

**Universidad de las Ciencias Informáticas
Facultad # 4**



**Título: Sistema de Gestión de Residencia. Módulo de gestión
para el control y asignación de avituallamiento y lavandería.**

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

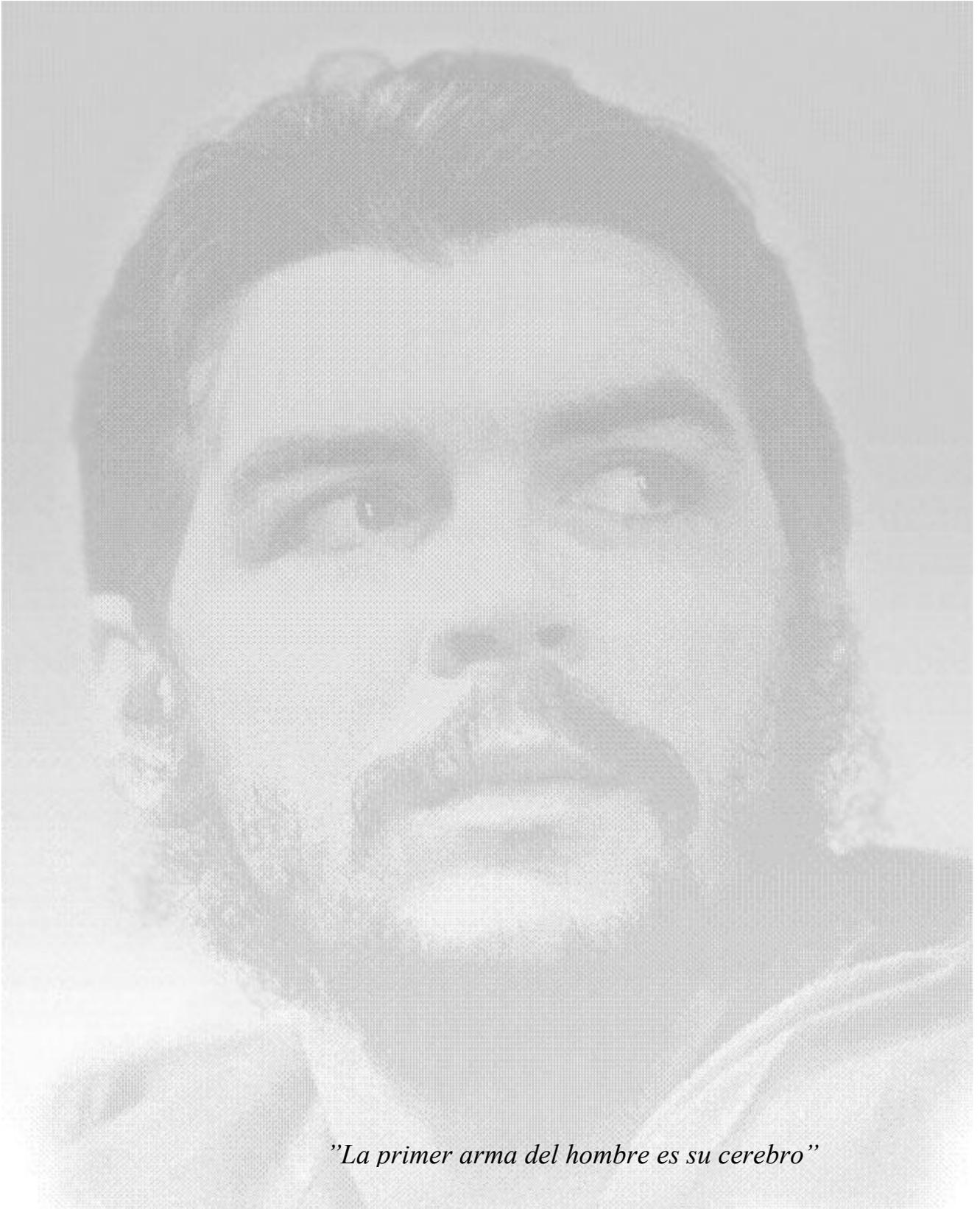
Autor: Ania Liset Aleaga Naranjo

Tutor: Msc. Rolando Quintana Aput

Consultante: Ing. Yosvany Medina Hernández

Ciudad de La Habana, Junio de 2008

“Año 50 de la Revolución”



"La primer arma del hombre es su cerebro"

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Ania Liset Aleaga Naranjo

Rolando Quintana Aput

Firma del Autor

Firma del Tutor

Datos de Contacto

Nombre y Apellidos: MSc Rolando Quintana Aput

Fecha de nacimiento: 19 de Junio de 1976

Institución: Universidad de las Ciencias Informáticas

País: Cuba.

Ciudadanía: Cubana.

Carnet de Identidad: 76061921366

Correo: rgaput@uci.cu

Situación laboral: Profesor Instructor, Departamento de la Especialidad de la facultad #4

Institución: Universidad de las Ciencias Informáticas

Dirección: Carretera San Antonio de los Baños, Torrens, Municipio Boyeros, Ciudad de La Habana, Cuba.

Código postal: 19370.

Resumen del Currículum

Ingeniero en Informática del Instituto Superior Politécnico José Antonio Echeverría, desde Julio del 2004 Máster en Gestión de Proyectos Informáticos desde Diciembre de 2007 de la universidad de Las Ciencias Informáticas. Al graduarse pasa a ser profesor de la Universidad de las Ciencias Informáticas, en la Disciplina de Técnicas de programación. Obtiene la categoría de Instructor en octubre del 2005 y actualmente está en el proceso de categorización optando por la categoría de Asistente. Tiene una experiencia como profesor de 5 cursos en las asignaturas, Introducción a la Programación, Programación I, Programación II, Seguridad Informática, Historia de la Informática y metodología de la Investigación además de cursos optativos sobre PHP y HTML. Ha trabajado en investigaciones relacionadas a la investigación de operaciones y competencias de las personas que ocupan roles en los proyectos multimedia. Tiene 2 artículos publicados en memorias de eventos científicos nacionales y uno internacional. Ha ejercido como tutor y miembro de tribunal de varias tesis de pregrado y en estos momentos participa como tribunal en tesis de postgrado.

AGRADECIMIENTOS

Agradezco:

A mis padres por su sacrificio y amor, sin ustedes nada de esto fuera posible.

A mis tíos Lety y Moya, por sus consejos en cada momento, por haber estado para mí siempre incondicionalmente.

A mis abuelos, a mi familia en general, gracias.

A mi tutor Rolando Quintana, sin su apoyo y orientación no hubieran sido posibles estos resultados, gracias.

A Yoerkis, por ser quien es y lo que significa su amistad para mí, porque lo quiero mucho mucho.

A Yanoska (mi chinita) y a Jorge Carlos (Guancho) por haberme regalado un quinto año tan feliz, por su apoyo en todo momento.

A todas las amistades que conocí en este tiempo, que en cada momento se preocuparon por mi tesis, a Carlos, Maylin, Orliandi, Analia, Asdrubal, Meylin, Daymí, Daylis...a todos gracias por este tiempo, por los buenos y duros momentos.

A Yusimi Rosales León por su ayuda en la captura de requisitos, en el trabajo con la tesis, muchas gracias, ojalá te pudieras graduar conmigo.

A todo el que de una forma u otra puso su granito de arena en mi formación como profesional.

A todos...gracias.

DEDICATORIA

Quisiera dedicar este momento tan especial a tita y a papá por ser mi fuente de inspiración y por ser ellos los que con su ejemplo me motivaron en cada momento durante estos 5 años lejos de casa.

A mi hermanita que tanto quiero y para quien soy un ejemplo en todo momento.

A mi tía Lety y mi tío Moya por haber sido mis segundos padres durante este tiempo, por su cariño incondicional, por su apoyo, por ser ellos...

A mis abuelitos que tanto se preocupan por mí y supieron estar pendientes en todo momento de mis estudios.

A todos mis seres queridos en general...

RESUMEN

La Universidad de las Ciencias Informáticas (UCI) tiene entre sus objetivos la informatización de todas sus áreas, con el propósito de mejorar el funcionamiento de los procesos que tienen lugar en el centro de altos estudios. El presente trabajo de diploma está dirigido al proceso de Avituallamiento y Lavandería de la residencia en la UCI. Para ello se realiza el análisis de una aplicación Web que automatice las actividades vinculadas a este proceso.

Se ha realizado un estudio profundo de las técnicas de captura y validación de requisitos, así como de las tendencias en el modelado de software y un análisis exhaustivo del negocio y funcionamiento del proceso en la residencia.

Para dar cumplimiento con calidad a este proceso, se ha utilizado la metodología de desarrollo de software RUP, usando UML como lenguaje de modelado y Visual Paradigm como herramienta CASE que permitirá la modelación gráfica del sistema.

Utilizando toda esta tecnología y siguiendo los pasos requeridos se ha especificado y documentado el negocio con todas sus actividades. De igual manera se ha llevado a cabo un levantamiento de requisitos tanto funcionales como no funcionales, los cuales definen las capacidades que el sistema debe cumplir, más las cualidades que el producto debe tener, cumpliendo con las necesidades reales del cliente. Y a partir del refinamiento de dichos requisitos se ha obtenido la modelación y especificación del sistema propuesto.

PALABRAS CLAVE

Requisito, Ingeniería.

INDICE

INTRODUCCION 1

CAPITULO 1: FUNDAMENTACION TEORICA 6

 1.1 INTRODUCCION 6

 1.2 CAPTURA DE REQUISITOS..... 7

 1.3 TÉCNICAS DE CAPTURA DE REQUERIMIENTOS UTILIZADAS EN EL PROCESO DE
DESARROLLO DE TODO TIPO DE SOFTWARE: 8

 1.4 DEFINICION DE REQUISITOS..... 11

 1.5 ESPECIFICACION DE REQUISITOS DE SOFTWARE 12

 1.6 VALIDACION DE REQUISITOS..... 13

 1.7 TRATAMIENTO DE REQUISITOS EN LA PROPUESTA PARA LA WEB 14

 1.7.1 Requisitos de datos 14

 1.7.2 Requisitos de interfaz (al usuario)..... 14

 1.7.3 Requisitos navegacionales 14

 1.7.4 Requisitos de personalización 14

 1.7.5 Requisitos transaccionales o funcionales internos 15

 1.7.6 Requisitos no funcionales 15

 1.8 SISTEMAS DE GESTION 15

 1.9 TECNICAS Y HERRAMIENTAS UTILIZADAS EN CADA FASE 15

 1.10 ANALISIS DE REQUERIMIENTOS 17

 1.10.1 Identificar Casos de Uso del sistema..... 18

 1.10.2 Dar detalle a los casos de uso descritos 19

 1.10.3 Definir una interfaz inicial del sistema (si es aplicable)..... 19

 1.11 JUSTIFICACION DE HERRAMIENTAS A UTILIZAR 20

 1.11.1 Software Libre 20

 1.11.2 Linux..... 21

 1.11.3 Metodologías de desarrollo 22

 1.12 Herramientas CASE 25

 1.13 LENGUAJE DE MODELACION 27

 1.14 ANALISTA DE SISTEMAS: ROL DEFINIDO POR RUP 29

 1.15 CONCLUSIONES 29

CAPITULO 2: CARACTERISTICAS DEL SISTEMA 30

 2.1 INTRODUCCION 30

 2.2 EXPEDIENTE DEL PROYECTO..... 30

2.3 PASOS A SEGUIR EN LA ESPECIFICACION DE LOS REQUISITOS.....	31
2.3.1 Propuesta general para la captura de requisitos.....	33
2.4 OBJETO DE AUTOMATIZACION.....	34
2.4.1 Situación problemática.....	34
2.4.2 Flujo actual de los procesos.....	34
2.4 SISTEMAS AUTOMATIZADOS VINCULADOS CON EL CAMPO DE ACCIÓN	35
2.5 INFORMACIÓN QUE SE MANEJA.....	35
2.6 MODELO DE NEGOCIO.....	36
2.6.1 Actor del negocio.....	37
2.6.2 Trabajador del negocio.....	37
2.6.3 Reglas del negocio.....	38
2.6.4 Diagrama de casos de uso del negocio.....	38
2.6.5 Descripciones de casos de uso del negocio.....	39
2.6.6 Diagramas de actividades.....	45
2.6.6 Modelo de objetos del negocio.....	50
2.7 ESPECIFICACION DE REQUERIMIENTOS DE SW	51
2.7.1 Requerimientos funcionales.....	52
2.7.2 Requerimientos no funcionales	53
2.8 SISTEMA	56
2.8.1 Actores del sistema.....	56
2.8.2 Diagrama de casos de uso del sistema.....	56
2.8.2 Especificación de los CUS.....	57
2.9 CONCLUSIONES	75
CAPÍTULO 3: ANÁLISIS DEL SISTEMA.....	76
3.1 INTRODUCCION	76
3.2 ANALISIS.....	76
3.4 DIAGRAMAS DE INTERACCION.....	80
3.5 DIAGRAMAS DE SECUENCIA PARA EL ANALISIS DEL SISTEMA.....	80
3.6 CONCLUSIONES	87
CONCLUSIONES.....	88
RECOMENDACIONES.....	89
REFERENCIAS BIBLIOGRAFICAS.....	90

INTRODUCCION

La comunicación actual entre dos o más personas es el resultado de siglos de desarrollo de medios de expresión, que no es más que el proceso de transmisión y recepción de ideas, información y mensajes. Esta permite ejercer cierta influencia sobre otras personas con el fin de obtener un objetivo específico, por lo que es prioritario que la misma sea efectiva. Con el de cursar del tiempo y el desarrollo de las civilizaciones, han seguido evolucionando los sistemas de comunicación, hasta lograr hoy la poderosa herramienta que da soporte a la comunicación: Internet.

Internet es uno de los fenómenos que más impacta al hombre de la sociedad tecnológica. El acelerado desarrollo en progresión, desconcierta y conmueve a muchos de forma sorprendente. Tiene un impacto profundo en el trabajo, el ocio y el conocimiento, gracias a esto millones de personas tienen acceso a un gran volumen de información y se acortan distancias. En las dos últimas décadas, la reducción de los tiempos de transmisión de la información a distancia y de acceso a la información ha supuesto un reto esencial de nuestra sociedad. La comunicación a través de las computadoras es uno de los avances más espectaculares dentro de las comunicaciones, se produce en el campo de la tecnología de los ordenadores. Desde la aparición de las computadoras digitales en la década de 1940, éstas se han introducido en los países desarrollados en prácticamente todas las áreas de la sociedad (industrias, negocios, hospitales, escuelas, transportes, hogares o comercios). Mediante la utilización de las redes informáticas y los dispositivos auxiliares, el usuario de un ordenador puede transmitir datos con gran rapidez. Estos sistemas pueden acceder a multitud de bases de datos. A través de la línea telefónica se puede acceder a toda esta información y visualizarla en pantalla o en un televisor convenientemente adaptado. El mundo empresarial no escapa de esta actividad y necesita involucrarse y promover el desarrollo de la informatización y sus actividades relacionadas, para facilitar el intercambio de información y contribuir de manera total a la transferencia de conocimientos y tecnología.

El software se ha convertido en el elemento clave de la evolución de los sistemas y productos informáticos. En los últimos 50 años el software ha pasado de ser una resolución de problemas especializada y una herramienta de análisis de información, a ser una industria por sí misma (PRESSMAN, 2005). La conveniente práctica de la Ingeniería de Software (IS) sustenta la idea de obtener un proceso de construcción de software que descansa sobre una organización de calidad, sistemático y que abarque todo el ciclo de vida del desarrollo de software. Esto evitaría los errores de un proceso de desarrollo inmaduro, centrado en la etapa de implementación y no en todo el ciclo de

vida, además de un software difícil de mantener, ya sea por su inaccesibilidad o por su alto costo.(FERNÁNDEZ, 2006)

En la actualidad la economía de todas las naciones desarrolladas depende del software(SOMMERVILLE, 2004). La gestión empresarial y administrativa no se encuentra exenta al mencionado proceso evolutivo del software, lo cual estimula el uso de las mismas en la automatización y modernización de sus procesos y actividades. Todo esto con el simple objetivo de controlar y perfeccionar mediante la tecnología, el manejo de sus recursos y materiales. Sin olvidar además, la marcada importancia de los métodos tradicionales y de organización en los que se basa la gestión empresarial y de mercado; con las cuales se obtienen aún notables resultados y que las nuevas tecnologías vienen a soportar.

En el ambiente de las comunicaciones se hace referencia al gran impacto de las llamadas tecnologías Web que ostentan características revolucionarias sobre múltiples sectores económicos y hasta de aspecto social y el funcionamiento de la sociedad en su conjunto.

El avance de Internet y las comunicaciones de los últimos años ha provocado un interés creciente por el desarrollo de propuestas metodológicas que ofrezcan un marco de referencia adecuado cuando se desarrolla un sistema de información Web.

La informática, y dentro de ella los sistemas Web propiamente dicho, se utilizan en diversidad de tareas y ramas de la sociedad, por ejemplo: elaboración de documentos, reportes y un sinnúmero de control de procesos, robots industriales, telecomunicaciones y vigilancia, además del desarrollo de juegos y multimedia.

El desarrollo de sistemas en todos los sectores de la sociedad cobra gran auge en la actualidad, tratando así de automatizar la mayor cantidad de procesos posibles, una sociedad para ser más eficaz, eficiente y competitiva, debe aplicar la informatización en todas las esferas y procesos. En este sentido, la creciente Industria cubana del Software lucha por convertirse en una de las mayores fuentes de ingreso del país. Juega un papel fundamental el capital humano que se está formando en la UCI, vinculando desde edades tempranas a los estudiantes con la producción de software.

La residencia universitaria cuenta ya con una matrícula de 10 000 estudiantes y alrededor de los 2000 profesores, con aproximadamente 135 edificios y una suma cercana a los 1300 apartamentos. Debido a la gran capacidad de alojamiento con que se cuenta es que surge la vicerrectoría de residencia, que cuenta con un vicerrector y 4 direcciones, tres de estas para la beca de estudiantes y la otra para los profesores. Se trabaja en base a la informatización de los procesos llevados a cabo

en el área de residencia, tanto estudiantil como de los profesores que en ella residen. Las áreas rectoras de todo el gran proceso que se lleva a cabo en dicha residencia son, precisamente, tres direcciones para los estudiantes y una para los profesores. Dentro del marco de actividades y procesos llevados a cabo en estas direcciones, tiene gran importancia el proceso de control de entrega y recepción de los medios de avituallamiento y de lavandería, debido además al gran y creciente volumen de internos. Surge entonces la necesidad de automatizar estos procesos e integrarlos a un sistema rector de todos los procesos de la residencia en la UCI.

Como parte de los procesos de avituallamiento y lavandería, se maneja un gran volumen de información, dicha información es enviada por varias vías de manera personal y por teléfono, por lo que es muy lento el flujo de información de un nivel hacia otro, producto de la ausencia de un sistema automatizado que agilice este proceso, se esta apostando por un sistema que posibilite la conexión entre todos los locales y la dirección. En este sentido surge el siguiente **problema**: ¿Como facilitar el diseño y la implementación del módulo de avituallamiento y lavandería del Sistema de Gestión de Residencia? Se planteó como **situación problémica** el hecho de que todo el trabajo es desarrollado de forma manual, por lo que se torna muy grande el cúmulo de información en papeles y son muy frecuentes los errores que se cometen.

Se propone entonces como **objetivo general** de la investigación obtener captura de requisitos y análisis del proceso de avituallamiento y lavandería de la residencia de la UCI utilizando para ello los artefactos y documentos identificados como necesarios según el negocio a automatizar.

Se definió como **objeto de estudio** los procesos de gestión de información llevados a cabo en la residencia universitaria, derivándose de esta forma como **campo de acción** los procesos de avituallamiento y lavandería que se desarrollan en la misma.

Se plantearon **objetivos específicos** para una mejor comprensión y organización del trabajo. Valorar las técnicas y artefactos a utilizar en la captura de requisitos. Proponer secuencia de pasos a seguir en la captura de requisitos a partir de experiencias prácticas y el análisis de las metodologías de desarrollo. Especificar flujo actual de procesos en el entorno de negocio del cliente. Obtener el modelo del negocio. Obtener los artefactos del análisis del proceso de avituallamiento y lavandería. Arrojando estos objetivos como **posibles resultados** obtener el modelo del negocio del proceso de avituallamiento y lavandería de la residencia de la UCI, así como la captura de requisitos y el análisis, obteniendo de esta forma un prototipo no funcional del sistema. Y a partir de esto dejar listos todos los documentos y artefactos que permitan iniciar la fase de Diseño.

Siempre que se quiera realizar un trabajo con éxito es muy importante planificarse tanto en cuanto a recursos como el tiempo necesario para obtener un producto de la calidad requerida. Las **tareas a cumplir** serían: Realizar consultas bibliográficas que permitan elaborar el marco teórico conceptual. Identificar las necesidades del cliente. Seleccionar las técnicas de captura de requisitos que más se adapten al cliente con el que se trabaja, a partir del estudio de todas las que se encuentran descritas en la literatura. Identificar los artefactos que se utilizarán para documentar a partir del estudio de la fase de captura de requisitos y análisis de la metodología RUP. Estudiar el proceso de Avituallamiento y lavandería para su modelado. Definir los requisitos funcionales y no funcionales el proceso de Avituallamiento y lavandería. Realizar el análisis del modelo del negocio del proceso de Avituallamiento y lavandería.

Con el objetivo de dar un correcto cumplimiento a las tareas propuestas, para una mejor comprensión de los procesos y el desarrollo de un estudio profundo de técnicas, herramientas y metodologías de desarrollo se utilizarán como métodos científicos de investigación: **Teórico y Empírico.**

Teórico:

- *Método Analítico-Sintético:* permite analizar y comprender las partes e interpretar la teoría con el fin de extraer los elementos más importantes que se relacionan con el objeto.
- *Método Inductivo-Deductivo:* permite, a través de un razonamiento lógico, llegar a obtener un grupo de conocimientos particulares y generales.
- *Método Histórico-Lógico:* permite observar la trayectoria de determinados fenómenos en diferentes períodos de la historia y, de esta forma, revelar las etapas principales de su desenvolvimiento y las conexiones históricas fundamentales.
- *Método de la Modelación:* permite la creación de modelos o patrones, es decir representar esquemáticamente el objeto de estudio.

Empírico:

- *Método de la Observación:* permite contemplar el desarrollo de la investigación en distintos momentos de la misma, conociendo el problema y el objeto de investigación, estudiando su curso natural y recoge la información de cada uno de los conceptos o variables definidas en la hipótesis.

- *Método la Entrevista:* permite obtener la mayor información posible acerca del tema de investigación, además de las experiencias, las ideas y los puntos de vistas de los entrevistados que aportan conocimientos específicos del tema.

El trabajo se desarrollará en tres capítulos. Se concibieron las siguientes actividades para un mejor cumplimiento de las tareas y objetivos específicos:

- División del levantamiento de requisitos en dos etapas.
- Validación de los procesos y casos de uso identificados.
- Descripción de diagramas de actividades de los procesos en Visual Paradigm.
- Centrar la atención en la descripción de los casos de uso y no tanto en el prototipo elemental de usuario.

CAPITULO 1: FUNDAMENTACION TEORICA

1.1 INTRODUCCION

En el presente capítulo se brinda una explicación de importantes conceptos de necesario dominio para la creación de una aplicación Web y para entender correctamente en qué consiste la misma, además de la ilustración de cómo se lleva a cabo el proceso de especificación de requisitos y la justificación de las herramientas a utilizar.

Se presentará el estado del arte de las propuestas que ofrecen las metodologías para la Web en el proceso de definición de requisitos. Se comenzará planteando la estructura básica de dicho proceso y las técnicas más comunes aplicadas de forma clásica en la ingeniería de requisitos.

Existen grupos que han trabajado y propuesto metodologías que ofrecen procesos, modelos y técnicas adecuadas para trabajar con sistemas basados en tecnologías Web; pero la gran mayoría de ellas enfocan su trabajo a la etapa de diseño del ciclo de vida, dando menor importancia a la ingeniería de requisitos.(ESCALONA, TORRES, MEJÍAS, GUTIÉRREZ, VILLADIEGO, 2007)

El tratamiento de requisitos es el proceso mediante el cual se especifican y validan los servicios que debe proporcionar el sistema así como las restricciones sobre las que se deberá operar. Consiste en un proceso iterativo y cooperativo de análisis del problema, documentando los resultados en una variedad de formatos y probando la exactitud del conocimiento adquirido.(FERREIRA, 2002) La importancia de esta fase es esencial puesto que los errores más comunes y más costosos de reparar, así como los que más tiempo consumen se deben a una inadecuada ingeniería de requisitos.

La definición de las necesidades del sistema es un proceso complejo, pues en él hay que identificar los requisitos que el sistema debe cumplir para satisfacer las necesidades de los usuarios finales y de los clientes. Para realizar este proceso, no existe una única técnica estandarizada y estructurada que ofrezca un marco de desarrollo que garantice la calidad del resultado. Existe en cambio un conjunto de técnicas, se debe tener en cuenta que la selección de las técnicas y el éxito de los resultados que se obtengan, depende en gran medida tanto del equipo de análisis y desarrollo, como de los propios clientes o usuarios que en ella participen.

Para su mejor comprensión el proceso de especificación de requisitos se puede dividir en tres grandes actividades(LOWE, 1999):

- 1- Captura de requisitos.

2- Definición de requisitos.

3- Validación de requisitos.

El proceso comienza con la realización de la captura de requisitos, el grupo de analistas toma la información suministrada por los usuarios y clientes. Esta información puede provenir de fuentes muy diversas: documentos, aplicaciones existentes, a través de entrevistas. En base a esta información, el equipo de desarrollo elabora el catálogo de requisitos. Finalmente con la validación de requisitos se realiza la valoración de los mismos, comprobando si existen inconsistencias, errores o si faltan requisitos por definir. El proceso de definición-validación es iterativo y en algunos proyectos complejos resulta necesario ejecutarlo varias veces. (KOCH, NORA, ESCALONA, MARÍA JOSÉ 2002)

Es necesario definir el término "requisito" que será empleado indistintamente en el presente documento. Este término ha sido concebido de varias maneras por diferentes autores aunque en sí, su significado es el mismo. Solo se hace referencia al concepto que proporciona la IEEE (Institute of Electrical and Electronics Engineers - Instituto de Ingenieros Eléctricos y Electrónicos), pues se considera que es una de las definiciones más completa.

De acuerdo con la IEEE un requisito es (a) una condición o capacidad que un usuario necesita para resolver un problema o lograr un objetivo, (b) una condición o capacidad que debe tener un sistema o un componente de un sistema para satisfacer un contrato, una norma, una especificación u otro documento formal, (c) una representación en forma de documento de una condición o capacidad como las expresadas en (a) o en (b). (BARESI L., 2001)

La forma más segura de garantizar que el software cumpla con lo establecido, es especificar detalladamente lo que se debe hacer y cómo se debe hacer, de esta manera se propicia un mecanismo para respaldar el trabajo realizado ante cambios y reclamos en etapas posteriores. Además, la especificación de los requisitos le brinda al desarrollador y al cliente los medios de evaluar la calidad una vez que el software ha sido entregado. (ESCALONA, M.J., TORRES, J., MEJÍAS, M, 2002)

1.2 CAPTURA DE REQUISITOS

La captura de requisitos es la actividad mediante la que el equipo de desarrollo de un sistema de software extrae, de cualquier fuente de información disponible, las necesidades que debe cubrir dicho sistema. El proceso de captura de requisitos puede resultar complejo, principalmente si el entorno de trabajo es desconocido para el equipo de analistas, y depende mucho de las personas

que participen en él. Por la complejidad que todo esto puede implicar, la ingeniería de requisitos ha trabajado en el desarrollo de técnicas que permitan hacer este proceso de una forma más eficiente y precisa. (KOCH, NORA, ESCALONA, MARÍA JOSÉ 2002)

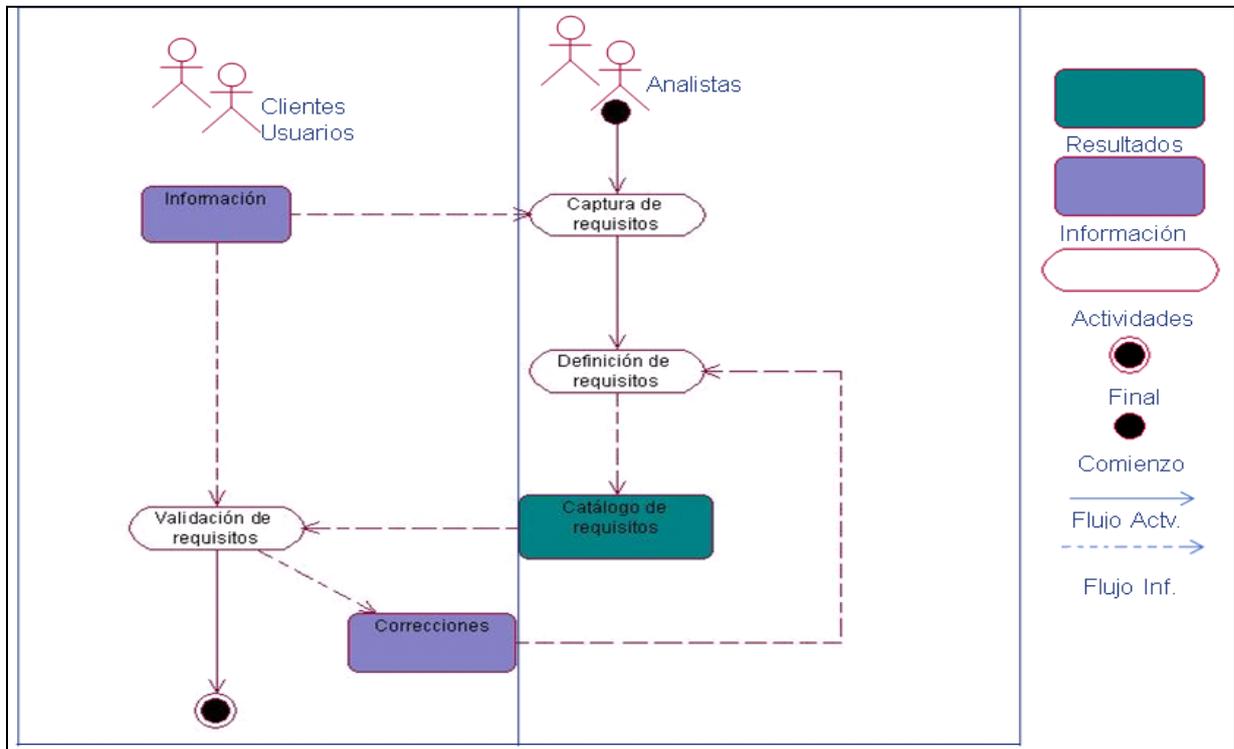


Fig. 1.1 Proceso de Ingeniería de Requisitos.

1.3 TÉCNICAS DE CAPTURA DE REQUERIMIENTOS UTILIZADAS EN EL PROCESO DE DESARROLLO DE TODO TIPO DE SOFTWARE:

La captura de requisitos es la actividad mediante la que el equipo de desarrollo de un sistema de software extrae, de cualquier fuente de información disponible, las necesidades que debe cubrir dicho sistema. (DÍEZ, 2001)

- **Entrevistas:** resultan una técnica muy aceptada dentro de la ingeniería de requisitos y su uso está ampliamente extendido. Las entrevistas le permiten al analista tomar conocimiento del problema y comprender los objetivos de la solución buscada. A través de esta técnica el equipo de trabajo se acerca al problema de una forma natural. Existen muchos tipos de entrevistas y son muchos los autores que han trabajado en definir su estructura y dar guías para su correcta realización (DURÁN A., 1999) (PAN, 2001) Básicamente, la estructura de la entrevista abarca tres

pasos: identificación de los entrevistados, preparación de la entrevista, realización de la entrevista y documentación de los resultados (protocolo de la entrevista).

A pesar de que las entrevistas son esenciales en el proceso de la captura de requisitos y con su aplicación el equipo de desarrollo puede obtener una amplia visión del trabajo y las necesidades del usuario, es necesario destacar que no es una técnica sencilla de aplicar (PAN, 2001). Requiere que el entrevistador sea experimentado y tenga capacidad para elegir bien a los entrevistados y obtener de ellos toda la información posible en un período de tiempo siempre limitado. Aquí desempeña un papel fundamental la preparación de la entrevista.

- **JAD (Joint Application Development/Desarrollo conjunto de aplicaciones):** esta técnica resulta una alternativa a las entrevistas. Es una práctica de grupo que se desarrolla durante varios días y en la que participan analistas, usuarios, administradores del sistema y clientes (IBM, 1997). Está basada en cuatro principios fundamentales: dinámica de grupo, el uso de ayudas visuales para mejorar la comunicación, mantener un proceso organizado y racional una filosofía de documentación WYSIWYG (What You See Is What You Get, lo que ve es lo que obtiene), es decir, durante la aplicación de la técnica se trabajará sobre lo que se generará.

Tras una fase de preparación del JAD al caso concreto, el equipo de trabajo se reúne en varias sesiones. En cada una de ellas se establecen los requisitos de alto nivel a trabajar, el ámbito del problema y la documentación. Durante la sesión se discute en grupo sobre estos temas, llegando a una serie de conclusiones que se documentan. En cada sesión se van concretando más las necesidades del sistema. Esta técnica presenta una serie de ventajas frente a las entrevistas tradicionales, ya que ahorra tiempo al evitar que las opiniones de los clientes se tengan que contrastar por separado, pero requiere un grupo de participantes bien integrados y organizados.

- **Brainstorming (Tormenta de ideas):** es también una técnica de reuniones en grupo cuyo objetivo es que los participantes muestren sus ideas de forma libre (RAGHAVAN, 1994). Consiste en la mera acumulación de ideas y/o información sin evaluar las mismas. El grupo de personas que participa en estas reuniones no debe ser muy numeroso (máximo 10 personas), una de ellas debe asumir el rol de moderador de la sesión, pero sin carácter de controlador. Como técnica de captura de requisitos es sencilla de usar y de aplicar, contrariamente al JAD, puesto que no requiere tanto trabajo en grupo como éste. Además suele ofrecer una visión general de las necesidades del sistema, pero normalmente no sirve para obtener detalles concretos del mismo, por lo que suele aplicarse en los primeros encuentros.

- **Concept Mapping:** Los concept maps son grafos en los que los vértices representan conceptos y las aristas representan posibles relaciones entre dichos conceptos. Estos grafos de relaciones se desarrollan con el usuario y sirven para aclarar los conceptos relacionados con el sistema a desarrollar. Son muy usados dentro de la ingeniería de requisitos, pues son fáciles de entender por el usuario, más aún si el equipo de desarrollo hace el esfuerzo de elaborarlo en el lenguaje de éste. Sin embargo, deben ser usados con cautela porque en algunos casos pueden ser muy sugestivos y pueden llegar a ser ambiguos en casos complejos, si no se acompaña de una descripción textual (PAN, 2001).

- **Sketches y Storyboards:** Está técnica es frecuentemente usada por los diseñadores gráficos de aplicaciones en el entorno Web. La misma consiste en representar sobre papel en forma muy esquemática las diferentes interfaces al usuario (sketches). Estos sketches pueden ser agrupados y unidos por enlaces dando idea de la estructura de navegación (storyboard).

- **Casos de Uso:** Aunque inicialmente se desarrollaron como técnica para la definición de requisitos (JACOBSON, I, 1995), algunos autores proponen casos de uso como técnica para la captura de requisitos (LIU, 2001; PAN, 2001). Los casos de uso permiten mostrar el contorno (actores) y el alcance (requisitos funcionales expresados como casos de uso) de un sistema. Un caso de uso describe la secuencia de interacciones que se producen entre el sistema y los actores del mismo para realizar una determinada función. Los actores son elementos externos (personas, otros sistemas, etc.) que interactúan con el sistema como si de una caja negra se tratase. Un actor puede participar en varios casos de uso y un caso de uso puede interactuar con varios actores.

La ventaja esencial de los casos de uso es que resultan muy fáciles de entender para el usuario o cliente, sin embargo carecen de la precisión necesaria (INSFRÁN, 2002; VILAIN, 2000) si no se acompañan con una información textual o detallada con otra técnica como pueden ser los diagramas de actividades (UML, 2001).

- **Cuestionarios y Checklists:** Esta técnica requiere que el analista conozca el ámbito del problema en el que está trabajando. Consiste en redactar un documento con preguntas cuyas respuestas sean cortas y concretas, o incluso cerradas por unas cuantas opciones en el propio cuestionario (Checklist). Este cuestionario será cumplimentado por el grupo de personas entrevistadas o simplemente para recoger información en forma independiente de una entrevista.

- **Comparación de terminología:** Uno de los problemas que surge durante la licitación de requisitos es que usuarios y expertos no llegan a entenderse debido a problemas de terminología. Esta técnica es utilizada en forma complementaria a otras técnicas para obtener consenso respecto

de la terminología a ser usada en el proyecto de desarrollo. Para ello es necesario identificar el uso de términos diferentes para los mismos conceptos correspondencia), misma terminología para diferentes conceptos (conflictos) o cuando no hay concordancia exacta ni en el vocabulario ni en los conceptos (contraste) (PAN, 2001).

1.4 DEFINICION DE REQUISITOS

Para la actividad de definición de requisitos en el proceso de ingeniería de requisitos hay un gran número de técnicas propuestas. Se describen brevemente las más relevantes para este trabajo (KOCH, NORA, ESCALONA, MARÍA JOSÉ 2002).

- **Lenguaje natural:** Resulta una técnica muy ambigua para la definición de los requisitos. Consiste en definir los requisitos en lenguaje natural sin usar reglas para ello. A pesar de que son muchos los trabajos que critican su uso, es cierto que a nivel práctico se sigue utilizando.

- **Glosario y ontologías:** La diversidad de personas que forman parte de un proyecto software hace que sea necesario establecer un marco de terminología común. Esta necesidad se vuelve más patente en los sistemas de información Web puesto que el equipo de desarrollo en ellas suele ser más interdisciplinario (KOCH, NORA, 2001). Por esta razón son muchas las propuestas que abogan por desarrollar un glosario de términos en el que se recogen y definen los conceptos más relevantes y críticos para el sistema. En esta línea se encuentra también el uso de ontologías, en las que no sólo aparecen los términos, sino también las relaciones entre ellos.

- **Plantillas o patrones:** Esta técnica, recomendada por varios autores (DURÁN A., 1999; ESCALONA, M.J., TORRES, J., MEJÍAS, M, 2002), tiene por objetivo el describir los requisitos mediante el lenguaje natural pero de una forma estructurada. Una plantilla es una tabla con una serie de campos y una estructura predefinida que el equipo de desarrollo va cumplimentando usando para ello el lenguaje del usuario. Las plantillas eliminan parte de la ambigüedad del lenguaje natural al estructurar la información; cuanto más estructurada sea ésta, menos ambigüedad ofrece. Sin embargo, si el nivel de detalle elegido es demasiado estructurado, el trabajo de rellenar las plantillas y mantenerlas, puede ser demasiado tedioso.

- **Escenarios:** La técnica de los escenarios consiste en describir las características del sistema a desarrollar mediante una secuencia de pasos (LIU, 2001). La representación del escenario puede variar dependiendo del autor. Esta representación puede ser casi textual o ir encaminada hacia una representación gráfica en forma de diagramas de flujo (WEIDENHAUPT, 1999). El análisis de los escenarios, hechos de una forma u otra, pueden ofrecer información importante sobre las necesidades funcionales de sistema (LOWE, 1999).

- **Casos de uso:** Como técnica de definición de requisitos es como más ampliamente han sido aceptados los casos de uso. Actualmente se ha propuesto como técnica básica del proceso RUP (KRUCHTEN, 1998). Sin embargo, son varios los autores que defienden que pueden resultar ambiguos a la hora de definir los requisitos (DÍEZ, 2001; INSFRÁN, 2002; VILAIN, 2000), por lo que hay propuestas que los acompañan de descripciones basadas en plantillas o de diccionarios de datos que eliminen su ambigüedad.

- **Lenguajes Formales:** Otro grupo de técnicas que merece la pena resaltar como extremo opuesto al lenguaje natural, es la utilización de lenguajes formales para describir los requisitos de un sistema. Las especificaciones algebraicas como ejemplo de técnicas de descripción formal, han sido aplicadas en el mundo de la ingeniería de requisitos desde hace años (PEÑA, 1998). Sin embargo, resultan muy complejas en su utilización y para ser entendidas por el cliente. El mayor inconveniente es que no favorecen la comunicación entre cliente y analista. Por el contrario, es la representación menos ambigua de los requisitos y la que más se presta a técnicas de verificación automatizadas.

1.5 ESPECIFICACION DE REQUISITOS DE SOFTWARE

La Especificación de Requisitos del Software (ERS) es el producto resultante sobre los requisitos e inicialmente se considera en la fase de Inicio, como un complemento para definir el alcance del sistema, luego se refina en un modo incremental, en las fases de Elaboración y Construcción (RUP, 2003).

La especificación, independientemente de la forma en que se realice, puede verse como un proceso de representación. Se centra en la colección y la organización de todos los requisitos que rodean el proyecto. Hace referencia al plan de Gestión de Requisitos para determinar la localización y organización correctas de los requisitos. Por ejemplo, se puede desear tener una ERS separada para describir los requisitos completos del software para cada característica en un comportamiento particular del producto. Esto puede incluir varios casos de uso del modelo de casos de uso del sistema, para describir los requisitos funcionales de esta característica, junto con el sistema relevante de requisitos detallados en especificaciones suplementarias (RUP, 2003).

Debido a que puede ser que los desarrolladores se encuentren con diversas herramientas para recolectar estos requisitos, es importante realizar la colección de requisitos teniendo en cuenta varios artefactos y diversas herramientas. Por ejemplo, puede ser que encuentre apropiado recoger requisitos textuales por ejemplo los requisitos no funcionales, con una herramienta de documentación textual en especificaciones suplementarias. Por otra parte, puede ser que

encuentre útil recoger algunos (o todos) los requisitos funcionales en los casos de uso y puede ser que encuentre práctico utilizar una herramienta apropiada a las necesidades de definir el modelo del casos de uso (RUP, 2003).

No cabe duda de que el modo de especificación tiene mucho que ver con la calidad de la solución. Muchos ingenieros se han visto forzados a lidiar con especificaciones incompletas, inconsistentes o engañosas y han experimentado la frustración y confusión que invariablemente provocan. La calidad, la fecha de entrega y el alcance del software son las que sufren las consecuencias (PRESSMAN, 2002).

1.6 VALIDACION DE REQUISITOS

Los requisitos una vez definidos necesitan ser validados. La validación de requisitos tiene como misión demostrar que la definición de los requisitos define realmente el sistema que el usuario necesita o el cliente desea (LOWE, 1999). Es necesario asegurar que el análisis realizado y los resultados obtenidos de la etapa de definición de requisitos son correctos. Pocas son las propuestas existentes que ofrecen técnicas para la realización de la validación y muchas de ellas consisten en revisar los modelos obtenidos en la definición de requisitos con el usuario para detectar errores o inconsistencias. (ESCALONA, M.J., TORRES, J., MEJÍAS, M, 2002)

Aún así, existen algunas técnicas que pueden aplicarse para ello:

- **Reviews o Walk-throughs:** Está técnica consiste en la lectura y corrección de la completa documentación o modelado de la definición de requisitos. Con ello solamente se puede validar la correcta interpretación de la información transmitida. Más difícil es verificar consistencia de la documentación o información faltante.

- **Auditorías:** La revisión de la documentación con esta técnica consiste en un chequeo de los resultados contra una checklist predefinida o definida a comienzos del proceso, es decir sólo una muestra es revisada.

- **Matrices de trazabilidad:** Esta técnica consiste en marcar los objetivos del sistema y chequearlos contra los requisitos del mismo (DURÁN A., 1999). Es necesario ir viendo qué objetivos cubre cada requisito, de esta forma se podrán detectar inconsistencias u objetivos no cubiertos.

- **Prototipos:** Algunas propuestas se basan en obtener de la definición de requisitos prototipos que, sin tener la totalidad de la funcionalidad del sistema, permitan al usuario hacerse una idea de la estructura de la interfaz del sistema con el usuario (OLSINA, 1998). Esta técnica tiene el

problema de que el usuario debe entender que lo que está viendo es un prototipo y no el sistema final.

1.7 TRATAMIENTO DE REQUISITOS EN LA PROPUESTA PARA LA WEB

El desarrollo de sistemas Web agrupa una serie de características que lo hacen diferente del desarrollo de otros sistemas. Por un lado, hay que tener en cuenta que roles muy diferentes de desarrolladores participan en el proceso: analistas, clientes, usuarios, diseñadores gráficos, expertos en multimedia y seguridad, etc. Por otro lado, la existencia en estos sistemas de una importante estructura de navegación obliga a un desarrollo preciso de este aspecto que garantice que el usuario no se “pierde en el espacio navegacional del sistema”. Estas ideas unidas al hecho que los sistemas Web suelen tratar con múltiples medios y es esencial que ofrezcan una interfaz adecuada en cada momento, obligan a que estos aspectos propios de la Web deban ser tratados de una forma especial en el proceso de desarrollo. (ESCALONA, M.J., TORRES, J., MEJÍAS, M, 2002).

Estas características especiales también hay que tenerlas en cuenta en la fase de especificación de requisitos. Por ello, la mayoría de las propuestas ofrecen diferentes clasificaciones de los requisitos. Esta constituye una clasificación de requisitos relevantes en sistemas Web.(ESCALONA, M.J., TORRES, J., MEJÍAS, M, 2002)

1.7.1 Requisitos de datos

También son denominados requisitos de contenido, requisitos conceptuales o requisitos de almacenamiento de información. Estos requisitos responden a la pregunta de qué información debe almacenar y administrar sistema.

1.7.2 Requisitos de interfaz (al usuario)

También llamados en algunas propuestas requisitos de interacción o de usuario. Responden a la pregunta de cómo interactuar el usuario con el sistema.

1.7.3 Requisitos navegacionales

Recogen las necesidades de navegación del usuario.

1.7.4 Requisitos de personalización

Describen cómo debe adaptarse el sistema en función de qué usuario interactúe con él y de la descripción actual de dicho usuario.

1.7.5 Requisitos transaccionales o funcionales internos

Recogen qué debe hacer sistema de forma interna, sin incluir aspectos de interfaz o interacción. También son conocidos en el ambiente Web como requisitos de servicios.

1.7.6 Requisitos no funcionales

Son por ejemplo los requisitos de portabilidad, reutilización, de entorno de desarrollo, de usabilidad, de disponibilidad, etc.

1.8 SISTEMAS DE GESTION

Gestionar información no es más que obtener la información adecuada, en la forma correcta, para la persona indicada, al costo adecuado, en el tiempo oportuno, en el lugar apropiado, para tomar la acción correcta.

Un sistema de gestión no es más que una estructura interactiva formada por personas, equipos y métodos destinados a crear un flujo de información capaz de proporcionar un fundamento adecuado para la toma de decisiones o sencillamente la solución a problemas específicos (JEFFERY L, 2003). Una planificación adecuada y el respaldo de la alta dirección pueden facilitar en gran medida la implantación en una organización de este tipo de sistemas. Cuando se gestiona información se incluyen los registros administrativos y los archivos, el soporte tecnológico de los recursos y el público a que se destina.

1.9 TECNICAS Y HERRAMIENTAS UTILIZADAS EN CADA FASE

Captura de requisitos (Rational Unified Process, 2003)

- Entrevistas y cuestionarios
- Tormenta de ideas (Brainstorming)
- Desarrollo conjunto de aplicaciones (Joint Application Development/)
- Mapas Conceptuales. (Concept Mapping)
- Interfaces de usuario y Estructura de la navegación.
- Listas de verificación. (Checklists)
- Comparación de terminología
- Glosario
- Sistemas existentes

- Grabaciones de video y de audio
- Observación
- Arqueología de documentos
- Aprendiz

Definición de requisitos (Rational Unified Process, 2003)

- Glosario
- Talleres de Trabajo basados en los Casos de Uso (Run Use Case WorkShop)
- Prototipos
- Lenguaje natural
- Plantillas o patrones
- Escenarios
- Lenguajes Formales

Validación de requisitos (Rational Unified Process, 2003)

- Prototipos
- Revisión de modelos. (Reviews o Walk-throughs)
- Auditorías
- Matrices de trazabilidad
- Casa de calidad
- (Quality Function Deployment)

ARTEFACTOS UML PROPUESTOS (JACOBSON, IVAR; BOOCH, GRADY; RUMBAUGH, JAMES, 1999)

Captura de requisitos

- Modelo del dominio
- Diagrama de Casos de Uso

Definición de requisitos

- Modelo del dominio
- Diagrama de Casos de Uso
- Diagrama de actividad (Por Caso de Uso)

DOCUMENTOS QUE DEBEN GENERARSE (Rational Unified Process, 2003)

Captura de requisitos

- Documento visión
- Glosario
- Documento de Especificación de Requisitos de software
- Documentos de Especificación de Casos de Uso
- Documento de las Fichas del proceso Actual

Definición de requisitos

- Glosario
- Documento de Especificación de Requisitos de software
- Documentos de Especificación de Casos de Uso

Validación de requisitos

- Documento de Especificación de Requisitos de software

1.10 ANALISIS DE REQUERIMIENTOS

Consiste en detectar y resolver conflictos entre requisitos. Se precisan los límites del sistema y la interacción con su entorno. Se trasladan los requisitos de usuario a requisitos del software (implementables) (FIGUEROA, 2000). En el análisis de requisitos se realizan tres tareas fundamentales:

➤ *Clasificación:*

- En funcionales vs. No funcionales (Capacidades vs. Restricciones).
- Por prioridades.
- Por coste de implementación.
- Por niveles (alto nivel, bajo nivel).

- Según su volatilidad/estabilidad.
- Si son requisitos sobre el proceso o sobre el producto.
 - Modelización Conceptual:
 - Ciertos aspectos de los requisitos se expresan mediante modelos de datos, de control, de estados, de interacción, de objetos, etc.
 - La meta es entender mejor el problema, más que iniciar el diseño de la solución.
 - El tipo de modelo elegido depende de la naturaleza del problema, la experiencia del analista y de la disponibilidad de herramientas.
 - Negociación:
 - En todo proceso de IR intervienen distintos individuos con distintos y, a veces, enfrentados intereses.
 - Estos conflictos entre requisitos se descubren durante el análisis. El conflicto no es rechazable, pues los conflictos son fuente de nuevos requisitos.
 - Los conflictos nunca se deben resolver “por decreto”, sino mediante un proceso de (re)negociación.

Actividades técnicas

1.10.1 Identificar Casos de Uso del sistema

Esta información se representa en un diagrama de casos de uso.

- ¿Cómo encontrar un actor?
 - Identifique los usuarios del Sistema:
 - ¿Por qué se diseña el sistema?
 - ¿Cuáles son los actores que el sistema va a beneficiar?
 - ¿Qué actores van a interactuar directamente con el sistema?
 - ¿Qué actores van a supervisar, mantener, recibir información del sistema?
 - Identifique los roles que juegan esos usuarios desde el punto de vista del sistema.
 - Identifique otros sistemas con los cuales exista comunicación.
- ¿Cómo encontrar un caso de uso?

Identifique las operaciones importantes del sistema a construir:

 - ¿Cuáles son las principales tareas de un actor?
 - ¿Qué información tiene el actor que consultar, actualizar, modificar?

¿Qué cambios del exterior debe informar el actor al sistema?

¿Qué información debe informársele al actor, con respecto a los cambios del sistema?

¿Cómo encontrar relaciones entre actores y casos de uso?

Identifique los casos de uso en los cuales se ve implicado un actor. Busque relaciones *extends* entre casos de uso:

¿Qué casos de uso son similares, diferenciándose en la forma en la cual hacen algunas operaciones?

¿Qué caso de uso redefine la forma en la cual se realiza una transacción dentro de otro caso de uso?

1.10.2 Dar detalle a los casos de uso descritos

Describir la información de entrada y salida de cada caso de uso. Descripción detallada del caso de uso:

- Descripción textual de su Objetivo.
- Variantes posibles para realizar este caso de uso. Diagramas de interacción de detalle (de secuencia o colaboración)
- Errores y excepciones posibles en el caso de uso.

Relacionar el caso de uso con la interfaz a usuario que lo representa. Especificar el diálogo que da solución al caso de uso.

1.10.3 Definir una interfaz inicial del sistema (si es aplicable)

- Dibujar las pantallas de interacción para los distintos actores-usuarios.
- Copiar el modelo mental del usuario.
- Revisar los elementos del modelo del mundo interesantes para el actor-usuario.
- Visualización típica de los elementos del modelo del mundo.
- Información relevante para el actor.
- Metáforas de interacción válidas.

Especificar el diálogo que da solución a cada caso de uso que se soluciona con la interacción con esta interfaz. Puede especificarse este diálogo de varias maneras, dependiendo de la

complejidad de la interfaz definida (en esta etapa se sugiere escoger el mínimo nivel de detalle posible, para dar más libertad de diseño en las etapas posteriores):

1. Por medio de una descripción textual de su funcionamiento.
 2. Por medio de diagramas de interacción que muestren la secuencia de operaciones entre los objetos de interfaz y los actores involucrados.
 3. Por medio de diagramas de estados, donde se muestre claramente los estados de la interfaz.
 4. Por medio de un prototipo funcional, en términos de la interacción con el usuario.
- Definir restricciones para la comunicación con actores y sistema.
 - Describir en el detalle del actor o de la relación con el caso de uso particular.

1.11 JUSTIFICACION DE HERRAMIENTAS A UTILIZAR

La ingeniería de requisitos es la rama de la ingeniería del software que se ocupa de la primera etapa en el proceso de desarrollo del software: la comprensión y formalización de las necesidades que debe satisfacer un sistema informático. Dentro de la ingeniería de requisitos se pueden distinguir dos fases: captura de requisitos y análisis de requisitos. Especialmente en la primera de ellas es esencial una cuidadosa interacción con el cliente que solicita el sistema. El ingeniero de requisitos debe realizar una verdadera tarea de investigación, similar en muchos aspectos a la labor de un científico experimental que interroga a la naturaleza en busca de la comprensión más profunda de un fenómeno, pero interrogando al cliente en lugar de a la naturaleza, para llegar a adivinar los deseos y necesidades que habitualmente el cliente no es capaz de describir más que en forma confusa, incompleta y desordenada.

1.11.1 Software Libre

Se debe pensar en SW libre como en libertad de expresión no como gratis. Se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el SW. De modo mas preciso, se refiere a cuatro libertades de los usuarios del SW:

- La libertad de usar el programa, con cualquier propósito (libertad 0).
- La libertad de estudiar cómo funciona el programa, y adaptarlo a tus necesidades (libertad 1). El acceso al código fuente es una condición previa para esto.
- La libertad de distribuir copias, con lo que puedes ayudar a tu vecino (libertad 2).

- La libertad de mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie. (libertad 3). El acceso al código fuente es un requisito previo para esto.

Un programa es software libre si los usuarios tienen todas estas libertades. Así pues, se debería tener la libertad de distribuir copias, sea con o sin modificaciones, sea gratis o cobrando una cantidad por la distribución, a cualquiera y a cualquier lugar. El ser libre de hacer esto significa (entre otras cosas) que no tienes que pedir o pagar permisos.

También deberías tener la libertad de hacer modificaciones y utilizarlas de manera privada en tu trabajo u ocio, sin ni siquiera tener que anunciar que dichas modificaciones existen. Si publicas tus cambios, no tienes por qué avisar a nadie en particular, ni de ninguna manera en particular.

La libertad para usar un programa significa la libertad para cualquier persona u organización de usarlo en cualquier tipo de sistema informático, para cualquier clase de trabajo, y sin tener obligación de comunicárselo al desarrollador o a alguna otra entidad específica. (FRANKLIN ST, 2006)

1.11.2 Linux

Dos características muy peculiares lo diferencian del resto de sistemas operativos que podemos encontrar en el mercado, la primera, es que es libre, esto significa que no se debe pagar ningún tipo de licencia a ninguna casa desarrolladora de software por el uso del mismo, la segunda, es que el sistema viene acompañado del código fuente.

El sistema lo forman el núcleo del sistema (kernel) más un gran número de programas / bibliotecas que hacen posible su utilización. Muchos de estos programas y bibliotecas han sido posibles gracias al proyecto GNU, por esto mismo, muchos llaman a Linux, GNU/Linux, para resaltar que el sistema lo forman tanto el núcleo como gran parte del software producido por el proyecto GNU.

Linux se distribuye bajo la GNU General Public License por lo tanto, el código fuente tiene que estar siempre accesible y cualquier modificación ó trabajo derivado tiene que tener esta licencia.

¿Por qué GNU/LINUX?

- *Multiplataforma:* Las plataformas en las que en un principio se puede utilizar Linux son 386-, 486-, Pentium, Pentium Pro, Pentium II, Amiga y Atari, también existen versiones para su utilización en otras plataformas, como amd64, Alpha, ARM, MIPS, PowerPC y SPARC.

- *Multitarea*: La palabra multitarea describe la habilidad de ejecutar varios programas al mismo tiempo. LINUX utiliza la llamada multitarea preventiva, la cual asegura que todos los programas que se están utilizando en un momento dado serán ejecutados, siendo el sistema operativo el encargado de ceder tiempo de microprocesador a cada programa.
- *Memoria virtual usando paginación (sin intercambio de procesos completos) a disco*: A una partición en el sistema de archivos, con la posibilidad de añadir más áreas de intercambio sobre la marcha.
- *Todo el código fuente está disponible*, incluyendo el núcleo completo y todos los drivers, las herramientas de desarrollo y todos los programas de usuario; además todo ello se puede distribuir libremente. Hay algunos programas comerciales que están siendo ofrecidos para Linux actualmente sin código fuente, pero todo lo que ha sido gratuito sigue siendo gratuito.
- TCP/IP, incluyendo ssh, ftp, telnet, NFS, etc.
- Software cliente y servidor Netware.
- Lan Manager / Windows Native (SMB), software cliente y servidor.
- Diversos protocolos de red incluidos en el kernel: TCP, IPv4, IPv6, AX.25, X.25, IPX, DDP, Netrom, etc.

1.11.3 Metodologías de desarrollo

Una metodología de desarrollo de SW consta de conceptos y diagramas, etapas de definición y entrega de cada una de ellas y actividades y recomendaciones.

➤ ***Desarrollo Guiado por la Funcionalidad (FDD)***:

FDD esta pensado para proyectos con tiempo de desarrollo relativamente cortos. Se basa en un proceso iterativo con iteraciones cortas que producen un software funcional que el cliente y la dirección de la empresa pueden ver y monitorizar (CALABRIA, 2003). Las iteraciones se deciden en base a las funcionalidades, que son pequeñas partes del software con significado para el cliente.

FDD se divide en 5 fases:

- Fase 1 Desarrollo de un modelo general.
- Fase 2 Construcción de la lista de funcionalidades.
- Fase 3 Plan de releases en base a las funcionalidades a implementar.

- Fase 4 Diseñar en base a las funcionalidades.
- Fase 5 Implementar en base a las funcionalidades.

FDD define métricas para seguir el proceso de desarrollo de la aplicación, útiles para el cliente y la dirección de la empresa, y que pueden ayudar, además para conocer el estado actual del desarrollo, a realizar mejores estimaciones en proyectos futuros.

Roles fundamental para FDD:

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.

➤ **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.

Aspectos fundamentales:

- 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. (SÁNCHEZ, 2004)

Características:

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.

Re fabricación: 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. (SÁNCHEZ, 2004)

➤ RUP

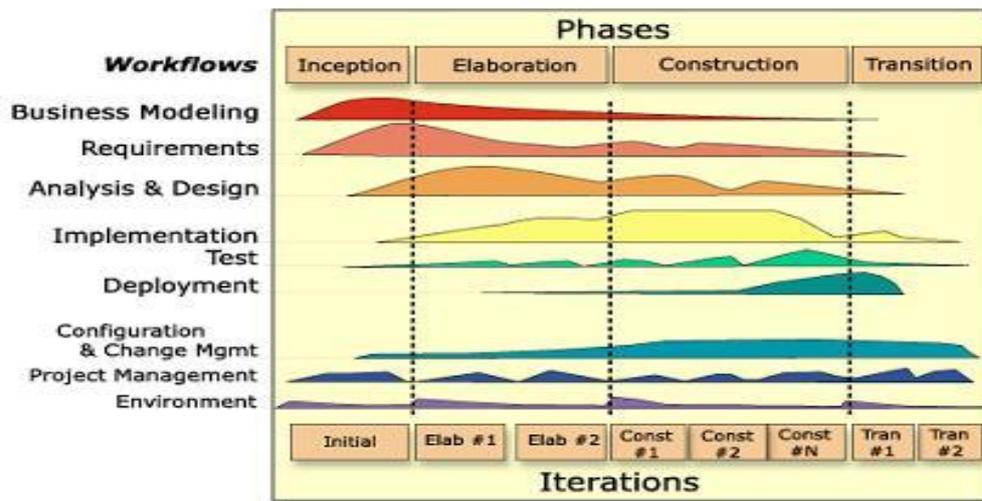


Fig. 1.2 Fases e Iteraciones de la Metodología RUP.

Se utilizará RUP (Rational Unified Process) como estándar para el modelado del negocio, el análisis y la documentación del sistema, que constituye un proceso de desarrollo de software y emplea el UML (Lenguaje Unificado de Modelado). (JACOBSON, IVAR; BOOCH, GRADY; RUMBAUGH, JAMES, 1999)

Como RUP es un proceso, en su modelación define como sus principales elementos:

Trabajadores (“quién”): Define el comportamiento y responsabilidades (rol) de un individuo, grupo de individuos, sistema automatizado o máquina, que trabajan en conjunto como un equipo. Ellos realizan las actividades y son propietarios de elementos.

Actividades (“cómo”): Es una tarea que tiene un propósito claro, es realizada por un trabajador y manipula elementos.

Artefactos ("qué"): Productos tangibles del proyecto que son producidos, modificados y usados por las actividades. Pueden ser modelos, elementos dentro del modelo, código fuente y ejecutables.

Flujo de actividades ("Cuándo"): Secuencia de actividades realizadas por trabajadores y que produce un resultado de valor observable.

RUP se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso. Divide el proceso de desarrollo en ciclos, teniendo un producto final, al final de cada ciclo. Cada ciclo se divide en fases que finalizan con un hito donde se debe tomar una decisión importante:

- *Inicio*: se hace un plan de fases, se identifican los principales casos de uso y se identifican los riesgos. El objetivo en esta etapa es determinar la visión del proyecto.
- *Elaboración*: se hace un plan de proyecto, se completan los casos de uso y se eliminan los riesgos. En esta etapa el objetivo es determinar la arquitectura óptima.
- *Construcción*: se concentra en la elaboración de un producto totalmente operativo y eficiente y el manual de usuario. En esta etapa el objetivo es llevar a obtener la capacidad operacional inicial.
- *Transición*: se implementa el producto en el cliente y se entrena a los usuarios. Como consecuencia de esto suelen surgir nuevos requisitos a ser analizados. El objetivo es llegar a obtener la versión del proyecto. (JACOBSON, IVAR; BOOCH, GRADY; RUMBAUGH, JAMES, 1999)

1.12 Herramientas CASE

Una herramienta CASE (Computer Aided Software Engineering) es un conjunto de métodos, utilidades y técnicas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todo el ciclo de vida de desarrollo de un Software. Las herramientas CASE representan una forma que permite modelar los procesos de negocios de las empresas y desarrollar los Sistemas de Información Gerenciales. (ANDRÉS, 2001)

➤ *Visual Paradigm*

Se utilizará esta herramienta debido a que ofrece un entorno de creación de diagramas para UML; presenta un diseño centrado en casos de uso y enfocado al negocio que generan un software de mayor calidad; posee un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación; capacidades de ingeniería directa (versión profesional) e inversa; modelo

y código que permanece sincronizado en todo el ciclo de desarrollo; disponibilidad de múltiples versiones para cada necesidad; disponibilidad de integrarse en los principales IDEs; disponibilidad en múltiples plataformas, y muy útil para la generación de código fuente en PHP, también con el Paradigm se generan script de las tablas de salidas para las clases persistentes.

Permite gestionar proyectos muy complejos con gran sencillez. Incluye herramientas muy interesantes para ingeniería inversa de bases de datos.

Incorpora una funcionalidad de análisis textual, facilitando la interacción directa con un enunciado escrito en lenguaje natural, es decir, permite relacionar elementos presentes en un enunciado con los diagramas UML correspondientes, aunque no contempla el tratamiento directo en el texto de la relación entre los componentes del enunciado.

La herramienta es colaborativa, es decir, soporta múltiples usuarios trabajando sobre el mismo proyecto; genera la documentación del proyecto automáticamente en varios formatos como Web o Pdf, y permite control de versiones.

Es un producto de calidad que soporta aplicaciones Web y también un conjunto de lenguajes en generación de código e ingeniería inversa en Java, C++, PHP, etc. En adición, la generación de código soporta C#, VB .NET, Delphi, Perl, Python, entre otros. La Ingeniería Inversa también soporta clases en Java, .dll y .exe en .NET, JDBC, y archivos de mapeo Hibernate.

➤ **Rational Rose**

Es la herramienta Case desarrollada por los creadores de UML que cubren todo el ciclo de vida de un proyecto: concepción y formalización del modelo, construcción de los componentes y certificación de las distintas fases (SOFTWARE, 2007). Permite una trazabilidad real entre modelo (análisis y diseño) y el código ejecutable.

Rational Rose domina el mercado de herramientas para el modelado, análisis, diseño y construcción orientada a objetos, posee todas las características que los desarrolladores, analistas, y arquitectos exigen, desarrollo basado en componentes con soporte para arquitecturas líderes en la industria y modelos de componentes, facilidad de uso e integración optimizada.

Rational Rose posee grandes ventajas como por ejemplo que utiliza la notación estándar en la arquitectura de software (UML), la cual permite a los arquitectos de software y desarrolladores visualizar el sistema completo utilizando un lenguaje común, además los diseñadores pueden modelar sus componentes e interfaces en forma individual y luego unirlos con otros componentes

del proyecto. Además de esto tiene como desventaja que es un software propietario y no corre sobre SO libres.

1.13 LENGUAJE DE MODELACION

Indistintamente se han representado los diseños de una manera más bien personal o con algún modelo gráfico. La falta de estandarización en la representación gráfica de un modelo impedía que los diseños gráficos realizados se pudieran compartir fácilmente entre distintos diseñadores, con este objetivo se crearon los Lenguajes de Modelación. Para la realización del presente trabajo se utilizará como lenguaje de modelado el Unified Modeling Language (UML).

Lenguaje Unificado de Modelación (UML) es un lenguaje gráfico para visualizar, especificar, construir, documentar y comunicar los artefactos de un sistema de software (JACOBSON, IVAR; BOOCH, GRADY; RUMBAUGH, JAMES, 1999). Utilizado para entender, diseñar, configurar, mantener y controlar la información sobre los sistemas a construir con la finalidad de describir modelos del sistema (del mundo real y del mundo del software), basados en los conceptos de objetos.

UML ofrece un estándar para describir un modelo del sistema, 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.

El UML especifica varios diagramas, de ellos los diagramas de Casos de Usos y los diagramas de iteración, que son los artefactos concretos a partir de los cuales creamos modelos. Los diagramas UML se visualizan por medio de las vistas (proyecciones visuales del modelo).

El UML es un lenguaje flexible de modelado que permite definir modelos de análisis (es el que se relaciona con la investigación del dominio y del ámbito del problema, pero no con la solución) y modelos del diseño (es el que se relaciona con la solución lógica del problema) (IRETA, 1999).

En lo que corresponde al desarrollo de programas UML, posee elementos gráficos para soportar la captura de requisitos, el análisis, el diseño, la implementación, y las pruebas, se compone de muchos elementos de esquematización que representan las diferentes partes de un sistema de software.

De forma general las principales *características* son:

- Lenguaje unificado para la modelación de sistemas.

- Tecnología orientada a objetos.
- El cliente participa en todas las etapas del proyecto.
- Corrección de errores viables en todas las etapas.
- Aplicable para tratar asuntos de escala inherentes a sistemas complejos de misión crítica, tiempo real y cliente/servidor.
- Facilita a los integrantes de un equipo multidisciplinario participar e intercomunicarse fácilmente.

Beneficios que ofrece UML:

- Contar con un mejor entendimiento del riesgo del proyecto antes de construir el sistema.
- Mejores tiempos totales de desarrollo.
- Permite especificar la estructura y el comportamiento del sistema y comunicarlo a todos los integrantes del proyecto.
- Permite documentar las decisiones de la arquitectura del proyecto.
- Se obtiene el "plano" del sistema.
- Mejor soporte a la planeación y al control del proyecto.
- Un aumento en la calidad del desarrollo.
- Reducción en los costos económicos.
- Soporta el Proceso Unificado de Rational (RUP).
- Posee características visuales que facilitan a integrantes de un equipo multidisciplinario participar e intercomunicarse fácilmente.

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.(JACOBSON, IVAR; BOOCH, GRADY; RUMBAUGH, JAMES, 1999)

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

1.14 ANALISTA DE SISTEMAS: ROL DEFINIDO POR RUP

El analista de sistema software es el encargado de captar la información suministrada por el cliente y modelarla de manera que el equipo de desarrollo pueda entenderla, con el fin de automatizarla; comprender la manera en que funcionan los negocios con el propósito de mejorar los procesos en la organización.

El analista de sistema nació 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 dentro de la organización.

Analista de procesos de negocio: Conduce y coordina el caso de uso del negocio que modela contorneando y delimitando la organización que es modelada; establece que actores del negocio y casos de uso del negocio existen y como trabajan entre ellos. Además es responsable de la arquitectura del negocio (JACOBSON, IVAR, 2000).

Analista de sistema: Conduce y coordina los requerimientos y los Casos de Uso modelando y delimitando la funcionalidad del sistema y delimitando el sistema; estableciendo que actores y casos de uso existen y como interactúan.

1.15 CONCLUSIONES

En el presente capítulo se realizó un estudio de las técnicas actuales para la captura de requisitos, así como para la especificación y validación de los mismos. De igual manera se justificó el uso de las distintas herramientas que servirán como soporte para la elaboración de la propuesta de solución al problema planteado.

CAPITULO 2: CARACTERISTICAS DEL SISTEMA

2.1 INTRODUCCION

El modelado del negocio permite comprender la estructura y la dinámica de la organización en la cual se va a implantar un sistema; así como identificar mejoras potenciales. En la realización de los casos de uso del negocio, se obtienen las actividades que serán objeto de automatización y es el punto de partida para identificar qué debe hacer el sistema y de ahí poder hacer una buena especificación de los requerimientos funcionales y no funcionales.

En este capítulo se presentará un estudio del flujo actual de los procesos, la información que se maneja y los artefactos resultantes correspondientes al flujo del modelo del negocio. Además, a partir de lo estudiado se procederá con el levantamiento de requisitos utilizando para ello como técnica de captura de requerimientos la entrevista libre, para luego comenzar con todo el proceso de validación de los mismos. El capítulo contará además con una breve descripción de los artefactos generados en el Expediente del Proyecto y una propuesta para la captura de requerimientos.

2.2 EXPEDIENTE DEL PROYECTO

Partiendo de la necesidad de crear en cada uno de los proyectos de la universidad un "Expediente del Proyecto" y de la experiencia alcanzada en este sentido, se seguirá la siguiente estrategia y principios en el proyecto Residencia:(RAMSÉS, 2003)

- Para su implantación se seguirá la Metas Generales propuesta por CMMI v1.2 para cada uno de sus niveles de madurez.
- La implantación del Expediente del Proyecto no puede implicar un motivo de retraso para el desarrollo de los proyectos.
- La implantación debe realizarse en el menor tiempo posible.

Dicho Expediente cuenta con numerosos documentos, muchos de ellos generados en la etapa de captura de requisitos.

➤ **Plantilla Modelo de CU del Negocio:**

Este documento contará con una breve descripción textual (especificación de CU) de cada uno de los casos de uso identificados en el modelamiento del negocio, así como los diagramas de actividades correspondientes a cada uno de estos casos de uso y actores y trabajadores

implicados en el negocio, además de contar con el modelo de objetos del negocio y las reglas generales establecidas en el mismo.

➤ **Plantilla Modelo de CU del Sistema:**

Contará con una breve descripción del sistema a implementar, además de la definición de todos los actores del sistema, un diagrama de CU del sistema y la descripción de cada uno de estos CU.

➤ **Plantilla Plan de Gestión de Requisitos:**

Detallará las reglas de confidencialidad, definiciones, abreviaturas y acrónimos. Para la gestión de requisitos, se describirán herramientas, ambientes e interfaces que serán utilizadas para el control de versiones de los requisitos generados durante todo el ciclo de vida. El programa de Gestión de Requisitos contará con la identificación de los requisitos, o sea describe como los elementos serán nombrados, marcados y numerados. Muy importante es dividir los requisitos en funcionales y no funcionales para una mejor comprensión de los mismos.

➤ **Plantilla Especificación de requisitos**

Definirá todos los requerimientos funcionales y no funcionales del sistema a construir, así como una descripción detallada de los mismos.

➤ **Plantilla Caso de uso detallado**

Esta plantilla se utilizará de forma opcional, se puede detallar en ella todos los casos de uso y luego pasar estas descripciones a la plantilla de CU del Sistema o Negocio según sea el caso.

2.3 PASOS A SEGUIR EN LA ESPECIFICACION DE LOS REQUISITOS

Se realizó un estudio de todas las técnicas y herramientas posibles a utilizar y del proceso en cuestión. El analista luego de encontrar datos interesantes se dispone a "analizar" los mismos utilizando herramientas para el análisis o técnicas de visualización. Este análisis le da una percepción especial acerca de los datos, luego el analista presenta esta percepción al usuario final.

1. Se escogió la entrevista libre como técnica de captura de requisitos, por ser esta la más adecuada con el negocio que se presentaba. Se realizaron entrevistas individuales a dirigentes-especialistas en el negocio (Visión de todo el negocio), además de entrevistas individuales con

especialistas funcionales directamente vinculados al negocio en cuestión (Conocimiento de las actividades de los procesos).

2. Se utilizó de igual manera el Desarrollo Conjunto de Aplicaciones (JDA), para ello se planificaron reuniones con los especialistas funcionales y dirigentes del negocio y por parte del equipo de desarrollo el o los analistas involucrados en la validación de los PEN (Procesos elementales del negocio) y el prototipo elemental de usuario.

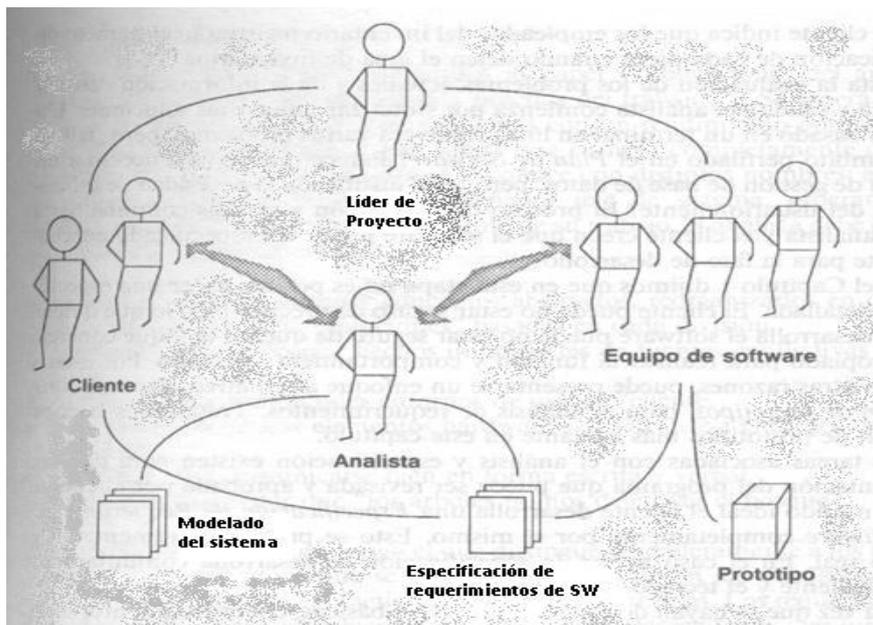


Fig. 2.1 Flujo de trabajo en la etapa de captura de requisitos y modelado del sistema.

3. Se optó por realizar todo el modelado con la herramienta CASE Visual Paradigm, precisamente porque la misma tiene la ventaja de ser multiplataforma. Se modeló el negocio y, luego de la captura de requisitos, se comenzó con el modelado del sistema. Indistintamente se continuaron planificando encuentros con los clientes con el objetivo de mostrar el grado de avance del producto y que, de esta forma, ellos corroboraran el trabajo realizado. Igualmente se efectuaban reuniones con todo el equipo de trabajo, muy fundamental en este sentido el encuentro entre analistas y programadores.

En el más simple de los escenarios, un análisis da como resultado un reporte de algún tipo. Pero en los escenarios reales los resultados pueden ser muy variados y complicados. Un buen gráfico que capture todas las relaciones en el modelo puede ser muy apropiado.

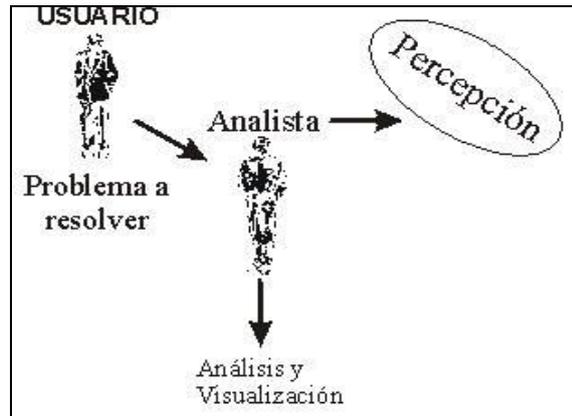


Fig. 2.2 Usuario-Analista.

2.3.1 Propuesta general para la captura de requisitos.

1- Captura de requisitos.

Es de vital importancia conocer las necesidades de los clientes, de ahí que lo primero que se debe hacer es entender como funciona el negocio. Se propone utilizar la técnica de captura de requisitos que más se adecue al negocio a tratar, siendo en este sentido las más recomendadas la *entrevista* y la *observación* (o método del aprendiz).

En esta etapa de la captura de requisitos se recomienda utilizar los siguientes documentos como soporte a todo el trabajo realizado:

- Modelo del dominio
- Diagrama de Casos de Uso

2- Definición de requisitos.

En numerosas ocasiones existen problemas de comprensión e interpretación de los requisitos debido a la manera en que se escriben. Se recomienda la utilización de *glosarios* y *ontologías* para la definición de los mismos, de forma tal que exista una terminología común entre todos los miembros del equipo de desarrollo.

Se propone entonces la utilización de los siguientes documentos como apoyo:

- Modelo del dominio
- Diagrama de Casos de Uso
- Diagrama de actividad (Por Caso de Uso)

3- Validación de requisitos.

Validar los requisitos, luego de haberlos definido correctamente, tiene el objetivo de probar que realmente se define lo que el sistema debe hacer, o sea, lo que el cliente desea.

Debe refinarse entonces en esta etapa:

- Documento de Especificación de Requisitos de software

2.4 OBJETO DE AUTOMATIZACION

2.4.1 Situación problemática.

Luego de haber definido la situación problemática, se proseguirá con la modelación de un sistema que permita automatizar el proceso de avituallamiento en la residencia universitaria, facilitando de esta forma el trabajo de las personas implicadas en el mismo y contribuyendo a la informatización de la universidad.

2.4.2 Flujo actual de los procesos.

El control y entrega de avituallamiento se lleva a cabo en todas las direcciones de residencia con que cuenta la universidad, tanto en las tres de estudiantes como en la dirección de residencia para especialistas y profesores.

El trabajo en las direcciones de residencia tiene mayor peso en el proceso de cambio de módulo de ropa, el mismo consiste en que los usuarios se presentan a los departamentos solicitando realizar el cambio de avituallamiento por uno limpio.

Otro proceso importante es el de gestionar las boletas de control de recepción y entrega de módulo, ya sea crearlas, actualizarlas o modificarlas.

Periódicamente se realizan reportes para un mejor control de los productos almacenados.

Todas estas informaciones deben llegar a las direcciones de residencia para de ahí pasar a la vicerrectoría de Residencia.

Además se realizan préstamos a los usuarios que así lo requieran en cualquiera de las direcciones.



Fig. 2.3 Flujo actual de información.

2.4 SISTEMAS AUTOMATIZADOS VINCULADOS CON EL CAMPO DE ACCIÓN

Hasta el momento se desconoce de la existencia de algún sistema que tenga la responsabilidad de resolver este tipo de problemas. En este sentido cobra vital importancia el desarrollo del presente trabajo. Se realizará para ello un estudio profundo del proceso de avituallamiento y lavandería, se recogerá de esta forma los detalles más significativos que permitan el modelado de un sistema capaz de cumplir con la demanda.

2.5 INFORMACIÓN QUE SE MANEJA

➤ Residencia para profesores y especialistas:

- *Entrega de productos químicos a La Flora:* Este documento registra las cantidades totales de los productos químicos que se llevan de la universidad a la lavandería “La Flora”.
- *Relación de avituallamiento por edificio:* Como su nombre lo dice se lleva el control de todo el avituallamiento en poder de profesores y especialistas, registrado según el edificio en el que vive.

- *Acta de entrega y recogida de productos textiles en la lavandería “La Flora”*: De forma análoga con el documento de *Entrega de productos químicos a La Flora* se recoge en un acta el total de productos entregados y recogidos en la lavandería, debido a que no necesariamente se recoge lo que se entrega.
- *Conciliación de avituallamiento*: Se tiene un control detallado de los productos textiles según el tipo, el color, los que se encuentran en el depósito o en almacén, etc.
- *Decisión de la autoridad facultada sobre la aplicación de la responsabilidad material*: Este se utiliza cuando existe algún problema como pérdidas, etc.
- *Acta de entrega*: Se utiliza en caso de realizar algún préstamo a personas ajenas a la universidad.
- *Tarjeta de Control de Entrega y Recepción de Avituallamiento*: No es más que la boleta donde se archivan todos los medios que posee el usuario.
 - **Residencia de estudiantes:**
- *Andar la residencia*: Se utiliza en caso de que un estudiante cause baja, licencia o traslado, es un hago constar de que se entregaron todos los medios antes de salir de la universidad.
- *Tarjeta de control de entrega y recepción de avituallamiento*: No es más que la boleta donde se archivan todos los medios que posee el usuario.
- *Acta de entrega y recogida de productos textiles en la lavandería “La Flora”*: Se recoge en un acta el total de productos entregados y recogidos en la lavandería, debido a que no necesariamente se recoge lo que se entrega.

2.6 MODELO DE NEGOCIO

El modelo de negocio no es más que un artefacto del flujo de trabajo Modelamiento del Negocio en la Fase de Inicio. Como principales objetivos están: comprender la estructura y la dinámica de la organización en la cual se va a implantar un sistema, comprender los problemas existentes actualmente dentro de la organización e identificar las mejoras potenciales, asegurar que los consumidores, usuarios finales y desarrolladores tengan un entendimiento común de la organización. Derivar los requerimientos del sistema que va a soportar la organización.

2.6.1 Actor del negocio

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

Actores identificados en el negocio.

Actor	Justificación
Usuario <div style="display: inline-block; vertical-align: middle; margin-left: 10px;">  <ul style="list-style-type: none"> Estudiante Profesor Especialista </div>	Es quien inicia la mayor parte de los CU, debido a que son los mayores beneficiarios del negocio
Técnico de Avituallamiento	Es responsable e interesado en emitir todos los reportes.

Tabla 2.1 Descripción de los actores del negocio.

2.6.2 Trabajador del negocio

Los trabajadores del negocio, son los posibles actores del sistema que se pretende construir. Define el comportamiento y responsabilidades (rol) de un individuo, grupo de individuos, sistema automatizado o máquina, que trabajan en conjunto como un equipo. Ellos realizan las actividades y son propietarios de elementos.

Trabajadores identificados en el negocio.

Trabajador	Justificación
Técnico	Son los encargados de llevar el control de la manipulación del avituallamiento que le corresponde a los estudiantes, velando porque les sean entregados desde primer año, debe registrar en una boleta para cada estudiante sus datos y el conjunto de medios entregados y establecer el control de los recursos dando de alta y baja a los mismos o registrando su estado y otras tareas.

Recepcionista Lavandería	Es el encargado de realizar el cambio de avituallamiento.
---------------------------------	---

Tabla 2.2 Descripción de los trabajadores del negocio.

3.6.3 Reglas del negocio

Las reglas del negocio no son más que un conjunto de reglas o normas que rigen el Negocio, de modo que el modelado debe realizarse teniendo en cuenta no violar ninguna de ellas.

- Cumplir con el tratamiento indicado en la legislación vigente relacionado con el asunto en caso de deterioro, hurto, pérdida o robo, haciendo constancia a través del Acta de responsabilidad Material firmada por las partes incurrentes.
- Entregar a los estudiantes matriculados en el acto de incorporación a la Residencia, la ropa de cama y otras lencerías en las cantidades asignadas y la calidad requerida, quedando en su posesión y custodia bajo acta de recepción oficial firmada debidamente.
- Brindar, por el personal designado, la atención permanente al cambio de ropa de cama y otras lencerías de los estudiantes, profesores y especialistas por servicios de lavandería mediante un sistema de control con periodicidad regular establecida.

2.6.4 Diagrama de casos de uso del negocio

El modelo de los casos de uso del negocio representa un proceso dentro del negocio que se estudia, por lo que se corresponde con una secuencia de acciones con un orden lógico y que producen un resultado observable para ciertos actores del negocio (IVAR JACOBSON, 2000).

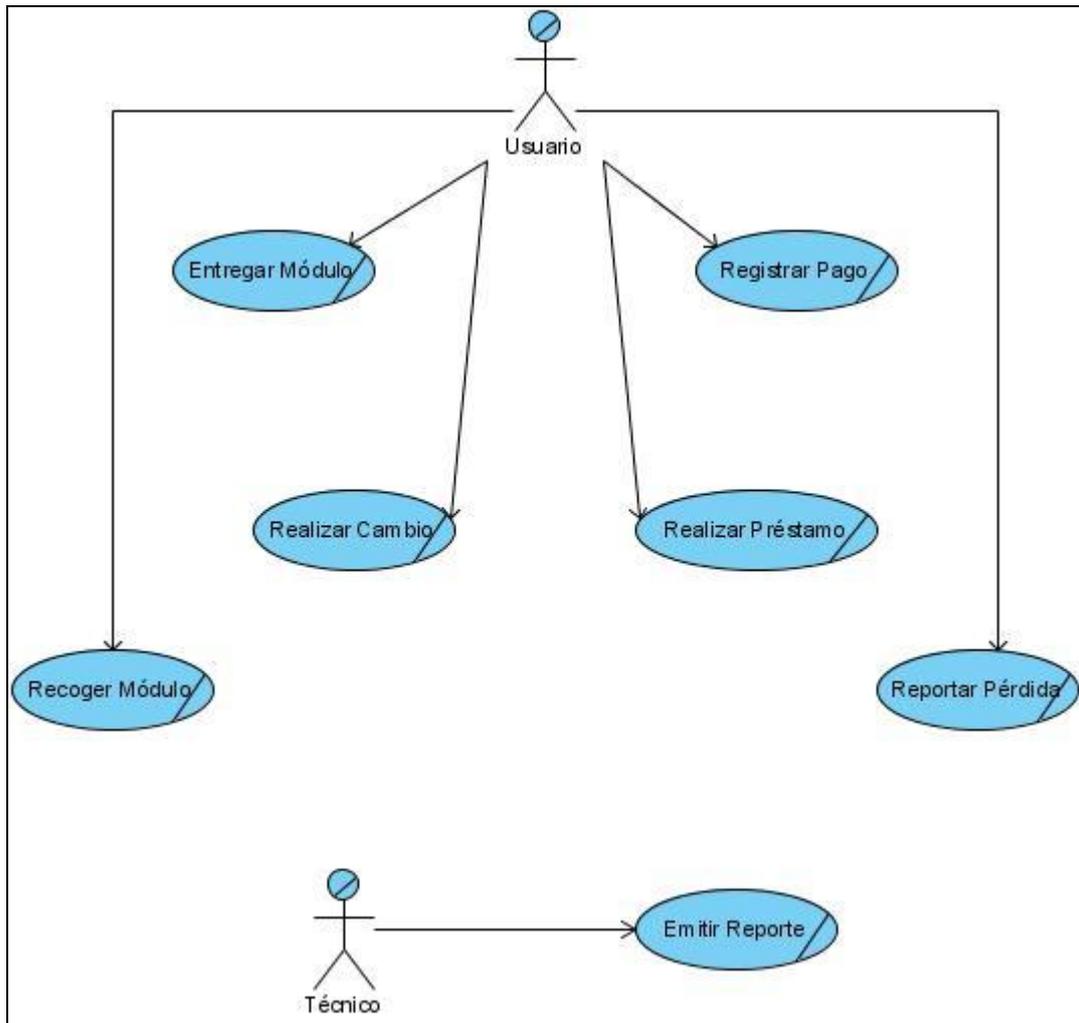


Fig. 2.4 Diagrama de casos de uso del negocio (DCUN).

2.6.5 Descripciones de casos de uso del negocio

A continuación se describirán cada uno de los CU identificados en el negocio, para una mejor comprensión de los mismos.

Caso de Uso:	Recoger módulo
Actores:	Usuario (inicia)
Trabajadores:	Técnico de avituallamiento

CAPITULO 2: CARACTERISTICAS DEL SISTEMA

Resumen:	El caso de uso se inicia cuando un usuario llega al departamento de avituallamiento y solicita el modulo que le corresponde, este es atendido por el Técnico de avituallamiento que lo registra creándole una ficha y le entrega lo asignado. El estudiante firma y recoge lo que se le entrega. Finalizando así el caso de uso.
Flujo Normal de Eventos	
Sección	
Acción del Actor	Respuesta del Negocio
1. El usuario acude al departamento de avituallamiento presentando el solapín.	<p>1.1. El Técnico de avituallamiento verifica que el usuario es de nuevo ingreso.</p> <p>1.2. Lo registra creándole una ficha, con los datos del estudiante y lo que se le entrega.</p> <p>1.3 El técnico de avituallamiento verifica el listado de medios que debe entregar.</p> <p>1.4 El Técnico de avituallamiento entrega los medios asignados.</p>
2. El usuario recoge el avituallamiento y firma su ficha de entrega, retirándose del local.	2.1. El Técnico de avituallamiento recoge la ficha de entrega.
Flujo Alternativo	
Acción del Actor	Respuesta del Negocio
1.1 Si el usuario no es de nuevo ingreso no se lleva a cabo el caso de uso.	
Caso de Uso:	Entregar modulo

CAPITULO 2: CARACTERISTICAS DEL SISTEMA

Actores:	Usuario (inicia)
Trabajadores:	Técnico de avituallamiento
Resumen:	El caso de uso se inicia cuando un usuario llega al departamento de avituallamiento y solicita le sean recogidos los medios, este es atendido por el Técnico de avituallamiento que cancela la ficha de los medios entregados y recoge lo que trajo el estudiante. El técnico firma la entrega. Finalizando así el caso de uso.
Flujo Normal de Eventos	
Sección	
Acción del Actor	Respuesta del Negocio
1. El usuario acude al departamento de avituallamiento presentando el solapín y los medios.	1.1. El Técnico de avituallamiento verifica la facultad a la que pertenece el usuario. 1.2. Busca la ficha del usuario, y pide todos los medios.
2. El usuario entrega el avituallamiento.	2.1 El técnico de avituallamiento llena el modelo de entrega y lo firma.
3. El usuario firma y se retira del local.	3.1 El técnico guarda el modelo de entrega y los medios.
Flujo Alterno	
Acción del Actor	Respuesta del Negocio
1.2 En caso de perdida ver caso de uso "Reportar perdida".	
Caso de Uso:	Reportar pérdida

CAPITULO 2: CARACTERISTICAS DEL SISTEMA

Actores:	Usuario(inicia)
Trabajadores:	Técnico de avituallamiento
Resumen:	El caso de uso se inicia cuando un usuario llega al departamento de avituallamiento y reporta la perdida total o parcial de los medios, este es atendido por el Técnico de avituallamiento, que registra la perdida en la ficha y le indica donde pagarlo. Finalizando así el caso de uso.
Flujo Normal de Eventos	
Sección	
Acción del Actor	Respuesta del Negocio
1. El usuario acude al departamento de avituallamiento para reportar la perdida.	1.1. El Técnico de avituallamiento verifica la facultad a la que pertenece el estudiante. 1.2. Busca la ficha del usuario.
2. El usuario especifica los medios perdidos.	2.1 El técnico de avituallamiento registra los medios perdidos en la ficha. 2.2 El técnico le indica donde debe pagarlos.
3. El usuario se retira a pagar los medios. Ver caso de uso "Registrar pago"	
Flujo Alternativo	
Acción del Actor	Respuesta del Negocio

CAPITULO 2: CARACTERISTICAS DEL SISTEMA

Caso de Uso:	Registrar Pago
Actores:	Usuario (inicia)
Trabajadores:	Técnico de avituallamiento
Resumen:	El caso de uso se inicia cuando un usuario llega al departamento de avituallamiento y entrega la constancia de pago de los medios perdidos, este es atendido por el Técnico de avituallamiento que registra el pago de los medios. Finalizando así el caso de uso.
Flujo Normal de Eventos	
Sección	
Acción del Actor	Respuesta del Negocio
1. El usuario acude al departamento de avituallamiento con la constancia de pago.	1.1. Comprueba la veracidad del documento. 1.2. Busca la ficha del estudiante. 1.3 Registra el pago con su firma.
2. El usuario se retira.	2.1 El técnico archiva los documentos.
Flujo Alternativo	
Acción del Actor	Respuesta del Negocio
Caso de Uso:	Realizar cambio
Actores:	Usuario (inicia)

CAPITULO 2: CARACTERISTICAS DEL SISTEMA

Trabajadores:	Recepcionista	
Resumen:	El caso de uso se inicia cuando un usuario llega al departamento de avituallamiento y solicita que le sea cambiada su ropa de cama.	
Flujo Normal de Eventos		
Sección		
Acción del Actor	Respuesta del Negocio	
1. El usuario acude al departamento de avituallamiento presentando el solapín.	1.1. El recepcionista comprueba que el usuario pertenece a esa residencia. 1.2. Si hay disponible ropa del tipo de la que desea cambiar el usuario se procede con el cambio.	
Flujo Alternativo		
Acción del Actor	Respuesta del Negocio	
	1.2 En caso de no haber ropa disponible, se le informa al usuario y el mismo se retira.	
Caso de Uso:	Emitir Reportes	
Actores:	Técnico (inicia)	
Trabajadores:	Técnico	
Resumen:	El caso de uso inicia cuando se necesita generar un reporte de cualquier tipo de datos o información, según las solicitudes de la dirección o vicerrectoría.	
Precondiciones:	Los datos deben estar actualizados.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Negocio	
1. El técnico busca los datos y	1.1 Con todos los documentos pertinentes se	

documentos necesarios para elaborar el reporte.	elabora el nuevo reporte.
Flujos Alternos	

2.6.6 Diagramas de actividades

Un diagrama de actividad describe un proceso que explora el orden de las tareas o actividades que logran los objetivos del negocio, establece cuando inicia, como se ejecuta y cuando termina. Permite la administración adecuada y una visión simplificada de la complejidad del proceso que se está modelando como descripción de la secuencia de actividades que se ejecutan. Para una mejor comprensión se modelarán a continuación los diagramas de actividades correspondientes a cada uno de los CU identificados en el negocio.

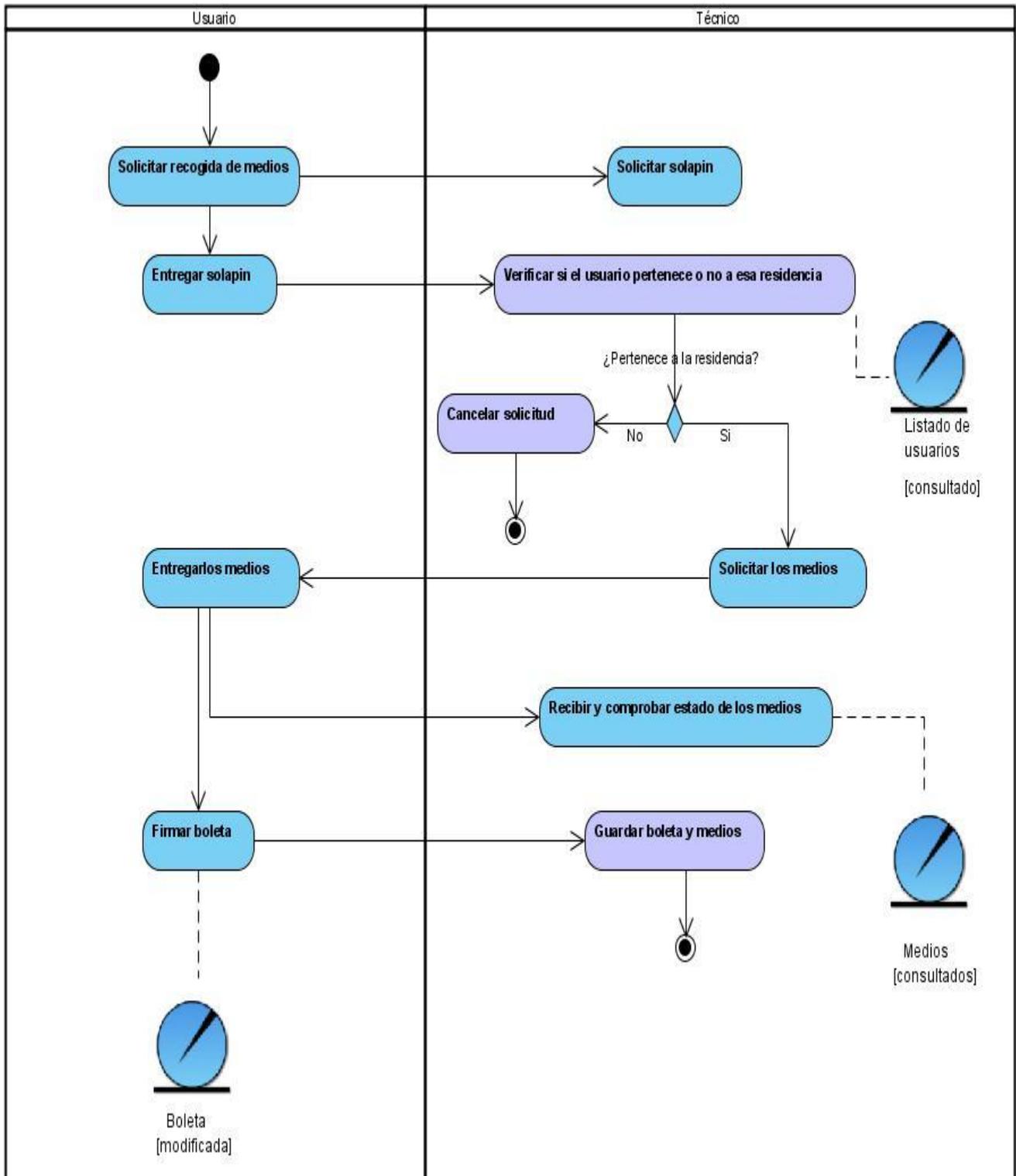


Fig. 2.5 Diagrama de actividades CU: “Entregar Módulo”.

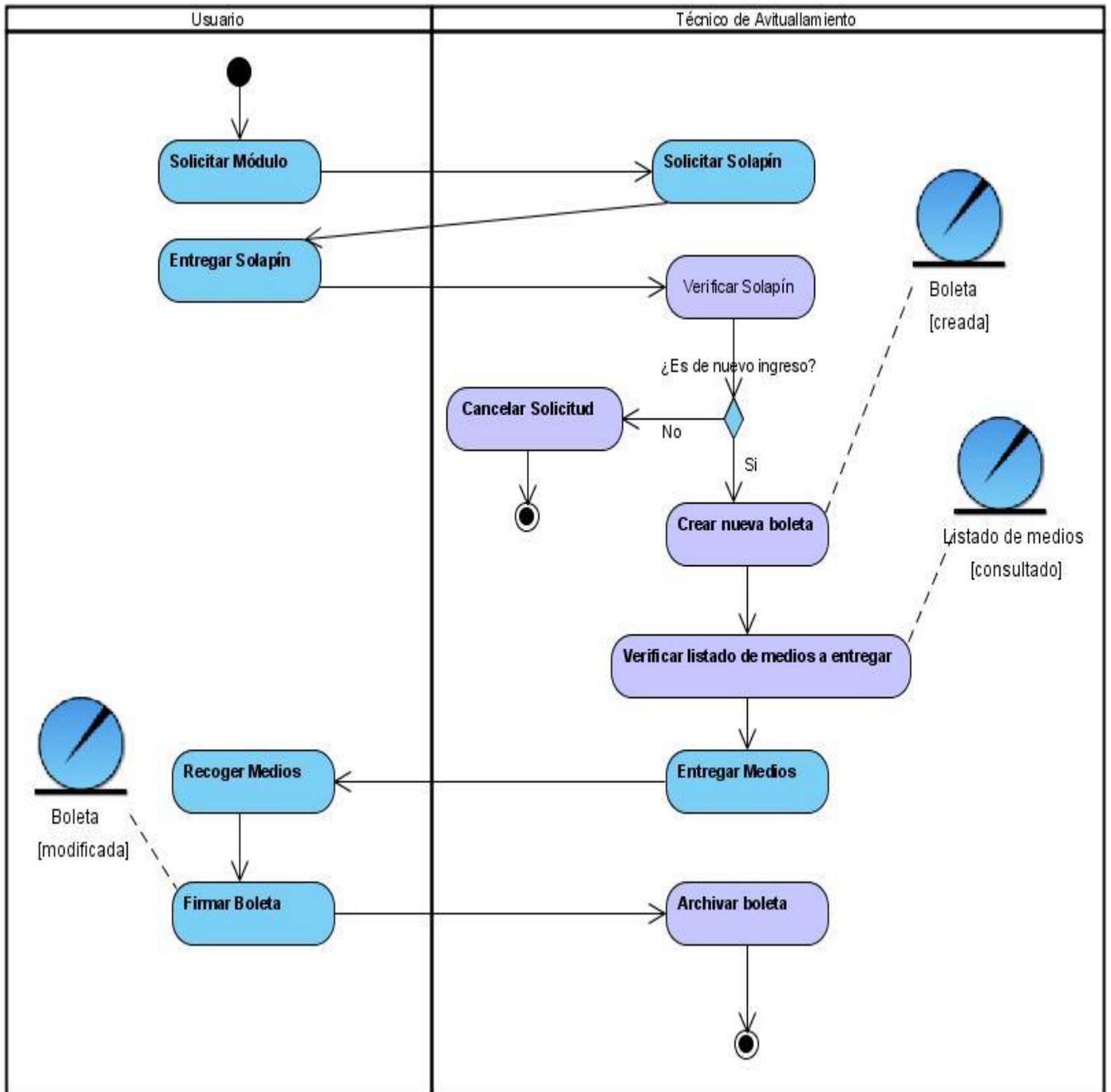


Fig. 2.6 Diagrama de actividades CU: "Recoger Módulo".

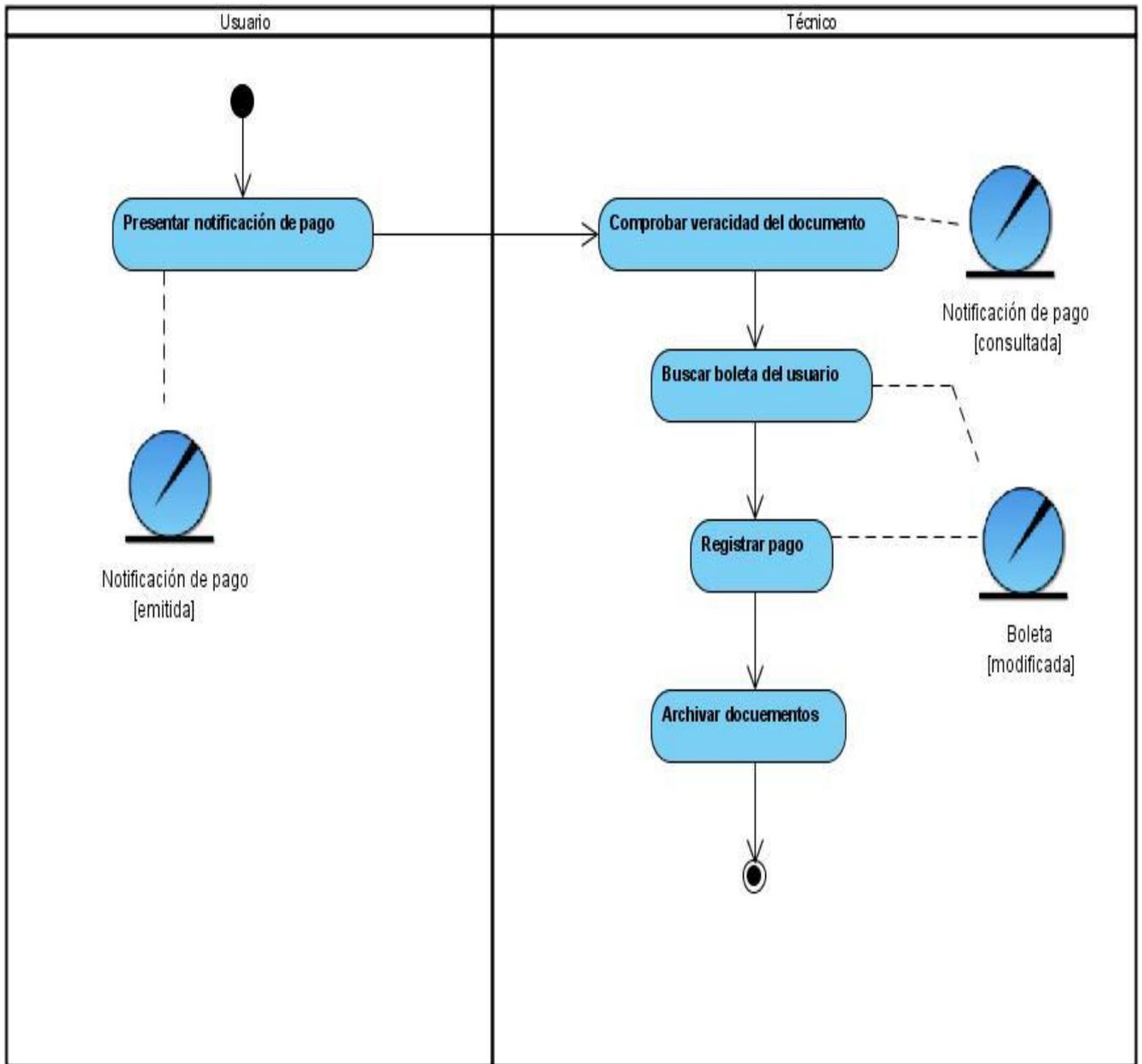


Fig. 2.7 Diagrama de actividades CU: "Registrar Pago".

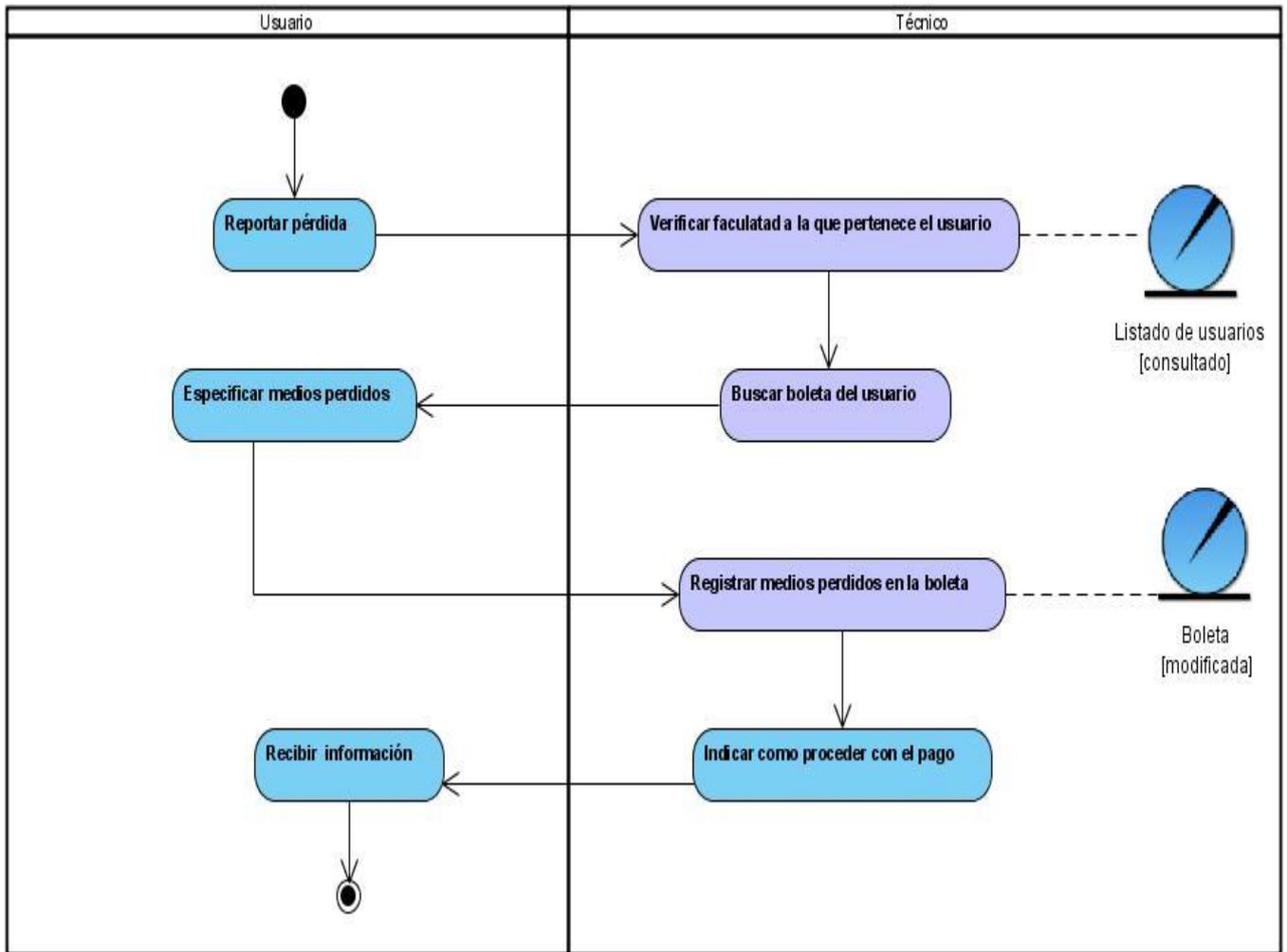


Fig. 2.8 Diagrama de actividades CU: “Reportar Pérdida”.

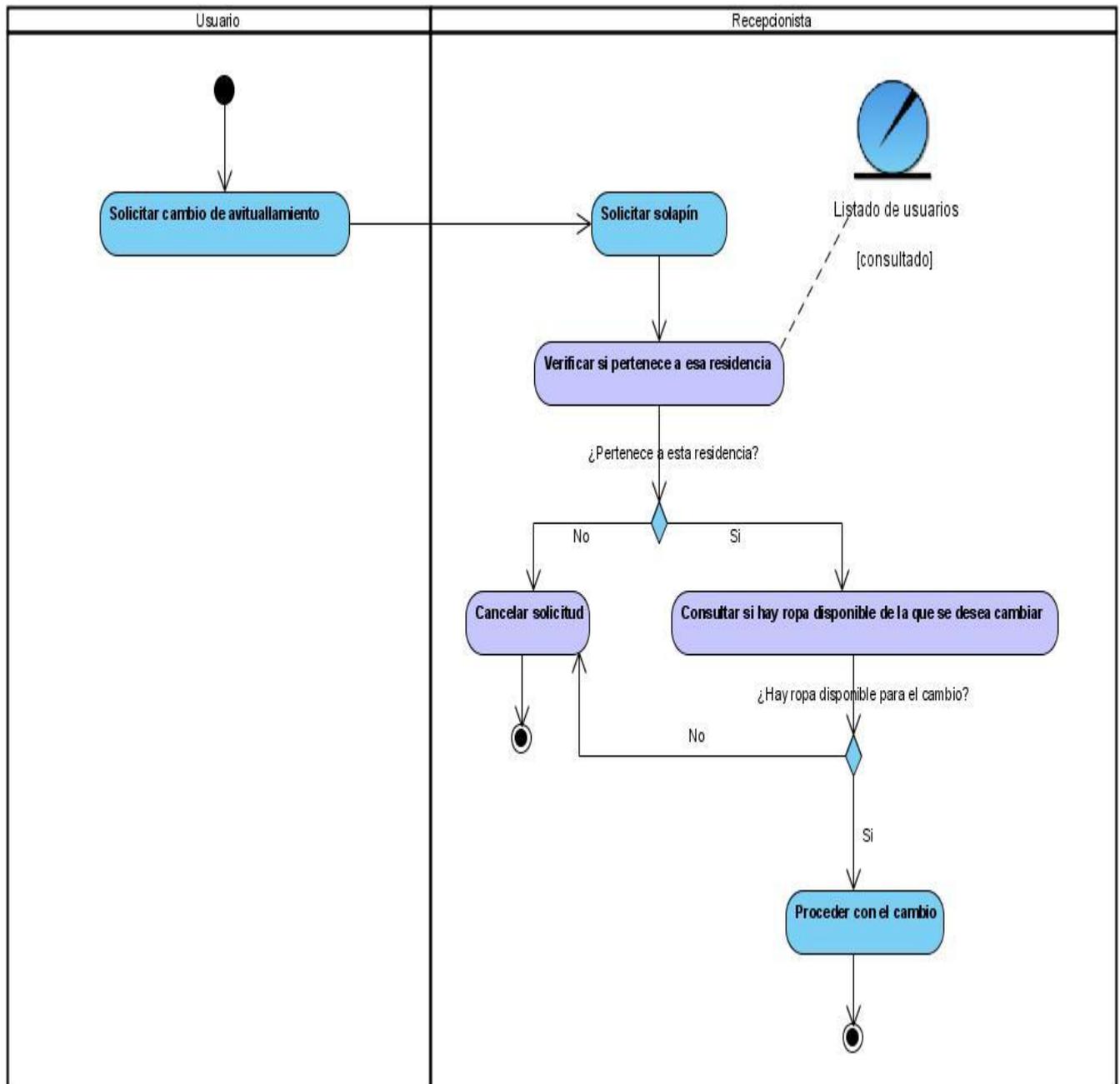


Fig. 2.9 Diagrama de actividades CU: “Realizar Cambio”.

2.6.6 Modelo de objetos del negocio

El modelo de Objetos del Negocio describe la realización de cada caso de uso del negocio, estableciendo los trabajadores y la información que en términos generales manipulan, asociados al caso de uso del negocio. Es una vía para entender cómo los trabajadores del negocio y las entidades necesitan ser relacionados y colaborar para realizar el negocio.

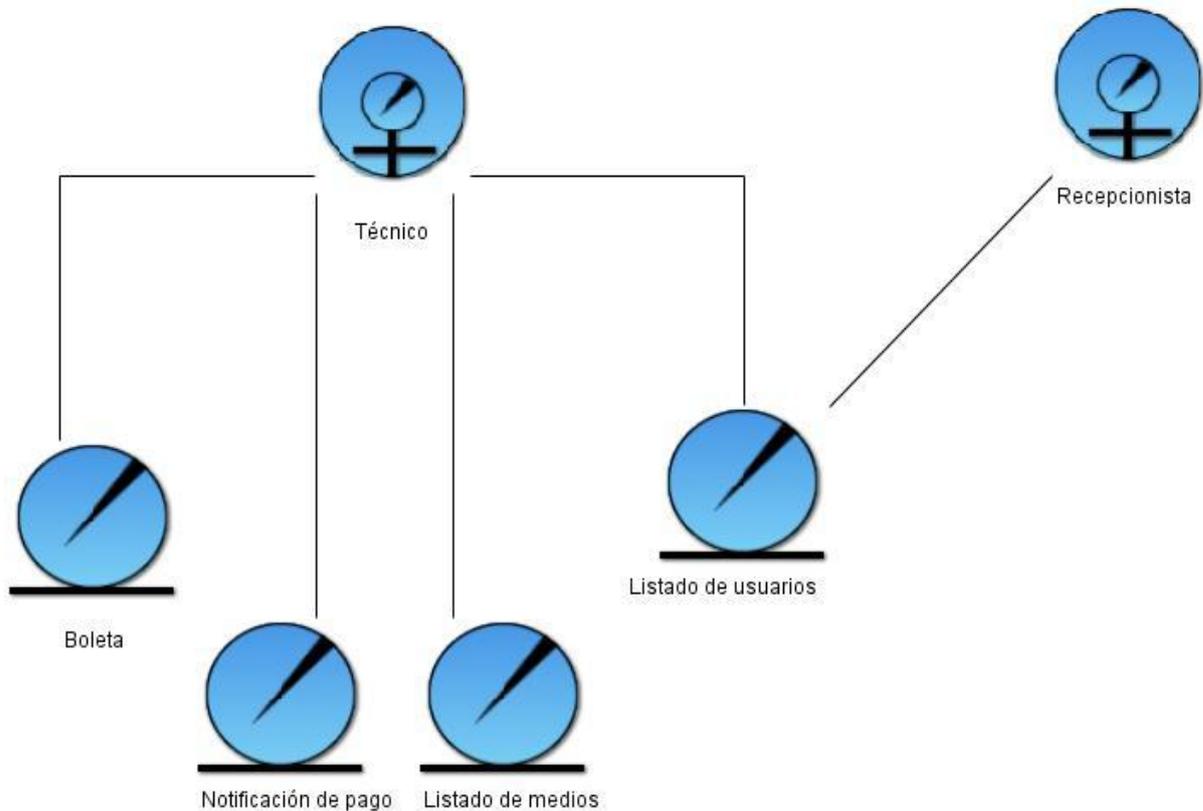


Fig. 2.10 Modelo de objetos del negocio.

2.7 ESPECIFICACION DE REQUERIMIENTOS DE SW

La importancia de una buena especificación puede evaluarse por los beneficios específicos que provee a todas las personas que intervienen en el proceso de desarrollo de un software (clientes/usuarios, desarrolladores y otros involucrados). Dado que permite establecer las bases de un acuerdo entre clientes/usuarios y desarrolladores en cuanto a lo que el producto software debe hacer, reduce el esfuerzo desarrollado para evitar rediseños, recodificación y repruebas, provee las bases para la estimación del costo y entrega del producto, así como una línea base para la validación y verificación, facilita la transferencia a usuarios nuevos o máquinas nuevas y sirve como una base para el aumento del producto final.

- De la definición de los requisitos dependerá la definición de las etapas subsecuentes del desarrollo del software.
- Permite gestionar las necesidades del proyecto en forma estructurada: Cada actividad consiste de una serie de pasos organizados y bien definidos.

- Mejora la capacidad de predecir cronogramas de proyectos, así como sus resultados ya que proporciona un punto de partida y actividades de mantenimiento, tales como estimación de costos, tiempo y recursos necesarios.
- Disminuye los costos y retrasos del proyecto porque durante todo el proceso de IR se van analizando y desarrollando los requisitos de manera que no queden errores para una correcta implementación.
- Mejora la calidad del software ya que esta tiene que ver con cumplir un conjunto de requerimientos.
- Mejora la comunicación entre equipos pues la especificación de requerimientos representa una forma de consenso entre clientes y desarrolladores.
- Evita rechazos de usuarios finales pues estos tienen la posibilidad de considerar sus requerimientos cuidadosamente y revisarlos dentro del marco del problema.

2.7.1 Requerimientos funcionales

La primera etapa de cualquier proceso de negocio comienza por la tarea de la identificación de los requisitos, para el logro de un mejor diseño, modelación y por ende un resultado exitoso ante el cliente que al final es el objetivo del trabajo realizado. Las actividades a automatizar son el punto de partida para definir lo que el sistema debe hacer.

El sistema será capaz de:

RF 1. Gestionar Boleta de control y entrega de avituallamiento.

1.1 Crear una nueva boleta: en caso de que un nuevo estudiante, profesor o especialista llegue a nuestra residencia.

1.2 Eliminar boleta: si alguien se va de baja, licencia o traslado del centro, permitiendo dejar un historial en la base de datos.

1.3 Actualizar Boletas: en caso de algún cambio de apto.

RF 2. Emitir Reportes Diarios:

2.1 Entrega de Productos Químicos a la lavandería "La Flora": actualizarlo luego de haber llevado los artículos a lavar.

2.2 Relación de avituallamiento por edificios.

2.3 Conciliación de Avituallamiento, Lavandería Residencia de profesores y especialistas.

2.4 Existencia en Lavandería.

2.5 Existencia en Almacenes.

2.6 Existencia en poder de los estudiantes.

2.7 Conciliar con el departamento de economía.

RF 3. Realizar cambio de ropa: presentando para ello el solapín, de forma intransferible.

RF 4. Realizar préstamos internos a personal vinculado con la escuela de una forma u otra (profesores externos de guardia y visitas)

RF 5. Mostrar Disponibilidad: mostrar en cada momento si la lavandería está prestando servicios y que ropa hay disponible en la misma.

RF 6. Verificar existencia de una persona en el sistema, o sea en caso de la misma solicitar un cambio de avituallamiento o de ser necesario gestionar su boleta de control.

RF 7. Adicionar persona no UCI, esto ocurre cuando una persona ajena al centro solicita un préstamo de avituallamiento, se requiere tener un registro en el sistema de los medios en manos de la misma.

RF 8. Verificar facultad panificada, se trata de conocer a que facultada corresponde el cambio de avituallamiento a diario.

RF 9. Verificar almacén, antes de comenzar con el cambio se debe consultar la existencia de los medios en el almacén.

RF 10. Gestionar almacén de cambio.

2.7.2 Requerimientos no funcionales

RNF 1. Usabilidad

La aplicación será utilizada por operadores que no necesariamente tendrán experiencia en el uso de la computadora, a estos se les dará un adiestramiento básico en el uso de la computadora y fundamentalmente en el uso del sistema. Estas personas tendrán un nivel de acceso en la aplicación, en la cual realizarán solamente las acciones que tienen asignadas.

RNF 2. Requerimientos de apariencia o interfaz externa.

La interfaz debe ser amigable y fácil de usar, de manera que esto no constituya una dificultad para los usuarios del sistema.

CAPITULO 2: CARACTERISTICAS DEL SISTEMA

Por el uso diario y constante que tendrá el software, la interfaz debe ser agradable, que favorezca el estado de ánimo del cliente y que combine correctamente los colores, tipo de letra y tamaño.

Deben utilizarse plantillas con un mismo estilo para que el usuario no se sienta perdido dentro de la aplicación.

RNF 3. Seguridad.

El sistema debe comunicarse usando un protocolo seguro, (http).

Mantener la integridad de la información, es decir que no se pierda durante su almacenamiento o transporte.

Los datos no pueden viajar de forma transparente por la red, deben ser encriptados.

Chequear si el usuario que está accediendo al sistema esta autenticado y brindarle servicio de autenticación.

Confidencialidad: La información del Avituallamiento será mostrada a todas personas de nuestra universidad puedan estar informadas sobre todo lo que acontece, pudiendo ser modificada solo por personal autorizadas.

Integridad: La información manejada por el sistema tendrá una cuidadosa protección para que no puedan ser modificadas ya que se alterarían inventarios de almacén y de la lavandería permitiendo desvío de de recursos y mal manejo del material con que contamos.

Disponibilidad: Los usuarios autorizados a poder hacer cambios, entrar datos, y eliminarlos se registraran en el sistema como lo que son y a partir de eso tendrán la posibilidad de gestionar lo que a ellos le corresponde pero no ha todo el sistema para la protección de los datos que en ella se encuentran.

RNF 4. Restricciones en el diseño y la implementación

Se debe realizar una aplicación que permita dar respuesta en el menor tiempo posible, garantizando la calidad del sistema y la conexión con el Web Service, la base de datos debe ser desarrollada en PostgreSQL. Para garantizar el desarrollo de la aplicación se utilizará como guía el Proceso Unificado de Desarrollo, obteniendo como documentación los distintos artefactos propuestos. Se utilizará para realizar los modelos del sistema UML (Unified Modelling Language) y como herramienta de apoyo a este Lenguaje de Modelación se utiliza Visual Paradigm.

Teniendo en cuenta las condiciones futuras de usabilidad del producto, se supone el software debe ser sencillo en cuanto a implementación para lograr velocidad de ejecución debido a que el área de trabajo o entorno donde será implantado requiere del procesamiento de información de forma muy consecutiva, en poco espacio de tiempo. Técnicos de avituallamiento y personal vinculado disponen de un control estricto sobre los medios y servicios que brindan porque lo que se debe lograr además restricciones seguras en cuanto a autenticación y control de acceso con el soporte y almacenamiento de información. Sobre la base de la relación de los trabajadores del área de la residencia y su poca vinculación con toda la nueva tecnología de la universidad, se cree poco adiestrados en el tema a los mismo por lo que se debe restringir la creación de la aplicación o interfaz externa lo mas simple, atractiva, usable, rápido y confiable posible.

RNF 5. Soporte

Para garantizar el soporte de esta aplicación, se documentará la misma con un manual de ayuda para los usuarios, así como la posibilidad de emitir sus quejas y sugerencias a los desarrolladores, por correo o por teléfono, realizar mantenimiento al sistema y darle solución a cualquier problema que surja con la aplicación.

RNF 6. Rendimiento

La aplicación debe estar concebida para un consumo mínimo de recursos, permitiendo de esta forma no afectar el correcto funcionamiento de la computadora en que se esté utilizando la misma.

RNF 7. Hardware

Se requiere que las computadoras tengan tarjeta de red.

Se requiere tengan al menos 64 MB de memoria RAM.

Se requiere al menos 100MB de disco duro.

Procesador 512 MHz como mínimo.

RNF 8. Legales

Velar porque todos las herramientas utilizadas para la creación del software sean libres o que estén avaladas.

2.8 SISTEMA

La modelación del sistema constituye una representación principalmente de los requisitos funcionales de software, donde se define la interrelación usuario-sistema como base a sus expectativas. Los artefactos actores, casos de uso y sus descripciones unido a los prototipos de interfaz conforman el Modelo del Sistema.

2.8.1 Actores del sistema

El artefacto actor del sistema puede intercambiar información con él, puede ser un recipiente pasivo de información o puede representar el rol que juega una o varias personas, un equipo o un sistema automatizado.

Los antiguos trabajadores del negocio son los actuales candidatos a actores del sistema y si algún actor del negocio va a interactuar con el sistema, entonces también será un actor del mismo.

Actor	Descripción
Técnico de avituallamiento	Encargado de administrar el sistema en cuanto a funcionalidades como: la gestión de las boletas y la emisión de reportes.
Recepcionista de lavandería	Encargado del proceso de cambio y préstamo de ropa.
Actor del sistema	Encargado de administrar el sistema, de dar los permisos de acceso al mismo.

Tabla 2.2 Actores del sistema.

2.8.2 Diagrama de casos de uso del sistema.

El modelado de casos de uso es la técnica más efectiva para modelar los requisitos del sistema. Los casos de uso se utilizan para modelar el funcionamiento que desea el cliente que tenga el sistema.

Por otro lado, los casos de uso son las funciones que proporciona un sistema para añadir valor a sus usuarios. Estos se han adoptado casi universalmente para la captura de requisitos de sistemas de software, sin embargo son más que simplemente una herramienta para la captura de requisitos; sino que dirigen todo el proceso de software [Jacobson00].

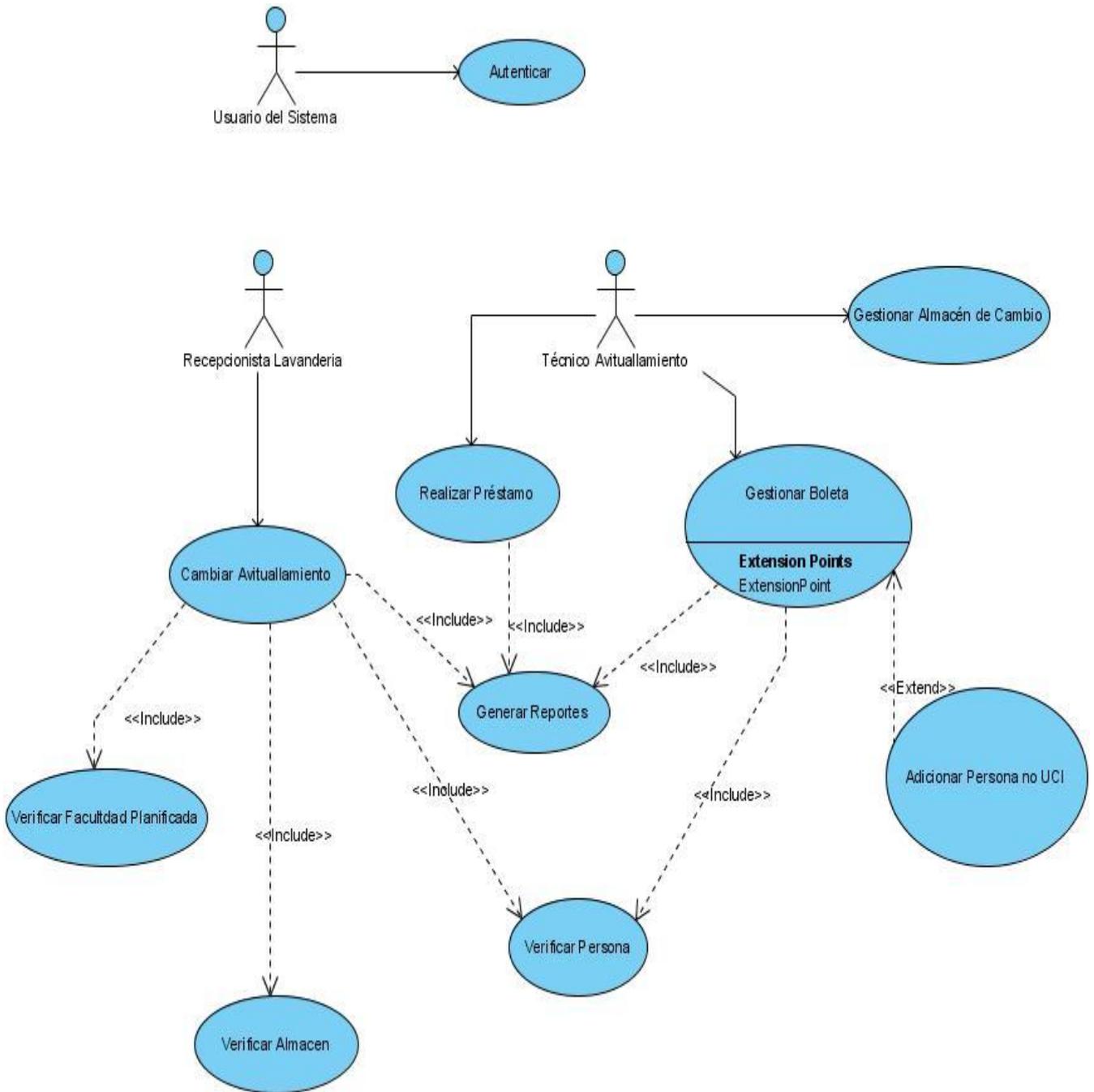


Fig. 2.11 Diagrama de Casos de Uso del Sistema.

2.8.2 Especificación de los CUS.

Caso de Uso:	Gestionar Boleta
Actores:	Técnico de Avituallamiento
Resumen:	El caso de uso comienza cuando un usuario se

CAPITULO 2: CARACTERISTICAS DEL SISTEMA

	presenta en el departamento para realizar alguna gestión con su boleta de control de entrega y recepción de avituallamiento.
Precondiciones:	El sistema debe estar actualizado.
Flujo Normal de Eventos	
Sección “Crear Boleta”	
Acción del Actor	Respuesta del Sistema
<p>1. El técnico presiona la opción de “Crear nueva boleta”.</p> <p>2. Llena el campo del “Nombre” y “No. de solapín”.</p> <p>2.2 El actor llena el resto de los campos que quedan en blanco y da clic en el botón “Crear”. (Ver Boleta de inscripción)</p> <p>2.4 El actor da clic en el botón “Imprimir” o “Cancelar”.</p>	<p>1.1 El sistema muestra el formulario “Tarjeta de Control de Entrega y Recepción de Avituallamiento”</p> <p>2.1 El sistema muestra la opción de comprobar si está o no en el “listado de prematrícula”, dando clic en el botón “Comprobar”.</p> <p>2.3 El sistema guarda los datos entrados, creando así una nueva boleta en la BD y muestra un mensaje: “¿Desea imprimir copia?”</p>
Prototipo de Interfaz	



Gestionar Boleta de Control de Entrega de Avituallamiento
Crear nueva Boleta

Nombre: **CI:**
Apellidos: **No. Solapín:**
Sexo: **Facultad:**

- | | |
|-------------------------------------|------------------------------------|
| <input type="checkbox"/> Sábanas | <input type="checkbox"/> Toalla |
| <input type="checkbox"/> Funda | <input type="checkbox"/> Percheros |
| <input type="checkbox"/> Colcha | <input type="checkbox"/> Almohada |
| <input type="checkbox"/> Mosquitero | <input type="checkbox"/> Sobrecama |

Sección “Eliminar Boleta”

Acción del Actor	Respuesta del Sistema
<p>3. El actor selecciona la opción de eliminar un residente del registro.</p> <p>3.2. Llena el campo del “No. de solapín”.</p> <p>3.6 Da clic en el botón “Aceptar”.</p>	<p>3.1 El sistema muestra un formulario en el que pide el número de solapín del residente.</p> <p>3.3 Verifica si los datos existen registrados en la BD.</p> <p>3.4 Elimina de la BD la boleta correspondiente al residente.</p> <p>3.5 Muestra un mensaje de información: “La boleta ha sido eliminada del sistema”.</p>

Prototipo de Interfaz



Gestionar Boleta de Control de Entrega de Avituallamiento
Eliminar Boleta

Nombre: CI:

Apellidos: No. Solapín:

Sexo: Facultad:

<input type="checkbox"/> Sábanas	<input type="checkbox"/> Toalla
<input type="checkbox"/> Funda	<input type="checkbox"/> Percheros
<input type="checkbox"/> Colcha	<input type="checkbox"/> Almohada
<input type="checkbox"/> Mosquitero	<input type="checkbox"/> Sobrecama

Sección “Actualizar Boleta”

Acción del Actor	
<p>4. El actor selecciona la opción de “Actualizar una boleta del registro”.</p> <p>4.2 El actor llena el campo “No. de solapín” y da clic en el botón “Comprobar”.</p> <p>4.5 El actor llena o cambia los campos que sean necesarios y da clic en el botón “Actualizar”.</p>	<p>4.1 El sistema muestra el formulario “Actualizar boleta” con el campo “No. de solapín” activado y el resto deshabilitados.</p> <p>4.3 Verifica si los datos existen registrados en la BD.</p> <p>4.4 Muestra el formulario “Actualizar boleta” con todos los campos habilitados.</p> <p>4.6 Se actualiza en la base de datos la boleta correspondiente al residente.</p>

Prototipo de Interfaz

Gestionar Boleta de Control de Entrega de Avituallamiento
Actualizar Boleta

Nombre: CI:

Apellidos: No. Solapín:

Sexo: Facultad:

Sábanas Toalla
 Funda Percheros
 Colcha Almohada
 Mosquitero Sobrecama

Flujo Alternativo

2.1 En caso de encontrarse ya en la BD, el sistema muestra un mensaje de error "Acción inválida, el usuario ya se encuentra registrado en el sistema".

3.3 Si la boleta no se encuentra en la BD, el sistema muestra un mensaje de que no existen esos datos y muestra otro formulario para que entre el número de solapín nuevamente o el nombre del residente en caso de que el número de solapín sea incorrecto.

4.3 Si la boleta del residente no se encuentra en la BD, el sistema muestra un mensaje de error "Acción inválida, el usuario no se encuentra registrado en el sistema" y muestra nuevamente el formulario "Actualizar boleta" con los campos "No. de solapín" o "Nombre del usuario".

Poscondiciones	
Mejoras	
Prioridad	Crítico.
Referencia	RF 1

CAPITULO 2: CARACTERISTICAS DEL SISTEMA

Caso de Uso:	Cambiar Avituallamiento.
Actores:	Recepcionista de Lavandería
Resumen:	El caso de uso comienza cuando un usuario se presenta a los locales de cambio de avituallamiento y solicita que se le cambie la ropa por otra limpia.
Precondiciones:	El sistema debe estar actualizado.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
<p>1. El actor presiona la opción de “Cambiar Avituallamiento”.</p> <p>2. El actor marca el solapín (por el código de barra) o teclea el número.</p> <p>3. Especifica el avituallamiento a cambiar llenando los campos, (si no desea cambiar algún tipo de medio puede dejar el campo en blanco) y da clic en el botón “Aceptar”.</p> <p>4. El actor da clic en el botón “Aceptar”.</p>	<p>1.1 El sistema muestra la interfaz de Cambio de Avituallamiento, incluyendo un formulario para insertar el número de solapín.</p> <p>2.1 Verifica si los datos existen registrados en la BD.</p> <p>2.2 El sistema muestra la interfaz con (Ver anexo1)</p> <p>3.1 El sistema verifica en la BD la existencia de los medios (Cantidad disponible por cada tipo).</p> <p>3.2 Muestra un mensaje de que ha sido cambiado el avituallamiento satisfactoriamente.</p> <p>4.1 Disminuye la cantidad en uno del tipo de ropa cambiado.</p>
Prototipo de Interfaz	



Cambiar Avituallamiento

Sábanas
 Funda
 Colcha
 Mosquitero

Toalla
 Percheros
 Almohada
 Sobrecama

Flujo Alternativo

2.1 Si el residente no se encuentra en la BD, el sistema muestra un mensaje de que no existen esos datos.

3.1 Si ya no queda disponible del tipo de ropa que desea cambiar el usuario el sistema muestra el mensaje “ya no quedan medios de ese tipo”.

3.1 Si el usuario pide más medios de la cantidad establecida el sistema muestra un mensaje de error y muestra nuevamente el formulario para especificar la cantidad de medios a cambiar.

Poscondiciones	
Mejoras	
Prioridad	Crítico.
Referencia	FR 2
Caso de Uso:	Realizar Préstamo
Actores:	Técnico de Avituallamiento
Resumen:	El caso de uso comienza cuando un usuario se presenta en el departamento para realizar alguna

CAPITULO 2: CARACTERISTICAS DEL SISTEMA

	gestión con préstamo de algún medio del modulo de avituallamiento.
Precondiciones:	El sistema debe estar actualizado.
Flujo Normal de Eventos	
Sección “Préstamo a estudiantes y profesores”	
Acción del Actor	Respuesta del Sistema
<p>1. El actor presiona la opción de “Realizar Préstamo de Avituallamiento”.</p> <p>2. Teclea el # de solapín o CI y da clic en el botón “Comprobar” para ver si existen los datos.</p> <p>3. Especifica los medios y la cantidad y da clic en el botón “Aceptar”.</p> <p>4. Da clic en el botón “Aceptar”.</p>	<p>1.1 El sistema muestra un formulario en el que pide el # de solapín o # CI (Llenar uno de los 2 campos obligatoriamente).</p> <p>2.1 El sistema verifica en la BD si es solapín, salga el # CI, o si es CI salga el solapín.</p> <p>2.2 Muestra un nuevo formulario para que el usuario especifique los medios que serán prestados.</p> <p>3.1 Le suma los medios prestados a los que se le dieron al inscribirse en el sistema. Y se actualiza la BD.</p> <p>3.2 Muestra un mensaje “Se ha realizado un préstamo satisfactoriamente”.</p>
Sección “Préstamo a Trabajadores”	
Precondiciones: El trabajador lleva papel de autorizo.	
Acción del Actor	Respuesta del Sistema
<p>5. El Actor solicita el préstamo de sábanas.</p> <p>6. Inserta el # de CI y da clic en el botón “Comprobar”.</p> <p>7. Da clic en el vínculo crear ficha.</p> <p>8. Inserta: nombre, Apellidos, dirección, CI, y especifica los medios, tipo y cantidad que serán prestados y da clic en el botón “Aceptar”.</p>	<p>5.1 El sistema muestra un formulario donde pide el número de CI.</p> <p>6.1 Verifica si no existe ficha con ese CI.</p> <p>6.2 Si no existe, el sistema muestra el mensaje “CI no encontrado”, el mismo formulario brindando la opción de insertar nuevamente el CI, y un vinculo “Crear Ficha”.</p> <p>7.1 Muestra el formulario “Crear Ficha” con botones Aceptar y Cancelar.</p> <p>7.1. Crea una ficha en la BD con todos los</p>

CAPITULO 2: CARACTERISTICAS DEL SISTEMA

datos y muestra un mensaje “Se ha realizado Satisfactoriamente un Préstamo”.

Prototipo de Interfaz



Realizar Préstamo

Nombre: CI:

Apellidos: No. Solapín:

Sexo: Facultad:

Sábanas

Toalla

Funda

Percheros

Colcha

Almohada

Mosquitero

Sobrecama

Flujo Alternativo

2.1 Si no encuentra el usuario con esos datos, muestra la interfaz con un mensaje “Datos Mal” y el mismo formulario dando la posibilidad de rectificar los datos el usuario puede entrar nuevamente los datos y da clic en el botón Comprobar o simplemente da clic en el botón Cancelar.

2.1 Si existe más de una persona con el mismo número de CI o el mismo nombre y apellidos, saldrá una interfaz con la foto de las personas, el nombre y los apellidos y un mensaje “Personas Encontradas con datos iguales”. El actor especifica quien es la persona seleccionándola y dando clic en un botón Aceptar.

CAPITULO 2: CARACTERISTICAS DEL SISTEMA

6.1 Si existe un CI igual, saldrá una interfaz con la foto de las personas, el nombre y los apellidos y un mensaje “Personas Encontradas con datos iguales”. El actor especifica quien es la persona seleccionándola y dando clic en un botón Aceptar en caso de que exista y se mostrará una interfaz con los datos del trabajador y su ficha de préstamo. En caso de que no exista la persona en la BD se le creara su ficha normalmente aunque exista otra con el mismo CI.

6.1 Si existe la persona en la BD, el sistema muestra la interfaz con sus datos, el actor especifica los medios que serán prestados y se da clic en el botón “Aceptar”.

Poscondiciones	
Mejoras	
Prioridad	Crítico.
Referencia	RF 3
Caso de Uso:	Autenticar
Actores:	Usuario del Sistema.
Resumen:	El caso de uso se comienza cuando el Usuario del Sistema desea autenticarse en nuestra aplicación, la interfaz debe tener acceso a un formulario definido para la Autenticación en el cual se introducen los datos necesarios para la validación del usuario. La interfaz tomara esos datos y comprobara su veracidad en un Repositorio de Información de usuarios, si el usuario no ha sido autenticado antes, los datos son enviados hacia la entidad que contiene todos los usuarios y sino la interfaz mostrara un mensaje de error usuario ya autenticado.
Precondiciones:	El sistema debe estar actualizado.

Flujo Normal de Eventos

Acción del actor	Respuesta del Sistema
<p>2. El actor selecciona el vínculo Autenticarse.</p> <p>3. Introduce el usuario para autenticarse y la contraseña y da clic en el botón “Autenticarse”.</p>	<p>1. El sistema muestra la interfaz “Página Principal.”</p> <p>2.1 El sistema muestra el formulario de autenticación</p>

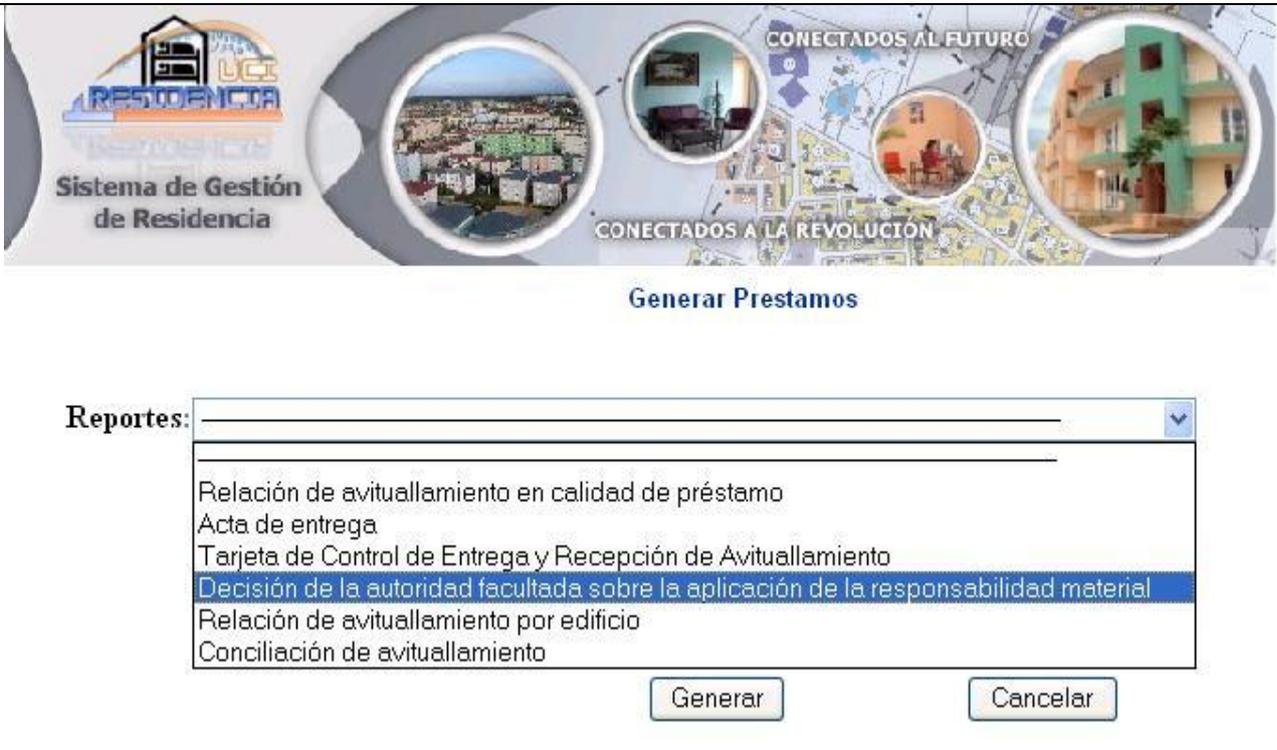
CAPITULO 2: CARACTERISTICAS DEL SISTEMA

	<p>3.1 El sistema chequea los datos en la BD verificando si existen.</p> <p>3.2 El sistema da los permisos según el resultado del chequeo, si es usuario carga la interfaz “Página Principal.” para usuario y si es administrador la interfaz “Página Principal.” y le da los permisos de administración.</p>
Flujo Alterno	
<p>3 Si el usuario no introduce ningún dato y da clic en el botón autenticarse debe mostrar un mensaje "Nombre de usuario y/o Contraseña en blanco"</p> <p>3.1 Si no existe el usuario el sistema muestra un mensaje de error: “Usuario o Contraseña Incorrecta” y muestra nuevamente el formulario indicando que debe ingresar nuevamente los datos. El Usuario puede ingresar los datos nuevamente y dar Aceptar o Cancelar.</p>	
Poscondiciones	Que se pueda registrar el usuario en el sistema.
Mejoras	
Prioridad	Crítico.
Referencia	TRF1
 <p>Autenticarse</p> <p>Usuario: <input type="text"/></p> <p>Contraseña: <input type="password"/></p> <p style="text-align: center;"> <input type="button" value="Aceptar"/> <input type="button" value="Cancelar"/> </p>	

CAPITULO 2: CARACTERISTICAS DEL SISTEMA

Caso de Uso:	Generar Reportes
Actores:	(inicia) CU: Cambiar Avituallamiento, Realizar Préstamo, Gestionar Boleta
Resumen:	El caso de uso inicia cuando es llamado por algún CU base, cuando se hace necesario emitir algún reporte.
Precondiciones:	El sistema debe estar actualizado.
Flujo Normal de Eventos	
Sección “Cambiar Avituallamiento”	
Acción del Actor	Respuesta del Sistema
<ol style="list-style-type: none"> 1. El actor selecciona el tipo de reporte que desea generar: “<i>Relación de avituallamiento por edificio</i>”, “<i>Conciliación de avituallamiento</i>”, da clic en el botón “Aceptar” 2. El actor da clic en el botón “Imprimir”. 	<ol style="list-style-type: none"> 1.1 El sistema muestra una planilla con el reporte generado. 2.1 El sistema envía la información a la impresora.
Sección “Realizar Préstamo”	
Acción del Actor	Respuesta del Sistema
<ol style="list-style-type: none"> 3. El actor selecciona el tipo de reporte que desea generar: “<i>Relación de avituallamiento en calidad de préstamo</i>”, “<i>Acta de entrega</i>”, da clic en el botón “Aceptar” 4. El actor da clic en el botón “Imprimir”. 	<ol style="list-style-type: none"> 3.1 El sistema muestra una planilla con el reporte generado. 4.1 El sistema envía la información a la impresora.
Sección “Gestionar Boleta”	
Acción del Actor	Respuesta del Sistema
<ol style="list-style-type: none"> 5. El actor selecciona el tipo de reporte que desea generar: “<i>Tarjeta de Control de Entrega y Recepción de Avituallamiento</i>”, “<i>Decisión de la autoridad facultada sobre la aplicación de la responsabilidad material</i>”, da clic en el botón “Aceptar” 6. El actor da clic en el botón “Imprimir”. 	<ol style="list-style-type: none"> 5.1 El sistema muestra una planilla con el reporte generado. 6.1 El sistema envía la información a la impresora.

Prototipo de Interfaz



Generar Prestamos

Reportes:

- Relación de avituallamiento en calidad de préstamo
- Acta de entrega
- Tarjeta de Control de Entrega y Recepción de Avituallamiento
- Decisión de la autoridad facultada sobre la aplicación de la responsabilidad material**
- Relación de avituallamiento por edificio
- Conciliación de avituallamiento

Flujos Alternos

Poscondiciones	
Mejoras	
Prioridad	
Referencia	
Caso de Uso:	Gestionar Almacén de Cambio.
Actores:	Técnico de Lavandería.
Resumen:	El caso de uso comienza cuando el administrador se dirige al almacén e indica que va a realizar alguna gestión con cierta cantidad de medios de avituallamiento.
Precondiciones:	El sistema debe estar actualizado.

Flujo Normal de Eventos

Sección “Ingresar Medios al Amacen”

Acción del Actor	Respuesta del Sistema
1. El técnico presiona la opción de “Ingresar Medios”.	1.1 El sistema muestra el formulario “Ingresar Medios” con los campos indicados para cada medio de avituallamiento (Ver Boleta de Control de Entrega de
2. Llena los campos correspondientes a	

CAPITULO 2: CARACTERISTICAS DEL SISTEMA

<p>los medios a ingresar y la cantidad indicada.</p> <p>y da clic en el botón “Ingresar”.</p> <p>2.5 El actor da clic en el botón “Aceptar”.</p>	<p>Avituallamiento y Boleta de Préstamo de Avituallamiento), además de la cantidad indicada.</p> <p>2.3 El sistema guarda los datos entrados incrementando la cantidad de los medios.</p> <p>2.4 El sistema muestra un mensaje “Se han ingresado correctamente los medios”</p>
Sección “Eliminar Medios del Almacén”.	
Acción del Actor	Respuesta del Sistema
<p>3. El actor selecciona la opción de eliminar medios de depreciación.</p> <p>3.2. Llena los campos por cada tipo de medios que se indicaron como baja y da clic en el botón Aceptar.</p> <p>3.5 Da clic en el botón Aceptar.</p>	<p>3.1 El sistema muestra un formulario en el que pide la cantidad de medios a eliminar por tipo. (Ver Boleta de Control de Entrega de Avituallamiento y Boleta de Préstamo de Avituallamiento)</p> <p>3.3 Elimina de la BD la cantidad de medios indicada.</p> <p>3.4 Muestra un mensaje de que han sido eliminados los datos.</p>
Sección “Actualizar la Cantidad Medios”	
Acción del Actor	
<p>4. El actor selecciona la opción de “Actualizar cantidad de medios”.</p> <p>4.2 El actor llena o cambia los campos que sean necesarios y da clic en el botón “Actualizar”.</p> <p>4.5 Da clic en el botón “Aceptar”.</p>	<p>4.1 El sistema muestra el formulario “Actualizar Medios” con los campos indicados por cada medio y la cantidad.</p> <p>4.3 Se actualiza en la base de datos la cantidad de medios correspondiente.</p> <p>4.4 Muestra mensaje “Datos Actualizados Satisfactoriamente”.</p>
Poscondiciones	
Mejoras	
Prioridad	Crítico.
Referencia	

Caso de Uso:	Verificar Almacén
Actores:	Recepcionista de Lavandería

CAPITULO 2: CARACTERISTICAS DEL SISTEMA

Resumen:	El caso de uso comienza cuando el recepcionista de lavandería verifica si existen medios disponibles (limpios) en la lavandería para realizar la rotación de Avituallamiento.	
Precondiciones:	El sistema debe estar actualizado.	
Escenario Cambiar Avituallamiento		
Flujo de Eventos		
Acción del Actor	Respuesta del Sistema	
<p>1. El actor especifica el avituallamiento a cambiar llenando los campos, (si no desea cambiar algún tipo de medio puede dejar el campo vacío) y da clic en el botón “verificar existencia de Medios”.</p> <p>2. El actor da clic en el botón “Aceptar”.</p>	<p>1.1 El sistema verifica en la BD la existencia de los medios (Cantidad disponible por cada tipo).</p> <p>1.2 En caso de estar disponibles estos medios muestra un mensaje “Medios Disponibles”.</p>	
Flujo Alterno		
1.2 Si ya no queda disponible del tipo de ropa que desea cambiar (Basta con que no exista un solo medio indicado) el sistema muestra el mensaje “ya no quedan medios de ese tipo”.		
Poscondiciones		
Mejoras		
Prioridad		
Referencia		

Caso de Uso:	Verificar Facultad Planificada.
Actores:	Recepcionista de Lavandería
Resumen:	El caso de uso comienza cuando un usuario se presenta a los locales de cambio de avituallamiento y solicita que se le cambie la ropa por otra limpia.
Precondiciones:	El sistema debe estar actualizado.

CAPITULO 2: CARACTERISTICAS DEL SISTEMA

Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
<p>1. El actor marca el solapín (por el código de barra) o teclea el número.</p> <p>2. El actor da clic en el botón “Aceptar”.</p>	<p>1.1 El sistema verifica si los datos pertenecen a la facultad indicada en ese día.</p> <p>1.2 Si pertenecen los datos, el sistema muestra un mensaje de “Puede Proceder al Cambio de Avituallamiento”.</p>
Flujo Alterno	
<p>1.2 Si el residente no pertenece en la facultad planificada, el sistema muestra un mensaje de “No pertenece a la facultad, Vuelva el día que le corresponde”.</p>	
Poscondiciones	
Mejoras	
Prioridad	Crítico.
Referencia	

Caso de Uso:	Verificar Persona.
Actores:	(inicia) CU: Cambiar Avituallamiento, Gestionar Boleta de Control de Entrega de avituallamiento
Resumen:	El caso de uso inicia cuando es llamado por algún CU base, cuando se hace necesario verificar si la persona existe.
Precondiciones:	El sistema debe estar actualizado.
Flujo Normal de Eventos	
Sección “Crear Boleta”	
Acción del Actor	Respuesta del Sistema
<p>1. Llena el campo del “Nombre” y “No. de solapín”.</p> <p>2 El actor llena el resto de los campos que</p>	<p>1.1 El sistema muestra la opción de buscar si está o no en el “listado de pre matrícula”, dando clic en el botón “Comprobar”.</p>

CAPITULO 2: CARACTERISTICAS DEL SISTEMA

quedan en blanco y da clic en el botón “Crear”. (Ver Boleta de Control de Entrega y Recepcion de avituallamiento)	
Sección “Eliminar Boleta”	
Acción del Actor	Respuesta del Sistema
<p>3. El actor selecciona la opción de eliminar un residente del registro.</p> <p>3.2. Llena el campo del “No. de solapín”.</p> <p>3.4 Da clic en el Botón Aceptar.</p>	<p>3.1 El sistema muestra un formulario en el que pide el número de solapín del residente.</p> <p>3.3 Busca en la BD si los datos existen.</p>
Sección “Actualizar Boleta”	
Acción del Actor	Respuesta del Sistema
<p>4. El actor selecciona la opción de “Actualizar una boleta del registro”.</p> <p>4.2 El actor llena el campo “No. de solapín” y da clic en el botón “Comprobar”.</p>	<p>4.1 El sistema muestra el formulario “Actualizar boleta” con el campo “No. de solapín” activado y el resto deshabilitados.</p> <p>4.3 Busca si los datos existen registrados en la BD.</p>
Sección “Cambio de Avituallamiento”	
Acción del actor	Respuesta del Sistema
<p>5. El actor selecciona la opción de “Cambiar Avituallamiento”</p> <p>5.2 El actor presiona el botón “Aceptar”.</p>	<p>5.1 El sistema muestra un formulario, solicitando no. de solapín.</p>
Prototipo de Interfaz	



Mejoras	
Prioridad	Crítico.
Referencia	RF 6

Caso de Uso:	Adicionar Persona No UCI.
Actores:	Técnico de Lavandería.
Resumen:	El caso de uso comienza cuando un usuario se presenta en el departamento para realizar alguna gestión con su boleta de control de entrega y recepción de avituallamiento.
Precondiciones:	El sistema debe estar actualizado.
Flujo Normal de Eventos	
Sección "Crear Boleta"	
Acción del Actor	Respuesta del Sistema
<ol style="list-style-type: none"> 1. El técnico presiona la opción de "Crear nueva boleta". 2. Llena los campos "Nombre" y "Apellidos" y da clic en el botón "Comprobar". 3. El actor llena los campos y da clic en el botón "Crear". (Ver Boleta de Control de Entrega y Recepción de avituallamiento). 	<ol style="list-style-type: none"> 1.1 El sistema muestra el formulario "Tarjeta de Control de Entrega y Recepción de Avituallamiento" 2.1 Si está en el Listado de Especialistas, muestra formulario "Tarjeta de Control de Entrega y Recepción de Avituallamiento".

2.4 El actor da clic en el botón "Imprimir" o "Cancelar".	3.1 El sistema guarda los datos entrados, creando así una nueva boleta en la BD y muestra un mensaje: "¿Desea imprimir copia?"
Flujos Alternos:	
2.1 Si no está en el Listado de Especialistas, muestra mensaje de error "No Se Encuentra en El Listado"	
Poscondiciones	
Mejoras	
Prioridad	Crítico.
Referencia	

2.9 CONCLUSIONES

A partir de la realización de los flujos de trabajo modelamiento del negocio y requerimientos, se tiene una mejor comprensión de cómo se desarrollará el sistema. De esta forma quedan sentadas las bases para comenzar con el análisis y diseño de la aplicación. En este capítulo se presentaron los artefactos generados para el correcto desarrollo de cada una de las actividades, de igual manera se propuso una secuencia de pasos a seguir para realizar la captura de requisitos a sistemas que puedan tener características semejantes a las del sistema modelado en este trabajo.

CAPÍTULO 3: ANÁLISIS DEL SISTEMA

3.1 INTRODUCCION

En este capítulo se expone la concepción general referente al análisis del sistema propuesto, donde el análisis tiene como objetivo mantener un modelo eficiente de la solución propuesta que sirva de base para el diseño.

3.2 ANALISIS

En el análisis se realiza un bosquejo del diseño del sistema, incluyendo su arquitectura. Puede no estar mantenido durante todo el ciclo de vida del software.

Diagrama de clases del análisis

El diagrama de clases del análisis posee tres estereotipos conceptuales sobre las clases: control, entidad e interfaz. Una clase de análisis es el resultado de la abstracción de una o varias clases y/o subsistemas del diseño del sistema.

1. *Interfaz:* La clase de interfaz es utilizada para la modelación de la interacción entre el sistema y sus actores (usuarios y sistemas externos). Dicha interacción frecuentemente implica la recepción y presentación de información y peticiones de y hacia los usuarios y sistemas externos. A través de una clase de interfaz se modelan las partes del sistema que dependen de sus actores, o sea, depuran y reúnen los requisitos en los límites del sistema. Por tal razón, un cambio en una interfaz de usuario o en una interfaz de comunicaciones queda generalmente aislado en una o más clases de interfaz (IVAR JACOBSON, 2000).
2. *Control:* Una clase de control es utilizada para representar la coordinación, secuencia, transacciones, y control de otros objetos, y se usan frecuentemente para el encapsulamiento del control de un caso de uso específico. Además, sirven como herramienta para la representación de derivaciones y cálculos complejos, como la lógica del negocio, que no permiten asociación con ninguna información definida, de larga duración, almacenada por el sistema, en otras palabras, una clase de entidad concreta. Constituyen también, un método eficiente para la modelación de los aspectos dinámicos del sistema, debido a su capacidad de manejar y coordinar las acciones y flujos de control principales, así como para la delegación de trabajo a otros objetos, o de entidad o de interfaz.
3. *Entidad:* Una clase de entidad es utilizada para la modelación de la información que posee una vida larga y que generalmente es persistente. Modela además, la información y el comportamiento asociado de algún fenómeno o concepto, como una persona, un objeto o un suceso del mundo real. Es muy común que una clase de entidad se derive directamente de una

clase de entidad del negocio o del modelo del dominio. A pesar de esto, una diferencia fundamental entre una clase de entidad u otra de entidad del negocio radica en que las primeras representan objetos manejados por el sistema, mientras que las otras representan objetos presentes en el negocio y en el dominio del problema. Además de todo lo anterior, las clases de entidad generalmente muestran una estructura de datos lógica y contribuyen a comprender de qué información depende el sistema.

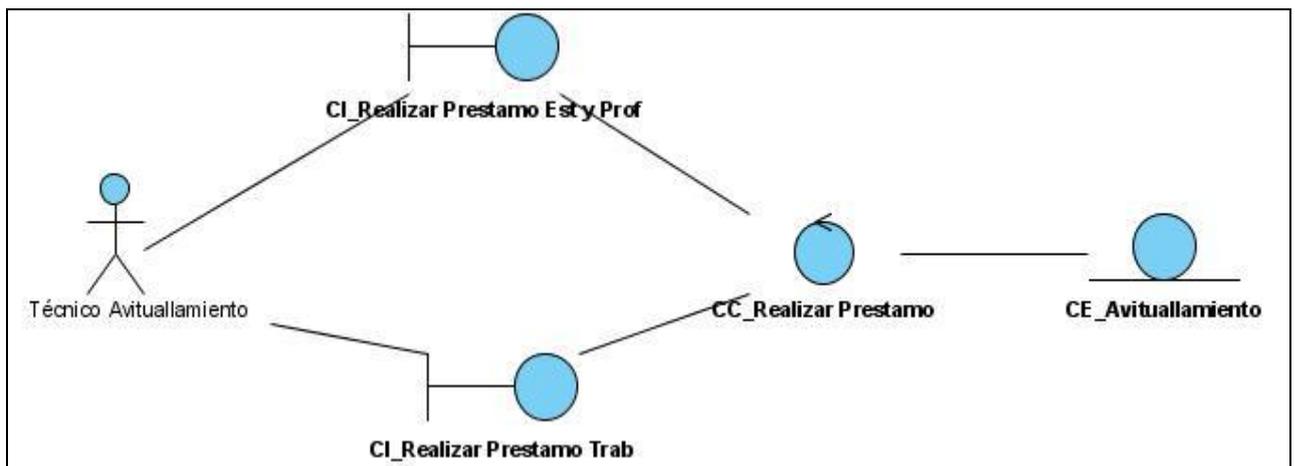


Fig. 3.1 Diagrama de clases del análisis: CU Realizar Préstamo.

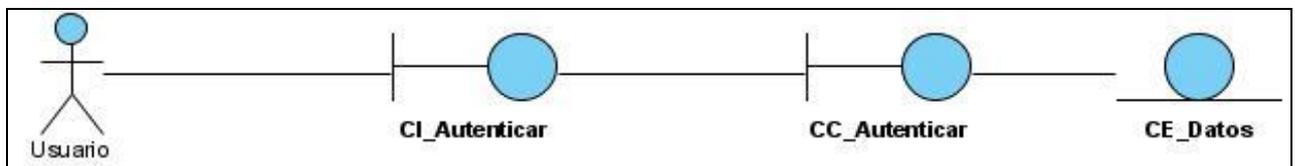


Fig. 3.2 Diagrama de clases del análisis: CU Autenticar usuario.

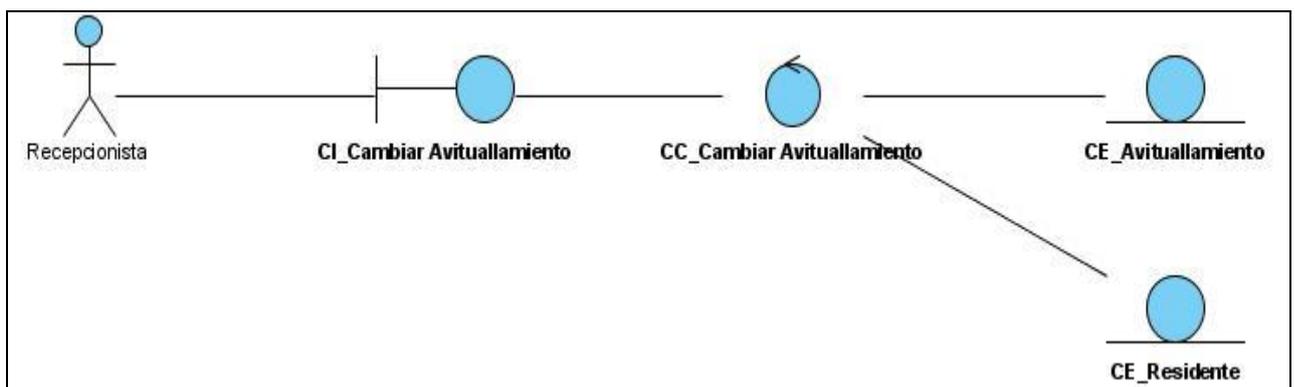


Fig. 3.3 Diagrama de clases del análisis: CU Cambiar Avituallamiento.

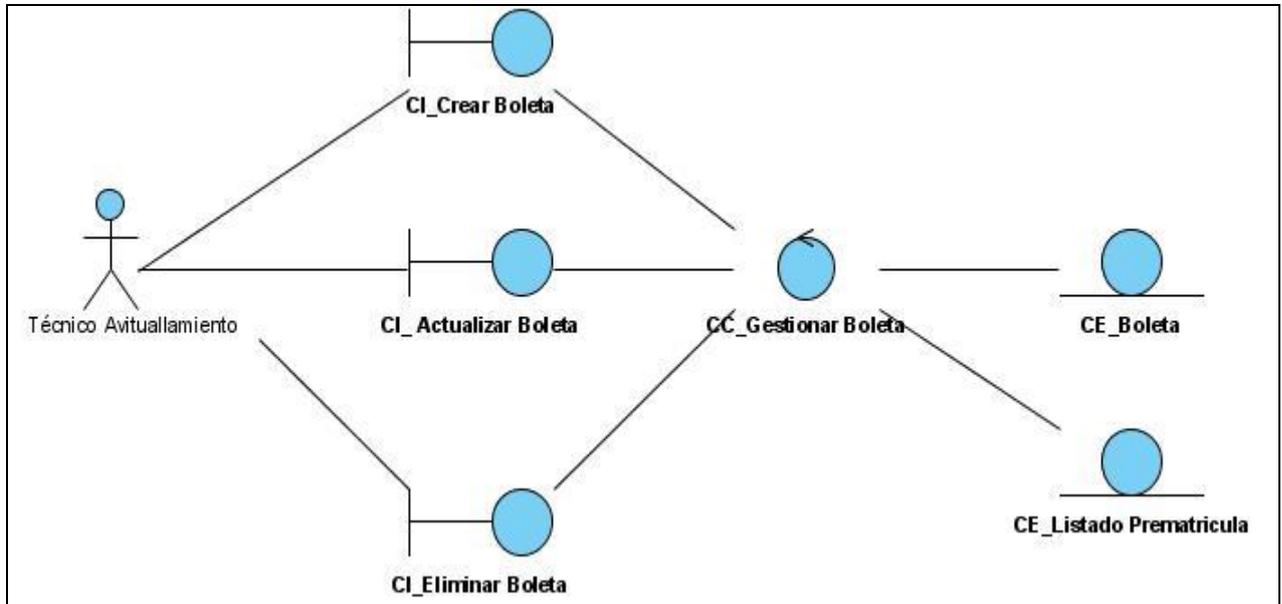


Fig. 3.4 Diagrama de clases del análisis: CU Gestionar Boleta.

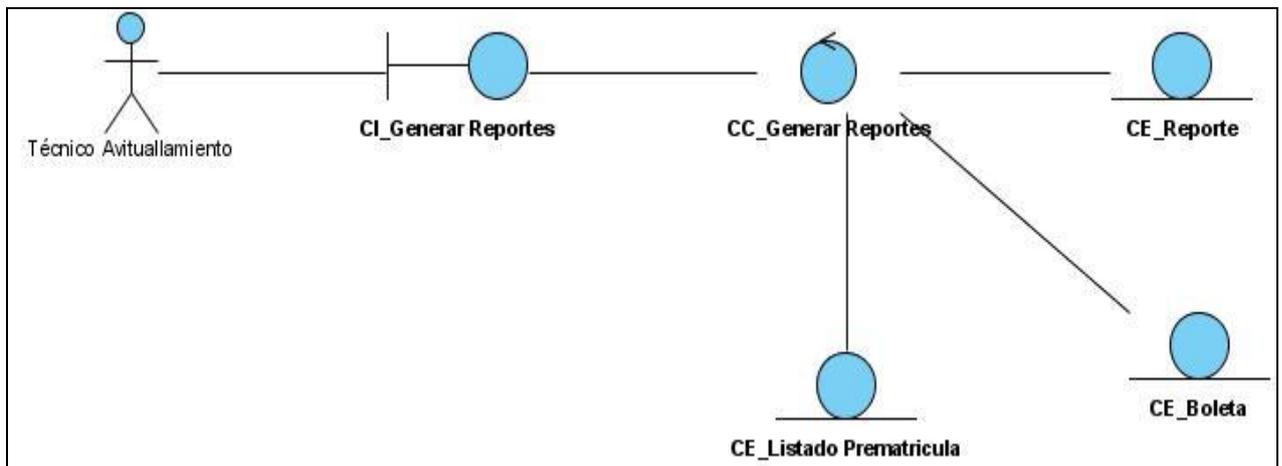


Fig. 3.5 Diagrama de clases del análisis: CU Generar Reportes.

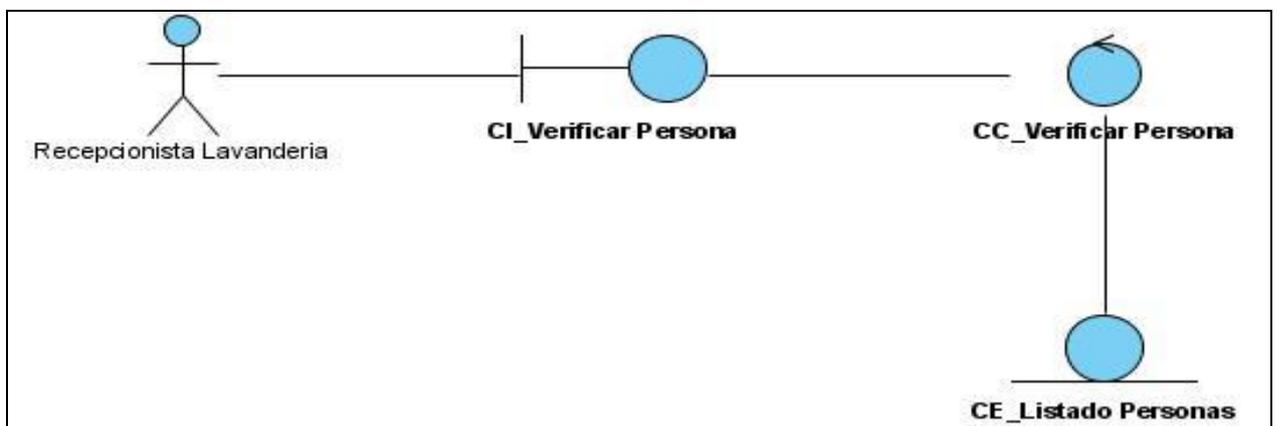


Fig. 3.6 Diagrama de clases del análisis: CU Verificar Persona.



Fig. 3.7 Diagrama de clases del análisis: CU Adicionar Persona no UCI.

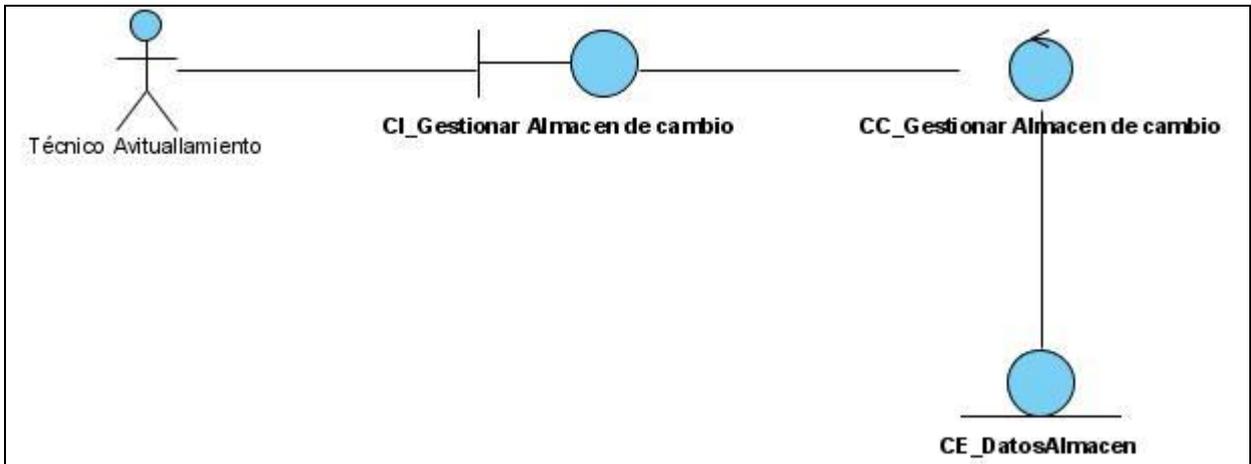


Fig. 3.8 Diagrama de clases del análisis: CU Adicionar Persona no UCI.

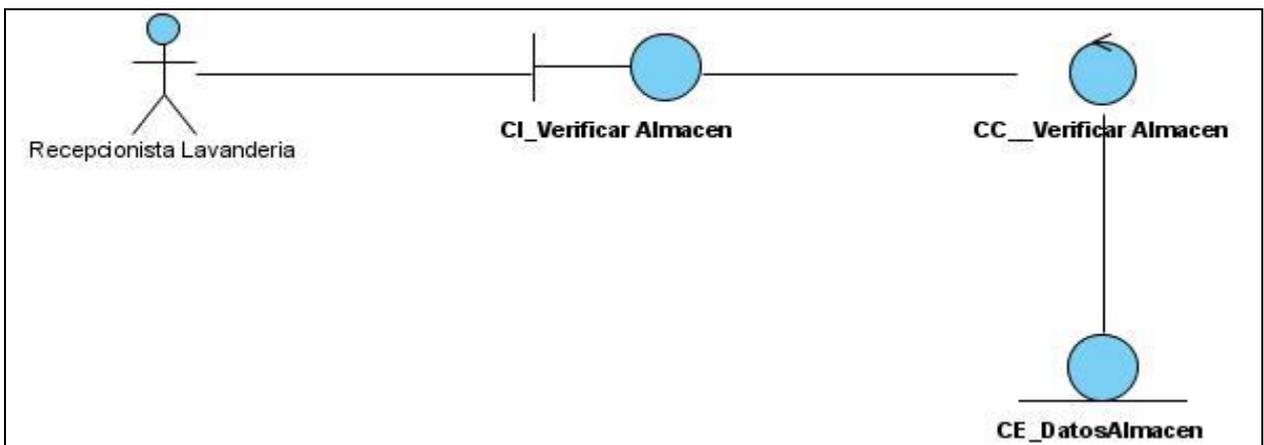


Fig. 3.9 Diagrama de clases del análisis: CU Verificar Almacén.

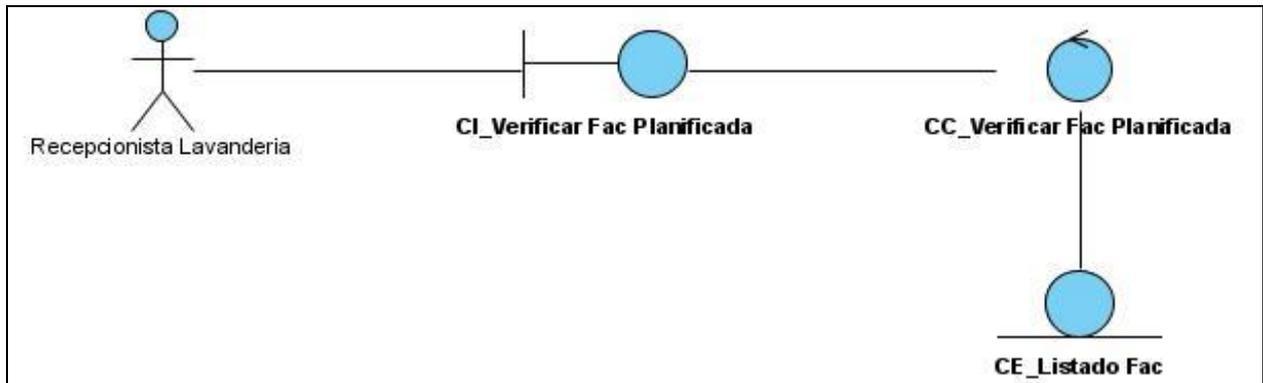


Fig. 3.10 Diagrama de clases del análisis: CU Verificar Facultad Planificada.

3.4 DIAGRAMAS DE INTERACCION

Los diagramas de interacción se utilizan para modelar los aspectos dinámicos de un sistema, lo que conlleva a modelar instancias concretas o prototípicas de clases, interfaces, componentes y nodos, junto con los mensajes enviados entre ellos, todo en el contexto de un escenario que ilustra un comportamiento. En el contexto de las clases describen la forma en que grupos de objetos colaboran para proveer un comportamiento. Mientras que un diagrama de casos de uso presenta una visión externa del sistema, la funcionalidad de dichos casos de uso se recoge como un flujo de eventos utilizando para ello interacciones entre sociedades de objetos.

Existen dos tipos de diagramas de interacción:

- Diagramas de secuencia (sequence diagrams).
- Diagramas de colaboración (collaboration diagrams).

Un diagrama de secuencia es un diagrama de interacción que destaca el orden temporal de los mensajes; un diagrama de colaboración es un diagrama de interacción que destaca la organización estructural de los objetos que envían y reciben mensajes.

3.5 DIAGRAMAS DE SECUENCIA PARA EL ANALISIS DEL SISTEMA

Un diagrama de secuencia de un sistema es una representación que muestra, en determinado escenario de un caso de uso, los eventos generados por actores externos, su orden y los eventos internos del sistema. A todos los sistemas se les trata como una caja negra; los diagramas se centran en los eventos que trascienden las fronteras del sistema y que fluyen de los actores del sistema (CRAING, 1999). En otras palabras, el diagrama de secuencia muestra gráficamente los eventos que fluyen de los actores al sistema. Un diagrama de secuencia muestra las interacciones

entre objetos ordenadas en secuencia temporal. Muestra los objetos que se encuentran en el escenario y la secuencia de mensajes intercambiados entre los objetos para llevar a cabo la funcionalidad descrita por el escenario. La elaboración de los diagramas secuencia del sistema forma parte de la investigación por el conocimiento del sistema, por lo que se incluye en el modelo de análisis.

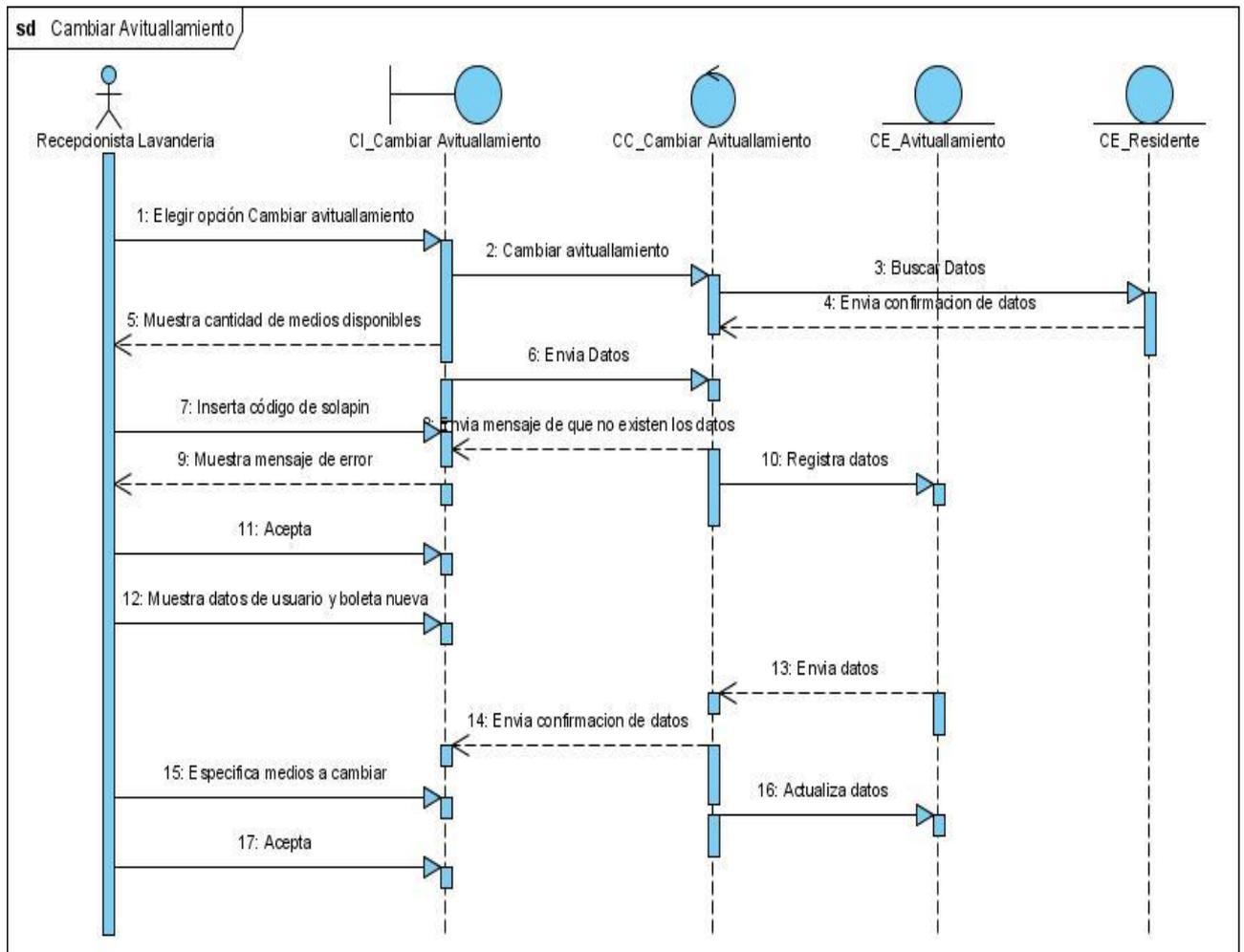


Fig. 3.6 Diagrama de Secuencia: CU Cambiar Avituallamiento.

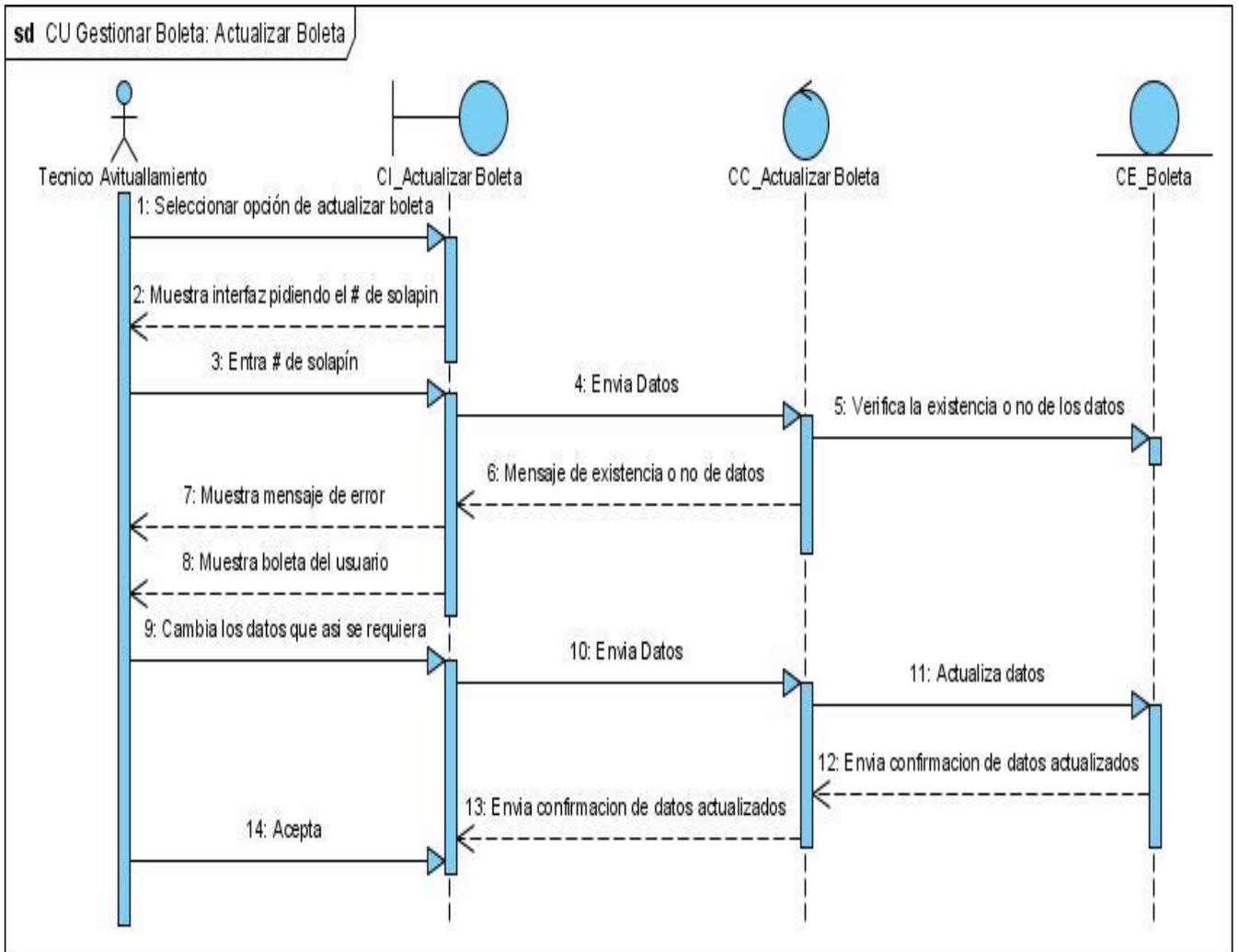


Fig. 3.7 Diagrama de Secuencia: CU Gestionar Boleta: Escenario Actualizar Boleta.

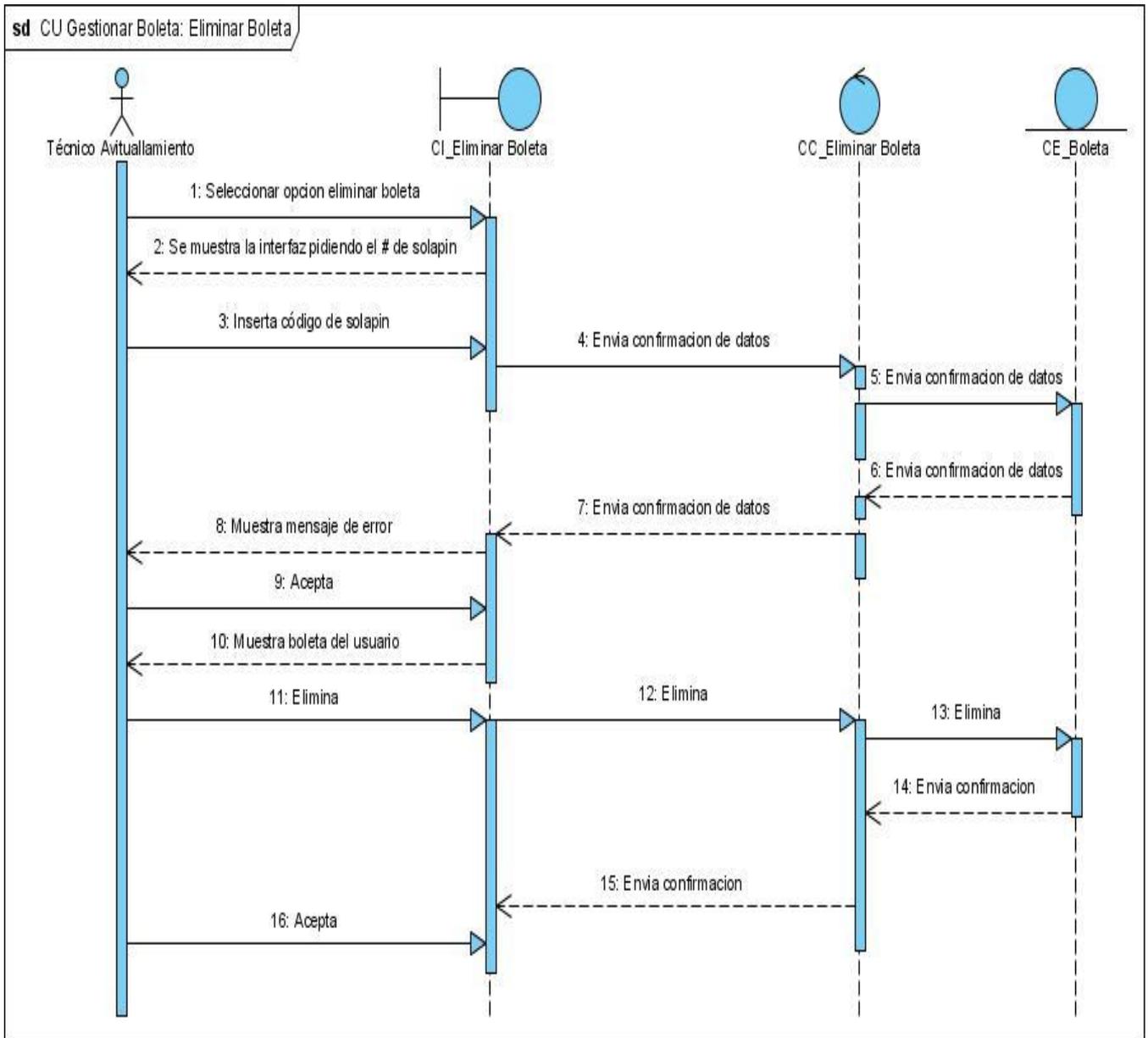


Fig. 3.8 Diagrama de Secuencia: CU Gestionar Boleta: Escenario Eliminar Boleta.

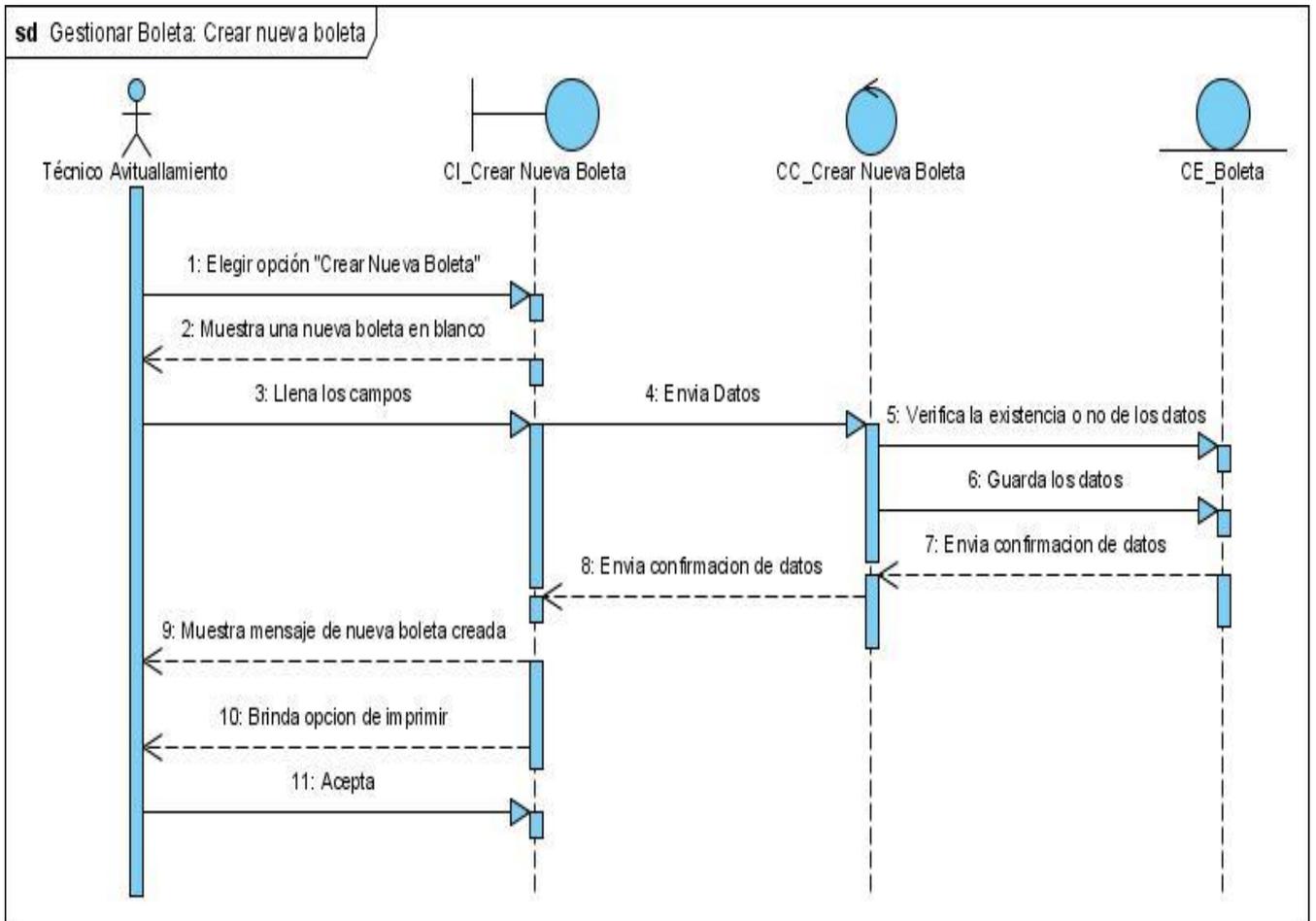


Fig. 3.8 Diagrama de Secuencia: CU Gestionar Boleta: Escenario Crear Boleta.

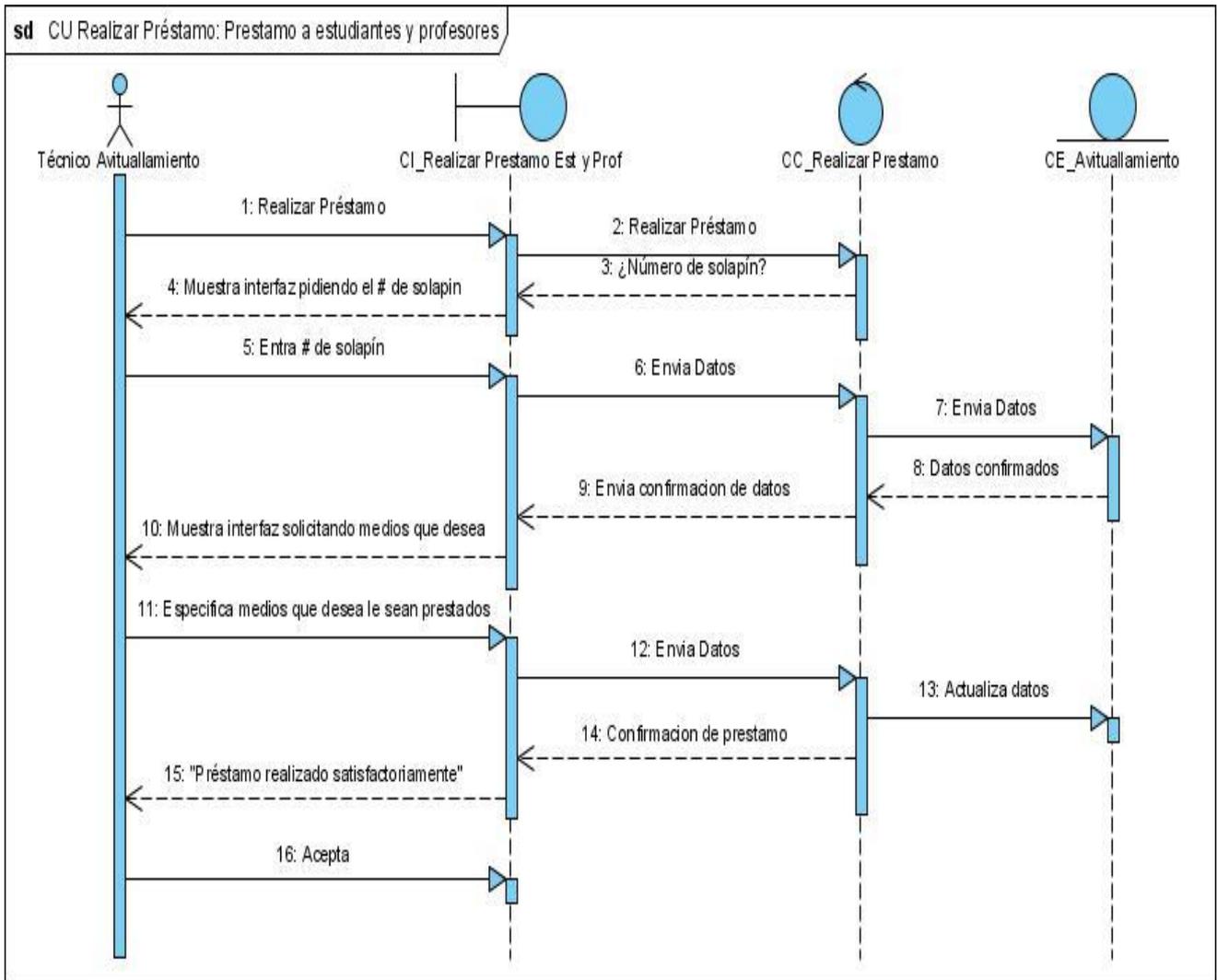


Fig. 3.9 Diagrama de Secuencia: CU Realizar Préstamo: Escenario Prof. y Est.

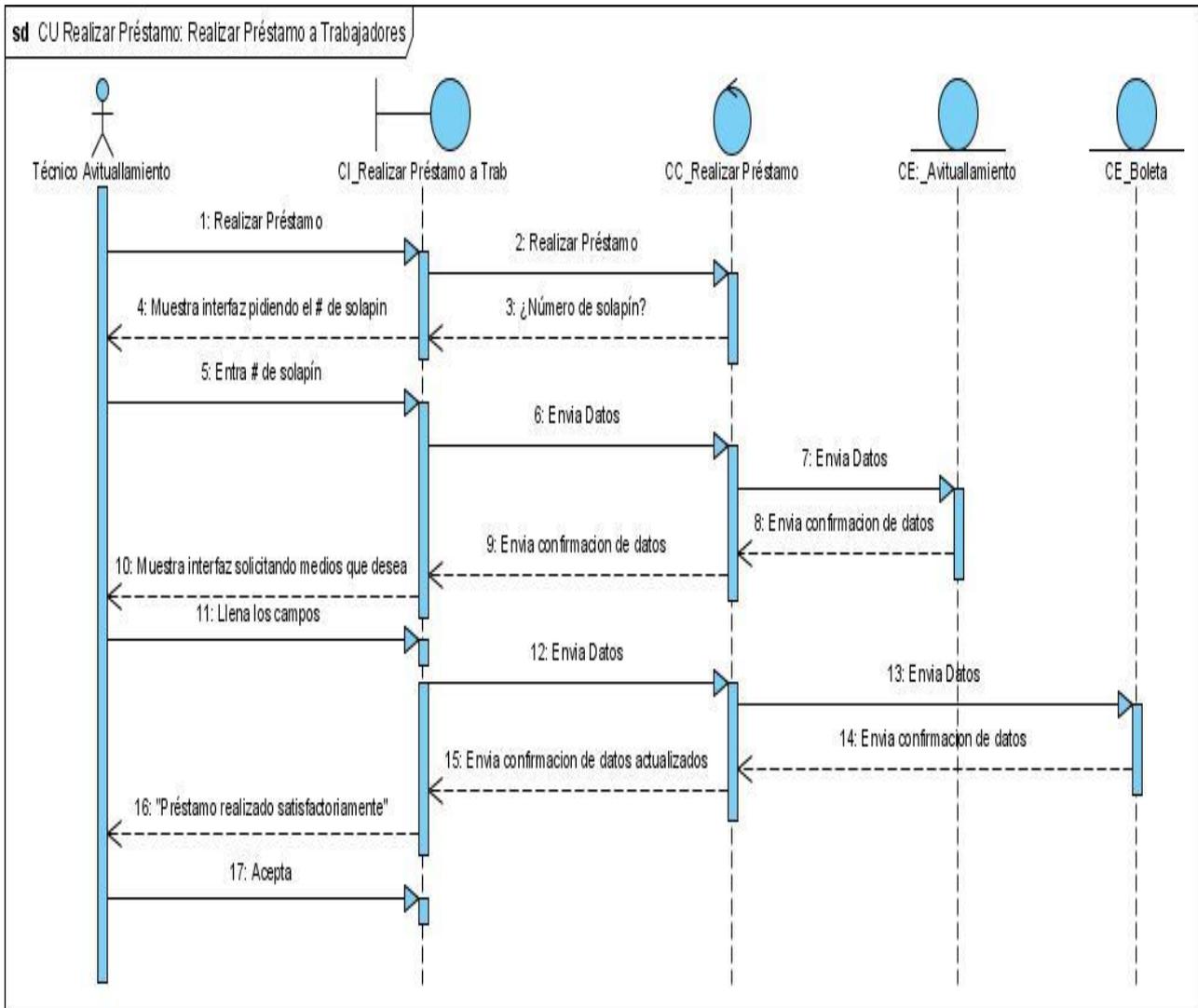


Fig. 3.10 Diagrama de Secuencia: CU Realizar Préstamo: Escenario Trabajadores.

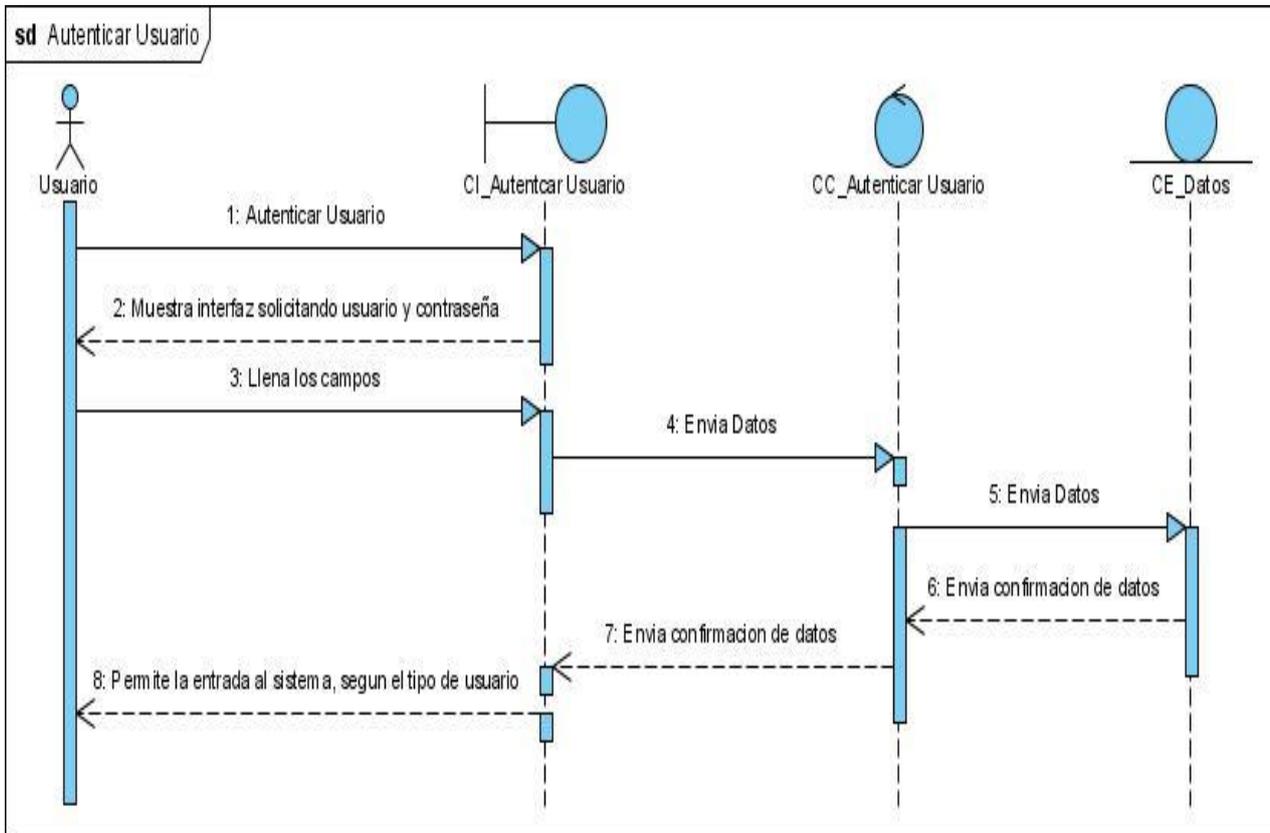


Fig. 3.11 Diagrama de Secuencia: CU Autenticar Usuario.

3.6 CONCLUSIONES

En este capítulo se mostraron los diagramas de clases del análisis para cada uno de los CU identificados, identificando las clases interfaces, controladoras y entidades. De igual manera se presentaron los diagramas de interacción correspondientes, en este caso diagramas de secuencia.

CONCLUSIONES

- ❖ A partir del estudio profundo realizado a los procesos que tienen lugar actualmente en la residencia universitaria, se arribó a la conclusión de que la principal causa de los problemas y situaciones existentes en el proceso de avituallamiento y lavandería en las direcciones de residencia es la carencia de una herramienta automatizada y especializada que garantice la actualización, procesamiento y obtención de información vinculada a este proceso.
- ❖ Se valoraron técnicas y artefactos necesarios para realizar una correcta captura de requisitos.
- ❖ Se realizó una propuesta de secuencia de pasos a seguir en la captura de requisitos a partir de experiencias prácticas y el análisis de las metodologías de desarrollo.
- ❖ Se estudió con profundidad el entorno de negocio del cliente con el objetivo de realizar un correcto modelado del mismo.
- ❖ Se obtuvo el modelo de negocio, requerimientos y análisis del proceso de avituallamiento y lavandería. En este sentido se dejaron listos todos los documentos y artefactos necesarios para comenzar con la etapa de diseño e implementación del sistema.

RECOMENDACIONES

Se recomienda profundizar en el diseño de la secuencia de pasos a seguir en la captura de requisitos con sus técnicas y artefactos y que la misma sea aplicada en tantos procesos de desarrollo de software como sea necesario hasta poder lograr su validación práctica.

Se debe tener en cuenta que el análisis presentado en este trabajo de diploma es el resultado de una iteración y, teniendo en cuenta que la metodología utilizada concibe un proceso de desarrollo de forma iterativa e incremental, los modelos obtenidos pueden estar sujetos a cambios durante todo el ciclo de desarrollo del producto.

Profundizar más en los procesos tratados en este trabajo, buscando siempre realizar una mejor y mayor comprensión de los mismos.

REFERENCIAS BIBLIOGRAFICAS

ANDRÉS, M. M. M. Herramientas CASE. 2001, n° Disponible en:
<http://www3.uji.es/~mmarques/f47/apun/node75.html>.

BARESI L., G. F., PAOLINI P. IEEE Computer Society: Extending UML for Modelling Web Applications. 2001, n°

CALABRIA, L. *Metodología FDD*. Universidad ORT Uruguay Facultad de Ingeniería, 2003.

CRAING, L. *UML y Patrones. Introducción al análisis y diseño orientado a objetos*. Pearson, 1999.

DÍEZ, A. IRqA y el desarrollo de proyectos: Experiencias Prácticas. I Jornadas de Ingeniería de Requisitos Aplicadas. 2001, n°

DURÁN A., B., B., RUIZ, A., TORO M. A Requirements Elicitation Approach Based in Templates and Patterns. Workshop de Engenharia de Requisitos. 1999, n°

ESCALONA, M. J., TORRES, J., MEJÍAS, M. Requirements capture workflow in Global Information Systems. 2002, n°

ESCALONA, T., MEJÍAS, GUTIÉRREZ, VILLADIEGO. The treatment of navigation in web engineering. 2007, n°

FERNÁNDEZ, E. *Asistente para la Gestión de Documentos de Proyectos de Explotación de Datos*. Tesis de Magister en Ingeniería del Software, Escuela de Postgrado. Instituto Tecnológico de Buenos Aires, 2006.

FERREIRA, L. Ingeniería de Requisitos en Aplicaciones para la Web – Un estudio comparativo. 2002, n°

FIGUEROA, P. Análisis de requerimientos. 2000, n°

FRANKLIN ST, F. F., BOSTON. *Free Software Foundation, Inc, La Definición de Software Libre*. 2006, Disponible en: <http://www.gnu.org/philosophy/free-sw.es.htm>.

IBM, O. Developing Object Oriented Software. 1997, n°

INSFRÁN, E., PASTOR, O., WIERINGA, R. Requirements Engineering-Based Conceptual Modeling. . 2002, vol. 7, n°

IRETA, C. U. <http://www3.uji.es/~mmarques/f47/apun/node75.html>. 1999.

- IVAR JACOBSON, G. B., JAMES RUMBAUGH. *El Proceso Unificado de Desarrollo de Software*. 2000.
- JACOBSON, I. *El proceso unificado de desarrollo de software*. 2000.
- . Modeling with use cases-Formalizing use-case modelling, *Journal of Object-Oriented Programming*. 1995, n°
- JACOBSON, I. B., GRADY; RUMBAUGH, JAMES. *El Proceso Unificado de Desarrollo de Software*. 1999. vol. I,
- KOCH, N. *Software Engineering for Adaptative Hypermedia Applications*. 2001, vol. 12, n°
- KOCH, N., ESCALONA, MARÍA JOSÉ *Ingeniería de Requisitos en Aplicaciones Web-Un estudio Comparativo*. 2002, n°
- KRUCHTEN, P. *The Rational Unified Process*. . 1998.
- LIU, L., YU, E. From Requirements to Architectural Design using Goals and Scenarios Proceedings of the 6th Micon Workshop. 2001, n°
- LOWE, D., HALL, W. *Hypermedia and the Web. An Engineering approach*. John Wiley & Son. 1999, n°
- OLSINA, L. *Metodología cualitativa para la evaluación y comparación de la calidad de sitios web*. Facultad de Ciencias Exactas. Universidad de la Pampa, 1998.
- PAN, D., ZHU, D., JOHNSON, K. *Requirements Engineering Techniques*. Internal Report. Department of Computer Science. 2001, n°
- PEÑA, R. *Diseño de programas. Formalismo y abstracción*. 1998.
- PRESSMAN. *Ingeniería de Software: un enfoque práctico*. 2005.
- RAGHAVAN, S., ZELESNIK, FORD, G. *Lectures Notes of Requirements Elicitacitation*. 1994, n°
- RAMSÉS, D. *DCS-UCI Estrategia para introducción del Expediente de Proyecto*. 2003, n°
- Rational Unified Process. 2003, n° Disponible en: <https://pid.dsic.upv.es/C1/Material>.
- RUP. *Rational Unified Process for Systems Engineering*. 2003, n°

SÁNCHEZ. *Programacion Extrema XP*. 2004,

SOMMERVILLE, I. *Software Engineering*. 2004, n°

UML. *Unified Modeling Language*. 2001, Disponible en: www.omg.org.

VILAIN, P., SCHWABE, D., SIECKENIUS, C. A diagrammatic Tool for Representing User Interaction in UML.

Lecture Notes in Computer Science. 2000, n°

WEIDENHAUPT, K., POHL, K., JARKE, M., HAUMER, P. *Scenarios in Systems Development: Current Practice*.

IEEE Software. 1999,

GLOSARIO DE TERMINOS

Auditoria: es el examen crítico y sistemático que realiza una persona o grupo de personas independientes del sistema auditado.

CASE: Computer Aided Software Engineering (Herramientas de ingeniería de software asistida por computadora).

Escenario: Un conjunto de variables que poseen un nivel de valor y un grado de ocurrencia.

IR: Ingeniería de Requisitos.

LINUX: Sistema operativo portable, flexible, potente, con entorno programable, multiusuario y multitarea, muy difundido.

Ontología: Formulación de un exhaustivo y riguroso esquema conceptual dentro de un dominio dado, con la finalidad de facilitar la comunicación y la compartición de la información entre diferentes sistemas.

Prototipo: Se puede referir a cualquier tipo de máquina en pruebas, o un objeto diseñado para una demostración de cualquier tipo.

Requisito: Condición o capacidad, necesidad o deseo que debe cumplir un sistema.

Reviews: Es una evaluación de una publicación, como una película, video juego, composición musical, un libro, o una pieza de hardware como un coche, electrodoméstico, o del ordenador.

Sistema: Conjunto de entes independientes entre sí mismos que se encuentran en interrelación con ellos mismos y con el ambiente que los rodea.

SO: Sistema Operativo.

UML: Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modelling Language) es el lenguaje de modelado de sistemas de software más conocido en la actualidad.