resumen	El CUS se inicia cuando algún usuario decide autenticarse, el sistema muestra una interfaz para que introduzca usuario y contraseña, el usuario entra los datos, el sistema los valida y de ser correctos permite el acceso a al sistema, cargando los servicios a los cuales tiene acceso según el rol asignado a su usuario, si no muestra un mensaje de error y da la opción de intentarlo nuevamente.
Referencia	R1
Precondiciones	Que exista conexión en el servidor.
Poscondiciones	Permitir el acceso al sistema, con los privilegios de su rol.
Prioridad	Primario
Descripción	
Interfaz	

Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El CU se inicia cuando algún usuario decide autenticarse e introduce usuario y contraseña.	1.1 El sistema valida los datos, permite el acceso al sistema, cargando los servicios asociados al rol que tenga asignado dicho usuario y termina el CU.
Flujo alterno de Eventos	
Línea 1.1a Datos no valido	
Acción del Actor	Respuesta del Sistema
	1.1a.1 El sistema muestra un mensaje de error en la ventana de autenticación, da la posibilidad de regresar al paso 1.1 del flujo normal de eventos o terminar el CU.

Tabla 7 Especificación del CU Gestionar roles _ usuarios

Caso de Uso	Gestionar roles _ usuarios.
Actores	Administrador
Propósito	Permitir al administrador del sistema, gestionar usuarios en el sistema y asignarle niveles de accesos.
Resumen	El CU se inicia cuando el administrador decide gestionar roles a usuario. El sistema permite al administrador adicionar nuevos Usuarios, determinar que rol jugarán los mismos en el sistema y determinar sobre que residencia o facultad ejercerán sus privilegios, así como modificarle o eliminar estos datos a un usuario determinado.
Referencia	R2
Precondiciones	Que los niveles de acceso ya estén predeterminados, con las funcionalidades a la que tiene acceso cada uno.
Poscondiciones	
Prioridad	Primario
Descripción	



Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
1. El CU se inicia cuando el administrador decide gestionar roles	1.2 El sistema muestra una interfaz con todos los usuarios del sistema y le da la opción de: 1.2.1 Adicionar usuario. 1.2.2 Modificar rol de un usuario 3.2.3 Eliminar un usuario.

Sección: Adicionar Usuario.

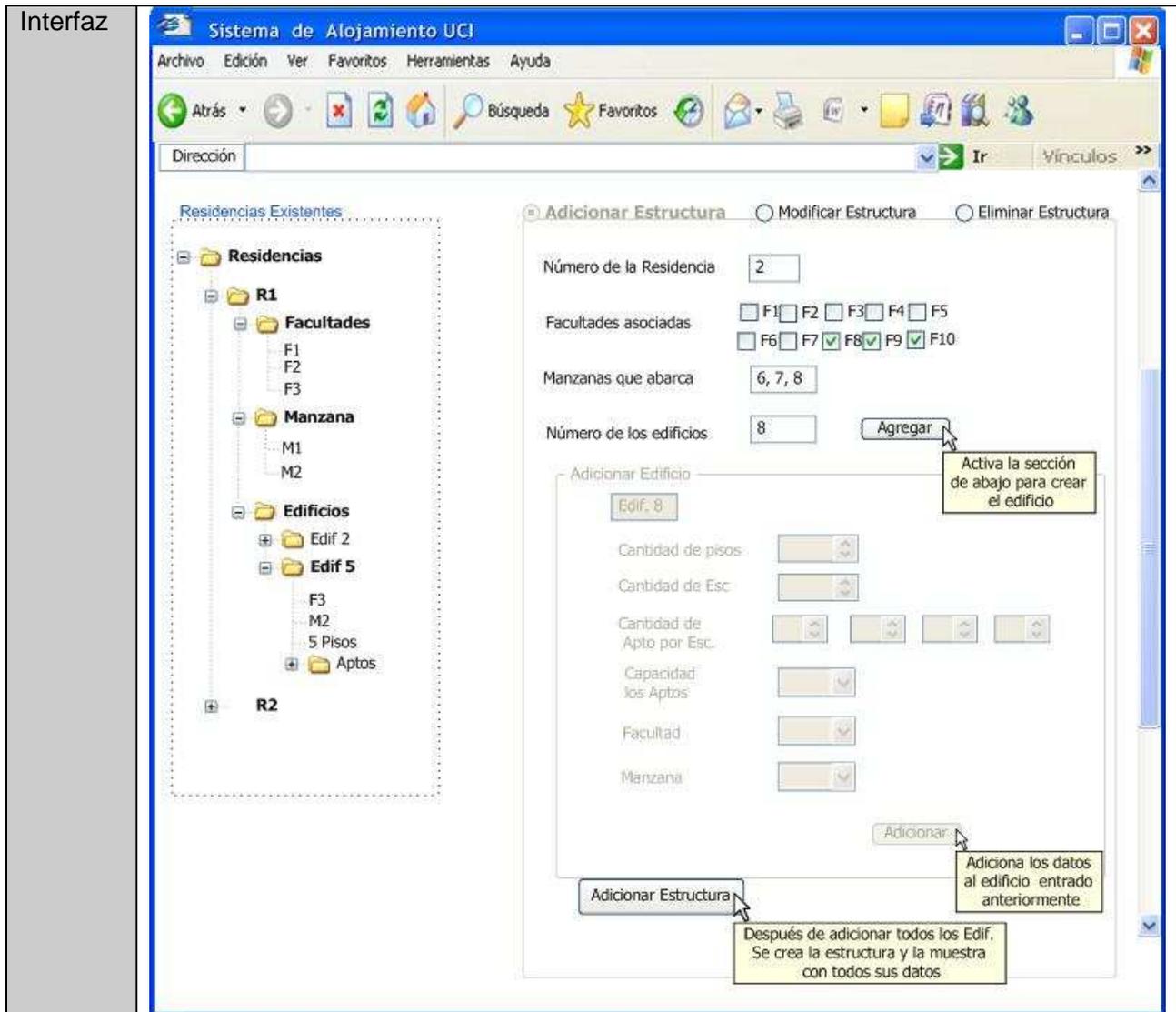
Acción del Actor	Respuesta del Sistema
2.1 El administrador escogió la opción 1.2.1 Adicionar Usuario	2.2 El sistema muestra una interfaz para que entre el usuario UCI y seleccione uno de los roles siguiente: Administrador general. Administrador de residencia. Usuario avanzado. Usuario.
2.3 El administrador entra el usuario y	2.4 El sistema agrega el nuevo usuario con

selecciona el rol Administrador General	privilegios sobre todo el sistema y termina el CU.
Flujo alternativo de Eventos	
Línea 2.3a Selecciona Rol Administrador de Residencia.	
Acción del Actor	Respuesta del Sistema
2.3a.1 El administrador entra el usuario y selecciona el rol Administrador de Residencia.	2.4a.2 El sistema muestra una interfaz para que se seleccione una residencia de las existentes.
2.5a.3 El administrador selecciona la residencia.	2.6a.4 El sistema agrega un nuevo usuario, con privilegios solo en su residencia y termina el CU.
Flujo alternos de Eventos	
Línea 2.3b Selecciona Rol Usuario Avanzado	
Acción del Actor	Respuesta del Sistema
2.3b.1 El administrador entra el usuario y selecciona el rol Usuario Avanzado o Usuario restringido.	2.4b.2 El sistema muestra una interfaz para que se seleccione una residencia y una facultad de las asociadas a la residencia.
2.5b.3 El administrador selecciona los datos	2.6b.4 El sistema agrega un nuevo usuario, con privilegios solo en su facultad y termina el CU.
Sección: Modificar rol de un usuario	
Acción del Actor	Respuesta del Sistema
3.2 El administrador escogió la opción 1.2.2 modificar el rol de un usuario y selecciona el usuario que desea modificar	3.3. El sistema muestra una interfaz para que se asigne un nuevo rol a ese usuario.
3.4 El administrador selecciona el nuevo rol y otros datos, si fuesen necesario en dependencia del rol.	3.5 El sistema actualiza los datos y termina el CU.
Sección: Eliminar Usuario	
Acción del Actor	Respuesta del Sistema
4.1 Si el administrador decide escoger la opción 2.1.3 eliminar usuario, selecciona	4.2 El sistema muestra una interfaz para validar si el administrador está seguro de la acción que va a

el que desea eliminar.	realizar.
4.3 Si el administrador acepta.	4.4 El sistema elimina el usuario de la BD.
Flujo alternos de Eventos	
Línea 4.3a Cancela la acción	
Acción del Actor	Respuesta del Sistema
4.3a.1 El administrador cancela la acción.	4.4a.2 El sistema finaliza el CU sin realizar ninguna acción.

Tabla 8 Especificación del CU CRUD Estructura

Caso de Uso	CRUD Estructura
Actores	Administrador
Propósito	Permitir a los administradores del sistema Adicionar, Modificar o Eliminar una estructura en la base de datos.
Resumen	El caso de uso se inicia cuando el administrador selecciona la opción CRUD estructura. Posteriormente el sistema le brinda la posibilidad de adicionar, modificar o eliminar una estructura, la misma deberá estar compuesta por: Una residencia, a la cual están asociadas varias facultades y que tienen varios edificios, cada uno con un número determinado de apartamentos y que abarca una o varias manzanas.
Referencia	R3, R4
Prioridad	Primario
Descripción	



Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
<p>1. El caso de uso se inicia cuando el administrador decide gestionar alguna acción sobre una estructura.</p>	<p>1.1 El sistema le brinda la posibilidad de escoger el tipo de acción que quiere realizar:</p> <p>1.2 Crear estructura.</p> <p>1.3 Modificar estructura.</p> <p>1.4 Eliminar estructura.</p>

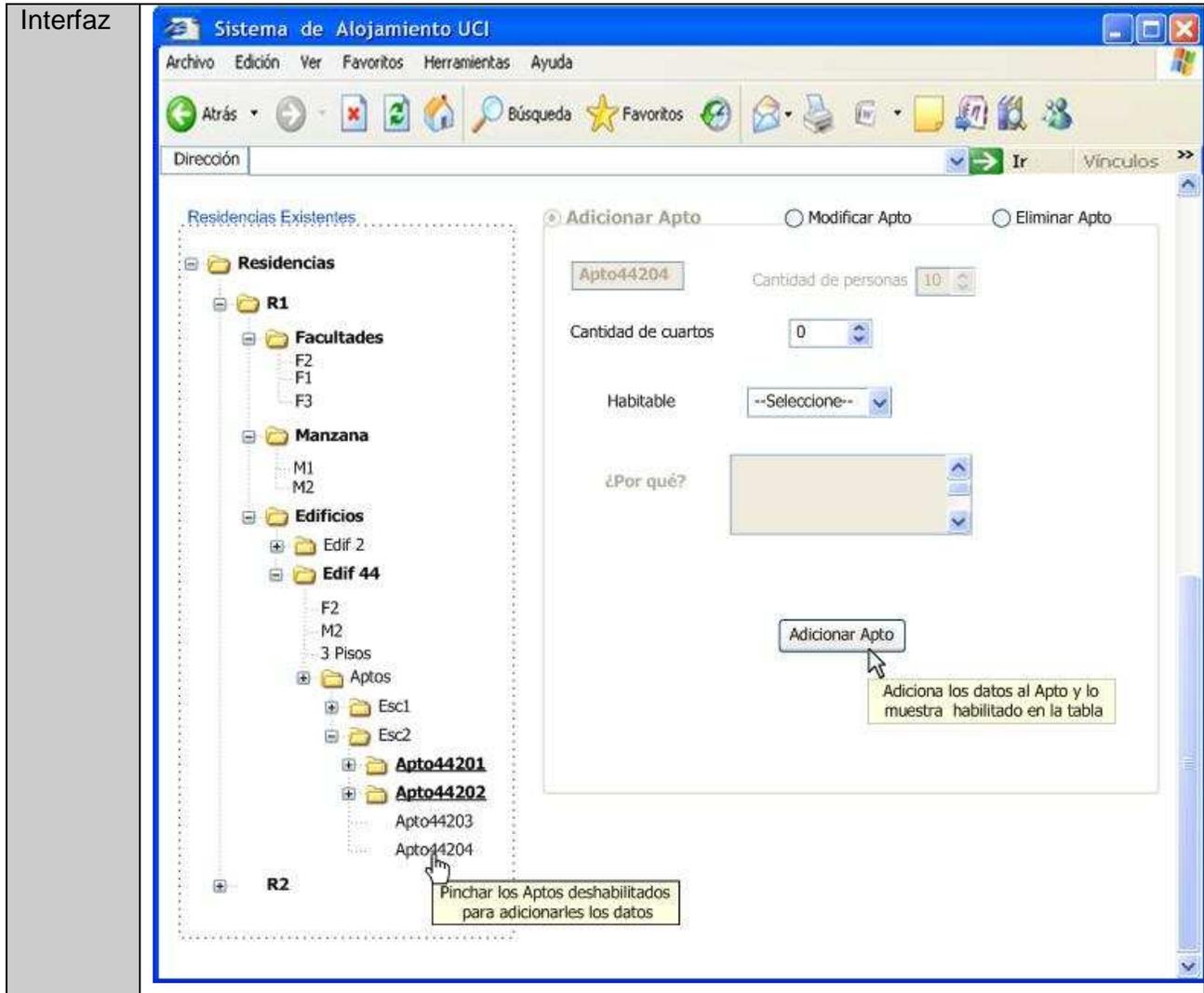
Sección: Crear Estructura.	
Acción del Actor	Respuesta del Sistema
2. El administrador escogió la acción 1.2 de crear estructura.	<p>2.1 El sistema muestra una interfaz para que se introduzcan los siguientes datos para la nueva estructura:</p> <ul style="list-style-type: none"> ▪ Número que identificará a la nueva residencia. ▪ Facultades que pertenecen a esta dirección de residencia. ▪ Manzanas que abarca la misma. ▪ Números de los edificios que la componen.
2.2 El administrador entra los datos y entra uno de los números de los edificios que componen la residencia.	<p>2.3 El sistema activa una sección para que sean introducido los datos de ese edificio, que en este caso son:</p> <ul style="list-style-type: none"> - Cantidad de pisos. - Cantidad de escaleras. - Cantidad de apartamento para cada escalera, (a cada escalera se le especifica la cantidad de apartamento) - Capacidad que tendrán todos los apartamentos del edificio. - Seleccionar la facultad a la que pertenece de las asociadas a esa residencia. <p><i>Nota:</i> (La cantidad de pisos y de apartamentos por escalera, son datos que no se almacenará, son solo para ayudar a generar los apartamentos).</p>
2.4 El administrador entra los datos del	2.5 El sistema valida los datos, agrega el nuevo

edificio.	edificio a la BD y genera automáticamente, para cada escalera, un conjuntos de números, igual a la cantidad de apartamento por escalera. (Nota: Los números de apartamentos están formados por el # del edificio, seguido del # de escalera y de un número de dos cifras consecutivo comenzando por 01. Ej: 5201, 5202,5210.)
	2.6 El sistema da la opción de regresar al paso 2.4 del flujo normal de eventos para adicionar más edificios a la estructura o crear la estructura con la cantidad de edificios entrados hasta el momento.
Flujo alternativo de Eventos	
Línea 2.5a Datos de edificios no válido.	
Acción del Actor	Respuesta del Sistema
	2.5a.1 El sistema muestra un mensaje de error y da la opción de: - Regresar al paso 2.4 o cancelar la acción.
Flujo alternativo de Eventos	
Línea 2.6a Datos de la estructura no válidos	
Acción del Actor	Respuesta del Sistema
	2.6 El sistema muestra una alerta del dato que se debe corregir y no la adiciona hasta que se modifique.
Sección: Modificar Estructura	
Acción del Actor	Respuesta del Sistema

3. El administrador escogió la acción 1.3 de modificar estructura.	3.1 El sistema muestra todas las residencias que existen hasta el momento con todos sus datos y los de cada uno, de sus edificios.
3.2 El administrador selecciona la residencia que desea modificar.	3.3 El sistema le da la opción de modificar cualquiera de sus datos, incluyendo todos los datos de cada uno de sus edificios, excepto la cantidad de pisos de un edificio y la cantidad de apartamento por escalera.
3.4 El administrador cambia los datos que desee de la misma.	3.5 El sistema actualiza los datos en la BD y termina el CU.
Sección: Eliminar Estructura	
Acción del Actor	Respuesta del Sistema
4. El administrador escogió la acción 1.4 de eliminar estructura.	4.1 El sistema le muestra todas las residencias existentes.
4.2 El administrador selecciona cual desea eliminar.	4.3 El sistema muestra un mensaje de advertencia para la acción a realizar y las implicaciones de la misma.
4.4 El administrador confirma si quiere eliminar o no la residencia.	4.5 Si el administrador da aceptar, el sistema elimina la residencia seleccionada, actualiza la BD y termina el CU.
Flujo alternativo de Eventos	
Línea 4.5a El administrador cancela la acción.	
Acción del Actor	Respuesta del Sistema
	4.5 Culmina el CUS sin ejecutar ninguna acción.

Tabla 9 Especificación del CU CRUD Apartamento (Apto)

Caso de Uso	CRUD Apto
Actores	Administrador
Propósito	Permitir a los administradores del sistema Adicionar, Modificar o Eliminar un apartamento a la BD.
Resumen	El caso de uso se inicia cuando el administrador selecciona la opción CRUD Apartamento. Posteriormente el sistema le brinda la posibilidad de adicionar un nuevo apartamento modificar o eliminar uno ya existente.
Referencia	R5
Precondiciones	Que el edificio, al cual pertenece el apartamento a tratar se haya creado con antelación.
Poscondiciones	
Prioridad	Primario
Descripción	



Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
1. El caso de uso se inicia cuando el administrador decide gestionar alguna acción sobre la estructura apartamento.	1.1 El sistema le brinda la posibilidad de escoger el tipo de acción que quiere realizar: 1.2 Adicionar apartamento. 1.3 Modificar apartamento. 1.4 Eliminar apartamento.

Sección: Adicionar apartamento.	
Acción del Actor	Respuesta del Sistema
2. El administrador escogió la acción 1.2 de adicionar apartamento.	2.1 El sistema muestra todos los edificios existentes por residencias y cada uno con la cantidad de escaleras que tiene.
2.2 El administrador escoge la escalera a la que le adicionará el nuevo apartamento.	2.3 El sistema muestra todos los números de los apartamentos de esa escalera, los apartamentos que ya les han adicionado los datos requeridos, aparecerán desactivados y los que faltan por adicionar, activados.
2.4 El administrador pincha el número del apartamento al cual desea adicionarle los datos.	2.5 El sistema muestra una interfaz para que entran los siguientes datos: -Cantidad de cuartos. -Si está habitable o no. -Si no está habitable una nota de por qué.
2.6 El administrador introduce los datos.	2.7 El sistema adiciona los datos al apartamento, y a partir de entonces, muestra el vínculo a ese número de apartamento desactivado, para esta acción.
	2.8 El sistema brinda la posibilidad de volver al paso 2.4 del flujo normal de eventos o terminar el CU.
Sección: Modificar Apartamento.	
Acción del Actor	Respuesta del Sistema
3. El administrador escogió la acción 1.3 de modificar apartamento.	3.1 El sistema muestra una interfaz con todos los apartamentos existentes organizados por edificios y residencias.
3.2 El administrador selecciona el que desea modificar.	3.3 El sistema muestra el apartamento y todos los datos del mismo, con posibilidad de cambiarlos.

3.4 El administrador cambia los datos que desee.	3.5 El sistema actualiza los datos en la BD.
Sección: Eliminar Apartamento.	
Acción del Actor	Respuesta del Sistema
4. El administrador escogió la acción 1.4 de eliminar apartamento.	4.1 El sistema muestra el apartamento con sus datos y da la posibilidad seleccionar el apartamento a eliminar.
4.2 El administrador selecciona el que desee eliminar.	4.3 El sistema muestra un mensaje de advertencia para la acción a realizar.
4.4 El administrador confirma si quiere eliminar o no el apartamento	4.5 Si el actor da aceptar el sistema elimina el apartamento de la BD.
Flujo alternativo de Eventos	
Línea4.5a El administrador cancela la acción	
Acción del Actor	Respuesta del Sistema
	4.7 El sistema finaliza el CUS sin ejecutar ninguna acción.

Tabla 10. Especificación del CU CRUD Criterios de ubicación para trabajadores

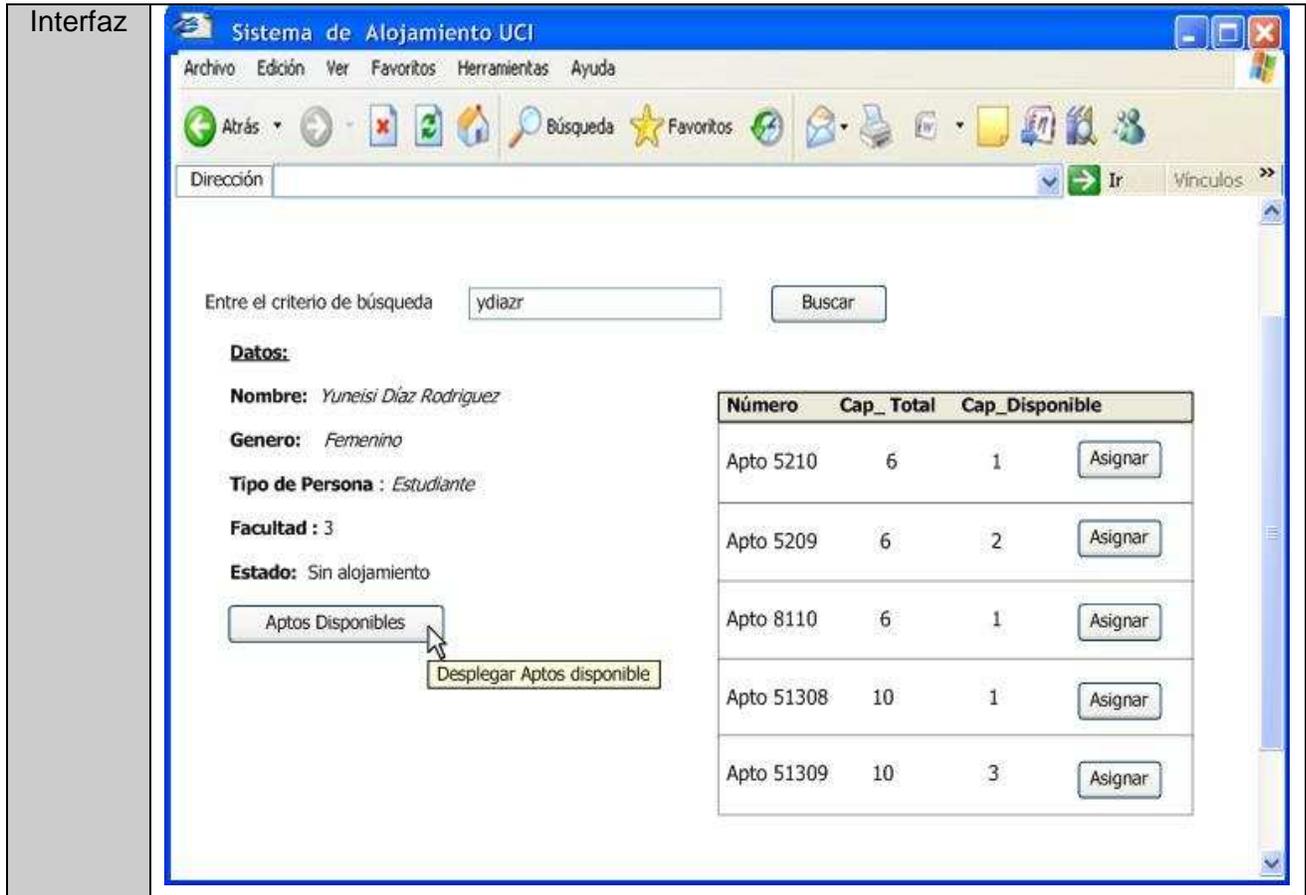
Caso de Uso	CRUD Criterios de ubicación para trabajadores
Actores	Administrador
Propósito	El propósito de este CUS es permitir a los jefes de Residencia, Adicionar Modificar o Eliminar los criterios a tener en cuenta a la hora de ubicar a los trabajadores en la residencia.
Resumen	El CUS se inicia cuando el Director de residencia ejecuta el CUS CRUD Criterios de ubicación para trabajadores. El sistema le brinda la opción de adicionar criterios de ubicación, dada la categoría de los trabajadores se definirá el tipo de apartamento en el que se alojará y sobre todo con cuantas personas compartirá el apartamento y el cuarto, además brinda la posibilidad de modificar o eliminar alguno de los criterios existentes.
Referencia	R11

Prioridad	Primario	
Descripción		
Interfaz		
Flujo Normal de Eventos		
	Acción del Actor	Respuesta del Sistema
	1.1 El CUS se inicia cuando el Administrador ejecuta el CUS CRUD Criterios de ubicación de trabajadores.	1.2 El sistema le brinda la opción de: 1.2.1 Adicionar criterio de ubicación. 1.2.2 Modificar criterio de ubicación. 1.2.3 Eliminar criterio de ubicación.
Sección: Adicionar criterio de ubicación		
	Acción del Actor	Respuesta del Sistema
	2.1 El Administrador escogió la acción 1.2.1 de adicionar.	2.2 El sistema muestra una interfaz para que se introduzcan todas las categorías de trabajadores existentes y se especifique para cada categoría: - El tipo de apartamento en que deben ser ubicados los trabajadores de esa categoría.

	<ul style="list-style-type: none"> - Con cuantas personas debe compartir el cuarto. -Con que otra categoría de trabajadores puede compartir alojamiento.
2.3 El Administrador introduce los datos.	2.4 El sistema adiciona el criterio a la BD y finaliza el CU.
Sección: Modificar criterio de ubicación	
Acción del Actor	Respuesta del Sistema
3.1 El Administrador escogió la acción 1.2.2 de modificar.	3.2 El sistema muestra una interfaz que para que seleccione de las categorías existentes la que desea modificar.
3.3 El Administrador selecciona una.	3.3 El sistema muestra dada esa categoría las condiciones en la que debe ser alojado y da la posibilidad de modificar cualquiera de sus datos.
3.4 El Administrador modifica los datos deseado.	3.5 El sistema actualiza los datos.
Sección: Eliminar criterio de ubicación	
Acción del Actor	Respuesta del Sistema
4.1 El Administrador escogió la acción 1.2.3 de eliminar.	4.2 El sistema muestra una interfaz con todos los criterios de ubicación existentes.
4.3 El Administrador selecciona el que desea eliminar.	4.4 El sistema muestra un mensaje preguntando si está seguro de la acción a realizar.
4.5 Si el Administrador confirma la acción.	4.6 El sistema elimina el criterio de ubicación.
Flujo alterno de eventos	
Línea 4.6a El administrador cancela la acción	
Acción del Actor	Respuesta del Sistema
	4.6 El sistema finaliza el CU sin ejecutar ninguna acción

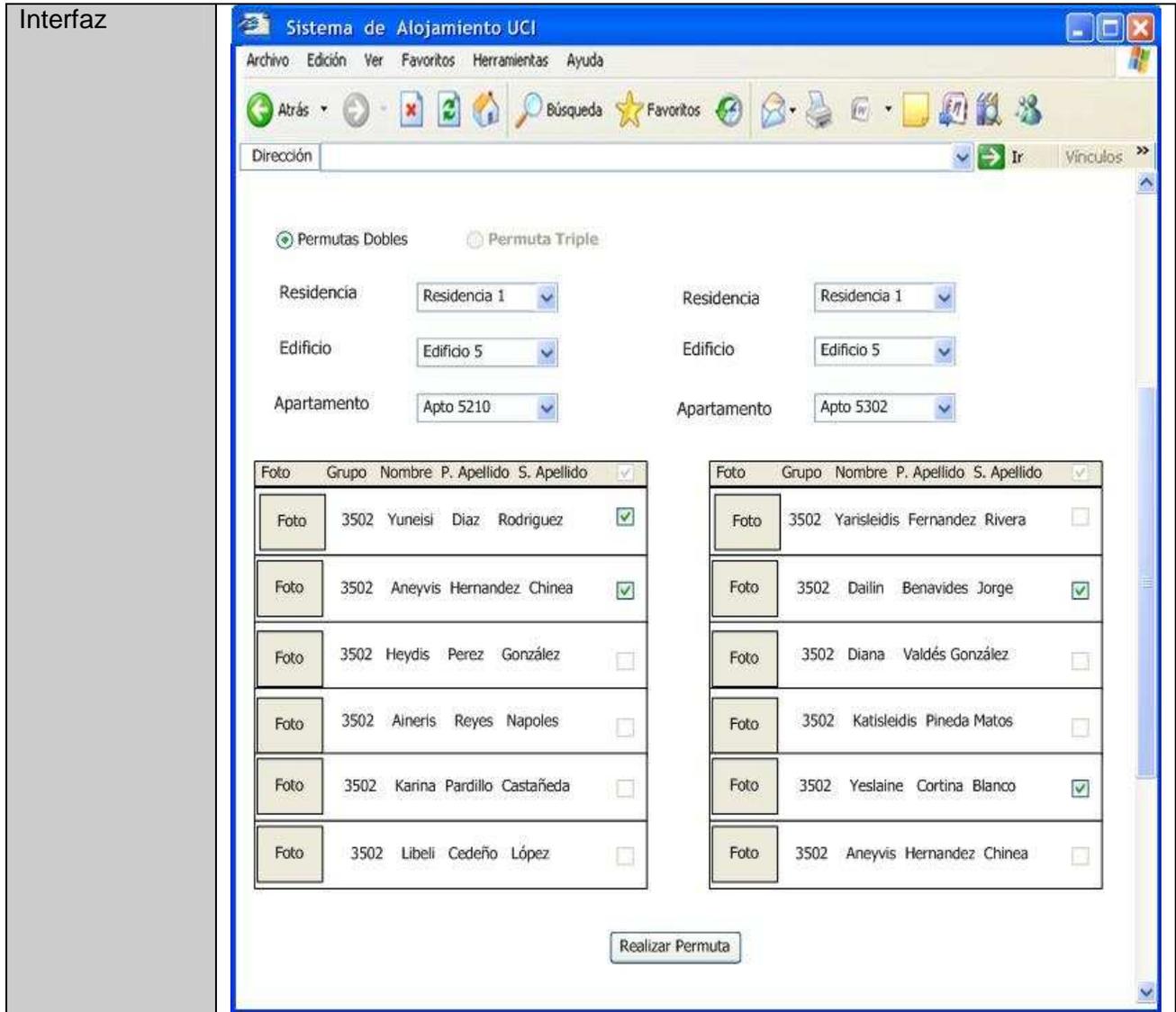
Tabla 11. Especificación del CU Gestionar alojamiento manual

Caso de Uso	Gestionar alojamiento manual	
Actores	Usuario Avanzado	
Propósito	Asignarles a los estudiantes y trabajadores internos una ubicación en la residencia de forma manual o modificársela de ya tener una.	
Resumen	El CU se inicia cuando el usuario avanzado decide gestionar alojamiento manual para algún becario, al cual le dará una ubicación, se le modificará si ya tiene una previa o se permutará con otro becario, en dependencia de las necesidades.	
Referencia	R6, R8	
Prioridad	Primario	
Descripción		
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El CU se inicia cuando el usuario avanzado decide gestionar alojamiento manual para algún becario	1.1 El sistema le brinda la posibilidad al usuario de escoger el tipo de acción que quiere realizar: 1.1 Alojarse manualmente. 1.2 Realizar permuta.	
Sección: Alojarse manualmente		



Acción del Actor	Respuesta del Sistema
2 El usuario avanzado escogió la acción 1.1 de alojar manualmente.	2.1 El sistema da la posibilidad de buscar la persona a ubicar.
2.2 El usuario avanzado introduce el criterio de búsqueda.	2.3 El sistema muestra, todas las personas con esas características y sus datos.
2.4 El usuario avanzado selecciona el usuario al cual desea asignarle ubicación.	2.5 El sistema muestra de los edificios asociados a su facultad, los apartamentos vacíos o con capacidades disponibles de acuerdo al sexo de la persona a ubicar.
2.6 El usuario avanzado selecciona el apartamento en el cual desea ubicarlo.	2.7 El sistema asigna la ubicación a la persona seleccionada y actualiza la BD.

Flujo alterno de Eventos	
Línea 2.4a La persona seleccionada ya posee una ubicación	
Acción del Actor	Respuesta del Sistema
2.4 a Si la persona seleccionada ya tiene una ubicación.	2.5a El sistema muestra un mensaje preguntando si desea modificarla: 2.5.1 Si el usuario avanzado acepta, el sistema regresa a la acción 2.6 del flujo normal de eventos. 2.5.2 Si no se finaliza el CU sin realizar ninguna operación.
Flujo alterno de Eventos	
Línea 2.5b La persona seleccionada es un trabajador interno	
Acción del Actor	Respuesta del Sistema
	2.5b Si el becario es un trabajador, el sistema muestra, de los edificios asociados a la residencia de profesores y trabajadores internos, los apartamentos que tienen capacidades libres.
Flujo alterno de Eventos	
Línea 2.5c No existen capacidades disponibles.	
	2.5c El sistema muestra un mensaje y finaliza el CU sin realizar ninguna operación.
Flujo alterno de Eventos	
Línea 2.7a Si se selecciona un apartamento totalmente vacío.	
	2.7a Si el apartamento seleccionado se encontraba vacío, el sistema además de asignar la ubicación al becario, marcará el apartamento según el sexo del mismo, o sea, solo se ubicarán en el apartamento, personas de ese mismo sexo.
Sección: Realizar Permutas	



Acción del Actor	Respuesta del Sistema
2 El usuario avanzado escogió la acción 1.2 de realizar permutas.	2.1 El sistema da la opción de seleccionar el tipo de permuta: 2.1.1 Doble. 2.1.2 Triple. 2.1.3 Mudanza.
2.2 El usuario avanzado selecciona la	2.3 El sistema muestra una interfaz para que se

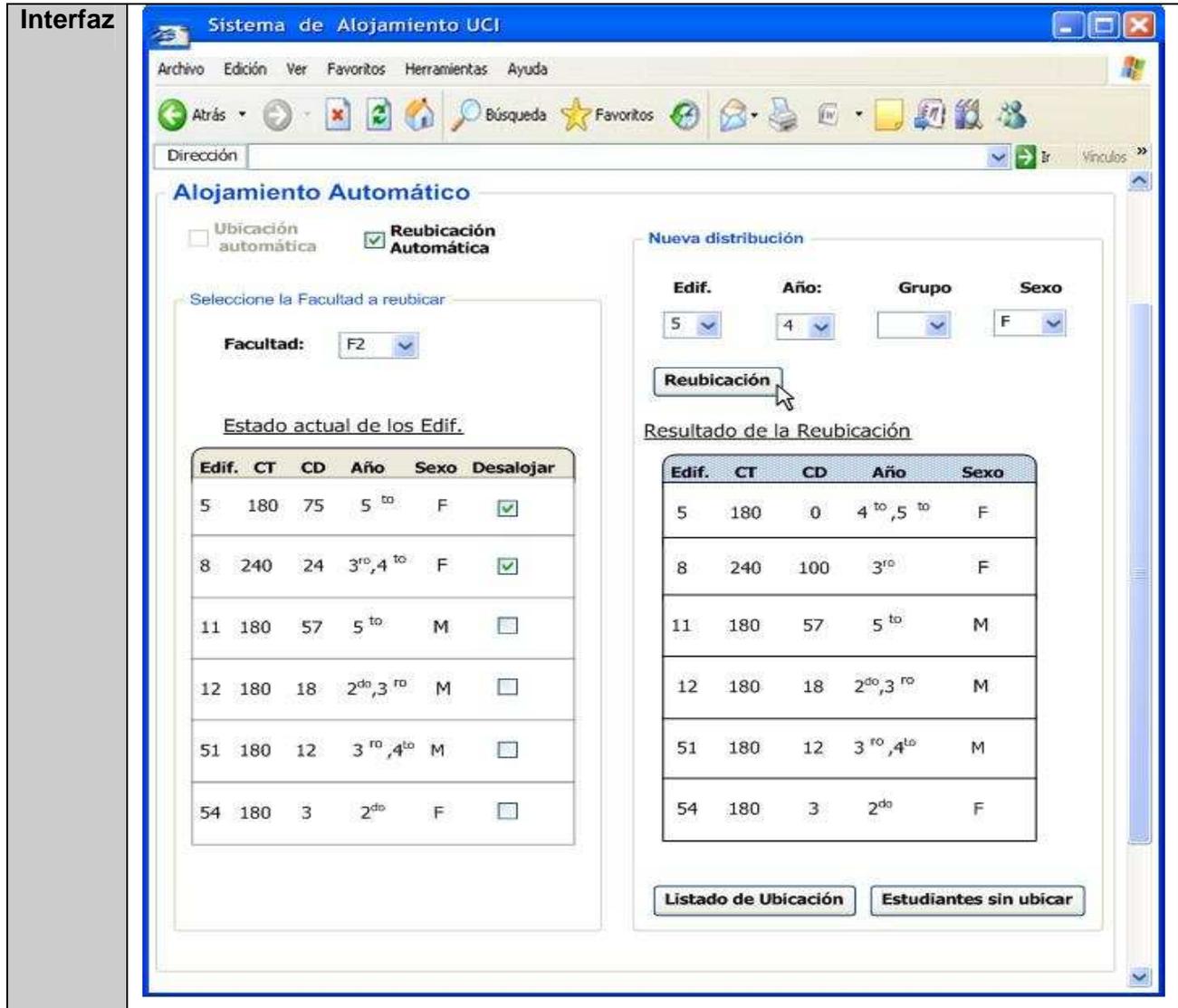
opción 2.1.1 permutas dobles.	seleccione las residencias, los edificios y los dos apartamentos entre cuales se va a realizar la permuta.
2.4 El usuario avanzado entra los datos.	2.5 El sistema muestra todos los becarios que viven en cada apartamento.
2.6 El usuario avanzado marca en cada apartamento los becarios que quiere permutar.	2.7 El sistema valida que ambos sean trabajadores o estudiantes y del mismo sexo.
	2.8 El sistema intercambia el alojamiento de los becarios, actualiza y muestra como quedaron los cambios.
Flujo alternativo de Eventos	
2.7a Datos no válidos	
Acción del Actor	Respuesta del Sistema
	2.8a El sistema muestra un mensaje de que no es válida la acción y da la opción de: Volver al paso 2.2 o finalizar el CUS sin realizar ninguna acción.
Flujo alternativo de Eventos	
Línea 2.2b Si la permuta es triple	
Acción del Actor	Respuesta del Sistema
2.2b El usuario avanzado selecciona la opción 2.1.2 permutas triples	2.3b El sistema muestra una interfaz para que se seleccione las residencias, los edificios y los tres apartamentos entre cuales se va a realizar la permuta.
2.4b El usuario avanzado entra los datos.	2.5b El sistema muestra todos los becarios que viven en cada uno de los tres apartamentos.
2.6b El usuario avanzado marca en cada apartamento los becarios que quiere	2.6 El sistema brinda la opción de escoger para cada becario, el apartamento para el cual lo va a permutar.

permutar.	
2.7 El usuario avanzado selecciona el apartamento para el que desea permutar a cada becario.	2.8 El sistema intercambia los becarios para los apartamentos seleccionados, actualiza y muestra como quedaron ubicados.
Flujo alterno de Eventos	
Línea 2.2c Mudanza	
Acción del Actor	Respuesta del Sistema
2.2c El usuario avanzado selecciona la opción 2.1.3 Mudanza	2.3c El sistema muestra una interfaz para que busque el apartamento que desea mudar y todos los apartamentos con capacidades disponibles.
2.4c El usuario avanzado entra el número del apartamento a mudar y selecciona uno de los disponibles	2.5c El sistema muestra una interfaz con todos los becarios de cada apartamento.
2.6c El usuario selecciona los estudiantes que desea mudar, de acuerdo a la cantidad de capacidades disponible en el otro apartamento.	2.7c El sistema asigna la nueva ubicación a los becarios, libera las capacidades utilizada anteriormente por los mismos, actualiza y muestra como quedan los cambios.

Tabla 12. Especificación del CU Gestionar alojamiento automático.

Caso de Uso	Gestionar alojamiento automático
Actores	Usuario avanzado
Propósito	Asignarle de forma automática alojamiento a grupos de personas.
Resumen	El CUS se inicia cuando se decide dar alojamiento automático a una x cantidad de personas, el sistema da la opción de seleccionar el grupo docente en caso de ser estudiantes o cargar una lista desde un fichero en caso de ser trabajadores y de acuerdo a los criterios de ubicación y las capacidades de los apartamentos, el sistema los ubica automáticamente. Mostrando al final la ubicación de cada

	becario y brindando la opción de imprimirla. El sistema permite además, reubicar de forma automática, liberando todas las capacidades de los edificios deseados y realizar nuevamente todo el proceso.
Referencia	R7
Prioridad	Primario
Descripción	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1 El CU se inicia cuando el usuario avanzado decide gestionar alojamiento automático para un grupo de becarios.	1.1 El sistema brinda la opción de realizar: 1.1.1 Alojamiento automático de estudiantes. 1.1.2 Alojamiento automático de trabajadores.
Sección: Alojamiento automático de estudiantes	



Acción del Actor	Respuesta del Sistema
2.1 El usuario avanzado selecciona la opción 1.1.1 Alojamiento automático de estudiantes.	2.2 El sistema brinda la opción de: 2.2.1 Realizar una ubicación automática. 2.2.2 Realizar una reubicación automática.
2.3 El Usuario avanzado selecciona la opción 2.2.1 Realizar una ubicación automática.	2.4 El sistema brinda la opción de seleccionar la facultad en la que se va realizar el alojamiento automático.

2.5 El usuario avanzado selecciona la facultad.	2.6 El sistema muestra una interfaz para que se seleccione el año o el grupo docente que quiere ubicar, y los edificios en los que desea se realice el alojamiento (especificando cual para las hembras y cual para los varones), de los asociados a la facultad que tiene capacidades disponibles.
2.7 El usuario avanzado selecciona los datos.	2.8 El sistema carga los listados de los grupos docentes del año seleccionado y comienza a ubicar en los edificios, comenzando por la primera capacidad disponible de los mismos y respetando el orden de los grupos. (Por ejemplo: Ubicará primero las hembras del 3101, luego a continuación las del 3102 y así sucesivamente con todos los grupos, al mismo tiempo realiza la misma operación con los varones en otro edificio.)
	2.9 Al concluir muestra un listado con la ubicación de cada estudiante y brinda la opción de imprimirlo.
Flujo alterno de Eventos	
Línea 2.3a Realizar una reubicación automática.	
Acción del Actor	Respuesta del Sistema
2.3a El usuario avanzado selecciona la opción 2.2.2 Realizar una reubicación automática.	2.4a El sistema brinda la opción de seleccionar la facultad en la que se va a reubicar los becarios de forma automática.
2.5a El usuario avanzado selecciona los datos.	2.6a El sistema muestra todos los edificios asociados a la facultad, con los datos de la capacidad total y la

	capacidades disponible en cada uno, y los años docente que tienen alojamiento en el mismo, brindado la opción de liberar todas sus capacidades.
2.7a El usuario avanzado selecciona los edificios en que desea reubicar.	2.8a El sistema libera todas las capacidades de los edificios seleccionados y brinda la opción de seleccionar el año docente o grupo que desea reubicar en cada uno, especificando además el sexo.
2.9a El usuario avanzado selecciona los datos.	2.10a El sistema carga los listados de los grupos docentes del año seleccionado y comienza a reubicar a todos los estudiantes en el edificio, respetando el orden de los grupos. Si alguno de estos estudiantes tenía una ubicación previa, el sistema se la cambia y libera la capacidad anterior.
	2.11a El sistema brinda la opción de visualizar el listado de las ubicaciones por edificios y los estudiantes que quedaron sin alojamiento, debido a este proceso.
Flujo alterno de Eventos	
Línea 2.9. Capacidades insuficientes	
Acción del Actor	Respuesta del Sistema
	<p>2.9a Si en uno de los edificio asignados, se agotan las capacidades disponibles antes de terminar de ubicar a todos los estudiantes de la lista, el sistema muestra:</p> <ul style="list-style-type: none"> - Estudiantes alojados, con sus respectivas ubicaciones. - Estudiantes sin alojamiento <p>Y brinda la opción de:</p> <ul style="list-style-type: none"> - Volver a la acción 2.4 del flujo normal de eventos y continua la ubicación en los nuevos edificios entrados. - Cancelar la acción y terminar el CU dejando esos

	estudiantes sin alojamiento.
Sección: Alojamiento automático de trabajadores	
Acción del Actor	Respuesta del Sistema
3.2 El usuario avanzado selecciona la opción 1.1.2 Alojamiento automático de trabajadores.	3.2 El sistema da la posibilidad de cargar la lista de trabajadores a ubicar, desde un fichero.
3.3 El usuario avanzado cargar la lista.	3.4 El sistema solicita que se seleccionen los criterios de ubicación que desea utilizar y muestra todos los edificios de la residencia de profesores con capacidades disponibles.
3.5 El usuario avanzado selecciona los criterios y los edificios en los cuales desea alojarlos.	3.6 El sistema valida que cada trabajador de la lista exista en la BD antes de ser alojado.
	3.8 El sistema comienza a ubicar en los edificios, de acuerdo a las capacidades de los apartamentos y los criterios de ubicación, al concluir muestra un listado con la ubicación de cada trabajador y brinda la opción de imprimirlo.
Flujo alternativo de Eventos	
Línea 3.6a Persona no identificada	
Acción del Actor	Respuesta del Sistema
	3.6 Si alguna persona del listado cargado no se encuentra en la BD, el sistema lo deja sin alojamiento y lo muestra como no identificado.
Flujo alternativo de Eventos	
Línea 3.8 Capacidades insuficientes	
Acción del Actor	Respuesta del Sistema
	3.8 Si de los edificio asignados, se agotan las capacidades disponibles antes de terminar de ubicar a

	<p>todos los trabajadores de la lista, el sistema muestra:</p> <ul style="list-style-type: none"> -Trabajadores alojados, con sus respectivas ubicaciones. - Trabajadores sin alojamiento. <p>Y brinda la opción de:</p> <ul style="list-style-type: none"> - Volver a al paso 3.5 del flujo normal de eventos. - Cancelar la acción y terminar el CU dejando esos trabajadores sin alojamiento.
--	--

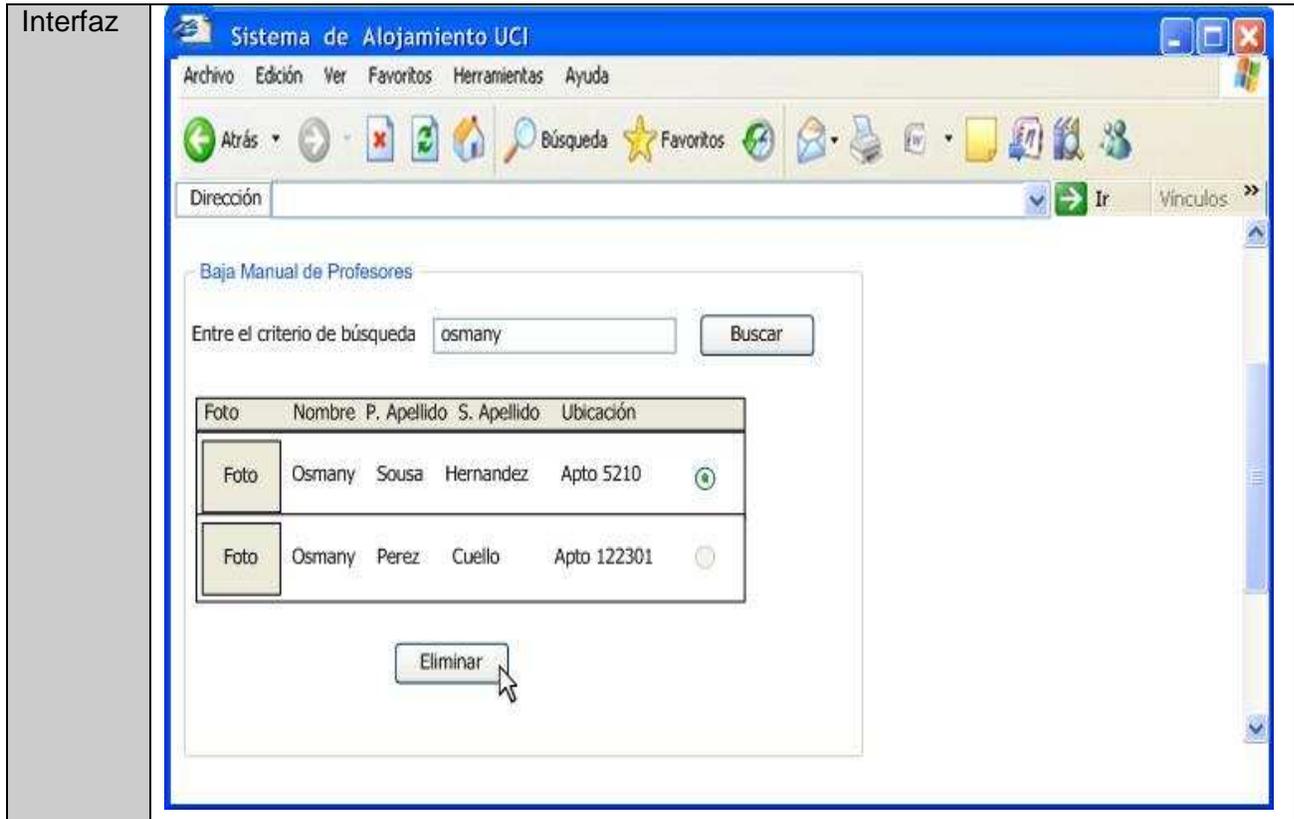
Tabla 13. Especificación del CU Baja automática de estudiantes

Caso de Uso	Baja automática de estudiantes
Actores	Akadosmos
Propósito	Dar baja en la aplicación de beca, a los estudiantes que están de licencia universitaria, que han causado baja en la universidad, o baja de una facultad por haberse trasladado hacia otra.
Resumen	<p>El CU se inicializa cuando Akadosmos nos notifica mediante un servicio web que se le solicita, que un estudiante se le ha dado licencia, baja universitaria, o traslado de facultad. El sistema una vez notificada una de estas acciones:</p> <p>Al estudiante de <u>baja universitaria</u>: Lo elimina automáticamente de la BD de beca, dejando como disponible su alojamiento.</p> <p>Al estudiante de <u>licencia</u>: Le Suspende el alojamiento, dejando el mismo como disponible, pero sin eliminarlos de la BD, con posibilidades de reubicar luego de un periodo de un año, como mínimo.</p> <p>(Nota: Este becario aparecerá en las búsquedas de personas como becario sin alojamiento por licencia.)</p> <p>Al estudiante trasladado de facultad: Se le suspende el alojamiento que tenía en la facultad de la cual se le dio baja, dejando el mismo como disponible y mostrar un mensaje para que se le gestione una nueva ubicación en su nueva facultad.</p>

Referencia	R9,R12	
Precondiciones	Que exista conectividad con Akademos Que Akademos esté brindando el servicio web que se le solicito.	
Poscondiciones		
Prioridad	Primario	
Descripción		
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El CU se inicializa cuando Akademos notifica que a un estudiante se le ha dado baja universitaria.	1.1 El sistema elimina el mismo automáticamente de la base de datos de beca, dejando como disponible su alojamiento.	
Flujo alterno de Eventos		
Línea 1.a Licencia Universitaria		
Acción del Actor	Respuesta del Sistema	
1.a El CU se inicializa cuando Akademos notifica que un estudiante se le ha dado licencia universitaria.	1.1a El sistema suspende el alojamiento del estudiante, dejando el mismo como disponible, pero sin eliminarlos de la base de datos, con posibilidades de reubicarlo luego de un período de un año, como mínimo.	
Flujo alterno de Eventos		
Línea 1.b Traslado de facultad		
Acción del Actor	Respuesta del Sistema	
1.b El CU se inicializa cuando Akademos notifica que un estudiante se ha trasladado de facultad.	1.1b El sistema suspende el alojamiento del estudiante, dejando el mismo como disponible, pero sin eliminarlos de la base de datos y muestra un mensaje para que se le gestione una nueva ubicación en su nueva facultad.	

Tabla 14. Especificación del CU Gestionar baja manual de trabajadores

Caso de Uso	Gestionar baja manual de trabajadores
Actores	Usuario avanzado.
Propósito	Este CU tiene como propósito eliminar de la base de datos aquellos profesores y trabajadores internos que no vayan a continuar en la universidad.
Resumen	El CU se inicia cuando un usuario avanzado decide dar baja a un trabajador, el sistema le da la posibilidad de buscarlo, el usuario avanzado introduce el criterio de ubicación, el sistema muestra todas las personas con esas características, el usuario avanzado selecciona el que desea eliminar, el sistema pregunta si está seguro de la operación a realizar y del usuario avanzado confirmar la operación, el sistema elimina al trabajador de la base de datos y libera su capacidad de alojamiento, quedando la misma como disponible.
Referencia	R10
Prioridad	Primario
Descripción	



Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El CU se inicia cuando el usuario avanzado decide gestionar la baja de algún trabajador interno	1.2 El sistema le da la posibilidad de buscarlo.
1.3 El usuario avanzado introduce el criterio de ubicación.	1.4 El sistema muestra los profesores con esas características.
1.5 El usuario avanzado selecciona el que desea eliminar y pincha el botón eliminar.	1.6 El sistema muestra un mensaje preguntando si está seguro que desea realizar la operación.
1.7 Si el usuario avanzado pincha el botón aceptar.	1.8 El sistema elimina el profesor con todos sus datos de la base de datos y muestra su ubicación como disponible.
Flujo Normal de Eventos	

Línea 1.7 Se cancela la acción	
Acción del Actor	Respuesta del Sistema
1.7 Si el usuario avanzado pincha cancelar	1.8 Se finaliza el CU sin realizar ninguna operación.

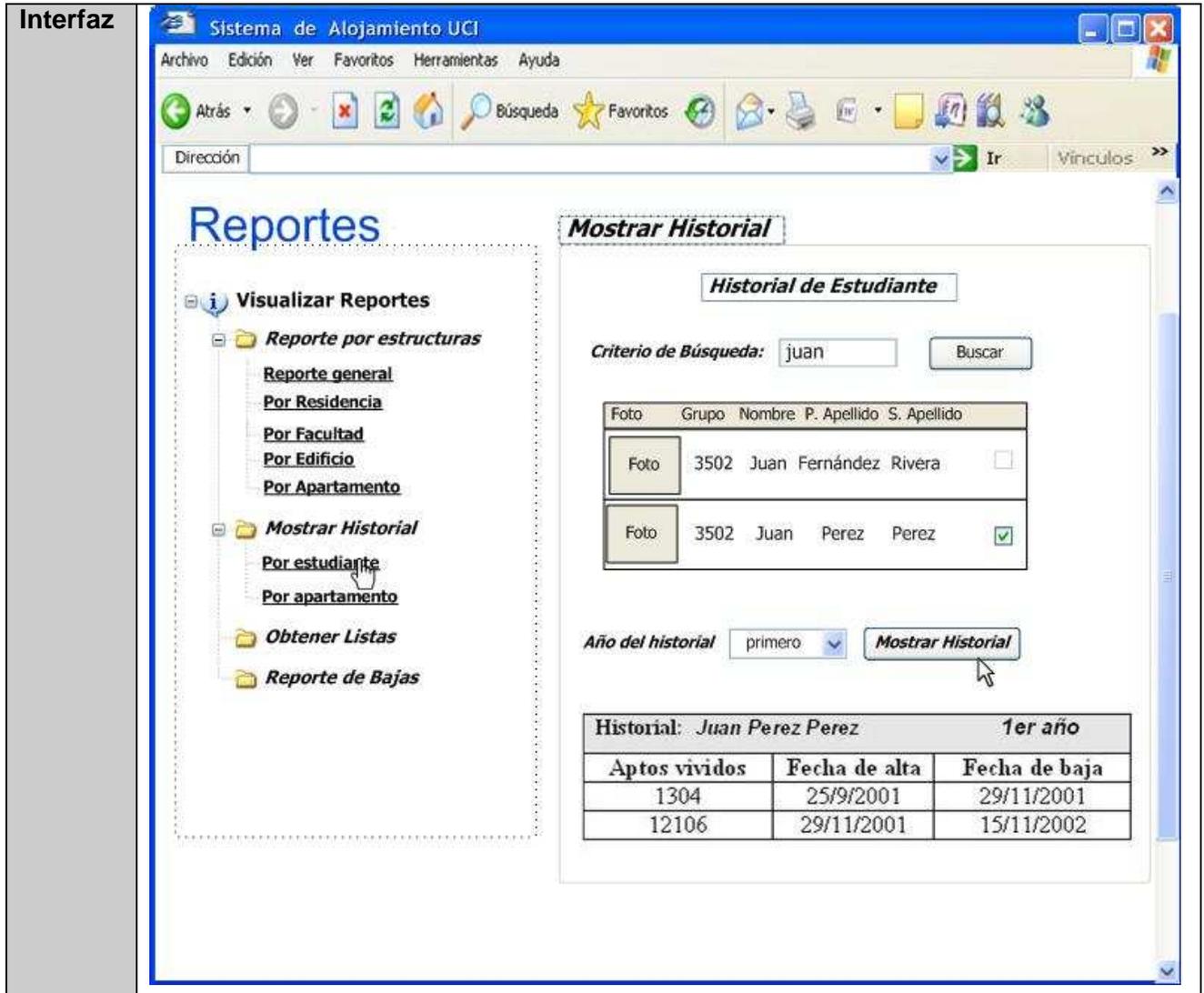
Tabla 15 Especificación del CU Registrar ubicaciones en un historial

Caso de Uso	Registrar ubicaciones en un historial	
Actores	Usuario avanzado	
Propósito	Este CU tiene como propósito crear un historial que almacene todas las ubicaciones que se les asignan a todos los estudiantes en el transcurso de los 5 años.	
Resumen	Este CU se inicia cada vez que se gestiona alojamiento o se gestiona alguna baja, de aquí el sistema registra para cada becario implicado en estas acciones los cambios en la ubicación con fecha de alta y baja en cada uno de los apartamentos.	
Referencia	R13	
Prioridad	Secundario	
Descripción		
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1.1 El CU se inicia cada vez que el usuario avanzado ejecuta uno de los siguientes CU: Gestionar alojamiento automático Gestionar alojamiento manual. Gestionar baja manual de profesores. Gestionar baja automática de estudiantes.	1.2 El sistema registrará (cada vez que se ejecute uno de los anteriores CU) para cada uno de los becarios involucrados, alguna de las siguiente entrada: Si se le dio alojamiento inicial: apartamento en el cual se alojó, con la fecha alta. Si se le gestionó un cambio de ubicación: Fecha de baja en el apartamento en el que estaba y el nuevo apartamento para el cual fue trasladado con fecha	

	<p>de alta.</p> <p>Si se le da baja: Fecha de baja en el apartamento que se encontraba.</p>
--	---

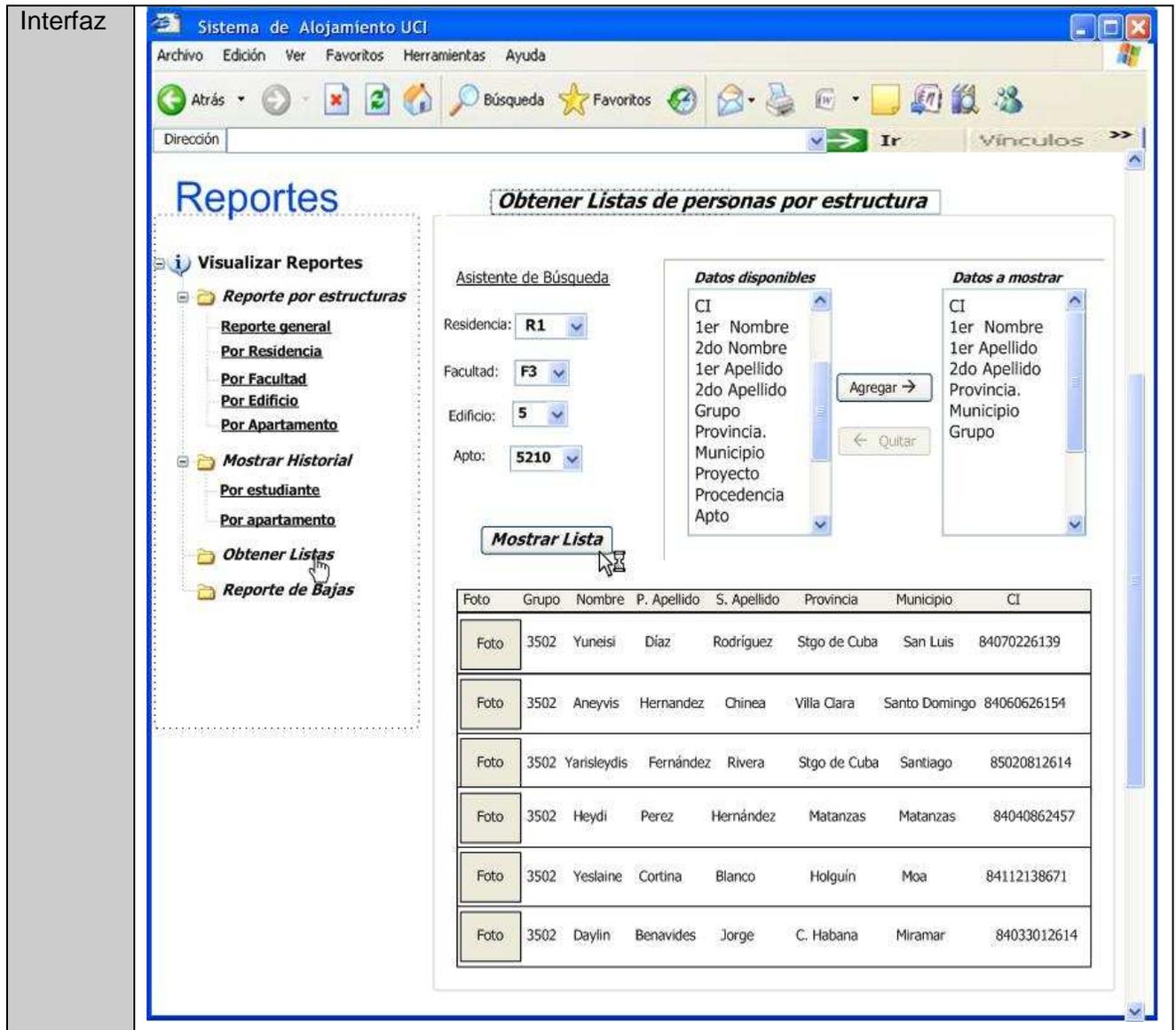
Tabla 17. Especificación del CU Visualizar reportes

Caso de Uso	Visualizar Reportes	
Actores	Usuario	
Propósito	Permitir a los usuarios del sistema visualizar una serie de reportes, de acuerdo a la información y los datos que deseen o necesiten conocer.	
Resumen	El CU se inicia cuando el usuario desea visualizar un reporte, el sistema le da la opción de seleccionar de los reporte más comunes el tipo que desee, el usuario selecciona uno de los reporte y brinda al sistema los datos requeridos para generarlo, el sistema visualiza el reporte solicitado.	
Referencia	R13, R14,R15, R16,R17	
Precondiciones		
Poscondiciones	El usuario visualiza el reporte	
Prioridad	Secundario	
Descripción		
Flujo Normal de Eventos		
	Acción del Actor	Respuesta del Sistema
	1 El CU se inicia cuando el usuario decide visualizar un reporte.	1.1 El sistema le muestra una interfaz con los diferentes tipos de reportes a visualizar: <ul style="list-style-type: none"> 1.1.1 Visualizar historial 1.1.2 Obtener listas 1.1.3 Visualizar reportes por estructura. 1.1.4 Visualizar reportes de bajas, licencias o traslados.
Sección: Visualizar historial		



Acción del Actor	Respuesta del Sistema
2.1 El usuario selecciona la acción 1.1.1 Visualizar un historial.	2.2 El sistema le brinda dos opciones: 2.1.1 Visualizar historial de un estudiante. 2.1.2 Visualizar historial de un apartamento.
2.3 El usuario selecciona la opción 2.1.1 obtener historial de un estudiante, entra los datos para buscarlo y selecciona de los años cursados por el mismo, el que desea	2.4 El sistema valida los datos y muestra todos los apartamentos en los cuales ha estado alojado el estudiante en ese período de tiempo, con la fecha de entrada y de salida en cada uno.

para el reporte.	
Flujo alterno de Evento	
Línea 2.4a Datos no válidos.	
Acción del Actor	Respuesta del Sistema
	2.4a Si no se obtiene ningún resultado en la búsqueda porque no existe dicho estudiante, el sistema muestra un mensaje y brinda las opciones de: Regresar al paso 2.3 del flujo normal de eventos o finalizar el CU sin realizar ninguna operación.
Flujo alterno de Evento	
Línea 2.3 b Visualizar historial de un apartamento.	
Acción del Actor	Respuesta del Sistema
2.3b El usuario selecciona la opción 2.1.2 obtener historial de un apartamento, entra el número del apartamento y el período de tiempo del que desea el reporte.	2.4b El sistema valida los datos y muestra todos los becarios que se han alojado en ese período de tiempo en ese apartamento y cual fue la fecha de entrada y de salida de cada uno, en el mismo.
Flujo alterno de Evento	
Línea 2.4c datos no válidos.	
Acción del Actor	Respuesta del Sistema
	2.4c Si el apartamento entrado no existe, el sistema muestra un mensaje y brinda las opciones de: Regresar al paso 2.3b del flujo alterno de evento o finalizar el CU sin realizar ninguna operación.
Sección: Obtener Listas	

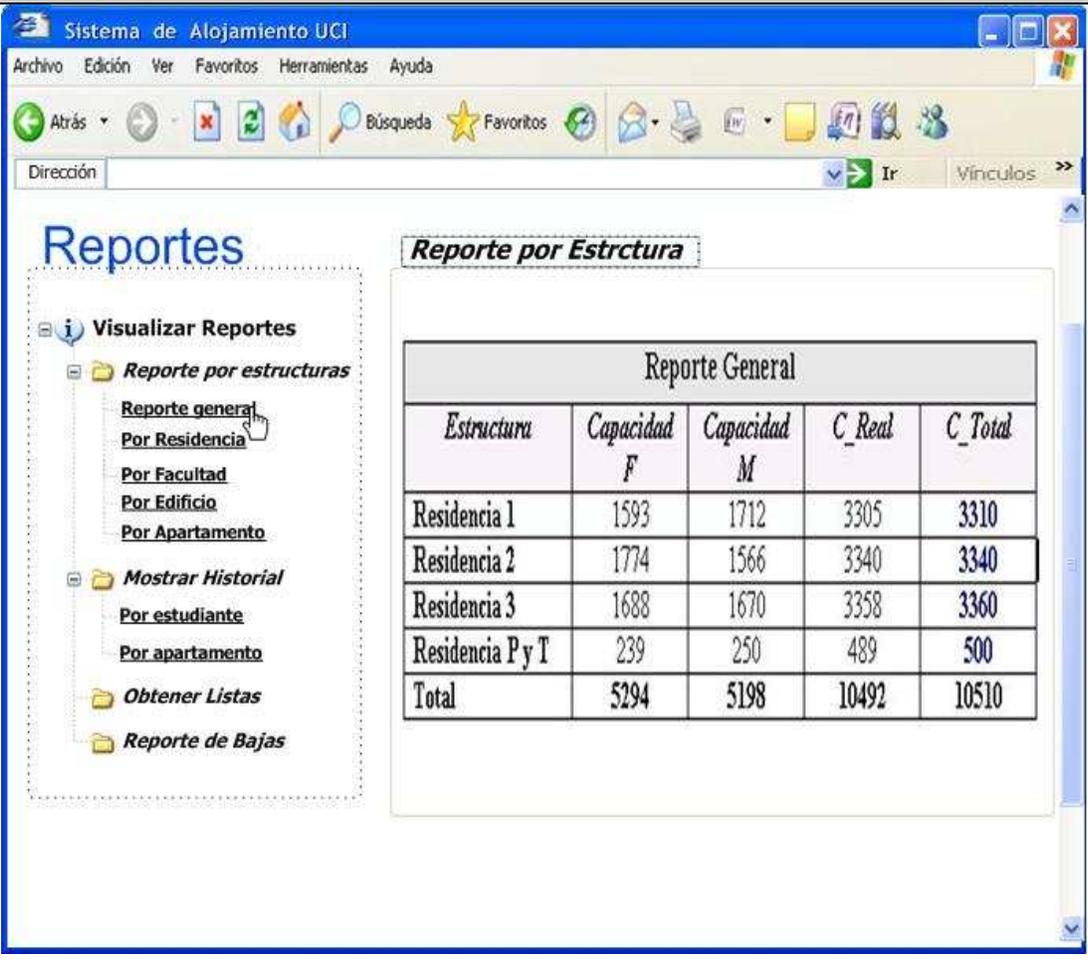


Acción del Actor	Respuesta del Sistema
3 1 El usuario selecciona la acción 1.1.2 Obtener una lista de becarios.	3.2 El sistema le muestra una interfaz con los siguientes parámetros de búsqueda: Residencia Facultad Edificio Apartamento

<p>3.3 El usuario escoge los parámetros por los cuales desea realizar la búsqueda.</p>	<p>3.4 El sistema le brinda la opción de seleccionar los datos que puede mostrar en el listado de las personas encontradas en la búsqueda: CI, Solapín, Nombres, Apellidos, Residencia, Facultad, Apartamento, Grupo, Provincia, Municipio, Procedencia.</p>
<p>3.5 El usuario escoge los datos.</p>	<p>3.6 El sistema muestra un listado de personas con todos los datos solicitados y brinda la opción de imprimirlos.</p>

Sección: Visualizar reportes por estructura.

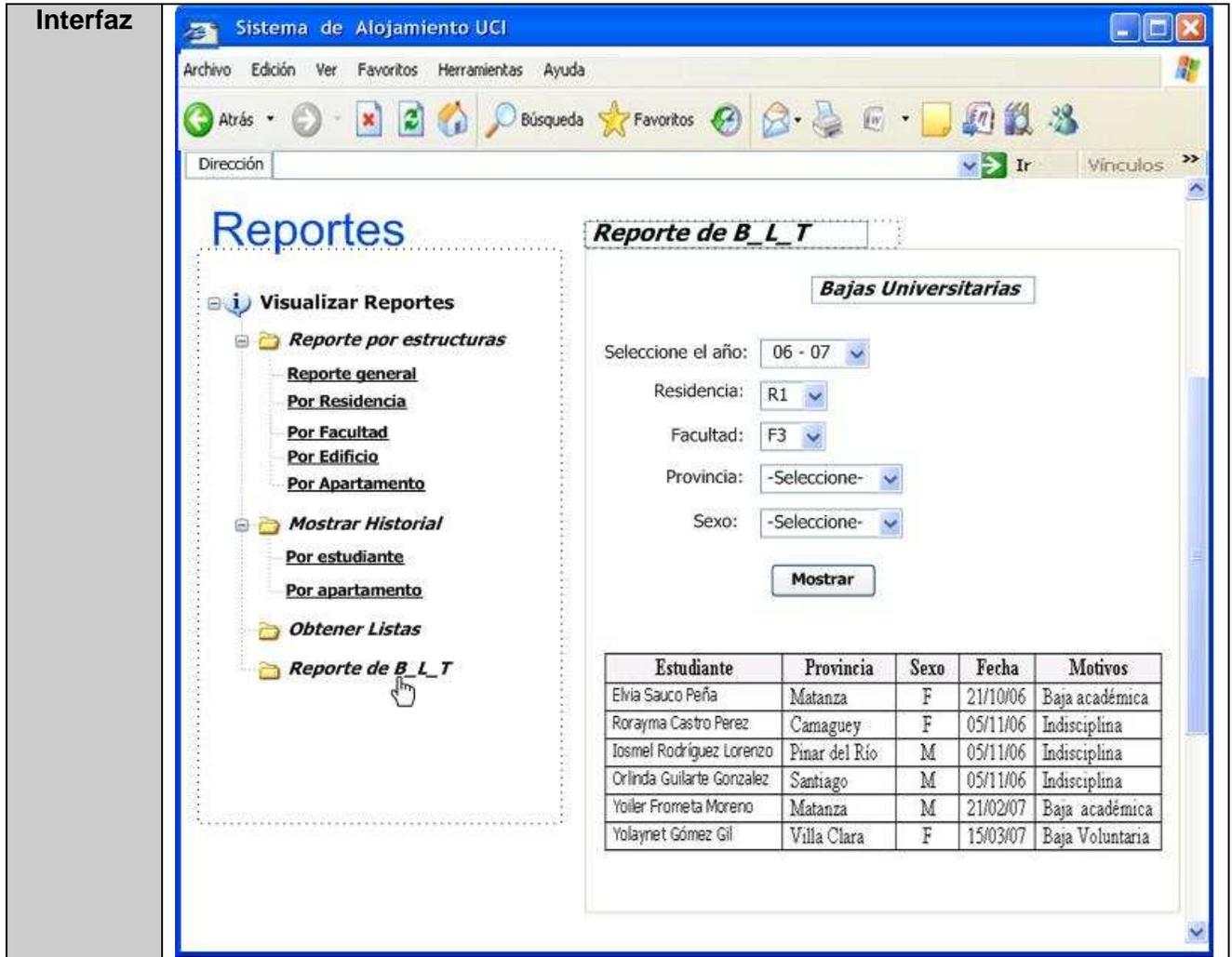
Interfaz



Reporte General				
Estructura	Capacidad F	Capacidad M	C_Real	C_Total
Residencia 1	1593	1712	3305	3310
Residencia 2	1774	1566	3340	3340
Residencia 3	1688	1670	3358	3360
Residencia P y T	239	250	489	500
Total	5294	5198	10492	10510

Acción del Actor	Respuesta del Sistema
4 El usuario selecciona la acción 1.1.3 Visualizar reportes por estructura.	4.1 El sistema le brinda la posibilidad al usuario de escoger el tipo de acción que quiere realizar: 4.1.1 Obtener reporte general. 4.1.2 Obtener reporte de residencia. 4.1.3 Obtener reporte por facultad. 4.1.4 Obtener reporte de edificio. 4.1.5 Obtener reporte de personas por estructura.
4.2 El usuario escogió la acción 4.1.2 de Obtener reporte general.	4.3 El sistema le muestra una tabla con todas las residencias y de cada una muestra, la capacidad total de la misma, las capacidades femeninas y masculinas y la capacidad total existente.
	4.4 El sistema brinda la opción de imprimir el reporte.
Flujo alternativo de Evento	
Línea 4.2a El usuario selecciona la opción 4.1.2.	
Acción del Actor	Respuesta del Sistema
4.2a El usuario escogió la acción de obtener reporte de residencia.	4.3a El sistema le muestra una interfaz con todas las residencias existentes.
4.4a El usuario selecciona la residencia de la cual desea obtener el reporte.	4.5a El sistema le muestra de la residencia seleccionada los edificios que pertenecen a ella, la capacidad total de personas, la capacidad real de sexo femenino y la de sexo masculino existente, capacidad total existente, así como las capacidades totales disponibles.
	4.6a El sistema brinda la opción de imprimir el reporte.
Flujo alternativo de Evento	
Línea 4.2b El usuario selecciona la opción 4.1.3.	

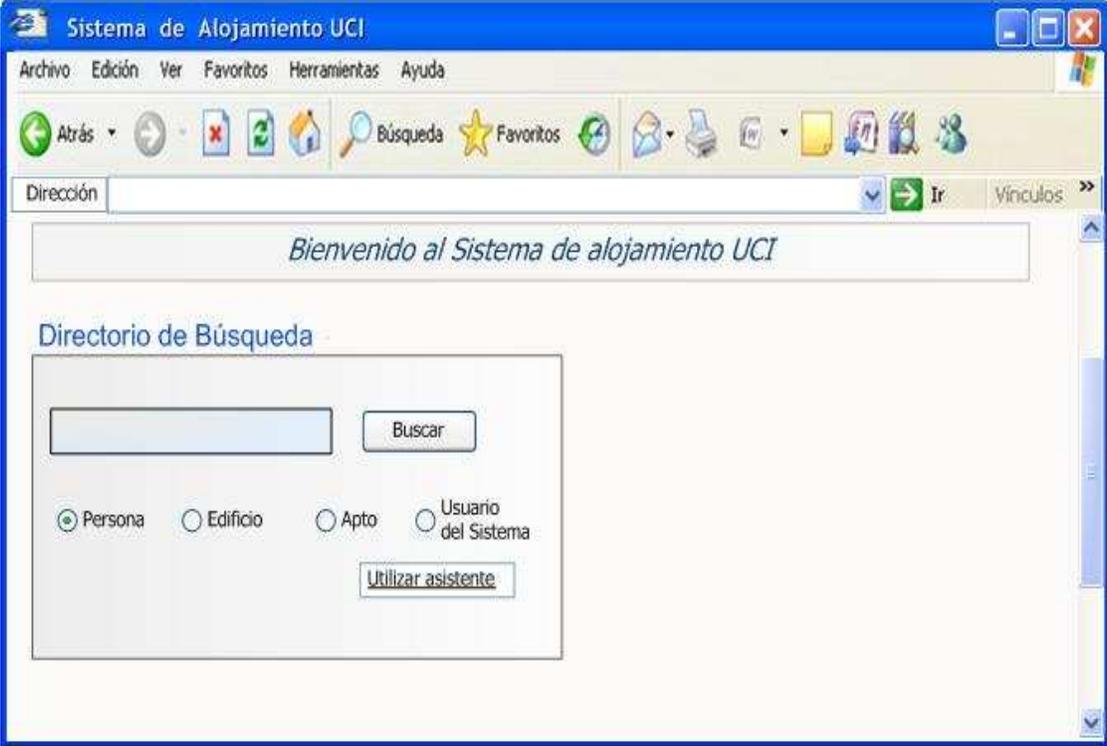
Acción del Actor	Respuesta del Sistema
4.2b El usuario escogió la acción de Obtener reporte por facultad.	4.3b El sistema le muestra una interfaz con todas las facultades existentes.
4.4b El usuario escoge la facultad de la que desea obtener el reporte.	4.5b El sistema le muestra de esta facultad, todos los edificios, la capacidad total de personas, la capacidad total de sexo femenino y la de sexo masculino, capacidades disponibles por sexos, así como las capacidades totales disponibles.
	4.6b El sistema brinda la opción de imprimir el reporte.
Flujo alternativo de Evento	
Línea 4.2c El usuario selecciona la opción 4.1.4.	
Acción del Actor	Respuesta del Sistema
4.2c El usuario escogió la acción de Obtener reporte de edificio.	4.3c El sistema muestra una interfaz con todas las residencias existentes.
4.4c El usuario selecciona la residencia en la que se encuentra el edificio del que desea obtener el reporte y luego selecciona el mismo.	4.5c El sistema brinda un reporte con la cantidad de apartamentos del mismo, capacidad total, las capacidades disponibles y los apartamentos no habitables.
Flujo alternativo de Evento	
Línea 4.2d El usuario selecciona la opción 4.1.5.	
Acción del Actor	Respuesta del Sistema
4.2d El usuario escogió la acción de Obtener reporte por apartamento	4.3d El sistema muestra un interfaz para que se pueda introducir el # del apartamento del cual desea obtener el reporte
4.4d El usuario introduce el # del apartamento	4.5d El sistema muestra la capacidad total del mismo, la cantidad de cuartos, las capacidades disponibles y el sexo de las personas que lo habitan.
Sección: Visualizar Reporte de Bajas, Licencias o Traslados	



Acción del Actor	Respuesta del Sistema
5 El usuario selecciona la acción 1.1.4 Visualizar Reporte de Bajas licencias o traslados	5.1 El sistema le brinda la posibilidad al usuario de escoger el tipo de acción que quiere realizar: 5.1.1 Obtener reporte de las bajas. 5.1.2 Obtener reporte de las licencias 5.1.3 Obtener reporte de los traslados. 5.1.4 Obtener reporte estadístico general.
5.2 El usuario escogió la acción 5.1.1 de obtener reporte de las bajas.	5.3 El sistema le muestra una interfaz para que el usuario seleccione el año del cual desea obtener el

	<p>reporte, y muestra además otros parámetros auxiliares para la búsqueda como: Residencia, Facultad, Sexo y Provincia.</p>
<p>5.4 El usuario selecciona el año y los parámetros por lo cual desea obtener el reporte.</p>	<p>5.5 El sistema le muestra una tabla con todos los estudiantes de bajas dado esos parámetros, con sus respectivos datos: nombre y apellidos, sexo, provincia, además de motivos y fecha de la misma.</p>
<p>Flujo alternativo de Evento</p>	
<p>Línea 5.2a El usuario escogió la acción 5.1.2 de obtener reporte de las licencias.</p>	
<p>Acción del Actor</p>	<p>Respuesta del Sistema</p>
	<p>5.5a El sistema le muestra una tabla con todos los estudiantes de licencias dado los parámetros seleccionados, con sus respectivos datos: nombre y apellidos, apartamento, grupo, sexo, provincia, además de motivos y fecha de la misma.</p>
<p>Flujo alternativo de Evento</p>	
<p>Línea 5.2b El usuario escogió la acción 5.1.3 de Obtener reporte de los traslados.</p>	
<p>Acción del Actor</p>	<p>Respuesta del Sistema</p>
	<p>5.5b El sistema le muestra una tabla con todos los estudiantes de traslado dado los parámetros seleccionados, con sus respectivos datos: nombre y apellidos, apartamento, grupo, sexo, provincia, además de motivos y fecha de la misma.</p>
<p>Flujo alternativo de Evento</p>	
<p>Línea 5.2c El usuario escogió la acción 5.1.4 de Obtener reporte estadístico general.</p>	
<p>Acción del Actor</p>	<p>Respuesta del Sistema</p>
<p>5.2c El usuario escogió la acción 5.1.4 de obtener reporte estadístico general.</p>	<p>5.3c El sistema le muestra una tabla con los datos estadísticos de las bajas, licencias y traslados en general.</p>

Tabla 18 Especificación del CU Buscar

Caso de Uso	Buscar
Actores	Usuario
Propósito	El CU permite realizar una búsqueda de cualquier parámetro del sistema.
Resumen	El CU se inicia cuando el usuario decide realizar una búsqueda, el sistema le da la opción de buscar persona, edificio, apartamento y usuario del sistema, el usuario selecciona lo que desea buscar, el sistema solicita los datos, la instructora los entra, el sistema muestra los resultados.
Referencia	R18
Precondiciones	Tiene que estar registrado la persona, el edificio, el apartamento y el usuario a buscar.
Poscondiciones	Mostrar datos del elemento buscado.
Prioridad	Secundario
Descripción	
Interfaz	

Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1 El CU se inicia cuando el usuario decide realizar una búsqueda.	1.1 El sistema le brinda la opción de seleccionar que desea buscar: Persona Edificio Apartamento Usuario del sistema
Sección: Buscar persona	
Acción del Actor	Respuesta del Sistema
2.1 El usuario selecciona realizar búsqueda de persona e introduce el nombre de la misma.	2.2 El sistema valida los datos y muestra un listado con todas las personas registradas con ese nombre.
Flujo alternativo de Evento	
Línea 2.4a Datos no válidos.	
Acción del Actor	Respuesta del Sistema
	2.4a.1 El sistema muestra categoría de personas con cantidad 0.
Sección: Buscar Edificio	
Acción del Actor	Respuesta del Sistema
3.1 El usuario selecciona realizar búsqueda de edificio y entra el número del mismo.	3.2 El sistema valida los datos, muestra el edificio y dice la facultad a la que pertenece, su capacidad total y las capacidades disponibles.
Flujo alternativo de Evento	
Línea 3.2 a Datos no válidos	
Acción del Actor	Respuesta del Sistema
	3.2a.1 El sistema muestra la categoría de edificio con cantidad 0
Sección: Buscar Apartamento	

Acción del Actor	Respuesta del Sistema
4.1 El usuario selecciona realizar búsqueda de apartamento y entra el número del mismo.	4.2 El sistema valida los datos, muestra el apartamento y dice la facultad a la que pertenece, su capacidad total y las capacidades disponibles y el sexo de las persas que lo habitan.
Flujo alternativo de Evento	
Línea 4.4a Datos no válidos	
Acción del Actor	Respuesta del Sistema
	4.2a.1 El sistema muestra la categoría de edificio con cantidad 0 y da la opción de regresar a la acción 4.1 del flujo normal de evento o terminar el CU.
Sección: Buscar Usuario del sistema	
Acción del Actor	Respuesta del Sistema
5.1 El usuario selecciona buscar usuarios del sistema y entra el usuario del mismo.	5.2 El sistema valida el usuario, muestra el usuario y el rol que tiene asignado.
Flujo alternativo de Evento	
Línea 5.2a Datos no válidos	
Acción del Actor	Respuesta del Sistema
	5.2a.1 El sistema muestra la categoría de usuario del sistema con cantidad 0 y da la opción de regresar a la acción 5.1 del flujo normal de evento o terminar el CU.

3.8 Análisis de los Resultados

En el trabajo se obtuvieron varios artefactos que son de vital importancia para la continuidad del mismo, es decir, para que los roles que siguen al analista puedan desempeñar su papel. Como parte de todo el proceso que se realizó, los resultados más importantes están dados en los requisitos funcionales y no funcionales y en el diagrama de casos de usos con sus especificaciones e interfaces.

Requisitos:

Como resultado de la captura de requisitos realizada, se obtuvieron 18 requisitos funcionales y se adicionaron los no funcionales en dependencia de las propiedades o cualidades que el sistema debe cumplir. Estos 18 requisitos funcionales sufrieron unas cuantas variaciones debido a que los mismos estaban sujetos a un procedimiento donde se adicionaron, combinaron y eliminaron requisitos, de la primera entrevista realizada que fue con el fin de negociación y de entender el entorno en el cual se desarrollaba el proceso, solo se pudieron obtener alrededor de 10 de ellos, los cuales estaban algo ambiguos y con una breve explicación que no daba mucha claridad de lo que realmente se quería, esta cifra representa un 56% del total que se capturo definitivamente. Al ver esos resultados se hacia necesario otras entrevistas para perfeccionar los requisitos ya existentes, a los cuales se le fueron sumando otros en dependencia del entendimiento. Se lograron reunir 17 requisitos representando un 94% de los existentes finalmente, estos contaban con un 100% de aprobación por parte de los clientes, logrando así un entendimiento entre ambas partes. Pero como parte del proceso de cambio se hizo necesario incorporar un nuevo requisito para darle cumplimiento a todas las funcionalidades de una forma eficiente, completándose así los 18 requisitos, para un 100% de aprobación por parte de los clientes.

Este 100% de aprobación lo arrojaron los avales realizados por una parte del personal de residencia, vinculado al proyecto, que se comportan como los clientes. En estos avales se recoge el grado de satisfacción de ellos con la captura de requisitos.

Ver anexos 5 y 6

Uso de métricas:

Las métricas son un buen medio para entender, monitorear, controlar, predecir y probar el desarrollo del software, ayudan a entender tanto le proceso técnico que se utiliza en el desarrollo del producto, como el propio producto. Al medir el producto, se intenta aumentar su calidad, evaluar la productividad del personal que lo desarrolla, evaluar los beneficios en términos de productividad y de calidad, también se

mide un producto con el objetivo de establecer una línea de base para la estimación, para ayudar a justificar el uso de nuevas herramientas o de formación adicional.

En este trabajo para determinar la especificidad (ausencia de ambigüedad) de los requisitos, se usa una métrica basada en la consistencia de la interpretación de los revisores para cada requisito. (Sugerida en: ISW un Enfoque Práctico).

Consiste en que los revisores escogidos, interpreten los requisitos y determinen en el caso de que existan, los requisitos con ambigüedad. Es usada de la siguiente forma:

Se tiene la suma de todos los requisitos representada por N_r , quedando de la siguiente manera:

$$N_r = N_f + N_{nf}$$

Donde:

N_f : es el número de requisitos funcionales

N_{nf} es el número de requisitos no funcionales.

Para determinar la especificidad usando la métrica basada en la consistencia de la interpretación de los revisores para cada requisito, quedando de la siguiente forma:

$$Q = N_{ui} / N_r$$

Donde N_{ui} es el número de requisitos para los que todos los revisores tuvieron interpretaciones idénticas. Cuanto más cerca de 1 esté el valor de Q, menor será la ambigüedad de los requisitos. Aplicado a los requisitos de este trabajo quedó de la siguiente manera.

Para un total de 32 requisitos, de estos 18 son funcionales y 14, no funcionales, los cuales fueron interpretados por 5 revisores, se obtuvo un total de 27 requisitos para los cuales los revisores tuvieron una interpretación idéntica.

$$Q = 27 / 32$$

$$Q = 0.84$$

Se hicieron algunos cambios en los requisitos para mejorar lo señalado, haciendo así una segunda versión de los mismos. Esta, fue entregada por segunda vez a los revisores, los cuales coincidieron en la interpretación de la mayoría de estos, completando así un valor de 0,97, el cual refleja la especificidad de los requisitos capturados.

Métricas aplicadas al diagrama de casos de uso del sistema.

Se aplicó un modelo de métricas orientado a objeto, al diagrama de casos de uso del sistema, para medir la calidad de la funcionalidad. Este modelo define cuatro atributos de propiedades de calidad, los cuales son: Consistencia, Correctitud, completitud y complejidad, cada uno de estos atributos está compuesto por diferentes factores que tienen asociado una métrica.

El atributo Completitud, mide el grado en que se ha logrado detallar todos los casos de uso relevantes, Consistencia mide el grado en que los casos de uso del sistema describen las interacciones adecuadas entre el usuario y el sistema, Correctitud mide el grado en que las interacciones actor / sistema soportan adecuadamente el proceso del negocio, mientras que el atributo complejidad, mide la claridad en la presentación de los elementos que describen el contexto y funcionalidades del sistema.

Los factores aplicados para cada atributo son:

Tabla 19 Atributo completitud

Factores de completitud	Métricas aplicadas
Factor 1. ¿Han sido definidos todos los roles relevantes de usuario encargados de generar/modificar o consultar información?	Métrica 1: No se omiten roles relevantes Error = 0 %
Factor 2. ¿Han sido considerados todos los sistemas externos con los cuáles interactuará el sistema?	Métrica 2: No se omiten sistemas externos Error =0 %
Factor 3. ¿Están definidos todos los requisitos que justifican la funcionalidad del caso de uso?	Métrica 3: No se omiten requisitos por caso de uso. Error = 0 %
Factor 4. ¿Se presenta una descripción detallada (descripción extendida esencial) de todos los casos de uso del sistema?	Métrica 4: Todos los casos de uso poseen una descripción extendida. Umbral = 0 %
Factor 5. ¿Se describen las condiciones de excepción	Métrica 5: Todos los casos de uso describen condiciones de excepción relevantes.

relevantes que debe contemplar cada flujo de eventos?	Error = 0 %
Factor 6. ¿Todos los casos de uso del sistema han sido clasificados de acuerdo a su relevancia (primario / secundario / opcional)?	Métrica 6: Todos los casos de uso han sido clasificados Error = 0 %
	Total 100 %

Tabla 20 Atributo consistencia

Factores de consistencia	Métricas aplicadas
Factor 7. ¿Representa el caso de uso una interacción observable por un actor?	Métrica 7: Todos los casos de uso representan una interacción observable por un actor Error = 0 %
Factor 8. ¿Existen acciones en el flujo de eventos asignadas a un responsable que no le corresponde?	Métrica 8: Todas las acciones del flujo de evento están asignadas al responsable que le corresponde. Error = 0 %
Factor 9. ¿Está adecuadamente redactado (en el lenguaje del usuario) el flujo de eventos?	Métrica 9: Un casos de uso no fue aceptado. Error = 8.3 %
Factor 10. ¿La descripción del flujo de eventos se inicia con la descripción de una acción externa originada por un actor o por una condición interna del sistema claramente identificable?	Métrica 10: No existe ningún caso de uso cuya descripción extendida no inicia con una acción externa o con una condición monitoreada por el sistema Error = 0 %
Factor 11. ¿Existe una adecuada separación entre el flujo básico de eventos y los flujos alternos y/o flujos subordinados?	Métrica 11: Todos los casos de uso complejos tienen separación entre el flujo básico y los flujos alternos.

	Error = 0 %
	Total 98,34 %

Tabla 21 Atributo correctitud

Factores de correctitud	Métricas aplicadas
Factor 12. ¿Existe para cada caso de uso por lo menos un usuario responsable?	Métrica 12: Todos los casos de uso tienen al menos un usuario responsable. Error = 0 %
Factor 13. ¿Representa el caso de uso requisitos comprensibles por el usuario?	Métrica 13: Todos los requisitos representados en los casos de uso son comprensibles por el usuario. Error = 0 %
Factor 14: ¿Las interacciones definidas introducen mejoras al proceso actual?	Métrica 14: Un caso de uso que debe ser modificado para mejorar el proceso actual. Error = 8,3 %
	Total 97,5 %

Tabla 22 Atributo complejidad

Factores de complejidad	Métricas aplicadas
Factor 15. ¿Los elementos dentro del diagrama están adecuadamente ubicados de manera que facilitan su interpretación?	Métrica 15: Ningún elementos del diagrama requiere reubicación Error = 0 %
Total 100%	

En el siguiente gráfico se muestran los resultados obtenidos de las métricas aplicadas, el mismo demuestra que el Diagrama de casos de uso del Sistema cumple con la calidad necesaria de funcionalidad, la cual está representada en un 98,95%.



Fig 6 Resultados de la métrica aplicada

El Diagrama de Casos de Uso del Sistema tiene definido todos los roles relevantes de usuarios, se consideraron todos los sistemas externos con los cuales interactuará el sistema. Se definieron los requisitos que justifican la funcionalidad de los casos de uso, a los cuales se le hizo una descripción detallada, donde se describen las condiciones de excepción relevantes en cada flujo de eventos, todos

estos casos de uso han sido clasificados de acuerdo a su relevancia. Todo esto arrojó como resultado un 100% de completitud.

Todos los casos de uso representan una interacción observable por un actor, no existe ninguno de estos cuya descripción extendida no inicia con una acción externa o con una condición monitoreada por el sistema, las acciones del flujo de evento están asignadas al responsable que le corresponde, existiendo una adecuada separación entre el flujo básico de eventos y los flujos alternos. Un caso de uso no tiene la mejor redacción al explicar el flujo de eventos, es muy extenso y se hace incómoda su interpretación. Reportando esto un 98,34 % de consistencia.

Todos los casos de uso tienen al menos un usuario responsable, los requisitos representados en los mismos son comprensibles por el usuario. Existe un caso de uso que puede ser modificado para ampliar las ventajas de las funcionalidades del mismo. Quedando un 97,5 % de correctitud. Todos los elementos del diagrama están adecuadamente ubicados facilitando su interpretación, reportando esto un 100% de complejidad.

Conclusiones

El correcto modelado del diagrama de CUS utilizando los patrones correspondientes antes mencionados, la descripción detallada de cada uno de los mismos en un lenguaje sencillo, su clasificación según la prioridad requerida y el modelado de los prototipos de interfaces no funcionales, permitieron reafirmar un entendimiento común con los clientes, mostrándole de forma más específica las funcionalidades del sistema propuesto de acuerdo a sus necesidades y las ventajas del mismo además de facilitar a los desarrolladores la continuidad del proceso de forma segura, basados en los resultados obtenidos con la aplicación del Modelo de Métricas OO para el diagrama de CUS el cual arrojó un resultado de calidad superior al 98 %.

Conclusiones Generales

- A partir del estudio realizado de las metodologías de desarrollo de software, se escogió RUP como la más adecuada para el desarrollo de este trabajo.
- El intercambio constante con los clientes y la aplicación de estrategias para la captura de requisitos, facilitó entender el negocio actual y describirlo, para realizar los diagramas correspondientes, y la obtención de la información necesaria para delimitar las funcionalidades, conformando así los requisitos del sistema.
- El DCUS, las especificaciones de los CUS y el diseño de los prototipos no funcionales de interfaz de usuario, muestran una descripción detallada de las funcionalidades del sistema.
- Los avales emitidos por los clientes muestran un alto grado de satisfacción con el sistema propuesto.
- La aplicación de métricas mostró un alto grado de especificidad en los requisitos y un alto grado de calidad en el DCUS a partir de los atributos completitud, consistencia, correctitud y complejidad.
- A través de la captura de requisitos y generación de los artefactos correspondientes al rol de analista se garantizó el entendimiento común entre clientes y desarrolladores, dando así cumplimiento al objetivo del trabajo.

Recomendaciones.

A pesar de que se les dio cumplimiento a todos los objetivos generales de este trabajo, en el transcurso de su desarrollo han surgido nuevas ideas que podrían ampliar y mejorar las funcionalidades del sistema propuesto, para ello se recomienda:

- Que a partir de los artefactos generados en el trabajo se continúe el ciclo de desarrollo para implementar el sistema.
- Realizar próximas iteraciones en las que se conciba la funcionalidad de Reportes, de forma genérica.
- Incorporar la funcionalidad de zonificación automática, una vez completada todas las capacidades de alojamiento de la residencia de la UCI, con el fin de mejorar el proceso de alojamiento

Referencias bibliográficas.

Adagio C/S. Disponible en:

http://www.gestionhotel.net/administracion_hotel_control_gestion/adagio_cs_manejo_servidos_hotel_restaurant.php

Adagio LT. Disponible en:

http://www.gestionhotel.net/pragrama_gestion_hotel_software/adagio_lt_control_hotel_gestion.php

Adagio. Sistema de Gestión Hotelera. Disponible en: <http://www.gestionhotel.net/>

Ayuda extendida del Rational Rose.

CALABRIA, L. *Metodología FDD*, 2003.

CICE. http://tesis.uci.cu/viewpage.php?page_id=21, 2006.

Conferencia de Ingeniería de Software. Patrones de diseño. 2005-2006.

Conferencia de Ingeniería de Software .Introducción a la Ingeniería de Software. 2005-2006.

Conferencia de Ingeniería de Software. Fase de Inicio. Modelo del Negocio., 2005-2006.

CONSULTORES, G. *Disciplina de administración del proyecto - M.S.F.*, 2006. [Disponible en:

<http://www.gpicr.com/msf.aspx>

Diccionarios online. Disponible en: <http://www.diccionarios.com/consultas.php>

GOTTLER PABLO, G. M. D., BIOUL, MAUREEN. *Ingeniería de Requisitos. Elicitación de requisitos*, 2003.

GUERRERO, H. C., <http://hancocchi.net/el-rol-del-analista-en-rup/>.

GUNNAR ÖVERGAARD, K. P. *Use Cases Patterns and Blueprints*, 2004.

INDUDATA. Disponible en: http://www.indudata.com/1rational_rose.htm

IVAR JACOBSON, G. B., JAMES RUMBARUGH *El Proceso Unificado de Desarrollo de Software*.

MARÍA JOSEFA ESCALONA, N. K. *Ingeniería de Requisitos en Aplicaciones para la Web – Un estudio comparativo*, 2002.

MOLPECERES, A. *Procesos de desarrollo: RUP, XP y FDD*, 2003.

Newhotel. Disponible en: http://www.newhotel.com/produtos.asp?Sld_ling=sp&Op_num=2&SubOp_num=0

PATRICIO LETELIER TORRES, E. A. S. L. *Metodologías Ágiles en el desarrollo del Software.*, 2003.

PRESSMAN, R. S. *Ingeniería del Software. Un enfoque práctico.*, 2005. p

Reserva en Línea de Hoteles en Cuba. Disponible en: <http://www.cuba.cu/reserva/hoteles/hoteles.php>

REYNOSO, C. *Métodos Heterodoxos en Desarrollo de Software*, 2004.

RUMBAUGH, J. Y. J. *El Lenguaje Unificado de Modelado*, 2000.

SANCHEZ, M. A. M. *Metodologías De Desarrollo De Software*, 2004. [Disponible en:
[http://www.informatizate.net/articulos/metodologias de desarrollo de software 07062004.html](http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html)

TORRES, J. L. Disponible en: <http://www.ewh.ieee.org/r9/guadalajara/boletin/sep01/requerimientos.htm>

Visio 2003. Disponible en: <http://www.microsoft.com/spain/office/products/visio/faq.msp#EHAAC>

Glosario de términos

CASE: Herramienta para la Ingeniería de Software Asistida por Ordenador.

Casos de Uso: Una descripción de un conjunto de secuencias de acciones, incluyendo variaciones, que un sistema lleva a cabo y que conduce a un resultado observable de interés para un actor determinado.

Estructura: Se denominan estructuras al grupo de (Residencia, Manzana, Edificio, Apartamento)

Especificaciones: Colección de escenarios de uso con descripciones en lenguaje natural.

Experticia: Persona que tiene muchos conocimientos en una materia.

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

Include: Relación de inclusión entre dos casos de uso.

Manzana: Es un área de la residencia, donde se encuentran, varios edificios. Es válido destacar que la residencia está formada por más de una manzana.

Residencia: Áreas en las que se encuentran varios edificios de la universidad, destinados al Alojamiento de personas, una residencia tiene el control del alojamiento 3 o 4 facultades.

Zonificación: Es la distribución de todos los edificios entre las residencias existentes, especificando cuales pertenecen a cada facultad.