

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

SISTEMA DE GESTIÓN DE INFORMACIÓN

T E S I S

QUE PARA OBTENER EL TITULO DE

INGENIERO EN COMPUTACIÓN

P R E S E N T A N :

CHÁVEZ SEGURA ÁNGEL RICARDO
GARCÉS CORELLA EDGAR
LANZA HERNÁNDEZ SONIA
VERTIZ GUERRERO JOSÉ ANTONIO

DIRECTOR DE TESIS
M.I. JUAN CARLOS ROA BEIZA



CIUDAD UNIVERSITARIA, MÉXICO, D. F. 2005

Edgar Garcés Corella

A mi esposa Dolores que ha sido y es mi motivación, mi persistencia y mi apoyo; a mi hijo Daniel que es mi compromiso a ser mejor y seguir adelante.

A mi madre Marcelina que me enseñó a perseverar aún ante la adversidad, a mi padre Rogelio que despertó y motivó mi interés en la ingeniería y a mis hermanos que pese a todo seguimos unidos.

A mis abuelos Trinidad y Guadalupe que me enseñaron la honestidad y el coraje de seguir un sueño.

A todos mis profesores y en general a toda aquella persona que me ha enseñado la importancia de trascender.

A mi Universidad y en particular a mi Facultad por forjar mi actitud y mi compromiso como profesionista.

Ángel Ricardo Chávez Segura

El desarrollo de esta tesis se la dedico a mis padres Jose Carlos Chávez Mata y María De los Ángeles Segura Barrios, a mis hermanos Carlos Adrián, Mayra Berenice y Alma Jessica ya que sin el apoyo de ellos no lo hubiera podido alcanzar esta meta que me fije y que hasta el momento estoy realizando.

También quiero agradecer a la luz de mi vida que es mi hijo Ricardo Chávez Garcia y a su mamá Zoila Garcia Contreras que aun que no estén conmigo pues sin ellos no hubiera sido posible todo este gran logro.

A mis amigos Jazmín Vázquez León y Víctor Cueto de la Facultad de Ingeniería que me apoyaron y cursamos juntos la carrera alentándome a terminar y titularme ya que ellos ya lo hicieron.

A todos mis profesores y asesores que a lo largo de mi vida me han ayudado y preparado, para llegar hasta donde ahora estoy.

A mis compañeros del trabajo del área de Recursos Humanos en Las Oficinas Centrales de PROFECO, en especial a mi gran amiga Guadalupe Tinajero Garcia y su hijo Johan, que también contribuyeron en que yo terminara este proyecto que es muy especial en mi vida y no solo a ellos si no a todos los que creyeron en mi pues es una forma de demostrarles que no los defraude y que seguí adelante gracias a su apoyo, esfuerzos y que no dejaron que renunciara a ser todo un Ingeniero.

También quiero agradecer especialmente a la Universidad Nacional Autónoma de México y a la Facultad de Ingeniería institución que me formó como profesionista.

Sonia Lanza Hernández

A ti papá… Que siempre vives en mi corazón y en mis pensamientos, gracias, porque desde ese hermoso cielo azul, estas siempre cerca de mi...

Algún día seré madre y sabré que por un hijo uno da la vida… Gracias MAMÁ por haber dado tu vida, para facilitarme la mía.

A la Universidad Nacional Autónoma de México por haber sido mi casa durante muchos años dándome la oportunidad de conocer gente muy valiosa, por haberme dado una formación crítica, conciente y con un firme carácter, obteniendo de ella otras tantas satisfacciones y como todo en la vida también pasando momentos difíciles con ella, ahora tengo un enorme orgullo de sentir que mi corazón es azul y mi piel dorada y por siempre puedo decir que: Por mi raza hablará el espíritu.

Mi último agradecimiento es para la vida… Por hacerme coincidir en espacio y tiempo con un Ángel de quien ahora soy una extensión. Gracias por estar siempre en los momentos y en las decisiones más importantes de mi vida, gracias por la fe que siempre me has tenido y por haberme impulsado para estar ahora escribiendo esta dedicatoria… Gracias MAESTRO, Gracias AMIGO, Gracias MARTÍN.

José Antonio Vertiz Guerrero

A mis padres Maria Ascensión y José Antonio al igual a mis hermanos Guadalupe, Sandra, Fernando y Claudia por todo lo que me han apoyado y la dedicación que me dieron para obtener una de las metas más anheladas, siendo esto para mí, la mejor de las herencias.

A mi esposa Emma por creer en mí y apoyarme en todo momento a la realización de este proyecto. A mis adorables hijas Grecia e Ivanna que fueron mi mayor motivación.

A mi abuelo Fernando; de quien en vida me dio todo su cariño.

A todos mis profesores y asesores que a lo largo de mi vida me han ayudado y preparado, para llegar hasta donde ahora estoy.

A mis compañeros de Petróleos Mexicanos quienes siempre me impulsaron a lograr este objetivo y me brindaron los medios para la elaboración de este trabajo, con un especial afecto al Ing. Jorge Molina García.

Especialmente a la Universidad Nacional Autónoma de México y a la Facultad de Ingeniería institución que me formó como profesionista.

Aquellos que de alguna manera estuvieron presentes cuando más los necesite y muy en especial a mi amigo el Ing. Joel Villavicencio Cisneros.

TODOS

Al personal del Programa de Apoyo a la Titulación del Palacio de Minería que nos ayudo a realizar y terminar con este proyecto.

A nuestro Director de Tesis el M.I. Juan Carlos Roa Beiza. Gracias por todo su apoyo.

ÍNDICE TEMÁTICO

INTRODUCCIÓN..... v

CAPÍTULO 1. ENTORNO DE LA EMPRESA..... 1

1.1 Conceptos básicos de la atención al cliente..... 2

1.1.1 ¿Qué es el servicio? 2

1.1.2 Conceptos básicos para inventarios..... 10

1.1.3 Misión y visión de este tipo de empresas..... 15

1.1.4 Aspectos logísticos para el resguardo de información..... 21

1.1.5 Organigrama de empresa tipo..... 32

CAPÍTULO 2. MARCO TEÓRICO..... 39

2.1 Conceptos fundamentales de la programación orientada a objetos 40

2.2 Metodología de diseño orientado a objetos..... 47

2.3 Sistemas Operativos..... 55

2.4 Características, ventajas y desventajas de la plataforma Java..... 63

2.5 Bases de datos relacionales (ventajas y desventajas)..... 79

2.6 Comunicaciones..... 85

CAPÍTULO 3. PLANTEAMIENTO DEL PROBLEMA Y PROPUESTA DE SOLUCIÓN..... 105

3.1 Problemática actual..... 106

3.2 Requerimientos generales y particulares..... 113

3.3 Búsqueda y análisis de la información..... 117

3.4 Áreas involucradas en el proceso e identificación del sistema..... 122

3.5 Opciones de solución y elección de la óptima..... 128

CAPÍTULO 4. DESARROLLO E IMPLANTACIÓN DEL SISTEMA 139

4.1 Aplicación de la metodología elegida..... 140

4.1.1 Diagrama de contexto Diagrama de casos de uso..... 140

4.1.2 Diagrama de flujo y procesos Diagrama de actividades..... 164

4.1.3 Diccionario de datos..... 176

4.1.4 Diagrama de entidad Relación Diagrama de clases..... 182

4.1.5 Normalización..... 192

4.2 Diseño y construcción del Back- End..... 194
4.3 Diseño y construcción de la Interfaz del Usuario (Front End)..... 212
4.4 Pruebas e integración del sistema..... 223
4.5 Generación de reportes para la toma de decisiones..... 244
4.6 Factibilidad técnica y operativa..... 248
CONCLUSIONES..... 254
MANUAL TÉCNICO..... 257
MANUAL DEL USUARIO..... 275
BIBLIOGRAFÍA..... 284

INTRODUCCIÓN

En la presente tesis se pretende dar solución al diseño y construcción de un sistema de gestión de información para el archivo de empresas inmobiliarias, este ayudará a optimizar el proceso de solicitud, control, envío y almacenamiento de la información generada, y su posterior localización física para consulta del contenido de la misma. Además poseerá un control para su eliminación cuando deje de ser relevante para la empresa. La motivación del desarrollo de dicho sistema surge, por la necesidad que tiene la empresa en agilizar el almacenamiento y la rápida consulta de la información contenida en el archivo, ya que se lleva actualmente el registro de información en bitácoras, las cuales tienen que ser capturadas manualmente o en hojas de cálculo por el responsable de esta función, el cual no lleva un control eficaz para después recuperar dicha información con la premura que se requiere en cada caso.

Este sistema deberá permitir el almacenamiento, registro y ubicación en forma organizada, de tal forma que cuando se requiera de esta pueda ser fácilmente localizada y consultada. Por lo cual el sistema propuesto debe cubrir lo siguiente: recepción automatizada de la solicitud de envío al archivo general, seguimiento del envío, registro de la ubicación física del material para la consulta posterior, además de proveer reportes de la ubicación, contenido y el control de destrucción en las fechas establecidas por el usuario.

La inmobiliaria requiere de un sistema que permita controlar de manera centralizada, la solicitud de envío de paquetes, autorización y seguimiento de su ubicación física así como su estatus.

En el primer capítulo se analizaran los aspectos generales relevantes para el buen funcionamiento y adecuada organización de una empresa inmobiliaria. Para el segundo capítulo dentro del marco teórico se detallaran las herramientas utilizadas en el desarrollo del sistema. En el tercer capítulo se plantea el problema y la propuesta de solución. En el cuarto capítulo se contempla el diseño la construcción e integración del sistema. Así mismo deberá poseer una base de datos del tipo relacional que contenga

toda la información antes mencionada, y que posea una interfaz de usuario que cualquier usuario que haya manejado un ambiente gráfico pueda operar sin ningún problema. Para ello deberá contar con todos los catálogos para que el operador elija de estos.

Esta se basará en una investigación de toda la situación en cada una de las áreas correspondientes, recopilando toda la información pertinente para cada caso, esta se analizará y clasificará de acuerdo al área para después dar una solución en forma precisa a la problemática que presentan. Se utilizará una metodología de desarrollo iterativo e incremental basado en la tecnología de orientación a objetos.

CONCEPTOS BÁSICOS DE LA ATENCIÓN AL CLIENTE

1.1 ¿Que es el servicio?

Servicio es:

- Anticipar y superar las expectativas del cliente
- Evitarle problemas y, si se presentan, solucionárselos con rapidez
- El conjunto de acciones que hacen más valioso un producto, cuando esto es reconocido por los clientes

Calidad del Servicio

Es muy importante que cada vez que se presta un servicio, se reconozca a la empresa por su excelencia, por lo que es, lo que hace y cómo lo hace.

¿Por qué es importante brindar al cliente un servicio de calidad?

- Porque el cliente es la razón de ser de la empresa
- Porque es la base para permanecer y seguir creciendo.
- Para que la empresa tenga éxito debe brindar un excelente servicio y ser reconocida por ello
- Porque es rentable
- Porque los errores cuestan más que dinero, cuestan reputación
- Un cliente descontento contará su mala experiencia a muchas personas, mientras que un cliente feliz sólo lo transmitirá a unas pocas

CONCEPTOS GENERALES SOBRE EL SERVICIO AL CLIENTE

Al abordar el servicio al cliente hay que conceptualizar adecuadamente tres aspectos interrelacionados del mismo:

- **Demanda de servicio.** Son las características deseadas por el cliente para el servicio que demanda

- **Meta de servicio.** Son los valores y características relevantes fijadas como objetivo para el conjunto de parámetros que caracterizan el servicio que el proveedor oferta a sus clientes.
- **Nivel de servicio.** Grado en que se cumple la meta de servicio

Lo anterior lleva a replantear el esquema tradicional de distribuir lo que se produce al esquema de distribuir lo que el cliente necesita. Para ello la empresa debe definir una filosofía de servicio, expresada en términos de: actitud, organización y responsabilidad que abra paso al establecimiento de una estrategia de servicio.

Los elementos a considerar para establecer una estrategia de servicio:

- **El cliente**. Hay que identificar con exactitud quién es el cliente y las necesidades y deseos que éste realmente tiene
- **La competencia**. Hay que identificar las fortalezas y debilidades de los competidores y con ello establecer un servicio al cliente mejor que la competencia, o sea, que proporcione ventaja competitiva
- **Los patrones**, **costumbre** y **posibilidades** de los clientes

En cada caso debe valorarse a partir de la apreciación del cliente cuáles son los elementos de mayor peso y alrededor de ellos diseñar el mejoramiento del nivel de servicio.

Diseño del servicio al cliente

El diseño del servicio al cliente lleva implícito el análisis de la organización que brindará el servicio y el de los clientes que lo recibirán. Para un adecuado diseño del servicio al cliente debe seguirse el procedimiento que aparece en la figura 1.1.1

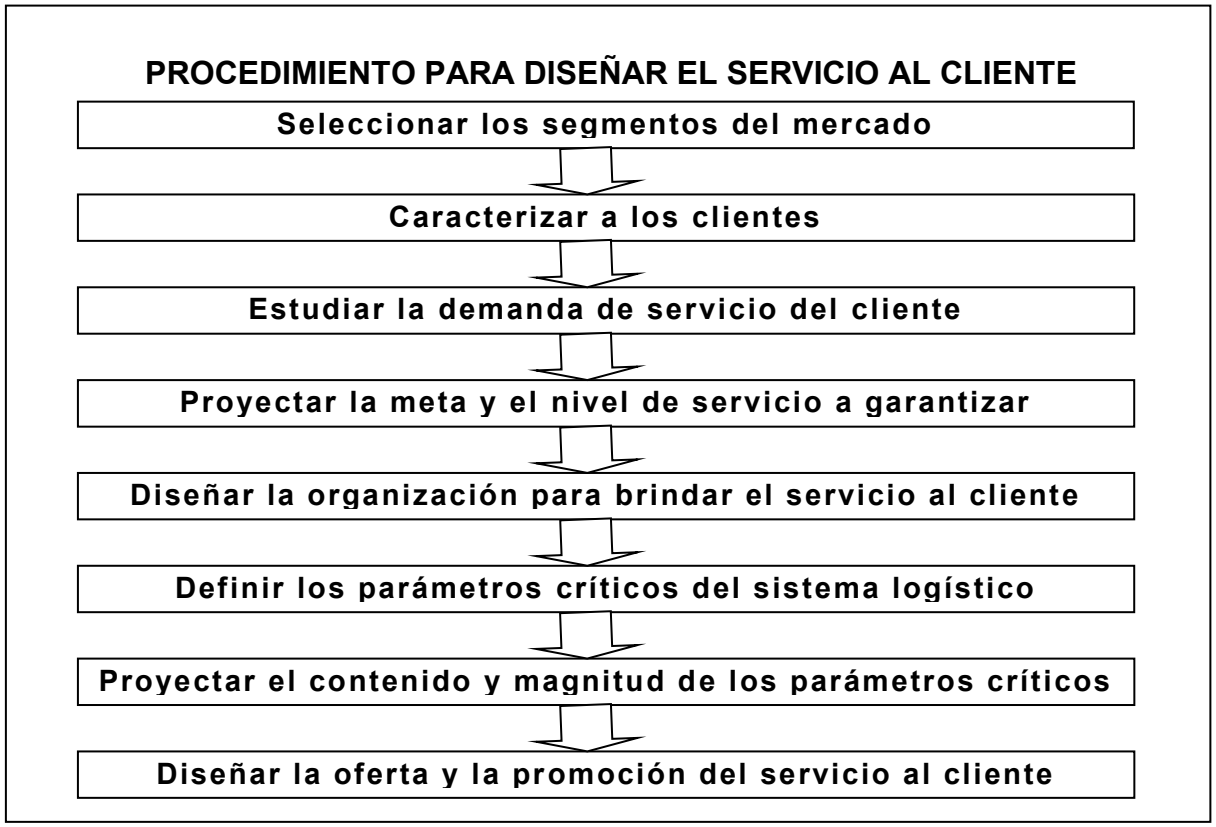


FIGURA 1.1.1. PROCEDIMIENTO DE DISEÑO DEL SERVICIO AL CLIENTE

- **Seleccionar los segmentos de mercado que son objetivos del sistema.** La segmentación se realiza mediante la selección de una o varias variables que deben ser: mensurable, accesible, sustancial, maniobrable, confiable, válida, estable, homogénea y generalizadora. En consecuencia, la selección está en función de las respuestas a la pregunta -¿Qué desea cada segmento de clientes? Usualmente diferentes clientes desean diferentes cosas. Cada segmento puede elaborarse con respuestas a preguntas tales como:

  - ¿Qué es importante para los clientes?
  - ¿Qué les gusta acerca del producto/servicio?
  - ¿Qué no les agrada?
  - ¿Cómo podría este cliente estar satisfecho?
  - ¿Cómo define este cliente la calidad?
  - ¿Quién es la competencia en este servicio?

- ¿Cuál es el tamaño de este segmento?
- ¿Qué porcentaje del negocio total está representado por ese segmento?

- **Caracterizar a los clientes.** Para definir de forma objetiva el nivel de servicio a ofrecerle a los clientes, es necesario clasificar a estos en grupos de acuerdo con sus características, deseos y posibilidades, de forma que se pueda garantizar el nivel de servicio que cada cliente demanda
- **Estudiar la demanda de servicio al cliente.** Para estudiar y determinar con exactitud cuál es la demanda de servicio al cliente que tiene el sistema logístico se debe:

  - seleccionar una muestra de clientes de cada segmento de mercado a abordar
  - recoger criterios sobre las características del servicio demandado y el grado de importancia de cada uno
  - recoger criterios sobre la valoración de cada característica del servicio que se está ofreciendo
  - determinar la demanda del servicio
  - valorar íntegramente la distancia entre el servicio esperado y el proporcionado Campo de Fuerza

- **Proyectar la meta y el nivel de servicio a garantizar.** La proyección del nivel de servicio puede hacerse a través de dos alternativas. La primera es determinar el nivel económico óptimo de servicio que se debe ofrecer, esta alternativa tiene el riesgo de no tener en cuenta la situación de la competencia. La segunda alternativa parte de fijar el nivel de servicio de la competencia y buscar el diseño de más bajo costo para ese nivel, de esta forma se tienen en cuenta a la competencia pero se corre el riesgo de no trabajar con el nivel de servicio óptimo desde el punto de vista económico. La meta y nivel de servicio se establece

individualmente para cada grupo de cliente en función de las características del servicio propuesto. El cumplimiento de las características del servicio se determina mediante los medidores del nivel de servicio por segmento de mercado, permitiendo evaluar la calidad ofertada del servicio prestado

- **Diseñar la organización para brindar el servicio al cliente.** El contenido de la organización para brindar el servicio al cliente abarca un amplio conjunto de aspectos que pueden estructurarse según la interrogante que responde (ver tabla 1).

INTERROGANTE	DEFINICIONES A LOGRAR
¿Estamos preparados?	• Características de la cultura de la organización para enfrentar un servicio al cliente superior • Necesidades de capacitación
¿Para quién?	• Caracterización de los grupos de clientes • Características de la atención a dar a cada grupo • Sistema de información y orientación al cliente
¿Qué?	• Contenido del servicio que se brinda • Características
¿Cómo?	• Procedimientos para que el cliente solicite el servicio • Procedimientos (tecnología) para brindar el servicio

TABLA 1. CONTENIDO DE LA ORGANIZACIÓN PARA BRINDAR EL SERVICIO AL CLIENTE

continúa página siguiente

¿Quién?	• Personal que se encarga de brindar el servicio: cantidad, funciones, características, estética, ética y calificación
¿Cuánto?	• Estructura organizativa • Magnitud de los parámetros relevantes del servicio
¿Cuando?	• Duración de los ciclos de respuesta y de ejecución del servicio
¿Con qué?	• Relación de medios a utilizar (equipos, utensilios, mobiliario, dispositivos, instalaciones, medios técnicos)
¿Dónde?	• Lugar para brindar el servicio. Su localización y disposición en planta
¿Por qué?	• Definir sólo tareas y acciones que agreguen valor al servicio al cliente • Objetivos y metas del servicio al cliente

TABLA 1. CONTENIDO DE LA ORGANIZACIÓN PARA BRINDAR EL SERVICIO AL CLIENTE

Una vez proyectado el contenido de la organización resulta conveniente elaborar el manual de servicio al cliente, el que resulta de gran utilidad como guía para los supervisores, pues sirve de herramienta para la evaluación sistemática y para capacitar al personal del sistema.

- **Definir los parámetros críticos del sistema logístico.** El lugar del cliente es clave en el diseño y funcionamiento del sistema logístico lo que pone de manifiesto la necesidad de definir los parámetros críticos del

sistema, que son los que están asociados a los problemas. Estos últimos son aquellos problemas cuya solución contribuye a alcanzar los objetivos fijados por el sistema logístico. Los parámetros críticos se convierten en las válvulas del sistema logístico, que permiten el cumplimiento del nivel del servicio al cliente

- **Proyectar el contenido y magnitud de los parámetros críticos del sistema logístico.** Una vez definidos los parámetros críticos del sistema, resulta imprescindible, para garantizar el adecuado funcionamiento del sistema logístico, determinar para cada uno de ellos cuál debe ser su contenido y magnitud, de forma que se garantice la obtención de los objetivos propuestos con la mayor racionalidad posible
- **Diseñar la promoción del servicio al cliente.** La promoción del servicio al cliente debe ser transparente para poder atraer a los clientes potenciales

Principios para el diseño del servicio al cliente:

- **Diferenciación del servicio para los distintos segmentos de mercado**. Para cada segmento de mercado debe diseñarse el nivel de servicio más adecuado, no siempre un único diseño de servicio al cliente es capaz de satisfacer todos los segmentos de mercado que debe atender la empresa
- **Competitividad**. El diseño del servicio que se realice debe además de satisfacer plenamente las necesidades de los clientes, garantizar la competitividad de la empresa de forma tal que pueda permanecer en el mercado
- **Racionalidad**. Lograr satisfacer las necesidades de los clientes y mantenerse en el mercado debe hacerse sobre la base de una adecuada racionalidad en la utilización de los recursos y procesos
- **Satisfacción del cliente**. Toda acción en la prestación del servicio debe estar dirigida a lograr satisfacción en el cliente. Esta satisfacción debe garantizarse en cantidad, calidad, tiempo y precio

- **Transparencia de la meta de servicio** tanto para el cliente como para quien brinda y apoya el servicio. El cliente tiene derecho a conocer qué puede esperar del servicio brindado por la empresa y quien brinda el servicio debe tener plena conciencia de cual es la meta de servicio a que puede comprometerse para no crear falsas expectativas en el cliente
- **Personalización**. El servicio se brinda no a un cliente indistinto sino a una persona o grupo específico y como tal debe tratarse

1.2 CONCEPTOS BÁSICOS PARA INVENTARIOS

Archivo muerto

Conjunto de documentos, sea cuales sea su fecha, su forma y su soporte material, producidos o recibidos por toda persona física o moral, y generados por algún servicio u organismo público o privado, en el ejercicio de su actividad, y que son conservados por sus creadores o por sus sucesores para sus propias necesidades o bien cedidos a la institución de archivos competente en razón de su valor. Institución responsable de la acogida, tratamiento, inventario, conservación y servicio de los documentos. Edificio o parte del edificio donde los documentos son conservados y servidos. En un fondo (Los fondos se pueden definir como soportes que contienen información de interés para ser guardada en los archivos) de archivo hay un conjunto de documentos procedentes de la actividad de una persona física o moral o de un organismo, cuya reunión es fruto de un proceso natural en el que el productor genera y conserva esos fondos de forma progresiva y constante. Los documentos tienen un valor probatorio de carácter jurídico-administrativo en un periodo determinado de su vida, durante el cual no son libremente accesibles.

Funciones del archivo:

- **Organización y puesta en servicio** de la documentación administrativa, durante el periodo de máxima utilidad para la gestión administrativa de las oficinas y para la toma de decisiones
- **Asegurar la transferencia** periódica al archivo de los documentos que ya no son de uso corriente por parte de las oficinas
- **Aplicar los principios y técnicas** modernas de valoración para, transcurrido un tiempo, seleccionar los documentos que por su valor van a ser conservados indefinidamente y destruir el resto
- **Clasificar los fondos y mantener** ordenada la documentación en diferentes etapas, de acuerdo con los principios de la archivística

- **Describir la documentación** para hacer fácilmente accesible la información, mediante los distintos instrumentos de descripción documental y valiéndose de las ventajas ofrecidas por las nuevas tecnologías
- **Instalar adecuadamente** los documentos mediante locales y equipamiento que garantice la conservación integra en el tiempo
- **Asegurar** que estas y cuantas funciones pueda desarrollar el archivo, queden firmemente establecidas y garantizadas mediante un reglamento de servicio

Todo ello orientado a que el archivo sea un recurso básico de información, al servicio de:

- La administración que emite los documentos
- Los ciudadanos
- La investigación y la difusión cultural

Características arquitectónicas:

- La estructura debe ser de hormigón armado, los elementos de madera no sustituibles estarán consolidados y tratados con productos ignífugos e insecticidas
- Los depósitos deben estar aislados del resto de las dependencias y dotados con medidas de protección contra riesgos por deterioro procedentes del exterior
- Las estanterías deben ser móviles y metálicas
- Las paredes deberán estar protegidas con pintura ignífuga
- En ellos no debe haber madera
- Las puertas deberán ser metálicas contrafuego
- Proveerles de una alarma que los bloquee automáticamente
- Los cables deberán tener una protección especial

Políticas

- **Todos los documentos** cuya vigencia ha concluido, serán sujetos a procesos de valoración, selección, transferencia secundaria o baja
- **Corresponde a los responsables** del área coordinadora de archivos, al archivo de concentración o al archivo histórico de las dependencias, según el caso:

    - Presentar ante el archivo las solicitudes de autorización de baja documental de aquellos archivos que ya no posean valores primarios (administrativo, legal y fiscal o contable), conforme a las normas establecidas
    - Efectuar el traslado de los archivos que serán transferidos, ordenados de acuerdo con los inventarios que los describen
    - Gestionar la eliminación de documentos autorizados en el acta de baja que autorice el Archivo
    - Integrar y conservar los expedientes de las bajas y transferencia de documentos que se realicen por un mínimo de cinco años

- **Corresponde al archivo**

    - Recibir, tramitar y dar seguimiento a las solicitudes y acciones de archivos generados por las dependencias
    - Realizar la valoración secundaria de los archivos promovidos para baja o transferencia
    - Recibir y custodiar aquellos archivos dictaminados para su conservación permanente
    - Expedir las actas de baja o transferencia de archivos

- **El proceso de valoración** secundaria por parte del archivo incluirá:

    - Las consultas que se consideren necesarias para incorporar el punto de vista de los productores y de los usuarios, para garantizar un análisis completo y riguroso de la documentación y de su potencial informativo
    - Realizar una visita de inspección para reconocer el estado del orden y conservación, así como para evaluar sus características documentales e informativas

- **Todo proceso de valoración** y selección deberá quedar debidamente documentado
- **Las transferencias documentales** se efectuarán cumpliendo tanto los requerimientos institucionales de la dependencia o entidad que transfiere el material, como las disposiciones reglamentarias a las que esta sujeta la documentación depositada en el archivo

Normas generales

Sin excepción, las solicitudes para transferencia del archivo, deberán presentarse con la siguiente documentación:

- Oficio de solicitud
- Inventario de baja o transferencia secundaria debidamente firmada y autorizada por quien corresponda

Inventario

El inventario deberá contener como elementos obligatorios:

- Logotipo de la dependencia o entidad
- Nombre de la dependencia o entidad (subfondo)
- Nombre de la unidad administrativa (sección)

- Nombre del área de procedencia del archivo (dirección de área, subdirección, departamento o equivalentes)
- Número secuencial (que indicará la secuencia de los elementos que conforman el inventario)
- Nombre del expediente o breve descripción de contenido
- Código de clasificación que identifique la serie y expediente
- Valor primario de los archivos (administrativo, legal, contable, fiscal, uno o varios)
- Periodo de trámite (años extremos que abarca la documentación)
- Vigencia documental (conforme al catálogo de disposición documental de la dependencia o conforme a disposiciones jurídicas preestablecidas). De ser el caso la vigencia documental incluirá el número de años a que corresponda en suma cuando se trate de archivos que cuenten con información reservada, para lo cual se hará la indicación que corresponda
- Hoja de cierre: (Leyenda)

    - El presente inventario consta de _____expedientes de los años de ___ (periodo), contenidos en ____ (expedientes o cajas), con un peso aproximado de _____Kg
    - Nombre y firma de los que formularon el inventario
    - Nombre y firma del visto bueno del titular del área productora de la documentación

Elaboración del inventario de archivos para baja o transferencia

Cuando los archivos estén integrados por documentación cuyo contenido es semejante, se aceptará que sean descritos por caja para su baja o transferencia. La descripción deberá contener la mayor cantidad de información posible acerca de los expedientes contenidos en la caja correspondiente.

1.3 MISIÓN Y VISIÓN

La planeación estratégica es un proceso estructurado para la búsqueda de respuesta a preguntas vitales para el diseño, organización y dirección de las empresas. El propósito de esta es tener un rumbo claro de hacia dónde dirigir nuestras acciones. Dentro del proceso de planeación estratégica, dos descripciones trascendentales son la definición de misión y la definición de visión de la empresa.

Misión

La misión, es una definición concreta de a qué actividad se dedica la empresa, la razón de ser de esta, cómo la lleva a cabo y qué le diferencia de las demás. Debe especificar el rol funcional que esta va a desempeñar en su entorno e indicar con claridad el alcance y dirección de sus actividades. Aunque esto puede parecer un paso simple es sorprendente la cantidad de empresas que no saben a qué se dedican o que definen su actividad de una manera muy vaga, con lo que los esfuerzos para conseguir los objetivos también son difusos y se diluyen persiguiendo cada vez una cosa o manteniéndose en una indefinición. La misión se caracteriza por:

- Ser un referente teórico que guía el quehacer de la empresa
- Ser una formulación explícita de propósito duradero
- Indicar la concepción que tiene la empresa hacia los demás (¿Qué somos?, ¿Qué hacemos?)

Debe incentivar en todos los miembros de la empresa el sentido de pertenencia. Así, un paso de suma importancia es redactar una misión, una definición que satisfaga las siguientes características:

- Debe contener una formulación de los objetivos de la empresa, en forma tal que los progresos en ese sentido puedan ser medidos. Significa que la

adopción de una misión debe cambiar el comportamiento de la empresa que la enuncia

- Debe diferenciarla de las otras empresas. Requiere establecer en forma individual, o única, lo que la empresa quiere no lo que la mayoría de las empresas quieren
- Debe definir el negocio en el que la empresa quiere estar, y no necesariamente aquel en el cual ya está. Significa encontrar un concepto unificador que amplíe la visión que la empresa tiene de sí misma y la enfoque correctamente
- Debe ser significativo para todos los participantes de la empresa. Lleva a establecer cómo se propone la empresa servir a sus clientes y proveedores, público en general, accionistas y empleados
- Debe ser estimulante e inspiradora. Debe motivar a todos aquellos cuya participación se requiera
- Debe indicar cuáles son los valores y principios que impulsan a nuestra empresa. Tiene que haber una serie de valores que identifiquen nuestra empresa (calidad, rapidez, servicio, etc.) y debemos señalarlos brevemente

La mejor prueba de que el enunciado de misión tiene valor, es que la empresa cambie a consecuencia de lo que en ella se establece. Si nada sucede al día siguiente ni en los siguientes años, significa que la misión de la empresa es otra.

Realizando este paso no sólo habremos definido nuestra actividad sino que lo habremos hecho con un enfoque adecuado, no restringiéndonos al mero ejercicio descriptivo sino pensando y poniendo de relieve, desde el principio, las ventajas, las diferencias y los valores que serán la directriz de la empresa.

Visión

La visión representa el futuro de la empresa y se realiza formulando una imagen ideal y poniéndola por escrito, a fin de crear el sueño (compartido por todos los que tomen parte en la iniciativa) de lo que debe ser en el futuro la empresa. La

visión, es pues, la narración de una idealización empresarial. Visualizar el futuro implica un permanente examen de la organización frente a sus clientes, su competencia, su propia cultura, y por sobre todo discernir entre lo que ella es hoy, y aquello que desea ser en el futuro, todo esto frente a sus capacidades y oportunidades. Para ello, a partir de esta premisa se debe escribir una visión que, de ser posible, contenga los siguientes puntos:

- **Confirmación de la declaración de la visión**. Quien desarrolla la visión debe decidir cual será su declaración de la misma, ejemplo: "seremos líderes en el diseño y fabricación de equipos para la industria alimentaría"
- **Comprensión del impacto ambiental**. Detectar las influencias del ambiente externo, importantes para el futuro de la empresa. Revisando los hechos importantes del ambiente se desarrolla la comprensión sobre como accionar y descubrir las oportunidades de influir en él
- **Definición de los clientes**. Definir los clientes que la empresa espera tener.
- **Selección de los grupos de productos y/o servicios de la empresa**. Integra aquellos que se desean generar. En este tema es necesario tener en cuenta algunas alternativas:

    - productos impulsados por el cliente
    - productos impulsados por la competencia
    - productos impulsados por los proveedores
    - productos sustitutos que reemplazan a los existentes
    - productos impulsados por la tecnología
    - productos basados en alguna fortaleza de la empresa

- **Estimación del potencial de la empresa** para facilitar cualquier decisión futura de inversión estratégica. Es necesario definir el potencial de crecimiento
- **Identificación de valores agregados**. Son un conjunto particular de destrezas, posicionamiento, experiencia o recursos para actuar con éxito.

Pueden ser de dos tipos: principales o secundarios, los primeros los suministra la empresa, los secundarios se obtienen a través de proveedores de valor agregado

- **Selección de los valores agregados principales y secundarios**. Si la empresa no posee los valores agregados para los productos que desea ofrecer, puede acudir a: subcontratación, creación de empresas temporales conjuntas, adquisición de otras empresas, inversiones para crear valor
- **Determinación de proveedores potenciales y las fuentes**. Conociendo los valores agregados que debemos buscar, habrá que debatir ampliamente para que la visión lleve su respaldo y compromiso.
- **Cuantificación de los criterios de éxito de los productos**. Aquí se crean las "metas alcanzables" e "incentivos apropiados" para poner en marcha la energía de la organización. La definición de estos criterios se realiza en términos de: funcionalidad del producto, apariencia, precio, margen, calidad etc.

La utilidad de la visión radica en:

- **Ser fuente de inspiración para el proyecto**, representa la esencia que guía la empresa, de ella se extraen fuerzas en los momentos difíciles y ayuda a trabajar por un motivo y en la misma dirección a todos los que se comprometen en el negocio
- **La visión** comienza a enfocar en una dirección las fuerzas creativas y sirve de guía en momentos de duda o dificultad

Por lo tanto, dentro del proceso de planeación estratégica, dos descripciones trascendentales son la definición de misión y la definición de visión de la empresa. Así, para esta inmobiliaria tenemos las siguientes definiciones:

Visión

Lo que la empresa quiere llegar a ser:

Ser una empresa competitiva a nivel nacional, que lleve a cabo proyectos integrales, enfocándonos a lograr la amplia satisfacción de nuestros clientes, proveedores, accionistas y empleados.

Misión

¿Qué se esta haciendo para lograr los objetivos fijados?

Desarrollar proyectos rentables dentro del territorio nacional, basados en la capacidad técnica desarrollada dentro de la empresa y de su personal altamente competente, que trabaje con ética profesional, calidad y seguridad.

Enfocándonos en proporcionar un valor agregado superior a nuestros clientes a través de incrementar nuestra competitividad.

- Entregar un servicio vital para la calidad de las personas, el desarrollo de las empresas y la comunidad
- Fomentar en todos con quienes trabajamos una cultura orientada al cliente, que sea ágil, con sentido de pertenencia y compromiso
- Superar las expectativas de nuestros clientes, con propuestas de valor que sean innovadoras y que respondan oportunamente a sus necesidades
- Ser reconocidos como miembros de una empresa que sabe escuchar y que está siempre dispuesta a aprender
- Ser una empresa preparada, con una reconocida excelencia operacional, confiabilidad y credibilidad
- Obtener una retribución acorde con la calidad de los productos y servicios entregados, creando valor en forma sostenida y sustentable

Los valores éticos que promueve la empresa son:

- Orientación al cliente
- Orientación a la rentabilidad
- Desarrollo de las personas
- Trabajo en equipo
- Ética en el trabajo
- Integración con la comunidad
- Compromiso con la calidad, seguridad y el medio ambiente

1.4 ASPECTOS LOGÍSTICOS PARA EL RESGUARDO DE INFORMACIÓN

Administración de la información como recurso

La información se define como la acción y efecto de informar; es decir enterar o instruir. La informática es la ciencia del tratamiento automático y racional de la información considerada como soporte de los conocimientos y las comunicaciones. Actualmente la información tiene una connotación de recurso primordial en la empresa siendo ésta promotor de la misma. La información puede llegar a ser el elemento decisivo, que en un momento dado, determine el éxito o el fracaso de una empresa. Con el fin de lograr la máxima utilidad de la información, ésta debe administrarse de manera correcta. Existen costos que se asocian con la producción, distribución, seguridad, almacenamiento, recuperación y desecho de la información. Su uso estratégico como un apoyo de la competitividad de la empresa no debe considerarse como un elemento gratuito aunque ésta aparentemente se encuentra siempre a nuestro alcance.

Administración de la información generada por computadora

La disponibilidad actual de las computadoras ha generado todo un incremento y una diversificación de la información, tanto para la sociedad general como para las empresas. La administración de la información que se genera por computadora, difiere en diversas formas de aquella que se obtiene manualmente. A menudo, se tiene una mayor cantidad de información si ésta se genera utilizando sistemas computacionales; los costos para crear y mantener la información que genera la computadora puede llegar a multiplicarse a velocidades impresionantes. Con frecuencia la información que se genera por computadora se trata con menos escepticismo que la obtenida por otros medios.

Resguardo

Se define como la guardia que se pone a una cosa para que esta no sufra pérdida y que a su vez es un documento que certifica algo.

Respaldo

Reserva para casos de urgencia; el término datos de respaldo se refiere a una copia extra, inteligible para la máquina, de un archivo o base de datos.

Restauración

El Procedimiento de restauración después de una falla de operación; respaldo y restauración es una combinación de procedimientos manuales y mecánicos, mediante los cuales es posible restaurar las informaciones perdidas en caso de falla del Hardware o del Software. Los respaldos periódicos de archivos, bases de datos y programas, así como las bitácoras que siguen las operaciones de la computadora, forman parte del programa de respaldo y restauración.

Registros y archivos

Registro es un grupo de hechos relacionados tratados como una unidad.

Manejo de los registros

Los registros en un sistema empresarial se acumulan. Al expansionarse la organización y alejarse la dirección de las transacciones diarias, el control de flujo de información respecto a las actividades del sistema se convierte en una función especializada la cual debe brindar:

- **Acceso**. Acceso rápido a los registros activos y a los inactivos que es política de la compañía retener
- **Retención**. Flujo eficiente y efectivo de información desde su creación y uso hasta su destrucción

- **Archivo**. Dispositivos de archivo adecuados y áreas de depósito accesibles para registros tanto temporales como permanentes

Acceso

El acceso a los registros se facilita por un ordenamiento lógico o de secuencia, tal como el ordenamiento numérico creciente por el número de seguro social, la fecha, el número de ítem o el número de departamento. La frecuencia de solicitud es un factor importante que afecta el acceso. Los registros activos como órdenes o pedidos de venta y cuentas por cobrar deben ser fácilmente accesibles a los responsables de su proceso. Los registros inactivos, como los contratos de la compañía pueden ser guardados en cajas de seguridad o tesoros bancarios y registrados electrónicamente para su control, ubicación y recuperación de los mismos.

Retención

La pregunta sobre qué registros hay que retener y por cuánto tiempo, en parte puede ser respondida por las necesidades o política interna de una sistema y requerimientos externos tales como reglamentaciones gubernamentales; éstas exigen que los registros sobre impuestos, seguridad social, transacciones accionarías, compensaciones laborales, liquidaciones de sueldos y comercio interestatal se conserven por los sistemas empresariales y las leyes exigen que esos registros estén a disposición de diversas oficinas federales y estatales para su inspección. Los registros de presupuestos, cálculos de ingeniería, planes y contratos comerciales deben ser considerados de naturaleza permanente y nunca ser destruidos. Los registros sobre las transacciones individuales de un sistema comercial -comprobantes de pagos, facturas, boletas de transporte, etc. – deben de ser conservados durante el tiempo que determine la política interna. Para evitar la acumulación de registros innecesarios, deben destruirse todos aquellos cuyo periodo de retención ha terminado. Aún entonces, la destrucción de cualquier registro requiere de autorización apropiada.

Archivo

Un archivo es un conjunto de registros con algunas características o función en común. Los archivos son importantes herramientas empresariales porque contienen las historias de actividades o de comunicaciones en un sistema empresarial. Archivar es la operación de insertar un registro en su lugar correspondiente en un archivo. En los sistemas manuales y en algunos tipos de aplicaciones de procesador electrónico, donde hay que insertar muchos registros en un archivo, la ordenación previa de las inserciones en la secuencia adecuada antes de archivar, ahorra tiempo al eliminar el ir y volver dentro del archivo.

Permanencia de archivos

Los archivos frecuentemente se caracterizan por ser o bien permanentes o temporales. El modo en que se archivan los registros es a menudo crucial para el funcionamiento de un sistema. Es obligatorio un arreglo lógico que cumpla las condiciones de seguridad normales. Los registros deben de estar guardados en muebles de acero, ya sea en armarios metálicos, estanterías metálicas o cajas de seguridad; el depósito debe ser a prueba de fuego e inundaciones. Cuando el volumen es grande y la frecuencia de consulta también lo es, un sistema de gestión de información automatizada para ayudar a controlar el acceso, retención y archivo desarrollando un inventario de registros y un calendario de retención.

Diseño de archivos

Los archivos y sus registros deber ser diseñados y organizados de modo que se les puedan leer o procesar fácilmente. A medida que aumentan las necesidades de registro en un sistema empresarial, también lo hace la dependencia entre un subsistema y los otros. Como varios programas de computación usan los mismos archivos para proceso, algunos archivos resultan más activos que otros. Al diseñar un archivo y sus registros asociados debe prestarse cuidadosa atención a cuatro factores básicos:

- Actividad del archivo
- Dimensión del archivo
- Entrada
- Salida

Actividad del archivo

Uno de los aspectos importantes al diseñar la organización de un archivo, es el del porcentaje de registros activos. Si se ha de procesar un porcentaje pequeño de registros a la vez, el archivo debe ser diseñado de modo de permitir un acceso directo a cualquier registro sin tener que revisar todos los demás registros del mismo.

Dimensión del archivo

El tamaño del archivo juega un papel muy importante en la determinación del diseño y organización del mismo, así como en el tipo y la magnitud del lugar requerido. No solo se debe considerar la dimensión actual del archivo, sino también su crecimiento estimado.

Entrada

Otra consideración importante en el diseño del archivo, es el de los tipos de documentos que constituyen la entrada al archivo. El archivo debe estar organizado de manera de procesar las entradas en forma eficiente.

Salida

Al procesar los archivos, generalmente se emiten informes o documentos de algún tipo. El archivo debe estar diseñado de manera que se obtenga fácilmente la información para la salida.

Identificación de archivos

El principio y el fin de cada archivo deben estar claramente identificados, tanto en los sistemas manuales como en los de procesado electrónico.

Registros de detalle

Los registros de detalle, que constituyen el volumen del archivo, contienen los datos necesarios para el proceso.

Rótulo final

El rótulo final o último registro que identifica el final de un archivo, también puede servir para separarlo del archivo siguiente. Éstos pueden contener información sumaria sobre la cantidad de registros en el archivo.

Dispositivos de almacenamiento de archivos

Generalmente la función del archivo determina el tipo de almacenamiento en que se aloja. Para lograr que sean tanto confidenciales como accesibles, se le puede guardar en muebles metálicos en la oficina de personal en sobres de papel, cajas, etc.

Registros de procesamiento electrónico y archivos

Como los costos de procesamiento de archivos en computación pueden ser altos, se ha de tratar más en detalle la organización de estos archivos, con miras al acceso y procesamiento eficiente de sus registros. En los sistemas de procesamiento electrónico, la información proveniente de documentos se captura electrónicamente y se almacenan en discos magnéticos, donde queda definitivamente alojada. Asimismo, hay dispositivos como el de reconocimiento óptico de caracteres o el de lectura de marcas sensibles, que permiten el procesamiento de documentos adecuadamente diseñados y rellenados, sin necesidad de trascripción. Todo sistema deberá contar con la documentación de

los procedimientos de resguardo y recuperación antes de entrar en producción. La misma será controlada por el área responsable de la Seguridad Informática para verificar que es clara, completa y contempla como mínimo la recuperación de los siguientes elementos:

- El reemplazo de los servidores críticos
- El sistema operativo y su configuración (parámetros, sistema de archivos, particiones, usuarios y grupos, etc.)
- Los utilitarios y paquetes de software de base necesarios para que la aplicación se ejecute
- Los programas que componen la aplicación
- Los archivos y/o bases de datos del sistema
- Horario de ejecución de la copia de resguardo

No se pondrá en producción ningún sistema que no cumpla este requerimiento. Todas las copias de resguardo deberán estar claramente identificadas, con etiquetas que indiquen como mínimo:

- Equipo al que pertenecen
- Fecha y hora de ejecución
- Frecuencia: anual, mensual, semanal, diario
- número de secuencia o lote
- Tipo de respaldo
- Nombre del sistema o aplicativo otros datos necesarios para su fácil reconocimiento

Se llevará un registro diario de las cintas en uso indicando al menos,

- Fecha de ejecución del resguardo
- Qué cintas integran el backup de los equipos
- Cantidad de veces que se usó la cinta
- Lugares asignados para su guarda

El área responsable de Seguridad Informática revisará periódicamente que se cumpla con este registro en tiempo y forma. Todos los procedimientos de respaldo deberán generar una bitácora en el equipo que permita la revisión del resultado de la ejecución, y dentro de lo posible, se realizarán con la opción de verificación de integridad (lectura posterior a la escritura). Los sitios donde se almacenen las copias de resguardo deberán ser físicamente seguros, con los controles físicos y ambientales según normas estándares; las cintas deben guardarse dentro de armario ignífugo. Se generarán en lo posible dos copias de resguardo, guardando una de ellas en un edificio diferente al del ámbito de procesamiento en un lugar que cumpla con los requerimientos mencionados y a distancia tal que la ocurrencia de cualquier contingencia en uno no afecte al otro. En caso de tener solo una copia esta debe ser llevada fuera del ámbito de procesamiento de la forma anteriormente mencionada. El traslado de las cintas debe ser realizado por personal debidamente autorizado, utilizando los accesos habilitados para movimiento de insumos. Se realizarán copias de resguardo del sistema completo de acuerdo a lo indicado en la frecuencia asignada a cada aplicación o sistema, previendo la conservación de estos respaldos por el período de tiempo también estipulado previamente conforme a la criticidad de la información. En el caso de utilizar respaldos incrementales se deberá tener en cuenta lo siguiente:

- Se documentará la identificación de secuencia de los respaldos incrementales
- Deberán existir controles para prevenir la carga de cintas en una secuencia equivocada
- Se realizará un respaldo del sistema completo cada siete días

Se efectuarán pruebas de recuperación de las copias de resguardo al menos una vez cada treinta días. Estas pruebas servirán para constatar que se puedan obtener correctamente los datos grabados en la cinta al momento de ser necesarios, de forma de garantizar su propósito. Las pruebas se deberán formalizar en un acta escrita y firmada por el responsable del sector técnico y el encargado de realizar la recuperación. Eventualmente el área responsable de la

Seguridad Informática presenciará las pruebas y firmará el acta. Los servidores críticos deberán contar con arreglo de discos redundantes a los efectos de que la información sensible no se vea afectada por potenciales desperfectos en los discos. Para el caso de aplicaciones críticas se implementarán técnicas de replicación automática, por hardware o software, de forma tal que si el equipo/base de datos principal deje de funcionar el equipo/base de datos espejo tome el control inmediatamente. Los períodos de retención de la información histórica son los siguientes:

- **Fuentes y base de datos**. Perpetuo
- **Lotes de transferencia**: Perpetuo
- **Actividades de los usuarios y pistas de auditoria**: 3 años

El resguardo de la información histórica se realizará utilizando soportes magnéticos de preferencia no reutilizables (discos ópticos). Los procedimientos de generación y grabación de estos archivos serán automáticos, a fin de evitar su modificación.

Integración de la información

Resguardo

- Archivos fuente
- Creación de directorios, subdirectorios y archivos
- Preparación de bases de datos
- Combinación de recursos

Es obligación del personal cuyas funciones así lo ameriten, implantar registros de las operaciones y producir la información necesaria relativa a sus actividades sin omitir ninguna etapa del proceso, que se produzca con oportunidad y veracidad, conservándola bajo su custodia, en previsión de usos indebidos o no autorizados. El personal encargado mantendrá en reserva la información que no deba ser de dominio público, así como abstenerse de aprovecharla para fines particulares.

La información privada sobre otras áreas a la que se accede por razón de la función, o aún de manera circunstancial, no deberá ser divulgada.

Asegurar la disponibilidad de la información

Una vez que la información ha sido capturada en un sistema de cómputo, debe ser almacenada de manera segura y estar disponible para los usuarios cuando la necesiten. La información también debe ser mantenida y utilizada de tal forma que su integridad no se vea comprometida. Hay dos componentes principales para asegurar la disponibilidad de la información y, por tanto, de minimizar el riesgo de confiarle información valiosa a la tecnología. Estos son asegurar que los sistemas operen para presentar la información cuando se requiera y contar con información de respaldo para prevenir fallas del sistema o pérdida de información. La información para los usuarios es uno de los propósitos principales de un sistema de cómputo. Empero, al mismo tiempo es importante asegurar su integridad. Solo aquellos usuarios con una "necesidad de saber" deben tener acceso a la información, particularmente cuando es sensible. Una vez más los permisos y contraseñas de acceso se pueden utilizar para restringirlo a quienes necesitan hacer cambios. Cuando a los usuarios se les permite efectuar cambios, se pueden utilizar técnicas de verificación para disminuir la posibilidad de errores.

Respaldo de la información

El resguardo final contra cualquier falla del sistema o pérdida de la información son los respaldos. Un respaldo regular requiere tomar este tipo de precauciones al menos una vez cada jornada laboral, o con mayor frecuencia si se trata de información crítica, como durante el desarrollo de una jornada electoral. La información puede ser respaldada en muy distintos formatos: discos removibles de diverso tipo, discos duros múltiples o cintas magnéticas, por ejemplo. La estrategia integral de tecnología informativa de la organización puede estructurar un régimen formal de respaldo. Lo ideal es que el respaldo funcione de manera automática para asegurar que los errores humanos no causen problemas. Sin embargo,

seguirá siendo necesaria una revisión periódica de los respaldos automáticos para asegurar que los errores de la computadora no causen problemas. La información "viva" puede ser respaldada al mismo tiempo que es creada, utilizando un disco duro como espejo, que puede estar localizado en el mismo servidor o en uno distinto. Al utilizar discos espejo, la misma información es almacenada de manera simultánea en dos o más discos. Esto significa que si un disco falla, la información puede ser restaurada de otro. Es preferible utilizar servidores separados para los discos espejo, ya que el segundo se puede utilizar si el primero falla completamente. Los programas, tanto los comerciales como los desarrollados internamente, también pueden ser respaldados para que se puedan recargar si las versiones originales se pierden o dañan. La mayoría de los programas vienen cargados en discos. Sin embargo es cada vez más frecuente que los programas se puedan descargar de Internet. En este caso las copias de respaldo deben ser almacenadas localmente, ya que no existe garantía de que estarán disponibles en línea en el futuro. Los discos de programa que son almacenados en una biblioteca administrada por una unidad u oficial responsable pueden ser fácilmente ubicados y utilizados, de ser necesario. Cuando se respalda la información de programas, se debe tener cuidado de no violar los permisos legales. La mayoría de las licencias o permisos legales permiten conservar copias de respaldo.

1.5 ORGANIGRAMAS

Son sistemas de organización que se representan en forma intuitiva y con objetividad. También son llamados cartas o gráficas de organización. Consisten en hojas o cartulinas en las que cada puesto de un jefe se representa por un cuadro que encierra el nombre de ese puesto (y en ocasiones de quien lo ocupa) representándose, por la unión de los cuadros mediante líneas, los canales de autoridad y responsabilidad.

Los organigramas señalan la vinculación que existe entre sí de los departamentos a lo largo de las líneas de autoridad principales.

¿Para qué sirve un organigrama?

Los organigramas revelan:

- La división de funciones
- Los niveles jerárquicos
- Las líneas de autoridad y responsabilidad
- Los canales formales de comunicación
- La naturaleza lineal o staff del departamento
- Los jefes de cada grupo de empleados, trabajadores, etc.
- Las relaciones existentes entre los diversos puestos de la empresa y en cada departamento o sección

Requisitos de un organigrama

- **Los organigramas deben ser,** ante todo, muy claros; por ello se recomienda que no contenga un número excesivo de cuadros y de puestos, ya que esto, en vez de ayudar a la estructura administrativa de la empresa, puede producir mayores confusiones

- **Los organigramas no** deben comprender ordinariamente a los trabajadores o empleados haciéndolos arrancar del Director, o Gerente General y terminarlos con los jefes o supervisores del último nivel
- **Los organigramas deben contener** nombres de funciones y no de personas. Cuando se desea que estos últimos figuren, conviene colocar dentro del mismo cuadro, con una letra mayor el nombre del puesto y con letra menor el nombre de la persona que lo ocupe

Ventajas de un organigrama

Saber las jerarquías de una empresa ya que las relaciones subordinado-superiores no existen debido a que se elabore el diagrama, sino más bien, a las relaciones de dependencia esenciales, mostrando líneas de autoridad para tomar decisiones, ya que el simple hecho de representar la estructura de una organización mediante un organigrama puede mostrar inconsistencias y complejidades. El organigrama también les muestra a los gerentes y al personal nuevo cómo encajan dentro de toda la estructura.

Limitaciones

Los organigramas están sujetos a limitaciones importantes. En primer lugar, un organigrama sólo muestra las relaciones formales de autoridad y omite las múltiples relaciones importantes informales que se encuentran en una empresa típicamente organizada. Muestra también las relaciones importantes de línea o formales. No muestra cuanta autoridad existe en cualquier punto de la estructura.

Otra dificultad con los organigramas es que quizá las personas confundan las relaciones de autoridad con la posición en la empresa. Aunque una buena elaboración del organigrama intenta hacer que los niveles en la gráfica estén de acuerdo con los niveles de importancia en la empresa, no siempre puede hacerse así. Este problema se puede manejar exponiendo con claridad las relaciones de autoridad y utilizando el mejor indicador de la posición –los niveles de sueldos y de bonos.

Clases de organigramas

Los organigramas pueden ser:

- Verticales
- Horizontales
- Circulares
- Escalares

Organigrama vertical

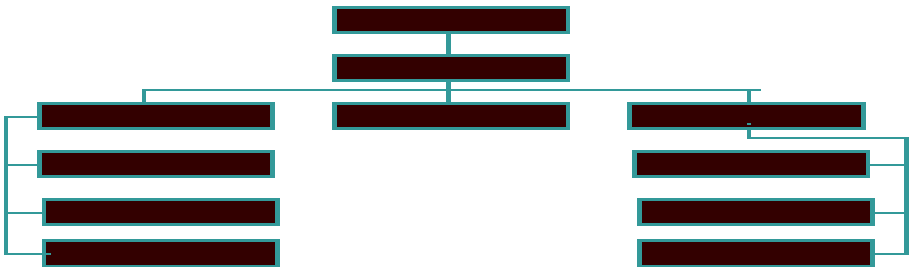
En los organigramas verticales, cada puesto subordinado a otro se representa por cuadros en un nivel inferior, ligados a aquel por líneas que representan la comunicación de responsabilidad y autoridad. De cada cuadro del segundo nivel se sacan líneas que indican la comunicación de autoridad y responsabilidad a los puestos que dependen de él y así sucesivamente.

Ventajas:

- Son las más usadas y por lo mismo, fácilmente comprendidas
- Indican en forma objetiva las jerarquías del personal

Desventajas:

Se produce el llamado “efecto de triangulación”, ya que, después de dos niveles, es muy difícil indicar los puestos inferiores, para lo que se requeriría hacerse organigramas muy alargados.



1.5.1 ORGANIGRAMA VERTICAL

Organigrama horizontal

Representan los mismos elementos del organigrama anterior y en la misma forma, sólo que comenzando el nivel máximo jerárquico a la izquierda y haciéndose los demás niveles sucesivamente hacia la derecha.

Ventajas:

- Siguen la forma normal en que acostumbramos leer
- Disminuyen en forma muy considerable el efecto de triangulación
- Indican mejor la longitud de los niveles por donde pasa la autoridad formal

Desventajas:

Son pocos usados en prácticas, y muchas veces, aun pudiendo hacerse una sola carta de toda la organización, resultan los nombres de los jefes demasiado apiñados y, por lo tanto, poco claros.



1.5.2 ORGANIGRAMA HORIZONTAL

Organigramas circulares

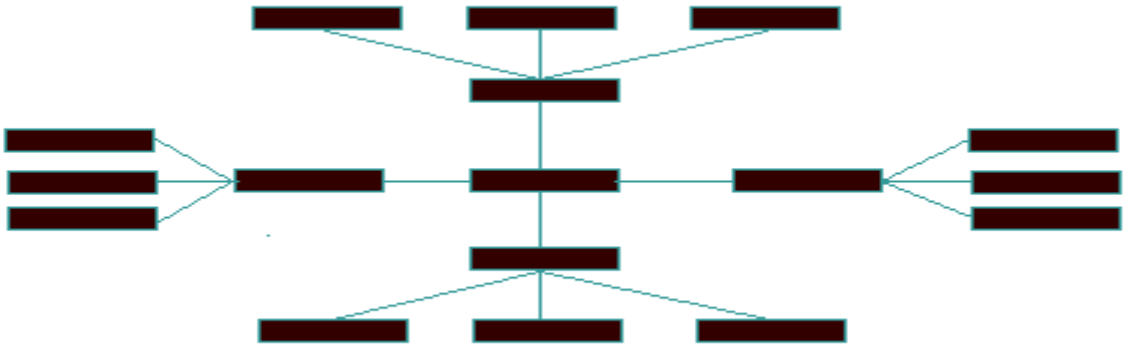
Formados por un cuadro central, que corresponde a la autoridad máxima en la empresa, a cuyo derredor se trazan círculos concéntricos, cada uno de los cuales constituye un nivel de organización. En cada uno de esos círculos se coloca a los jefes inmediatos, y se les liga con líneas, que representan los canales de autoridad y responsabilidad.

Ventajas:

- Señalan muy bien, forzando a ello, la importancia de los niveles jerárquicos
- Eliminan, o disminuyen al menos, la idea del status más alto o más bajo
- Permiten colocar mayor número de puestos en el mismo nivel

Desventajas:

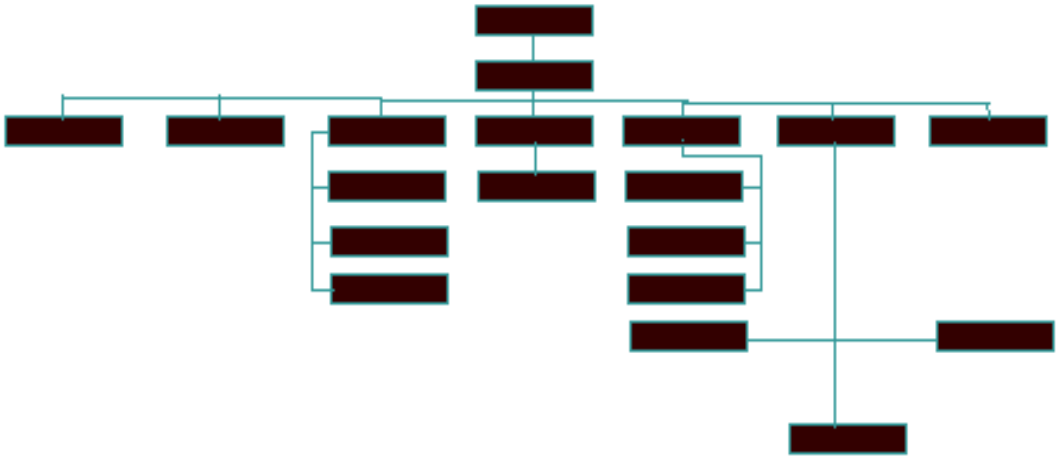
Resultan confusos y difíciles de leer; porque no permiten colocar con facilidad niveles donde hay un solo funcionario y que fuerzan demasiado los niveles.



1.5.3 ORGANIGRAMA CIRCULAR

Organigrama Mixto

En este tipo de organigramas se puede mezclar los tres tipos de organigramas anteriores (Vertical, Horizontal, Circular) en uno sólo, cada empresa, cada organización utiliza este tipo de organigramas debido a su alto volumen y complejidad de puestos que tienen bajo su administración y con ello buscan la optimización del espacio en el que se encuentran trabajando.



1.5.3 ORGANIGRAMA CIRCULAR

Ejemplo de organigrama (véase figura 1.5.4)

Gerencia de Contabilidad y Finanzas

Actividades que pueden realizarse dentro de este departamento:

- Autorizar gastos
- Proteger bienes y valores
- Promover eficiencia del personal
- Prevenir fraudes
- Detectar desperdicios innecesarios, tanto de materia, tiempo, costo, etc.
- Actualiza la tecnología
- Maneja presupuestos
- Maneja recursos financieros y económicos
- Representa intereses de los diversos grupos de una institución



FIGURA 1.5.4 EJEMPLO ORGANIGRAMA

2.1. CONCEPTOS FUNDAMENTALES DE LA PROGRAMACIÓN ORIENTADA A OBJETOS

La programación orientada a objetos (POO) se basa en los siguientes conceptos:

El elemento fundamental de la POO es, como su nombre lo indica, el objeto que es la unidad que permite combinar datos y funciones que operan sobre esos datos. Dentro de un objeto residen los datos (atributos) que caracterizan el estado del objeto. Así como funciones o subrutinas (métodos) que operan sobre ellos. Los métodos dentro del objeto son el único medio de acceder a los datos privados de un objeto. Los datos están ocultos, y eso asegura que no se puedan modificar accidentalmente por funciones externas al **objeto.Estructura De un Objeto**

Un objeto puede considerarse como una especie de cápsula dividida en tres partes:

- Relaciones
- Propiedades
- Métodos

Cada uno de estos componentes desempeña un papel totalmente independiente.

Relaciones

Las relaciones entre objetos son, precisamente, los enlaces que permiten a un objeto relacionarse con aquellos que forman parte de la misma organización. Las hay de dos tipos:

- **Relaciones jerárquicas**. Son esenciales para la existencia misma de la aplicación porque la construyen. Son bidireccionales, es decir, un objeto es padre de otro cuando el primer objeto se encuentra situado inmediatamente encima del segundo en la organización en la que ambos forman parte; asimismo, si un objeto es padre de otro, el segundo es hijo del primero

Con formato: Derecha: 0.04 cm, Punto de tabulación: 15.56 cm, Derecha + No en 15 cm

Con formato: Centrado, Posición: Horizontal: Centro, Con relación a: Margen

  - **Una organización jerárquica simple**: puede definirse como aquella en la que un objeto puede tener un solo padre
  - **Una organización jerárquica compleja**: un hijo puede tener varios padres
- **Relaciones semánticas**. Se refieren a las relaciones que no tienen nada que ver con la organización de la que forman parte los objetos que las establecen. Sus propiedades y consecuencia solo dependen de los objetos en sí mismos (de su significado) y no de su posición en la organización

Propiedades

Las propiedades distinguen un objeto determinado de los restantes que forman parte de la misma organización y tiene valores que dependen de la propiedad de que se trate. Todo objeto puede tener cierto número de propiedades, cada una de las cuales tendrá, a su vez, uno o varios valores. En POO, las propiedades corresponden a las clásicas "variables" de la programación estructurada. Son, por lo tanto, datos encapsulados dentro del objeto, junto con los métodos (programas) y las relaciones (punteros a otros objetos). Las propiedades de un objeto pueden tener un valor único o pueden contener un conjunto de valores más o menos estructurados (matrices, vectores, listas, etc.). Además, los valores pueden ser de cualquier tipo (numérico, alfabético, etc.) si el sistema de programación lo permite. Pero existe una diferencia con las "variables", y es que las propiedades se pueden heredar de unos objetos a otros. En consecuencia, un objeto puede tener una propiedad de maneras diferentes:

- **Propiedades propias**. Están formadas dentro de la cápsula del objeto
- **Propiedades heredadas**. Están definidas en un objeto diferente, antepasado de éste (padre,"abuelo", etc.). A veces estas propiedades se llaman propiedades miembro porque el objeto las posee por el mero hecho de ser miembro de una clases

Con formato: Derecha: 0.04 cm, Punto de tabulación: 15.56 cm, Derecha + No en 15 cm

Con formato: Centrado, Posición: Horizontal: Centro, Con relación a: Margen

Métodos

Podemos definir método como un programa procedural escrito en cualquier lenguaje, que está asociado a un objeto determinado y cuya ejecución sólo puede desencadenarse a través de un mensaje recibido por éste o por sus descendientes. Son sinónimos de 'método' todos aquellos términos que se han aplicado tradicionalmente a los programas, como procedimiento, función, rutina, etc. Sin embargo, es conveniente utilizar el término 'método' para que se distingan claramente las propiedades especiales que adquiere un programa en el entorno POO, que afectan fundamentalmente a la forma de invocarlo (únicamente a través de un mensaje) y a su campo de acción, limitado a un objeto y a sus descendientes, aunque posiblemente no a todos. Si los métodos son programas, se deduce que podrían tener argumentos, o parámetros. Puesto que los métodos pueden heredarse de unos objetos a otros, un objeto puede disponer de un método de dos maneras diferentes:

- **Métodos propios**. Están incluidos dentro de la cápsula del objeto
- **Métodos heredados**. Están definidos en un objeto diferente, antepasado de éste (padre,"abuelo", etc.). A veces estos métodos se llaman “miembro” porque el objeto los posee por el mero hecho de ser miembro de una clase

Organización de los objetos

Los objetos forman siempre una organización jerárquica, en el sentido de que ciertos objetos son superiores a otros de cierto modo. Existen varios tipos de jerarquías: serán simples cuando su estructura pueda ser representada por medio de un "árbol". En otros casos puede ser más compleja. En cualquier caso, sea la estructura simple o compleja, podrán distinguirse en ella tres niveles de objetos.

- **La raíz de la jerarquía.** Se trata de un objeto único y especial. Este se caracteriza por estar en el nivel más alto de la estructura y suele recibir un nombre muy genérico, que indica su categoría especial, como por ejemplo objeto madre, Raíz o Entidad

Con formato: Derecha: 0.04 cm, Punto de tabulación: 15.56 cm, Derecha + No en 15 cm

Con formato: Centrado, Posición: Horizontal: Centro, Con relación a: Margen

- **Los objetos intermedios**. Son aquellos que descienden directamente de la raíz y que a su vez tienen descendientes. Representan conjuntos o clases de objetos, que pueden ser muy generales o muy especializados, según la aplicación. Normalmente reciben nombres genéricos que denotan al conjunto de objetos que representan. En un conjunto reciben el nombre de clases o tipos si descienden de otra clase o subclase
- **Los objetos terminales**. Son todos aquellos que descienden de una clase o subclase y no tienen descendientes. Suelen llamarse casos particulares, instancias o ítems porque representan los elementos del conjunto representado por la clase o subclase a la que pertenecen

Clase

Clase es la descripción abstracta de un conjunto de objetos. Consta de métodos y datos que resumen características comunes del conjunto de objetos. Se pueden definir muchos objetos de una misma clase. Es decir una clase es la declaración de un tipo de objetos.

Subclases

Descripción de una subclase, basada en la estructura de otra clase, la cual se denomina superclase. Se puede mirar como una extensión de la superclase, donde se pueden tomar las características de la superclase, y adicionarle nuevas características. Este concepto suele confundirse con el concepto de herencia, pero no siempre puede decirse que una subclase es una herencia, ya que una subclase puede redefinir (override) todo lo de su superclase, luego no heredaría nada. En la herencia se reciben las definiciones de la superclase y se usan conjuntamente con nuevas definiciones; heredar es compartir atributos entre una clase y sus subclases, si no comparte no hay herencia.

Con formato: Derecha: 0.04 cm, Punto de tabulación: 15.56 cm, Derecha + No en 15 cm

Con formato: Centrado, Posición: Horizontal: Centro, Con relación a: Margen

Herencia

Propiedad que permite a los objetos ser construidos a partir de otros objetos. Dicho de otra forma la capacidad de un objeto para utilizar estructuras de datos y métodos presentes en sus antepasados. El objetivo principal de la herencia es la reutilización, poder utilizar código desarrollado con anterioridad. La herencia se basa en la división de clases básicas en subclases. Dicha división se fundamenta en el concepto de jerarquía. La herencia supone una clase base y una jerarquía de clases que contiene las clases derivadas. Las clases derivadas pueden heredar el código y los datos de su clase base, añadiendo su propio código especial y datos, incluso cambiar aquellos elementos de la clase base que necesitan ser diferentes. No debe confundirse la relación de los objetos con las clases, con la relación de una clase base con sus clases derivadas. Los objetos existentes en la memoria de la computadora expresan las características exactas de su clase. Las clases derivadas heredan características de su clase base, pero añaden otras características propias, nuevas. Las clases derivadas pueden a su vez servir como clases base para definir nuevas clases y de esta forma enfatizar la reutilización.

Existen dos tipos de herencia:

- **Herencia simple.** Una clase puede tener sólo un ascendente. Es decir una subclase puede heredar datos y métodos de una única clase base
- **Herencia múltiple.** Una clase puede tener más de un ascendente inmediato, adquirir datos y métodos de más de una clase

La mayor parte de los lenguajes orientados a objetos permiten la herencia simple no así la herencia múltiple porque pueden surgir ambigüedades. Al aplicar la herencia múltiple si las clases utilizadas para definir una clase nueva tienen un método con el mismo nombre aparecerán problemas de ambigüedad que deberán resolverse con una operación de prioridad que el lenguaje de programación deberá soportar y entender. Cuando se define una subclase a partir de una clase base, la subclase no tiene que heredar todas las características de la clase base,

Con formato: Derecha: 0.04 cm, Punto de tabulación: 15.56 cm, Derecha + No en 15 cm

Con formato: Centrado, Posición: Horizontal: Centro, Con relación a: Margen

puede seleccionarse únicamente las características de utilidad, esto se conoce como herencia selectiva.

Polimorfismo

Una de las características fundamentales de la POO es el polimorfismo, que no es otra cosa que la posibilidad de construir varios métodos con el mismo nombre, pero con relación a la clase a la que pertenece cada uno, con comportamientos diferentes. Esto conlleva la habilidad de enviar un mismo mensaje a objetos de clases diferentes. Estos objetos recibirían el mismo mensaje global pero responderían a él de formas diferentes; por ejemplo, un mensaje "+" a un objeto ENTERO significaría suma, mientras que para un objeto STRING significaría concatenación ("pegar" cadenas uno seguido al otro).

Encapsulamiento y Ocultación

Cada objeto es una estructura compleja en cuyo interior hay datos y programas, todos ellos relacionados entre sí, como si estuvieran encerrados conjuntamente en una cápsula. Esta propiedad (encapsulamiento), es una de las características fundamentales en la POO. Los objetos son inaccesibles, e impiden que otros objetos, los usuarios, o incluso los programadores conozcan cómo está distribuida la información o qué información hay disponible. Esta propiedad de los objetos se denomina ocultación de la información. Esto no quiere decir, sin embargo, que sea imposible conocer lo necesario respecto a un objeto y lo que contiene. Si así fuera no se podría hacer gran cosa con él. Lo que sucede es que las peticiones de información a un objeto, deben realizarse a través de mensajes dirigidos a él, con la orden de realizar la operación pertinente. La respuesta a estas órdenes será la información requerida, siempre que el objeto considere que quien envía el mensaje está autorizado para obtenerla. El hecho de que cada objeto sea una cápsula facilita enormemente que un objeto determinado pueda ser transportado a otro punto de la organización, o incluso a otra organización totalmente diferente que precise de él. Si el objeto ha sido bien construido, sus métodos seguirán

Con formato: Derecha: 0.04 cm, Punto de tabulación: 15.56 cm, Derecha + No en 15 cm

Con formato: Centrado, Posición: Horizontal: Centro, Con relación a: Margen

funcionando en el nuevo entorno sin problemas. Esta cualidad hace que la POO sea muy apta para la reutilización de programas.

Modularidad

Proceso de crear partes de un todo que se integran perfectamente entre sí para que funcionen por un objetivo general, y a las cuales se les pueden agregar más componentes que se acoplen perfectamente al todo, o extraerle componentes sin afectar su funcionamiento. En el caso que se requiera actualizar un módulo, no hay necesidad de hacer cambios en otras partes del todo. También se puede cambiar su finalidad si se acomodan esos módulos para darle otro objetivo.

Con formato: Derecha: 0.04 cm, Punto de tabulación: 15.56 cm, Derecha + No en 15 cm

Con formato: Centrado, Posición: Horizontal: Centro, Con relación a: Margen

2.2 METODOLOGÍAS DE DISEÑO ORIENTADO A OBJETOS

Durante finales de la década de 1980 y principios de la de 1990 se inició el campo de la metodología orientada a objetos con la publicación de varios métodos de desarrollo de este tipo (Shlaer-88, Coad-91, Booch-91, Rumbaugh-91) así como la de varios libros de diseño de lenguajes de programación. El libro Objectory (Jacobson-92) publicado un poco después se basa en trabajos anteriores haciendo diferencia con su enfoque sobre los casos de uso y el proceso de desarrollo. Durante los siguientes años aparecieron muchas otras metodologías, añadiendo algunos nuevos conceptos pero, en general, teniendo gran similitud entre las propuestas de los distintos autores, pues partían de uno o más de los métodos existentes haciendo extensiones o cambios menores. Los autores originales actualizaban además sus trabajos, incorporando en algunos casos, ideas de otros autores. Surgió con esto una serie de conceptos comunes, aunque con algunas discrepancias para las distintas metodologías que provocaban confusión entre los desarrolladores que iniciaban la utilización de este tipo de metodologías. Hubo algunos intentos de unificar los conceptos entre métodos (Fusion por Coleman-94), pero como en este no se involucró a los autores originales, permaneció como un método más. El primer intento exitoso de combinar y reemplazar los métodos existentes llegó cuando Rumbaugh y Booch se unieron en Rational Software, obteniendo una primera propuesta en 1995, momento en el que llegó además Jacobson para integrar lo que fue llamado Lenguaje Unificado de Modelado (UML). Este grupo comenzó a trabajar posteriormente con metodólogos y desarrolladores de otras compañías para generar una propuesta a los miembros del Object Management Group (OMG), siendo UML sometido finalmente a la consideración del OMG en septiembre de 1997. La metodología Rational Unified Process (RUP) fue desarrollada también por los principales autores de UML (Booch, Rumbaugh y Jacobson) en el que se basa. Cabe destacar que las metodologías desarrolladas inicialmente por cada uno de estos autores tienen en menor o mayor grado, modelos que los mismos retomaron para integrar el lenguaje UML.

Con formato: Derecha: 0.04 cm, Punto de tabulación: 15.56 cm, Derecha + No en 15 cm

Con formato: Centrado, Posición: Horizontal: Centro, Con relación a: Margen

Rational Unified Process (RUP)

El proceso de software propuesto por RUP tiene tres características esenciales:

- **Proceso dirigido por Casos de Uso.** Los Casos de Uso son una técnica de captura de requisitos que fuerza a pensar en términos de importancia para el usuario y no sólo en términos de funciones que sería bueno contemplar. Se define un Caso de Uso como un fragmento de funcionalidad del sistema que proporciona al usuario un valor añadido. Los Casos de Uso representan los requisitos funcionales del sistema. Basándose en los Casos de Uso se crean los modelos de análisis y diseño, luego la implementación que los lleva a cabo, y se verifica que efectivamente el producto implemente adecuadamente cada Caso de Uso. Todos los modelos deben estar sincronizados con el modelo de Casos de Uso

- **Proceso centrado en la arquitectura.** La arquitectura de un sistema es la organización o estructura de sus partes más relevantes, lo que permite tener una visión común entre todos los involucrados (desarrolladores y usuarios) y una perspectiva clara del sistema completo, necesaria para controlar el desarrollo
- **La arquitectura involucra los aspectos estáticos y dinámicos** más significativos del sistema, está relacionada con la toma de decisiones que indican cómo tiene que ser construido el sistema y ayuda a determinar en qué orden. Además la definición de la arquitectura debe tomar en consideración elementos de calidad del sistema, rendimiento, reutilización y capacidad de evolución por lo que debe ser flexible durante todo el proceso de desarrollo. La arquitectura se ve influenciada por la plataforma de software, sistema operativo, gestor de bases de datos, protocolos, consideraciones de desarrollo como sistemas heredados. Muchas de estas restricciones constituyen requisitos no funcionales del sistema. En el caso de RUP además de utilizar los Casos de Uso para guiar el proceso se presta especial atención al establecimiento temprano de una buena

Con formato: Derecha: 0.04 cm, Punto de tabulación: 15.56 cm, Derecha + No en 15 cm

Con formato: Centrado, Posición: Horizontal: Centro, Con relación a: Margen

arquitectura que no se vea fuertemente impactada ante cambios posteriores durante la construcción y el mantenimiento. Cada producto tiene tanto una función como una forma. La función corresponde a la funcionalidad reflejada en los Casos de Uso y la forma la proporciona la arquitectura. Existe una interacción entre los Casos de Uso y la arquitectura, los Casos de Uso deben encajar en la arquitectura cuando se llevan a cabo y la arquitectura debe permitir el desarrollo de todos los Casos de Uso requeridos, actualmente y en el futuro. Esto provoca que tanto arquitectura como Casos de Uso deban evolucionar en paralelo durante todo el proceso de desarrollo de software véase figura 2.2.1.

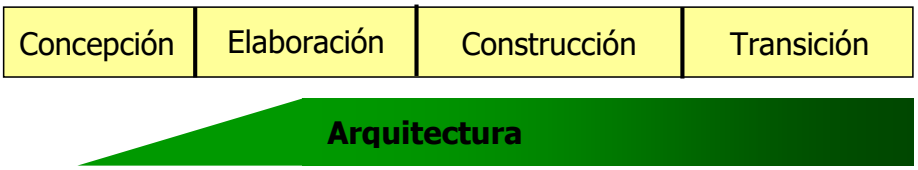


FIGURA 2.2.1: EVOLUCIÓN DE LA ARQUITECTURA DEL SISTEMA

Proceso iterativo e incremental

El proceso iterativo e incremental consta de una secuencia de iteraciones. Cada iteración aborda una parte de la funcionalidad total, pasando por todos los flujos de trabajo relevantes y refinando la arquitectura. Cada iteración se analiza cuando termina para determinar si han aparecido nuevos requisitos o han cambiado los existentes. Toda la retroalimentación de la iteración pasada permite reajustar los objetivos para las siguientes iteraciones, continuando con esta dinámica hasta que se haya finalizado por completo con la versión actual del producto.

RUP divide el proceso en cuatro fases, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor hincapié en los distintas actividades, véase figura 2.2.2.

Con formato: Derecha: 0.04 cm, Punto de tabulación: 15.56 cm, Derecha + No en 15 cm

Con formato: Centrado, Posición: Horizontal: Centro, Con relación a: Margen

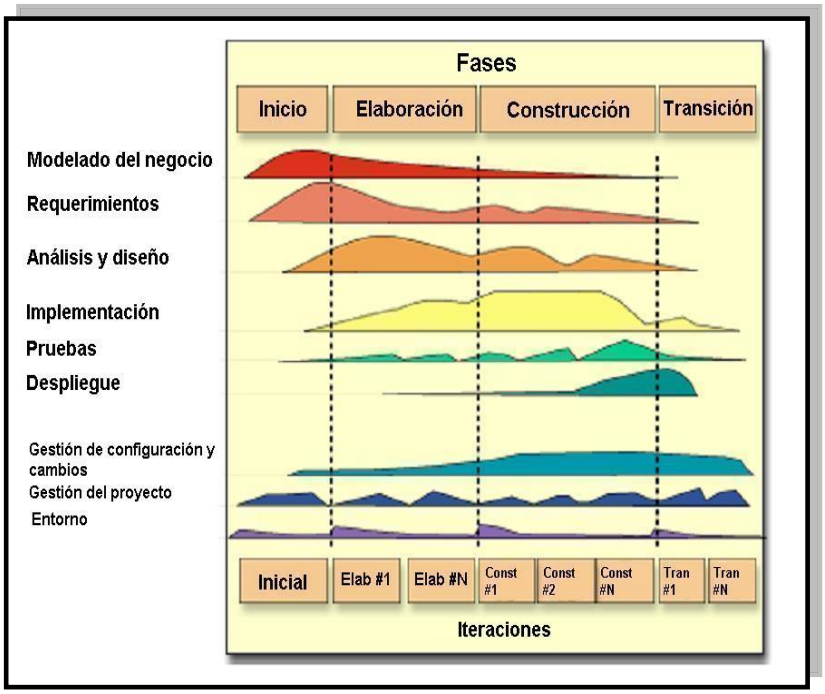


FIGURA 2.2.2 ESFUERZO EN FLUJOS DE TRABAJO SEGÚN FASE DEL PROYECTO

Las primeras iteraciones (en las fases de Inicio y Elaboración) se enfocan hacia la comprensión del problema y la tecnología, la delimitación del ámbito del proyecto, la eliminación de los riesgos críticos, y al establecimiento de una línea base de la arquitectura. Durante la fase de inicio las iteraciones ponen mayor énfasis en actividades de modelado del negocio y de requisitos. En la fase de elaboración, las iteraciones se orientan al desarrollo de la línea base de la arquitectura, abarcan más los flujos de trabajo de requerimientos, modelo de negocios (refinamiento), análisis, diseño y una parte de implementación orientado a la línea base de la arquitectura. En la fase de construcción, se lleva a cabo la construcción del producto por medio de una serie de iteraciones. Para cada iteración se selecciona algunos Casos de Uso, se refina su análisis y diseño y se procede a su

Con formato: Derecha: 0.04 cm, Punto de tabulación: 15.56 cm, Derecha + No en 15 cm

Con formato: Centrado, Posición: Horizontal: Centro, Con relación a: Margen

implementación y pruebas. Se realiza una pequeña cascada para cada ciclo. Se realizan tantas iteraciones hasta que se termine la implementación de la nueva versión del producto. En la fase de transición se pretende garantizar que se tiene un producto preparado para su entrega a la comunidad de usuarios.

Flujos de trabajo

Un flujo de trabajo es una relación de actividades que nos producen resultados observables. A continuación se indican las principales tareas en cada flujo:

- **Modelado del negocio**. Se debe llegar a un mejor entendimiento de la organización donde se va a implantar el producto. Los objetivos del modelado de negocio son:
    - Entender la estructura y la dinámica de la organización para la cual el sistema va ser desarrollado (organización objetivo)
    - Entender el problema actual en la organización objetivo e identificar potenciales mejoras
    - Asegurar que clientes, usuarios finales y desarrolladores tengan un entendimiento común de la organización objetivo
- **Requisitos**. Se establece qué tiene que hacer exactamente el sistema que será construido. En esta línea los requisitos son el contrato que se debe cumplir, de modo que los usuarios finales tienen que comprender y aceptar los requisitos que especifiquemos. Los objetivos son:
    - Establecer y mantener un acuerdo entre clientes sobre lo que el sistema podría hacer
    - Proveer a los desarrolladores un mejor entendimiento de los requisitos del sistema
    - Proveer una base para la planeación de los contenidos técnicos de las iteraciones
    - Proveer una base para estimar costos y tiempo de desarrollo del sistema

Con formato: Derecha: 0.04 cm, Punto de tabulación: 15.56 cm, Derecha + No en 15 cm

Con formato: Centrado, Posición: Horizontal: Centro, Con relación a: Margen

- Definir una interfaz enfocada a las necesidades y metas del usuario

Los requisitos se dividen en dos grupos. Los requisitos funcionales representan la funcionalidad del sistema. Se modelan mediante diagramas de Casos de Uso. Los requisitos no funcionales representan aquellos atributos que debe exhibir el sistema, pero que no son una funcionalidad específica. Por ejemplo requisitos de facilidad de uso, fiabilidad, eficiencia, portabilidad, etc.

- **Análisis y Diseño**. Su objetivo es traducir los requisitos a una especificación que describe cómo implementar el sistema. Los objetivos del análisis y diseño son:

  - Transformar los requisitos al diseño del futuro sistema
  - Desarrollar una arquitectura para el sistema
  - Adaptar el diseño para que sea consistente con el entorno de implementación, diseñando para el rendimiento

El análisis consiste en obtener una visión del sistema que se preocupa de ver qué hace, de modo que sólo se interesa por los requisitos funcionales. Por otro lado el diseño es un refinamiento del análisis que tiene en cuenta los requisitos no funcionales, en definitiva cómo cumple el sistema sus objetivos.

- **Implementación.** En este flujo de trabajo se implementan las clases y objetos en ficheros fuente, binarios, ejecutables y demás. Se deben hacer además las pruebas de unidad. Cada implementador es responsable de probar las unidades que produzca. El resultado final de este flujo de trabajo es un sistema ejecutable. En cada iteración habrá que hacer lo siguiente:

  - Planificar qué subsistemas deben ser implementados y en que orden deben ser integrados, formando el Plan de Integración
  - Si encuentra errores de diseño, los notifica
  - Se prueban los subsistemas individualmente

Con formato: Derecha: 0.04 cm, Punto de tabulación: 15.56 cm, Derecha + No en 15 cm

Con formato: Centrado, Posición: Horizontal: Centro, Con relación a: Margen

- Se integra el sistema siguiendo el plan

- **Pruebas.** Es el encargado de evaluar la calidad del producto que estamos desarrollando, pero no para aceptar o rechazar el producto al final del proceso de desarrollo, sino que debe ir integrado en todo el ciclo de vida. Esta disciplina brinda soporte a las otras disciplinas. Sus objetivos son:

  - Encontrar y documentar defectos en la calidad del software
  - Provee la validación de los supuestos realizados en el diseño y especificación de requisitos por medio de demostraciones concretas
  - Verificar las funciones del producto de software según lo diseñado
  - Verificar que los requisitos tengan su apropiada implementación

Las actividades de este flujo comienzan en el proyecto con el plan de prueba (el cual contiene información sobre los objetivos generales y específicos de las prueba en el proyecto, así como las estrategias y recursos con que se dotará a esta tarea), o incluso antes con alguna evaluación durante la fase de inicio, y continuará durante todo el proyecto.

- **Despliegue.** El objetivo de este flujo de trabajo es producir con éxito el producto y distribuirlo a los usuarios. Las actividades implicadas incluyen:

  - Probar el producto en su entorno de ejecución final
  - Empaquetar el software para su distribución
  - Distribuir el software
  - Instalar el software
  - Proveer asistencia y ayuda a los usuarios
  - Formar a los usuarios y al cuerpo de ventas
  - Migrar el software existente o convertir bases de datos

Con formato: Derecha: 0.04 cm, Punto de tabulación: 15.56 cm, Derecha + No en 15 cm

Con formato: Centrado, Posición: Horizontal: Centro, Con relación a: Margen

- **Gestión del proyecto.** La Gestión del proyecto es el arte de lograr un balance al gestionar objetivos, riesgos y restricciones para desarrollar un producto que sea acorde a los requisitos de los clientes y los usuarios. Los objetivos de este flujo de trabajo son:

    - Proveer un marco de trabajo para la gestión de proyectos de software intensivos
    - Proveer guías prácticas realizar planeación, contratar personal, ejecutar y monitorear el proyecto
    - Proveer un marco de trabajo para gestionar riesgos

La planeación de un proyecto posee dos niveles de abstracción: un plan para las fases y un plan para cada iteración.

- **Configuración y control de cambios.** Su finalidad es mantener la integridad de todos los artefactos que se crean en el proceso, así como la información del proceso evolutivo que han seguido

- **Entorno.** La finalidad de este flujo de trabajo es dar soporte al proyecto con las adecuadas herramientas, procesos y métodos. Brinda una especificación de las herramientas que se van a necesitar en cada momento, así como definir la instancia concreta del proceso que se va a seguir. En concreto las responsabilidades de este flujo de trabajo incluyen:

    - Selección y adquisición de herramientas
    - Establecer y configurar las herramientas para que se ajusten a la organización
    - Configuración del proceso
    - Mejora del proceso
    - Servicios técnicos

Con formato: Derecha: 0.04 cm, Punto de tabulación: 15.56 cm, Derecha + No en 15 cm

Con formato: Centrado, Posición: Horizontal: Centro, Con relación a: Margen

2.3 SISTEMAS OPERATIVOS

Sistema Operativo. Programa o conjunto de programas que permiten administrar los recursos de hardware y software de una computadora. En este caso se hablará del sistema operativo basado en ventanas siendo este la última versión **Windows XP**. Este sistema operativo es la mejora mas importante técnicamente desde Windows 9x, y unifica las versiones separadas que hubo estos años: WINDOWS 9x/ME para usuarios hogareños contra Windows NT/2000 para usuarios corporativos con requerimientos de trabajo en redes de alto nivel. En realidad es la continuación de Windows NT/2000. Se destaca en este producto su alto grado de integración con las redes e Internet, además de proveer una nueva interface gráfica que se hace notar ni bien se comienza a utilizar. Los cambios de interfaz son básicamente estéticos. La diferencia real con sus predecesores esta dada por el soporte LAN, software de grabación de discos compactos, multimedia, escritorio remoto y manejo de usuarios. Algo muy importante es el hecho de que Microsoft con esta versión de su sistema operativo. ha puesto especial énfasis en los controladores. Windows XP ahora es muchísimo mas renuente que sus predecesores a instalar controladores no certificados para el mismo. Con esto se pretende reducir al máximo las ya tan conocidas (y sufridas) "pantallas azules", aduciendo que la mayoría de las causas de inestabilidad de las versiones anteriores estaba dada por el uso de controladores no certificados, obsoletos o mal desarrollados. Se destaca la búsqueda inteligente que hace el sistema operativo. Al momento de instalar un dispositivo nuevo, escaneando unidades en busca de los controladores correctos. Las siguientes tablas muestran todas las características principales de Windows XP: El nuevo estándar informático de eficacia y fiabilidad pone en sus manos las herramientas avanzadas de productividad de la era digital: (véase siguiente tabla)

Con formato: Derecha: 0.04 cm, Punto de tabulación: 15.56 cm, Derecha + No en 15 cm

Con formato: Centrado, Posición: Horizontal: Centro, Con relación a: Margen

CARACTERÍSTICA	DESCRIPCIÓN	VENTAJA
Fiable		
Basado en el motor de Windows	Windows XP integra la base de códigos de Windows NT y Windows 2000, que presenta una arquitectura informática de 32 bits y un modelo de memoria completamente protegida.	Windows XP Professional ofrece una experiencia informática fiable a todos los usuarios comerciales.
Comprobador de controladores de dispositivos mejorado	Basado en el comprobador de controladores de dispositivos de Windows 2000, proporciona incluso mayores pruebas de carga para los controladores de dispositivos.	Los controladores de dispositivos que superan estas pruebas son los más sólidos disponibles, lo que garantiza la máxima estabilidad del sistema.
Soporte colateral de DLL	Proporciona un mecanismo para instalar y ejecutar colateralmente varias versiones de componentes individuales de Windows.	Esto ayuda a resolver el problema "DLL hell (demonio)", al permitir que una aplicación escrita y probada con una versión de un componente del sistema siga utilizando la misma versión.
Rendimiento		
Arquitectura multitarea preferente	Su diseño permite que varias aplicaciones se ejecuten simultáneamente, al tiempo que garantiza una gran respuesta y estabilidad del sistema.	Ejecución de las aplicaciones más exigentes con excelentes tiempos de respuesta del sistema.
Memoria escalable y soporte de procesador	Admite hasta 4 gigabytes (GB) de memoria RAM y hasta dos multiprocesadores simétricos.	Los usuarios que necesiten el más alto nivel de rendimiento podrán trabajar con el hardware más reciente.

Con formato: Derecha: 0.04 cm, Punto de tabulación: 15.56 cm, Derecha + No en 15 cm

Con formato: Centrado, Posición: Horizontal: Centro, Con relación a: Margen

CARACTERÍSTICA	DESCRIPCIÓN	VENTAJA
Seguro		
Sistema de cifrado de archivos (EFS) con soporte para varios usuarios	Cifra todos los archivos con una clave generada aleatoriamente. Los procesos de cifrado y descifrado son transparentes para el usuario. EFS permite que varios usuarios tengan acceso a un documento cifrado.	El más alto nivel de protección contra piratas informáticos y robo de datos.
Seguridad IP (IPSec)	Ayuda a proteger los datos transmitidos a través de una red. IPSec es una parte importante de la seguridad de las redes virtuales privadas (VPN), que permiten a las organizaciones transmitir datos de forma segura a través de Internet.	Los administradores de tecnologías de la información podrán crear redes virtuales privadas seguras con rapidez y facilidad.
Fácil de utilizar		
Entorno de usuario adaptable	Se adapta al método de trabajo de cada usuario. Las aplicaciones utilizadas más frecuentemente se muestran primero. Al abrir varios archivos en la misma aplicación, las ventanas abiertas se consolidan en un único botón de la barra de tareas. Para despejar el área de notificaciones, los elementos que no se utilizan quedan ocultos	Un entorno de trabajo más ordenado permite al usuario ser más eficiente. Los usuarios pueden buscar los datos y las aplicaciones importantes que necesiten con rapidez y facilidad. Todas estas configuraciones se pueden controlar a través de directivas, por lo que los administradores de tecnologías de la información pueden decidir cuáles son las características más adecuadas para su entorno.
Publicar información en el Web fácilmente	Se pueden publicar fácilmente archivos y carpetas en cualquier servicio Web que utilice el protocolo WebDAV.	Los usuarios podrán publicar información importante en servidores Web de la intranet de la empresa.

Con formato: Derecha: 0.04 cm, Punto de tabulación: 15.56 cm, Derecha + No en 15 cm

Con formato: Centrado, Posición: Horizontal: Centro, Con relación a: Margen

CARACTERÍSTICA	DESCRIPCIÓN	VENTAJA
Revoluciona el trabajo de los usuarios remotos		
Escritorio remoto	Permite a los usuarios crear sesiones virtuales en sus equipos de escritorio mediante el protocolo de escritorio remoto (RDP, Remote Desktop Protocol) de Microsoft.	Permite a los usuarios tener acceso a los datos y las aplicaciones alojados en sus equipos de escritorio desde otro equipo que ejecute Windows 95 o una versión posterior y que esté conectado a su máquina a través de una red.
Administrador de credenciales	Almacén protegido para la información de contraseñas. Permite que los usuarios escriban una vez sus nombres de usuario y contraseñas, y que a partir de entonces el sistema proporcione esta información en las visitas siguientes.	Los usuarios que no estén conectados a un dominio o que necesiten tener acceso a recursos de varios dominios sin relaciones de confianza, podrán tener acceso fácilmente a los recursos de la red.
Visualización sin conexión	Hace posible la visualización sin conexión de páginas Web con gráficos enteras.	Los clientes pueden ver las páginas Web importantes desconectados de Internet.
Administrador de sincronización	Permite a los usuarios comparar y actualizar archivos y carpetas sin conexión con los de la red.	Después de trabajar con archivos y carpetas sin conexión, los cambios se reproducen automáticamente en la red, lo que garantiza que la versión más reciente esté siempre disponible.
Soporte para redes inalámbricas	Proporciona acceso protegido, además de mejoras en el rendimiento, para redes inalámbricas.	Facilita notablemente las tareas a los usuarios que necesitan desplazarse entre redes inalámbricas y admite altos niveles de cifrado para mejorar la seguridad.

Con formato: Derecha: 0.04 cm, Punto de tabulación: 15.56 cm, Derecha + No en 15 cm

Con formato: Centrado, Posición: Horizontal: Centro, Con relación a: Margen

CARACTERÍSTICA	DESCRIPCIÓN	VENTAJA
NLA (Network Location Awareness)	Proporciona un servicio subyacente que permite al sistema operativo y a las aplicaciones determinar cuándo ha cambiado una máquina las ubicaciones de red.	Las aplicaciones y los servicios de red se pueden escribir para que reaccionen de forma inteligente cuando se detecte un cambio de ubicación en la red.
Asistentes para la configuración del acceso remoto más sencillos	Guían al usuario por los pasos de configuración del acceso remoto en una red o una red privada virtual (VPN).	Hacen más fácil la conexión con una red empresarial o con Internet.
Ayuda y soporte técnico		
Asistencia remota	Asistencia remota permite al usuario compartir el control de su equipo a través de una red o de Internet. Un administrador o un amigo pueden ver la pantalla del usuario y controlar el puntero y el teclado para ayudar a solucionar un problema técnico. Los departamentos de tecnologías de la información pueden crear soluciones personalizadas, en API publicadas con HTML, para adaptar Asistencia remota a sus necesidades.	Reduce el tiempo que los administradores de sistemas pasan en los escritorios de los usuarios. Muchas tareas administrativas y de solución de problemas se pueden realizar ahora desde los escritorios de los administradores.
Restauración de controladores de dispositivos	Al instalar ciertas clases de nuevos controladores de dispositivos, Windows XP Professional mantiene una copia del controlador antiguo, que se puede volver a instalar si se produce algún problema.	Si un controlador de dispositivos nuevo ocasiona un error de funcionamiento en Windows XP Professional, un administrador puede volver a instalar fácilmente el controlador anterior.

Con formato: Derecha: 0.04 cm, Punto de tabulación: 15.56 cm, Derecha + No en 15 cm

Con formato: Centrado, Posición: Horizontal: Centro, Con relación a: Margen

CARACTERÍSTICA	DESCRIPCIÓN	VENTAJA
Comunicación y redes		
Servidor de seguridad de conexión a Internet	Cliente de servidor de seguridad que protege empresas pequeñas de los ataques comunes en Internet.	Los propietarios de pequeñas empresas podrán conectar de forma segura sus equipos Windows XP Professional con Internet.
Asistente para configuración de red	El asistente recorre los pasos clave, como los archivos e impresoras compartidos, las conexiones con Internet compartidas y la configuración del servidor de seguridad de conexión a Internet.	Las pequeñas empresas pueden instalarse en la red de forma productiva y segura sin necesidad de expertos.
Conexión compartida a Internet (ICS)	Conecta la red de oficinas pequeñas con Internet mediante una conexión telefónica o de banda ancha. Proporciona servicios de traducción, direccionamiento y resolución de nombres de direcciones de red para que todos los equipos de la red de una empresa pequeña compartan una única conexión.	Los propietarios de pequeñas empresas pueden conectar los sistemas de sus equipos con Internet mediante una única conexión, con el resultado de ahorros potenciales en costo de ISP y hardware.

Con formato: Derecha: 0.04 cm, Punto de tabulación: 15.56 cm, Derecha + No en 15 cm

Con formato: Centrado, Posición: Horizontal: Centro, Con relación a: Margen

CARACTERÍSTICA	DESCRIPCIÓN	VENTAJA
Administración e implementación		
Compatibilidad con aplicaciones	A medida que se publiquen nuevas correcciones para aplicaciones, las actualizaciones estarán disponibles a través del servicio Windows Update.	Los usuarios disfrutarán de mayor compatibilidad con aplicaciones en comparación con Windows 2000 Professional y, a medida que se publiquen más correcciones de aplicaciones, estarán disponibles automáticamente.
Herramienta de migración de estados de usuario	Ayuda a los administradores a migrar los datos de usuario y la configuración de aplicaciones/sistema operativo desde un equipo antiguo a uno con Windows XP Professional.	Después de una migración, los administradores de tecnologías de la información pueden reducir el número de llamadas de ayuda y los usuarios finales sufrirán menos periodos de inactividad, ya que conservarán el entorno operativo con el que están familiarizados.
Instalación desatendida	Permite especificar un mayor número de opciones que las versiones anteriores de Windows y permite disponer de un mayor grado de seguridad mediante el cifrado de contraseñas en los archivos de respuesta.	Permite a los administradores un mayor grado de flexibilidad en las secuencias de comandos de rutinas de configuración desatendidas, lo que garantiza las opciones de instalación más flexibles disponibles actualmente.

Con formato: Derecha: 0.04 cm, Punto de tabulación: 15.56 cm, Derecha + No en 15 cm

Con formato: Centrado, Posición: Horizontal: Centro, Con relación a: Margen

CARACTERÍSTICA	DESCRIPCIÓN	VENTAJA
Kit de administración de Internet Explorer 6	El Kit de administración de Internet Explorer (IEAK 6) permite personalizar más Internet Explorer 6, lo que facilita la implementación y el mantenimiento del explorador.	Los administradores disponen de un mayor grado de flexibilidad en la implementación de Internet Explorer 6.
Herramienta de preparación del sistema (SysPrep)	SysPrep ayuda a los administradores a clonar configuraciones de equipo, sistemas y aplicaciones. Una única imagen, que incluya el sistema operativo y aplicaciones comerciales, se puede restaurar para varias configuraciones de equipos diferentes.	SysPrep permitirá a los administradores reducir el número de imágenes que mantienen del sistema operativo, además de reducir el tiempo empleado en implementar un sistema de escritorio típico.
Administrador de instalación	Asistente gráfico que guía a los administradores en el diseño de comandos de instalación.	Ahora es más fácil crear archivos de respuesta para instalaciones desatendidas.
Instalación remota del sistema operativo	Windows XP Professional se puede instalar a través de la red (incluidas las imágenes SysPrep). **Nota:** esta característica requiere el servicio Active Directory™.	Ahorra tiempo y reduce el costo de implementación al permitir que los administradores estandaricen los entornos de escritorio para que cumplan los requisitos organizacionales.
Instrumental de administración de Windows (WMI)	Proporciona una infraestructura estándar para la supervisión y administración de los recursos del sistema.	Permite a los administradores supervisar y controlar el sistema a través de secuencias de comandos y aplicaciones de otros fabricantes.
Consola de administración de Microsoft (MMC)	Proporciona un entorno centralizado y coherente para las herramientas de administración.	Los administradores de tecnologías de la información podrán crear consolas de aplicación personalizadas.

Con formato: Derecha: 0.04 cm, Punto de tabulación: 15.56 cm, Derecha + No en 15 cm

Con formato: Centrado, Posición: Horizontal: Centro, Con relación a: Margen

2.4 CARACTERISTICAS, VENTAJAS, DESVENTAJAS DE LA PLATAFORMA JAVA

Características

La plataforma básica Java es la plataforma mínima en la que los desarrolladores pueden confiar para ejecutar cualquier aplicación o "applet" de Java. Está orientada a las redes, computadoras de escritorio y estaciones de trabajo. Incluye la máquina virtual y un subgrupo de todos los APIs requeridos para ejecutar aplicaciones llamado Java Applet API o Java Base API. Las aplicaciones con estos subgrupos pueden confiar en que sus programas serán ejecutados en cualquier ordenador sin incorporar grupos de directorios adicionales. Existe otro grupo de APIs llamado Standard Extension API y que ha sido definido por Javasoft en cooperación con otras empresas para extender las funciones básicas (véase Fig. 2.41).

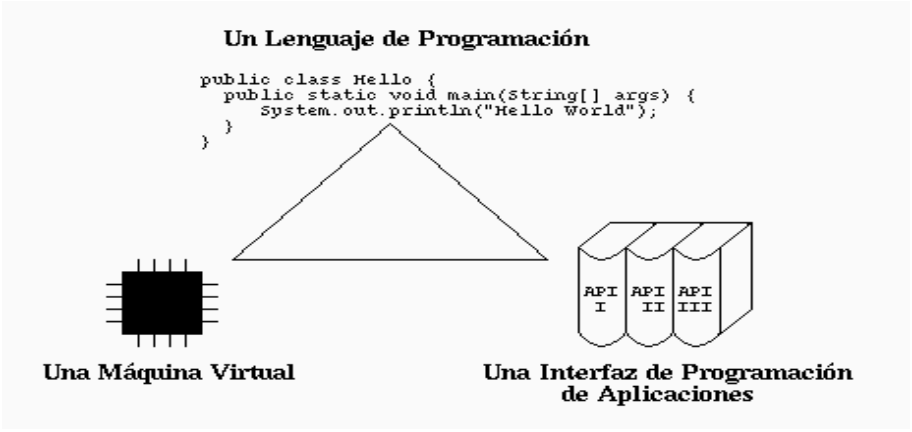


FIGURA 2.4.1 LA PLATAFORMA VIRTUAL DE JAVA

Con formato: Derecha: 0.04 cm, Punto de tabulación: 15.56 cm, Derecha + No en 15 cm

Con formato: Centrado, Posición: Horizontal: Centro, Con relación a: Margen

Arquitectura neutral

Para establecer Java como parte integral de la red (véase figura 2.4.2), el compilador Java compila su código a un fichero objeto de formato independiente de la arquitectura de la máquina en que se ejecutará. Cualquier máquina que tenga el sistema de ejecución (run-time) puede ejecutar ese código objeto, sin importar en modo alguno la máquina en que ha sido generado. Actualmente existen sistemas run-time para Solaris 2.x, SunOs 4.1.x, Windows 95, Windows NT, Linux, Irix, Aix, Mac, Apple.

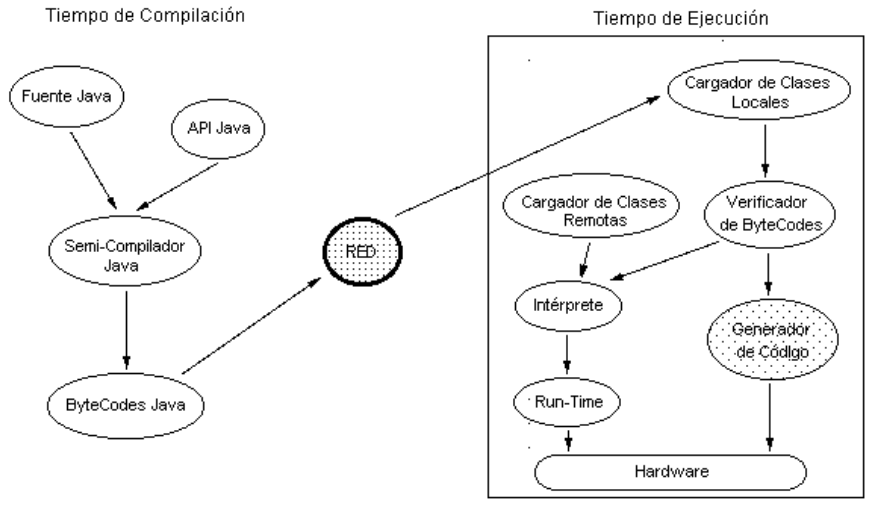


FIGURA 2.4.2 ARQUITECTURA NEUTRAL

Java es una plataforma virtual que provee:

- **El lenguaje Java.** Un lenguaje de programación de alto nivel
- **La Java Virtual Machine.** Una máquina virtual con su propio conjunto de instrucciones
- **La Java Platform API.** Una interfaz para la programación de aplicaciones

Con formato: Derecha: 0.04 cm, Punto de tabulación: 15.56 cm, Derecha + No en 15 cm

Con formato: Centrado, Posición: Horizontal: Centro, Con relación a: Margen

El código fuente Java se compila a un código de bytes de alto nivel independiente de la máquina. Este código (byte-codes) está diseñado para ejecutarse en una máquina hipotética que es implementada por un sistema run-time, que sí es dependiente de la máquina. Representación de los elementos que forman parte de la arquitectura de Java sobre una plataforma genérica, (véase la figura 2.4.3).

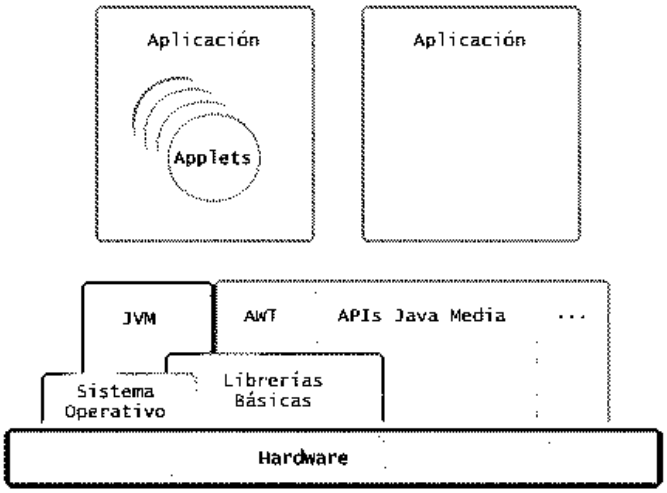


FIGURA 2.4.3 PLATAFORMA GENÉRICA

En ella podemos ver que lo verdaderamente dependiente del sistema es la JVM (Java Virtual Machine - Máquina Virtual Java) y las librerías fundamentales, que también nos permitirían acceder directamente al hardware de la máquina. Además, habrá API's (Application Programming Interface – Interfaz de Programación de Aplicaciones) de Java que también entren en contacto directo con el hardware y serán dependientes de la máquina, como ejemplo de este tipo de APIs podemos citar:

- **Java 2D**. Gráficos 2D y manipulación de imágenes
- **Java Media Framework**. Elementos críticos en el tiempo: audio, video...
- **Java Animation.** Animación de objetos en 2D

Con formato: Derecha: 0.04 cm, Punto de tabulación: 15.56 cm, Derecha + No en 15 cm

Con formato: Centrado, Posición: Horizontal: Centro, Con relación a: Margen

- **Java Telephony**: Integración con telefonía
- **Java Share**. Interacción entre aplicaciones multiusuario
- **Java 3D.** Gráficos 3D y su manipulación

El lenguaje Java es orientado a objetos

Trabaja con sus datos como objetos y con interfaces a esos objetos. Soporta las características propias del paradigma de la orientación a objetos: encapsulación, clases, métodos, subclases, enlace dinámico, encapsulamiento, herencia y polimorfismo. Las plantillas de objetos son llamadas, clases y sus copias, instancias. Estas instancias, necesitan ser construidas y destruidas en espacios de memoria. Incorpora funcionalidades como la resolución dinámica de métodos. Java mediante una interfaz específica llamada RTTI (RunTime Type Identification - Identificación de tipos en tiempo de corrida), que define la interacción entre objetos excluyendo variables de instancias o implementación de métodos. Las clases en Java tienen una representación en el run-time (tiempo de corrida), que permite a los programadores interrogar por el tipo de clase y enlazar dinámicamente la clase con el resultado de la búsqueda.

Seguro

El código Java pasa muchas pruebas antes de ejecutarse en una máquina. El código se pasa a través de un verificador de byte-codes que comprueba el formato de los fragmentos de código y aplica un probador de teoremas para detectar fragmentos de código ilegal-código que falsea punteros, viola derechos de acceso sobre objetos o intenta cambiar el tipo o clase de un objeto. Si los byte-codes pasan la verificación sin generar ningún mensaje de error, entonces sabemos que:

- El código no produce desbordamiento de operadores en la pila
- El tipo de los parámetros de todos los códigos de operación son conocidos y correctos
- No ha ocurrido ninguna conversión ilegal de datos, tal como convertir enteros en punteros

Con formato: Derecha: 0.04 cm, Punto de tabulación: 15.56 cm, Derecha + No en 15 cm

Con formato: Centrado, Posición: Horizontal: Centro, Con relación a: Margen

- El acceso a los campos de un objeto se sabe que es legal: pública, privada y protegida
- No hay ningún intento de violar las reglas de acceso y seguridad establecidas

El Cargador de Clases también ayuda a Java a mantener su seguridad, separando el espacio de nombres del sistema de ficheros locales, de los recursos procedentes de la red. Las clases importadas de la red se almacenan en un espacio de nombres privado, asociado con el origen. Cuando una clase del espacio de nombres privado accede a otra clase, primero se busca en las clases predefinidas (del sistema local) y luego en el espacio de nombres de la clase que hace la referencia. Esto imposibilita que una clase suplante a una predefinida.

Portable

Más allá de la portabilidad básica por ser de arquitectura independiente, Java implementa otros estándares de portabilidad para facilitar el desarrollo. Además, Java construye sus interfaces de usuario a través de un sistema abstracto de ventanas de forma que las ventanas puedan ser implantadas en entornos Unix, Pc o Mac.

Distribuido

Java se ha construido con extensas capacidades de interconexión TCP/IP (Protocolo de control de transmisión/Protocolo de Internet). Existen librerías de rutinas para acceder e interactuar con protocolos como HTTP (Protocolo de transferencia de hipertexto) y FTP (Protocolo de transferencia de archivos). Esto permite a los programadores acceder a la información a través de la red con tanta facilidad como a los ficheros locales. Java en sí no es distribuido, sino que proporciona las librerías y herramientas para que los programas puedan ser distribuidos, es decir, que se corran en varias máquinas, interactuando.

Con formato: Derecha: 0.04 cm, Punto de tabulación: 15.56 cm, Derecha + No en 15 cm

Con formato: Centrado, Posición: Horizontal: Centro, Con relación a: Margen

Dinámico

Java se beneficia todo lo posible de la tecnología orientada a objetos. Java no intenta conectar todos los módulos que comprenden una aplicación hasta el tiempo de ejecución (véase figura 2.4.4). Las librerías nuevas o actualizadas no paralizarán las aplicaciones actuales.

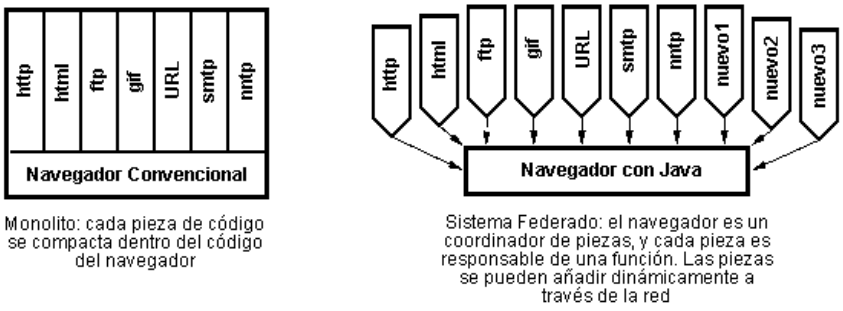


FIGURA 2.4.4 MÓDULOS DINÁMICOS

Java también simplifica el uso de protocolos nuevos o actualizados. Si el sistema ejecuta una aplicación Java sobre la red y encuentra una pieza de la aplicación que no sabe manejar, Java es capaz de traer automáticamente cualquiera de esas piezas que el sistema necesita para funcionar. Evita que los módulos de bytecode, los objetos o nuevas clases tengan que estar trayéndolos de la red cada vez que se necesiten, implementa las opciones de persistencia, para que no se eliminen cuando se limpie la memoria caché de la máquina.

Multiproceso (Multithread)

Al ser multi-proceso (multithreaded), Java permite muchas actividades simultáneas en un programa. Los threads (a veces llamados procesos ligeros), son básicamente pequeños procesos o piezas independientes de un gran proceso. Al estar los procesos construidos en el lenguaje, son más fáciles de usar y más robustos que sus homólogos en C o C++. El beneficio de ser multiproceso

Con formato: Derecha: 0.04 cm, Punto de tabulación: 15.56 cm, Derecha + No en 15 cm

Con formato: Centrado, Posición: Horizontal: Centro, Con relación a: Margen

consiste en un mejor rendimiento interactivo y mejor comportamiento en tiempo real. Aunque el comportamiento en tiempo real está limitado a las capacidades del sistema operativo subyacente (Unix, Windows, etc.), aún supera a los entornos de flujo único de programa (single-threaded) tanto en facilidad de desarrollo como en rendimiento.

Simple

- Sintaxis familiar a la de C y C++
- Se eliminan las características complejas de C y C++
- No hay aritmética de punteros
- No hay herencia múltiple

Robusto

- Java siempre verifica los índices
- Java hace recolección automática de basuras
- Java no tiene aritmética de punteros

Multiplataforma

La plataforma Java se incrusta en plataformas reales como:

- Windows'95 y 'NT
- Power/Mac
- Unix (Solaris, Linux, ...)

La API de la Plataforma Java consiste en una biblioteca de clases que provee de:

- Manejo de archivos
- Interfaces gráficas
- Acceso a la red Internet
- Acceso a bases de datos

Con formato: Derecha: 0.04 cm, Punto de tabulación: 15.56 cm, Derecha + No en 15 cm

Con formato: Centrado, Posición: Horizontal: Centro, Con relación a: Margen

Ventajas de la plataforma Java

Implementación de la Plataforma Java

- Un compilador de Java para traducir código fuente al conjunto de instrucciones de la máquina virtual de Java
- Un intérprete o un compilador JIT para ejecutar las instrucciones de la máquina virtual
- Una implementación de la API de Java

Liberación explícita de la memoria

- La proporción aumenta a medida que la complejidad aumenta: para mil líneas es fácil, para diez mil es difícil, para cien mil líneas es imposible
- El mantenimiento de los programas es aún más difícil

En Java los programadores son más productivos porque:

- Se necesita programar menos pues hay abundantes bibliotecas de clases
- No se necesita reescribir los programas cuando se cambia la plataforma
- No se pierde tiempo liberando explícitamente la memoria
- Se pierde menos tiempo depurando los programas
- Java es compilado

Algunas sugerencias para adoptar Java como tecnología corporativa, serían:

- **Usar Java en el desarrollo de la interfase del cliente**; Java es suficientemente estable para desarrollar una interface portable. Utilizar herramientas de programación más estables en los servidores, porque son la parte crítica
- **Portar o crear un servidor no-crítico en Java**, de forma que tanto cliente como servidor estén escritos en Java

Con formato: Derecha: 0.04 cm, Punto de tabulación: 15.56 cm, Derecha + No en 15 cm

Con formato: Centrado, Posición: Horizontal: Centro, Con relación a: Margen

- **Utilizar Java en proyectos de envergadura** tanto en el cliente como en el servidor, para valorar la efectividad de Java

La Intranet está creciendo actualmente más rápido que Internet. Las organizaciones corporativas están adoptando la metodología Internet para proporcionar soluciones a sus usuarios y clientes. Java tiene todas las cartas para ser una herramienta de inestimable valor en el desarrollo de aplicaciones corporativas. Java es la mejor plataforma para suministrar servidores de aplicaciones al mercado por estas razones:

- **Es una mejora de los lenguajes tradicionales de tercera generación**. Java es más simple y de más nivel que el lenguaje C++ y permite, además, enfrentarse fácilmente a complejos problemas de programación
- **Es mejor que los lenguajes de cuarta generación**, ya que incluye diversas clases de librerías y puede extenderse fácilmente con reglas de negocio reutilizables al más alto nivel. Es universal. Donde hay un cliente conectado a la red, Java está ahí
- **Está maduro**, tanto a nivel de velocidad como de rendimiento con un entorno de ejecución y desarrollo potente y flexible
- **No permite el manejo directo** del hardware ni de la memoria inclusive no permite modificar valores de punteros; de modo que se puede decir que es virtualmente imposible colgar un programa Java
- **El intérprete siempre tiene el control**. Inclusive el compilador es suficientemente inteligente como para no permitir usar variables sin inicializarlas, modificar valores de punteros directamente, acceder a métodos o variables en forma incorrecta, utilizar herencia múltiple, etc.
- **También está diseñado específicamente para trabajar sobre una red**, de modo que incorpora objetos que permiten acceder a archivos en forma remota. Además, con el JDK (Java Development Kit: Herramientas de desarrollo de Java) vienen incorporadas muchas herramientas, entre ellas un generador automático de documentación que, con un poco de atención

Con formato: Derecha: 0.04 cm, Punto de tabulación: 15.56 cm, Derecha + No en 15 cm

Con formato: Centrado, Posición: Horizontal: Centro, Con relación a: Margen

al poner los comentarios en las clases, crea inclusive toda la documentación de las mismas en formato HTML

- **Java es un lenguaje interpretado**. Al compilar un programa Java, lo que genera es un seudo código definido por la compañía Sun, para una máquina genérica. Luego, al correr sobre una máquina dada, el software de ejecución Java simplemente interpreta las instrucciones, emulando a dicha máquina genérica. El intérprete Java (run-time) puede ejecutar directamente el código objeto. Enlazar un programa, normalmente, consume menos recursos que compilarlo, por lo que los desarrolladores con Java pasarán más tiempo desarrollando y menos esperando por el ordenador. No obstante, el compilador actual del JDK es bastante lento

Para conseguir ser un lenguaje independiente del sistema operativo y del procesador que incorpore la máquina virtual utilizada, es tanto interpretado como compilado. Esto no es ningún contrasentido, el código fuente escrito con cualquier editor se compila generando el byte-code. Este código intermedio es de muy bajo nivel, pero sin alcanzar las instrucciones de máquina propias de cada plataforma. El byte-code corresponde al 80% de las instrucciones de la aplicación. Ese mismo código es el que se puede ejecutar sobre cualquier plataforma. Para ello hace falta el run-time, que sí es completamente dependiente de la máquina y del sistema operativo, que interpreta dinámicamente el byte-code y añade el 20% de instrucciones que faltaban para su ejecución. No sufre pérdidas de memoria

- No sufre errores de acceso a la memoria
- Con la integración de múltiples componentes, los problemas pueden diagnosticarse más fácilmente
- Ha sido diseñado desde el principio teniendo en cuenta la Web
- Es rápido. La llegada de nuevas máquinas virtuales y tecnologías ha permitido a Java reducir la diferencia de ejecución de aplicaciones

Desarrollo de software basado en componentes, sin las limitaciones de otros entornos, lo que permite ahorrar costos e incrementar la fiabilidad. Disminución en

Con formato: Derecha: 0.04 cm, Punto de tabulación: 15.56 cm, Derecha + No en 15 cm

Con formato: Centrado, Posición: Horizontal: Centro, Con relación a: Margen

los costos finales de los proyectos, por el menor costo de los ciclos de desarrollo, y por la facilidad de desplegar el software en cualquier plataforma. La plataforma Java proporciona ventajas para los departamentos de administración de sistemas de la compañía. Los controles de versiones y actualizaciones se simplifican ya que las aplicaciones Java son capaces de mantenerse en un lugar centralizado a través del cuál se puede distribuir a cada usuario lo que necesite usar. En aquellos entornos con aplicaciones de varios fabricantes que necesitan ser ejecutadas en múltiples plataformas, el número de éstas que necesitan ser soportadas se reduce a una. Cuando las aplicaciones se ejecutan en una plataforma Java, las compañías pueden aprovechar la interactividad de Internet, transfiriendo tareas normalmente realizadas por los empleados, a sus clientes y proveedores. Así, pueden reducir el tiempo invertido en introducir pedidos al suministrar a sus clientes formularios a través de la Web que puedan rellenar directamente. Esto es posible por el uso de estándares, dado que los clientes serán capaces de utilizar formularios independientemente del sistema operativo utilizado por su ordenador. La plataforma Java ofrece múltiples ventajas, no sólo para los usuarios finales sino también para los desarrolladores y personal de soporte.

Ventajas para el usuario

La plataforma Java ofrece la posibilidad de acceder a aplicaciones a través de la Web con acceso instantáneo a los programas. Desde el principio, las aplicaciones están disponibles para su ejecución en cualquier sistema operativo liberando a los usuarios de tener que elegir un cierto sistema operativo. Con Java, los más pequeños sistemas pueden ser utilizados para ejecutar aplicaciones especializadas.

Ventajas para el desarrollador

El lenguaje Java es un conjunto de APIs reducido, bien conocido y documentado. Los desarrolladores pueden escribir sus aplicaciones sólo una vez para ejecutarlas en cualquier lugar. Además, los entornos de desarrollo Java generan un único

Con formato: Derecha: 0.04 cm, Punto de tabulación: 15.56 cm, Derecha + No en 15 cm

Con formato: Centrado, Posición: Horizontal: Centro, Con relación a: Margen

código binario en cualquier sistema operativo. En vez de desarrollar aplicaciones en múltiples plataformas, los desarrolladores son capaces de construir sus aplicaciones así como proporcionar una aplicación para una única plataforma instalada en todos los equipos. Además de desarrollar aplicaciones orientadas a objetos reutilizables y compartidos, reduce los costes de producción permitiendo a los desarrolladores concentrarse en la creación de objetos para resolver problemas inesperados. Se compila la aplicación una única vez y los ejecutables en byte-code obtenidos son válidos para cualquier plataforma. El código fuente queda a salvo.

Desventajas del lenguaje Java

- **Compatibilidad**. Aunque Java es compatible con muchas plataformas, no lo es con todas, entonces, si su plataforma de hardware y software no es compatible con Java, para poder hacer uso del mismo, deberá cambiarse de plataforma. Además puede que el sistema operativo o navegador no sea compatible con la última versión de Java, por lo tanto deberá asegurarse de tener actualizada la plataforma
- **Rendimiento.** Java puede ser que no alcance las exigencias de rendimiento de las aplicaciones en las que la velocidad de la ejecución es de una vital importancia
- **Aprendizaje**. Sí bien Java es un lenguaje simple, se necesita un cierto tiempo de aprendizaje para aprovechar el potencial que brinda Java y desarrollar sus aplicaciones
- **Plug-ins** (programa de complementos) son herramientas válidas que permiten incorporar características nuevas al navegador sin necesidad de que el vendedor conceda su permiso
- **Agentes.** Son utilidades que se pueden emplear para buscar material en la intranet y hacer que envíen un aviso cuando haya nueva información específica disponible
- **Velocidad.** Evidentemente la interpretación o incluso compilación just-in-time (justo a tiempo) del byte-code produce aplicaciones más lentas que en

Con formato: Derecha: 0.04 cm, Punto de tabulación: 15.56 cm, Derecha + No en 15 cm

Con formato: Centrado, Posición: Horizontal: Centro, Con relación a: Margen

el caso de la ejecución directa de un binario. El recolector de basura puede suponer una sobrecarga adicional al procesador

- **La generalidad** tiene como inconveniente que no se aprovecha totalmente la potencia de la máquina y del sistema operativo. Por ejemplo, el aspecto de una aplicación Java puede resultar simple y poco atractivo en comparación con las aplicaciones nativas
- **Los programas en Java se ejecutan más lento** que sus equivalentes en C o C++
- **La plataforma estándar** ocupa como mínimo 16 MB en memoria principal y unos 30 MB en disco
- **Corre bien** en Pentium 166MHz con 32 MB
- **La programación de applets** es frágil

Applets

Una applet (mini aplicación) es un pequeño programa en Java transferido dinámicamente a través de Internet. Presentan un comportamiento inteligente, pudiendo reaccionar a la entrada de un usuario y cambiar de forma dinámica. Java cambia en este aspecto, haciendo posible que los programadores ejerzan un control sobre los programas ejecutables de Java que no es posible encontrar en otros lenguajes. Al contrario de lo que ocurre en una aplicación al uso, un applet no puede ser ejecutado directamente desde el sistema operativo. Gracias a la tecnología de OLE (Object Linking and Embedding - Encapsulación y Enlace de Objetos lógicos), los applets tienen cada vez mayor prevalencia como herramienta. Un applet bien diseñado puede ser llamado y ejecutado desde muchas aplicaciones diferentes.

La lógica de aplicación se implementa en una librería de servlets que se ejecutan en el servidor del JLE. Estos servlets son el Back-End de un Java Web Server que actúa como Front-End de la aplicación. Este servidor Web se configura para que sea un servidor seguro garantizando en cierta medida la seguridad de los datos intercambiados, como puede ser la cuenta de acceso y la contraseña del usuario,

Con formato: Derecha: 0.04 cm, Punto de tabulación: 15.56 cm, Derecha + No en 15 cm

Con formato: Centrado, Posición: Horizontal: Centro, Con relación a: Margen

o los contenidos de los cursos. La información viaja cifrada por la red evitando los ataques de usuarios malintencionados que pretendan apoderarse o alterar la información de los contenidos, o que pretendan suplantar la personalidad de algún usuario legal del sistema. La interfaz de usuario está diseñada para poder ser usada desde cualquier navegador de WWW (World Wide Web – red mundial) y esta basada mayoritariamente en: formularios HTML, Applets Java y plug-ins específicos para presentar contenidos especiales. Los applets son más difíciles de programar que los scripts en Javascript y requerirán unos conocimientos básicos o medios del lenguaje Java. La principal ventaja de utilizar applets consiste en que son mucho menos dependientes del navegador que los scripts en Javascript, incluso independientes del sistema operativo del PC donde se ejecutan. Además, Java es más potente que Javascript, por lo que el número de aplicaciones de los applets podrá ser mayor. Como desventajas en relación con Javascript cabe señalar que los applets son más lentos de procesar y que tienen espacio muy delimitado en la página donde se ejecutan, es decir, no se mezclan con todos los componentes de la página ni tienen acceso a ellos. Es por ello que con los applets de Java no podremos hacer directamente cosas como abrir ventanas secundarias, controlar Frames, formularios, capas, etc. En los últimos tiempos, los creadores de páginas Web tienden a mejorar el aspecto de las mismas empleando todo tipo de componentes multimedia. El objetivo último que persiguen es conseguir efectos de animación e interactividad entre ellas. Hoy en día el uso de Applets se ha concentrado principalmente en dos tipos de diseños:

- **Información en Tiempo Real** : Este tipo de aplicación se refiere aquella que requiere ser colocada dentro de un navegador y debe ser actualizada constantemente, en este caso el Applet contiene la lógica necesaria para conectarse a una Base de Datos y/o rediseñar su propia interfase, lo anterior sin la necesidad de navegar a lo largo de varios documentos HTML / XHTML, o bien, recargar una página

Con formato: Derecha: 0.04 cm, Punto de tabulación: 15.56 cm, Derecha + No en 15 cm

Con formato: Centrado, Posición: Horizontal: Centro, Con relación a: Margen

- **Manipulación de Gráficas**: Mediante un Applet también es posible llevar acabo la manipulación excéntrica de gráficas para ser colocadas dentro del diseño de páginas HTML / XHTML

Javascript

Es una versión de Java directamente interpretada, que se incluye como parte de una página HTML, lo que lo hace muy fácil y cómodo para aplicaciones muy pequeñas, pero que en realidad tiene muchas limitaciones:

- No soporta clases ni herencia
- No se precompila
- No es obligatorio declarar las variables
- Verifica las referencias en tiempo de ejecución
- No tiene protección del código, ya que se baja en ASCII
- No todos los navegadores lo soportan completamente

Se trata de un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento. Gracias a su compatibilidad con la mayoría de los navegadores modernos, es el lenguaje de programación del lado del cliente más utilizado. Con Javascript podemos crear efectos especiales en las páginas y definir interactividades con el usuario. El navegador del cliente es el encargado de interpretar las instrucciones Javascript y ejecutarlas para realizar estos efectos e interactividades, de modo que el mayor recurso, y tal vez el único, con que cuenta este lenguaje es el propio navegador. Javascript es el siguiente paso, después del HTML, que puede dar un programador de la Web que decida mejorar sus páginas y la potencia de sus proyectos. Entre las acciones típicas que se pueden realizar en Javascript tenemos dos vertientes. Por un lado los efectos especiales sobre páginas Web, para crear contenidos dinámicos y elementos de la página que tengan movimiento, cambien de color o cualquier otro dinamismo. Por el otro, javascript nos permite ejecutar instrucciones como respuesta a las acciones del usuario, con lo que podemos crear páginas interactivas con

Con formato: Derecha: 0.04 cm, Punto de tabulación: 15.56 cm, Derecha + No en 15 cm

Con formato: Centrado, Posición: Horizontal: Centro, Con relación a: Margen

programas como calculadoras, agendas, o tablas de cálculo. También nos permite la programación de pequeños scripts, pero también de programas más grandes, orientados a objetos, con funciones, estructuras de datos complejas, etc. Además, Javascript pone a disposición del programador todos los elementos que forman la página Web, para que éste pueda acceder a ellos y modificarlos dinámicamente.

Scripts

Un lenguaje script permite incorporar código fuente para la programación del lado cliente, directamente en la página HTML, y el plug-in que interpreta ese lenguaje se activará automáticamente cuando se cargue en el navegador. Se cargan muy rápidamente porque van incluidos en la página que envía el servidor. El inconveniente es que el código del programador está expuesto a la vista de cualquiera, aunque tampoco se pueden hacer demasiadas filigranas con un lenguaje script. Estos lenguajes se utilizan fundamentalmente para hacer más atractivos las interfaces gráficas de las páginas, porque disponen de elementos gráficos y pueden llegar a resolver el 80% de los problemas que se plantean en la programación de la parte cliente del sistema cliente-servidor. Cuando se pide una página Web, se reciben ficheros gráficos, el código de la página (con o sin scripts), código Java precompilado y componentes ActiveX (controles). Java está diseñado para ejecutar los applets en una caja cerrada (sand box), envuelta en una capa de seguridad que impide cualquier acceso a disco o a memoria que se encuentre fuera de esa caja cerrada. Así que, la parte más propensa a introducir elementos no deseados en la máquina cliente son los componentes ActiveX. Diseñar la aplicación para que esté distribuida entre servidores y clientes, y la línea de partida debe ser el diseño modular.

Con formato: Derecha: 0.04 cm, Punto de tabulación: 15.56 cm, Derecha + No en 15 cm

Con formato: Centrado, Posición: Horizontal: Centro, Con relación a: Margen

2.5 BASES DE DATOS RELACIONALES

A través de los años los sistemas de administración de Bases de Datos han evolucionado hacia Sistemas de Administración de Base de Datos Relacionales (RDBMS). Una base de datos relacional es un modelo organizado de entidades que posee características que tienen las relaciones entre ellas. Una base de datos relacional bien diseñada provee información de un negocio o un proceso y su uso más común es para almacenar y recuperar información. Entre las mayores ventajas de RDBMS están la forma en la que almacena y recupera información y cómo mantiene la integridad de la misma. Las estructuras RDBMS son fáciles de comprender y construir, pues son lógicamente representadas utilizando Diagramas Entidad-Relación. Las bases de datos relacionales tienen las siguientes características principales:

- **Estructuras**. Son objetos que almacenan o acceden a los datos de la base de datos (Tablas, vistas e índices)
- **Operaciones**. Son acciones usadas para definir las Estructuras o manipular los datos de las mismas (SELECT, CREATE)
- **Reglas de integridad**. Gobiernan los tipos de acciones permitidas en los datos y la estructura de la BD. Protegen los datos y estructuras de la BD. (Llaves primarias y foráneas)

Modelo lógico

En la fase de diseño de un ciclo de desarrollo se crea un modelo lógico de BD llamado Entidad-Relación (ER). Un modelo ER es visual y muestra la estructura, características e interacción entre los datos que están siendo modelados. Este modelo ER contiene:

- **Entidades y Atributos**. Una Entidad en un modelo lógico es como un sustantivo en gramática: una persona, un lugar o cosa. Las características de una entidad son conocidas como sus Atributos. Un atributo es

Con formato: Derecha: 0.04 cm, Punto de tabulación: 15.56 cm, Derecha + No en 15 cm

Con formato: Centrado, Posición: Horizontal: Centro, Con relación a: Margen

información detallada de una entidad que la califica, identifica, clasifica o cuantifica. Existen además atributos obligatorios y opcionales

- **Relaciones e identificadores únicos**. Las relaciones entre las entidades representan la dependencia entre ellas. Dichas relaciones se representan utilizando líneas que apuntan a las entidades e indican el tipo de relación. Se utilizan líneas continuas para relaciones identificantes (La entidad A hereda Atributo A a la entidad B como parte de su llave primaria) y líneas discontinuas para relaciones no identificantes (La entidad C hereda Atributo C a la entidad D como llave foránea). Debe ser posible además identificar en forma única a cada registro o elemento de una entidad (Llave primaria) así como la relación de atributos heredados de otras entidades (Llaves foráneas), ver figura 2.5.1

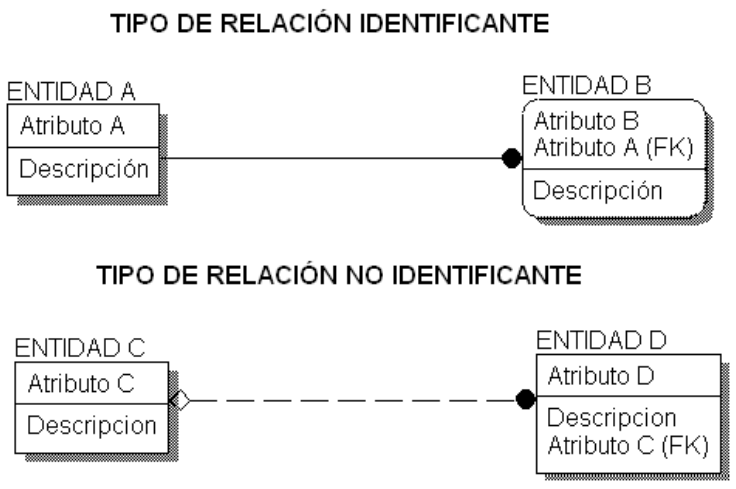


FIGURA 2.5.1 TIPOS DE RELACIONES

- **Cardinalidad** en las relaciones
- **Uno a uno**. Cada elemento de una entidad es representada con un elemento único en otra entidad

Con formato: Derecha: 0.04 cm, Punto de tabulación: 15.56 cm, Derecha + No en 15 cm

Con formato: Centrado, Posición: Horizontal: Centro, Con relación a: Margen

- **Uno a muchos**. Cada elemento de una entidad puede ser representada por muchos elementos en otra entidad
- **Muchos a muchos**. Un elemento de una entidad es representado por uno o mas elementos en otra entidad y cada elemento de la segunda entidad puede ser representada por uno o mas elementos de la primera entidad

Normalización

Cuando se han establecido las relaciones entre entidades, se debe realizar la normalización del diseño, esto es, eliminar la información redundante de las entidades. Las reglas principales para normalización son las siguientes:

- **Primera Forma Normal (FN1)**. No debe haber atributos con nombre repetido. Si esto sucede se debe crear una entidad hijo y relacionarla con la primera
- **Segunda Forma Normal (FN2)**. Un atributo debe ser dependiente de su llave primaria completa, no solo de una parte de la misma
- **Tercera Forma Normal (FN3)**. No debe haber atributos que dependen de otros que no son llave

Modelo físico

Se crea un modelo físico utilizando el modelo lógico de la BD para crear una Base de Datos y los objetos de la misma representados por las entidades y las relaciones. En el modelo físico, cada entidad se convierte en una tabla y los atributos en columnas de la misma. Las relaciones entre las entidades son parte de una o más restricciones (constraints) entre las tablas. La implementación física podría requerir de combinar, separar o crear nuevas entidades, así como crear procedimientos almacenados (stored procedures), funciones y disparadores (triggers) para cumplir las reglas del negocio.

Con formato: Derecha: 0.04 cm, Punto de tabulación: 15.56 cm, Derecha + No en 15 cm

Con formato: Centrado, Posición: Horizontal: Centro, Con relación a: Margen

Structured Query Language (SQL)

E.F. Codd propuso el modelo relacional como una nueva forma de representar la información para su tratamiento. Con esto sentó las bases de un lenguaje algebraico llamado SQUARE para la manipulación de datos. Este lenguaje fue mejorándose y ampliándose hasta que se denominó SEQUEL y, por último en 1976, SQL. Durante esa década tanto el modelo relacional como el lenguaje propuesto por Codd estuvieron siendo validados, mientras tanto, una serie de firmas empezaron a comercializar productos basados en los mismos. En los primeros años de la década de los 80 IBM comercializó su primer producto relacional, el SQL/DS y el DB2, mas adelante; fue en esta década cuando ANSI publicó los primeros estándares relacionales.

Álgebra relacional

SQL es un lenguaje de manipulación de datos basado en el lenguaje algebraico al cual se le ha añadido una semántica que lo hace más próximo al lenguaje natural. SQL utiliza una serie de operadores (algebraicos) que actúan sobre las relaciones o tablas de un esquema relacional. En una operación SQL pueden intervenir una o varias tablas y uno o varios operadores algebraicos. Cada operador puede operar sobre una o dos tablas pero siempre sobre la totalidad de registros que forman la tabla. El resultado de una operación SQL es una tabla, que a su vez, puede ser sometida a nuevas operaciones SQL. Si bien SQL utiliza los operadores algebraicos matemáticos, estos operan distinto a como lo hacen en la teoría de conjuntos. A estos operadores algebraicos se les denomina operadores básicos, y aunque existen además operadores avanzados, en realidad son composiciones de los operadores básicos.

Los operadores algebraicos básicos son:

- **Unión** (Union). La unión de dos relaciones compatibles R_1 y R_2 es una nueva relación R_3, también compatible, cuyo esquema es igual al esquema

Con formato: Derecha: 0.04 cm, Punto de tabulación: 15.56 cm, Derecha + No en 15 cm

Con formato: Centrado, Posición: Horizontal: Centro, Con relación a: Margen

de R_1 y R_2, y cuya extensión está formada por la agrupación, sin repetición, de las extensiones de R_1 y R_2

- **Diferencia** (Minus). La diferencia de dos relaciones compatibles R_1 y R_2 es una nueva relación R_3, también compatible, cuyo esquema es igual al esquema de R_1 y R_2, y cuya extensión está formada por aquellos registros de la relación R_1 que no se encuentran en la relación R_2
- **Selección** (Select). La selección sobre una relación R_1 mediante una cualificación Q es una nueva relación R_2, cuyo esquema es igual a R_1, y cuya extensión está formada por todos aquellos registros de R_1 que satisfacen la cualificación Q
- **Proyección** (Project). La proyección sobre una relación R_1 con esquema $R_1.a_i$, $R_1.a_{j,...}$, $R_1.a_z$ mediante un subesquema $S(R_1) \Xi R_1.a_m$, $R_1.a_{n,...}$, $R_1.a_p$ donde $a_m >= a_i$ y $a_p <= a_z$ es una nueva relación R_2, cuyo esquema es igual al subesquema $S(R_1)$, y cuya extensión es igual a todos los registros de R_1 sin repetición sobre el subesquema $S(R_1)$
- **Producto Cartesiano** (Product). El producto cartesiano de dos relaciones R_1 y R_2 no necesariamente compatibles es una nueva relación R_3, cuyo esquema es igual a la concatenación de los esquemas R_1 y R_2, y cuya extensión está formada por el conjunto de registros que se obtiene de concatenar cada uno de los registros de R_1 con todos y cada uno de los registros de R_2

Los operadores algebraicos avanzados son:

- **Intersección** (Intersect). La intersección de dos relaciones compatibles R_1 y R_2 es una nueva relación R_3, también compatible, cuyo esquema es igual al esquema de R_1 y R_2, y cuya extensión está formada por el conjunto de registros que son comunes a R_1 y R_2
- **Reunión** (Join). La reunión de dos relaciones R_1 y R_2 no necesariamente compatibles, pero en las que existe al menos un atributo con el dominio común, sobre una cualificación Q, es una nueva relación R_3 cuya intención está formada por la concatenación de las intenciones de R_1 y R_2, y cuya

Con formato: Derecha: 0.04 cm, Punto de tabulación: 15.56 cm, Derecha + No en 15 cm

Con formato: Centrado, Posición: Horizontal: Centro, Con relación a: Margen

extensión está formada por los registros que resultan del producto cartesiano de R_1 x R_2 que satisfacen la cualificación Q

- **División** (Division). La división de una relación R_1 con esquema $R_1.a_i$, $R_1.a_{j,...,}$ $R_1.a_z$ entre una relación R_2 de esquema $R_1.a_m$, $R_1.a_{n,...,}$ $R_1.a_p$ donde $a_m >= a_i$ y $a_p <= a_z$, es una nueva relación R_3 cuyo esquema es igual a la diferencia del esquema de R_1 menos el esquema de R_2, y cuya extensión es igual a todos los registros de R_1 sin repetición para las cuales está presente toda la extensión de la relación R_2

Hoy en día existe en el mercado un gran número de RDBMS cuyo lenguaje de manipulación de datos es SQL con más o menos mejores introducidas por los fabricantes. SQL se ha convertido en un estándar gracias a las mejoras que ha experimentado en los últimos años que le han conferido una gran potencia y sencillez de uso y aprendizaje. Este lenguaje es más como una herramienta de comunicación que un lenguaje de programación. Existen categorías de sentencias SQL, siendo Data Definition Lenguaje (DDL) y Data Manipulation Language (DML) las más comunes. La primera está formada por las sentencias utilizadas para definir, modificar o eliminar objetos de la base de datos (CREATE, ALTER, DROP), mientras que la segunda permite acceder, crear y manipular datos existentes en las estructuras de la base de datos (SELECT, INSERT, UPDATE y DELETE).

Con formato: Derecha: 0.04 cm, Punto de tabulación: 15.56 cm, Derecha + No en 15 cm

Con formato: Centrado, Posición: Horizontal: Centro, Con relación a: Margen

2.6 COMUNICACIONES

Para que se necesitan las redes

Los equipos electrónicos que ejecutan tareas que van desde la computación y el desarrollo de la seguridad hasta el control ambiental pueden generar mayores beneficios al formar parte de sistemas integrados. Las ventajas de tener una serie de dispositivos individuales trabajando en forma conjunta crecen a medida que su cantidad se multiplica. Al mismo tiempo el reto de poder brindar los enlaces necesarios también aumenta.

Funciones básicas

Para superar las limitaciones físicas de los elementos básicos de una red, existen dispositivos cuyas funciones son las de extender las topologías de red. Estos elementos son: concentradores (hubs), repetidores, puentes (bridges), encaminadores (routers) y pasarelas (gateways). Los dispositivos de interconexión de redes proporcionan algunas de (o todas) las siguientes funciones básicas:

Definición de segmentos dentro de la red: Al dividir la red en segmentos se consigue aumentar las prestaciones de la red ya que cada tramo soporta sólo su propio tráfico y no los de los otros segmentos.

Separación entre redes: Mediante estos dispositivos las grandes redes se pueden componer de otras más pequeñas interconectadas entre sí, de forma transparente para el usuario. Varias redes físicas pueden combinarse para formar una única red lógica.

En la figura 2.6.1 se representa la relación de los dispositivos de interconexión con los niveles del modelo de referencia OSI (Organización Internacional para la estandarización).

Con formato: Derecha: 0.04 cm, Punto de tabulación: 15.56 cm, Derecha + No en 15 cm

Con formato: Centrado, Posición: Horizontal: Centro, Con relación a: Margen

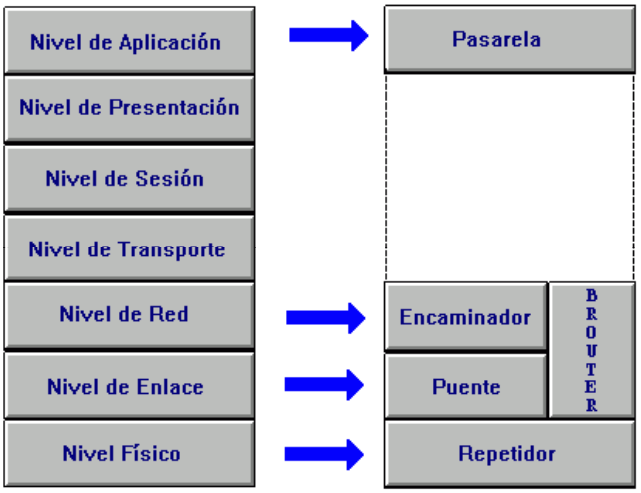


FIGURA 2.6.1 RELACIÓN DE DISPOSITIVOS DE INTERCONEXIÓN

Una red debe ser:

- **Confiable.** Estar disponible cuando se le requiera, poseer velocidad de respuesta adecuada
- **Confidencial.** Proteger los datos sobre los usuarios de ladrones de información
- **Integral** en su manejo de información

Tipos de redes de datos

Las redes de comunicación se han clasificado por su tamaño, complejidad y alcance en: Redes Multiproceso(LIN), Redes de Área Local(LAN), Redes de Área Metropolitana(MAN), Redes de Área Amplia (WAN), Redes de Área Global(GAN).

Redes Multiproceso: Redes empleadas en procesos industriales, para el control y automatización de los sistemas de producción; comúnmente llamadas Red Local Industrial o Red Multiproceso (LIN).

Con formato: Derecha: 0.04 cm, Punto de tabulación: 15.56 cm, Derecha + No en 15 cm

Con formato: Centrado, Posición: Horizontal: Centro, Con relación a: Margen

Redes de Área Local: Se puede definir una LAN como un sistema de comunicaciones local, capaz de facilitar el intercambio de datos, voz, fax, videoconferencias, difusión de video, telemetría y cualquier otra forma de comunicación electrónica.

Las LAN se caracterizan por lo siguiente:

- **Un medio de comunicación común** a través del cual todos los dispositivos pueden compartir información, programas y equipos independientes del lugar físico donde se encuentra el usuario o el dispositivo
- **Una velocidad de transmisión** muy elevada que puede adaptarse a las necesidades de los usuarios y de los equipo (hasta 100Mbps)
- **Una distancia** entre estaciones relativamente corta, de unos metros y varios kilómetros (2000 o 3000mts) aunque la distancia puede ser mucho mayor utilizando dispositivos especiales (hasta 25km)
- **Empleo de cables** de conexión normales
- **Todos los dispositivos** pueden comunicarse con el resto y algunos de ellos pueden funcionar independientemente
- **Flexibilidad**, el usuario administra y controla su propio sistema

Redes de Área Metropolitana. Este tipo de redes, engloban dos o más LAN´s, teniendo un área de cobertura mayor.

Redes de Área Amplia. Están constituidas de dos o más MAN o LAN y pueden definirse como aquellas que abarcan los límites de un país.

Redes de Área Global. Son redes corporativas que atienden necesidades de empresas a nivel Nacional e Internacional (Internet).

Red de Área Extendida (Wide Área Network). Es un sistema de comunicación entre computadoras, que permite compartir información y recursos, con la característica de que la distancia entre las computadoras es amplia (de un país a

Con formato: Derecha: 0.04 cm, Punto de tabulación: 15.56 cm, Derecha + No en 15 cm

Con formato: Centrado, Posición: Horizontal: Centro, Con relación a: Margen

otro, de una cuidad a otra, de un continente a otro). Es comúnmente dos o más redes de área local interconectadas, generalmente a través de una amplia zona geográfica.

Red Regional. Es una red que conecta redes de área extendida en una determinada área geográfica. Estas redes están interconectadas a otras redes de nivel superior con enlaces T1 de líneas telefónicas (o vía satélite), capaces de transmitir 1.54 Megabytes por segundo.

Red Columna Vertebral (Backbone Network). También llamada Red de Transporte (Carrier Network). Este tipo de red cubre, por lo general, un país o un continente. Sirve como apoyo a las empresas que poseen redes locales y no pueden costear la inversión en la infraestructura y mantenimiento de una red de área extendida propia. Es una red de alto rendimiento formada por líneas telefónicas especiales de alta velocidad (enlaces T3 que puede transmitir 4.5 Megabytes por segundo), cables de fibra óptica y enlaces vía satélite.

Red Internacional (INTERNETworking - Telaraña de Área Mundial (World Wide Web). Es una enorme red de redes que se enlaza a muchas de las redes científicas, de investigación y educacionales alrededor del mundo así como a un número creciente de redes comerciales.

Topología de Redes

Se define como Topología, a la forma física en que es posible conectar las estaciones de trabajo dentro de un LAN. Las topologías sirven para poner orden al caos potencial que se puede producir al colocar las terminales de cómputo de forma indiscriminada en la Red.

Existen tres formas posibles de conexión

- **Punto a punto.** En la que sólo se unen dos terminales de cómputo adyacentes, sin pasar a través de una estación intermedia

Con formato: Derecha: 0.04 cm, Punto de tabulación: 15.56 cm, Derecha + No en 15 cm

Con formato: Centrado, Posición: Horizontal: Centro, Con relación a: Margen

- **Multipunto.** En la que dos o más terminales de cómputo comparten un solo medio de transmisión
- **Lógica.** En la cual las terminales de cómputo se pueden comunicar entre sí, haya o no conexión física directa entre ellas

Topologías en Bus

Características:

- Consiste de varios nodos ligados a un medio de transmisión común, este medio es también conocido como una línea troncal, bus o segmento de red
- Las señales viajan en ambos sentidos
- Los mensajes se envían por todo el canal de distribución, para que una estación pueda recibirlo, reconociendo su propia dirección
- Utiliza unidades repetidoras para su conectividad y ampliación
- Su operación se basa en el Método de Acceso CSMA/CD (Carrier Sense Multiple Access/Collision Detection)
- Velocidad de transmisión de 10Mbps

Una de sus ventajas más relevante es que si llega a fallar una estación de trabajo, esto no provocará la caída de la red, además de tenerse una gran facilidad en su expansión. Véase la conexión de esta topología en la figura 2.6.2.

Con formato: Derecha: 0.04 cm, Punto de tabulación: 15.56 cm, Derecha + No en 15 cm

Con formato: Centrado, Posición: Horizontal: Centro, Con relación a: Margen

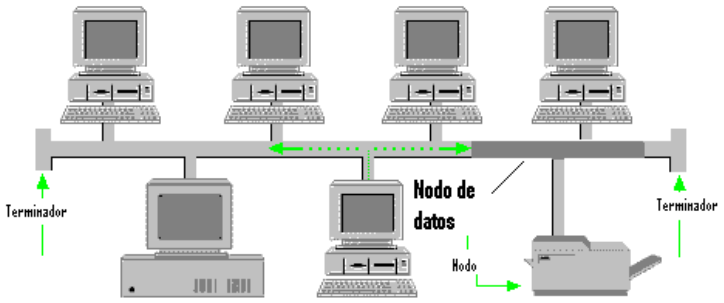


FIGURA 2.6.2 TOPOLOGÍA DE CANAL (BUS)

Topología de Anillo

Características:

- Conecta a varias estaciones de trabajo sobre un medio de transmisión formando un circuito de conexión punto a punto
- Los datos viajan al rededor del anillo en un solo sentido y pasa a través de cada nodo
- Su operación se basa en el Método de Acceso Token Passing
- Su velocidad de transmisión es de 4 ó 16 Mbps

Una de sus ventajas es que no hay un concentrador o centro de cableado, el nodo provee esta función. Una de sus desventajas es que si alguna estación llega a fallar, provocará la caída total de toda la red y presenta dificultad para su crecimiento. Véase la figura 2.6.3 como se efectúa la conexión Topología de Anillo

Con formato: Derecha: 0.04 cm, Punto de tabulación: 15.56 cm, Derecha + No en 15 cm

Con formato: Centrado, Posición: Horizontal: Centro, Con relación a: Margen



FIGURA 2.6.3 TOPOLOGÍA DE ANILLO

Topología de Estrella (Star)

En una topología estrella todos y cada uno de los nodos de la red se conectan a un concentrador, véase figura 2.6.4. Los datos en estas redes fluyen del emisor hasta el concentrador. Este controla realiza todas las funciones de red además de actuar como amplificador de los datos. Esta configuración se suele utilizar con cables de par trenzado aunque también es posible llevarla a cabo con cable coaxial o fibra óptica.

Tanto Ethernet como LocalTalk utilizan este tipo de topología

- Gran facilidad de instalación
- Posibilidad de desconectar elementos de red sin causar problemas
- Facilidad para la detección de fallo y su reparación

Inconvenientes de la topología de estrella

- Requiere más cable que la topología de bus
- Un fallo en el concentrador provoca el aislamiento de todos los nodos a él conectados
- Se han de comprar concentradores

Con formato: Derecha: 0.04 cm, Punto de tabulación: 15.56 cm, Derecha + No en 15 cm

Con formato: Centrado, Posición: Horizontal: Centro, Con relación a: Margen

Conectar un conjunto de computadoras en estrella es uno de los sistemas más antiguos, equivale a tener una computadora central (el servidor de archivos o Server), encargada de controlar la información de toda la red. Dicha información abarca desde los mensajes entre usuarios, datos almacenados en un archivo en particular, manipulación de archivos, etc. Para poder instalar este tipo de red, cada una de las computadoras utilizadas como estaciones de trabajo necesitan de una tarjeta de conexión para lograr la interfase con la computadora central. Los tres tipos de conexión mencionados son los principales para comunicar una serie de computadoras de la misma familia. Uno de los tipos mas antiguos de topologías de redes es la de estrella, la cual usa el mismo método de envío y recepción de mensajes de una topología LAN en estrella deben pasar a través de un dispositivo central de conexiones conocido como concentrador de cableado, el cual controla el flujo de datos. La topología en estrella tiene un nodo central desde el que se irradian todos los enlaces hacia los demás nodos. Por el nodo central, generalmente ocupado por un concentrador, pasa toda la información que circula por la red. Esta topología se caracteriza por existir en ella un punto central, o más propiamente un nodo central. Todos los elementos de la red se encuentran conectados directamente mediante un enlace punto a punto al nodo central de la red.

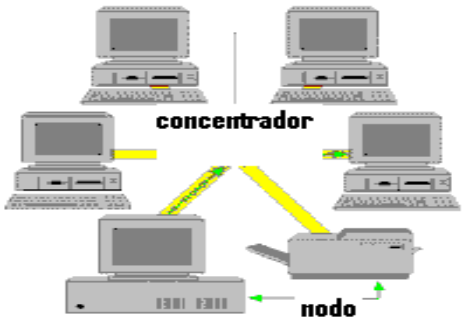


FIGURA 2.6.4 TOPOLOGÍA ESTRELLA

Con formato: Derecha: 0.04 cm, Punto de tabulación: 15.56 cm, Derecha + No en 15 cm

Con formato: Centrado, Posición: Horizontal: Centro, Con relación a: Margen

Componentes de una red

Para obtener la funcionalidad de un red son necesarios diversos dispositivos de ésta, que se conectan entre si de maneras específicas. A continuación presentamos los dispositivos básicos que conforman una red.

- **Servidor (Server).** Es la máquina principal de la red. Se encarga de administrar los recursos de ésta y el flujo de la información. Algunos servidores son dedicados que realizan tareas específicas. Estación de trabajo (Workstation): es una computadora que se encuentra conectada físicamente al servidor por medio de algún tipo de cable. En la mayor parte de los casos esta computadora ejecuta su propio sistema operativo y, posteriormente, se añade al ambiente de red
- **Impresora de red.** Impresora conectada a la red de tal forma que más de un usuario pueda imprimir en ella
- **Sistema operativo de red.** Es el sistema (software) que se encarga de administrar y controlar en forma general a la red. Existen varios sistemas operativos multiusuario a decir Unix, Netware, Windows NT, Linux, etcétera. Según la forma de interacción de los programas en la red, existen dos formas de arquitectura lógica
- **Cliente-servidor.** Modelo de proceso en el que las tareas se reparten entre programas que se ejecutan en el servidor y otros en la estación de trabajo del usuario. El cliente es la entidad que solicita la realización de una tarea al servidor
- **Redes de pares (peer-to-peer; punto a punto):** modelo que permite la comunicación entre usuarios (estaciones) directamente, sin tener que pasar por un equipo central para la transferencia
- **Concentradores (Hubs).** El término concentrador describe la manera en que las conexiones de cableado de cada nodo de una red se centralizan y conectan en un único dispositivo (véase figura 2.6.5). Se suele aplicar a concentradores Ethernet, TokenRing y FDDI (Fiber Distributed Data Interface) soportando módulos individuales que concentran múltiples tipos

Con formato: Derecha: 0.04 cm, Punto de tabulación: 15.56 cm, Derecha + No en 15 cm

Con formato: Centrado, Posición: Horizontal: Centro, Con relación a: Margen

de funciones en un solo dispositivo. El tipo de hub Ethernet más popular es el hub 10BaseT. En este sistema la señal llega a través de cables de par trenzado a una de las puertas, siendo regenerada eléctricamente y enviada a las demás salidas.

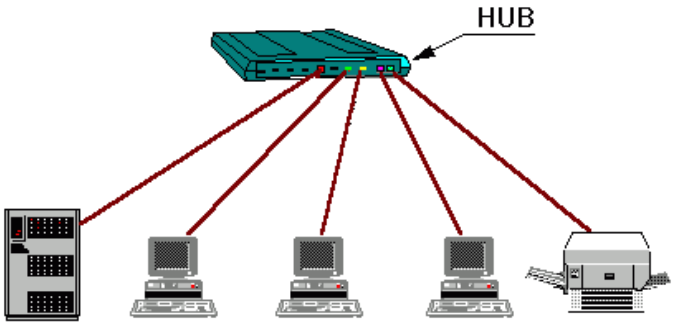


FIGURA 2.6.5 CONEXIÓN DE CABLEADO DE UN HUB

Repetidores. El repetidor es un elemento que permite la conexión de dos tramos de red, teniendo como función principal regenerar eléctricamente la señal, para permitir alcanzar distancias mayores manteniendo el mismo nivel de la señal a lo largo de la red. Un repetidor interconecta múltiples segmentos de red en el nivel físico del modelo de referencia OSI. Por esto sólo se pueden utilizar para unir dos redes que tengan los mismos protocolos de nivel físico. Debido a esto existen más riesgos de colisión y más posibilidades de congestión de la red (véase la figura 2.6.6). Se pueden clasificar en dos tipos:

- **Locales.** Cuando enlazan redes próximas
- **Remotos.** Cuando las redes están alejadas y se necesita un medio intermedio de comunicación

Con formato: Derecha: 0.04 cm, Punto de tabulación: 15.56 cm, Derecha + No en 15 cm

Con formato: Centrado, Posición: Horizontal: Centro, Con relación a: Margen

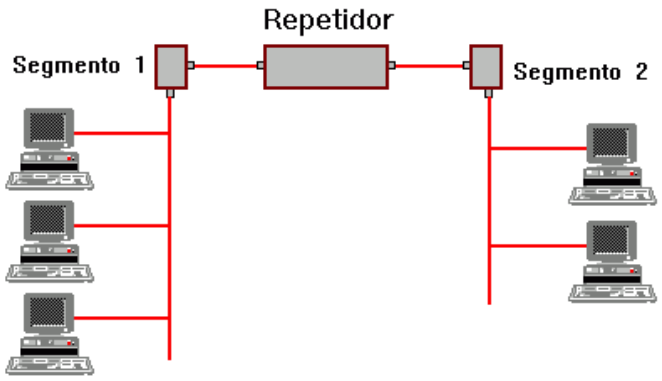


FIGURA 2.6.6 REPETIDOR

Normalmente la utilización de repetidores está limitada por la distancia máxima de la red y el tamaño máximo de cada uno de los segmentos de red conectados.

Ventajas:

- Incrementa la distancia cubierta por la RAL
- Retransmite los datos sin retardos
- Es transparente a los niveles superiores al físico

Desventajas:

- Incrementa la carga en los segmentos que interconecta los repetidores son utilizados para interconectar RALs que estén muy próximas, cuando se quiere una extensión física de la red. La tendencia actual es dotar de más inteligencia y flexibilidad a los repetidores, de tal forma que ofrezcan capacidad de gestión y soporte de múltiples medios físicos, como Ethernet sobre par trenzado (10BaseT), ThickEthernet (10Base5), ThinEthernet (10Base2), TokenRing, fibra óptica, etc.

Con formato: Derecha: 0.04 cm, Punto de tabulación: 15.56 cm, Derecha + No en 15 cm

Con formato: Centrado, Posición: Horizontal: Centro, Con relación a: Margen

Puentes (Bridges)

Son elementos inteligentes, constituidos como nodos de la red, que conectan entre sí dos subredes, transmitiendo de una a otra el tráfico generado no local. Al distinguir los tráficos locales y no locales, estos elementos disminuyen el mínimo total de paquetes circulando por la red por lo que, en general, habrá menos colisiones y resultará más difícil llegar a la congestión de la red. Las redes conectadas a través de un puente que aparenta ser una única red (véase figura la 2.6.7), ya que realizan su función transparentemente; es decir, las estaciones no necesitan conocer la existencia de estos dispositivos, ni siquiera si una estación pertenece a uno u otro segmento. Un puente ejecuta tres tareas básicas: Aprendizaje de las direcciones de nodos en cada red, Filtrado de las tramas destinadas a la red local y envío de las tramas destinadas a la red remota. Se distinguen dos tipos de puentes:

- **Locales.** Sirven para enlazar directamente dos redes físicamente cercanas
- **Remotos o de área extensa**: Se conectan en parejas, enlazando dos o más redes locales, formando una red de área extensa, a través de líneas telefónicas

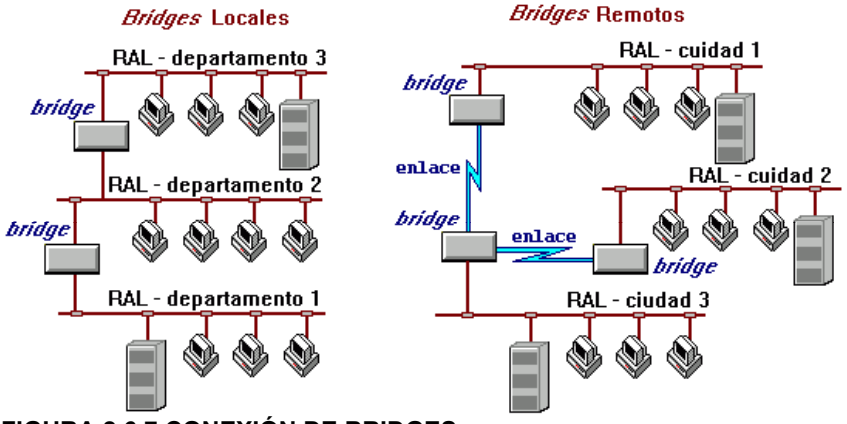


FIGURA 2.6.7 CONEXIÓN DE BRIDGES

Con formato: Derecha: 0.04 cm, Punto de tabulación: 15.56 cm, Derecha + No en 15 cm

Con formato: Centrado, Posición: Horizontal: Centro, Con relación a: Margen

Se puede realizar otra división de los bridges en función de la técnica de filtrado y envío (bridging) que utilicen: Spanning Tree Protocol Bridge o Transparent Protocol Bridge (Protocolo de Árbol en Expansión o Transparente, STP). Estos bridges deciden qué paquetes se filtran en función de un conjunto de tablas de direcciones almacenadas internamente. Se emplea normalmente en entornos Ethernet. Source Routing Protocol Bridge (Bridge de Protocolo de Encaminamiento por Emisor, SRP). El emisor ha de indicar al bridge cuál es el camino a recorrer por el paquete que quiere enviar. Se utiliza normalmente en entornos TokenRing. Source Routing Transparent Protocol Bridge (Bridge de Protocolo de Encaminamiento por Emisor Transparente, SRTP). Este tipo de bridges pueden funcionar en cualquiera de las técnicas anteriores.

Ventajas de la utilización de bridges:

- **Fiabilidad.** Utilizando bridges se segmentan las redes de forma que un fallo sólo imposibilita las comunicaciones en un segmento
- **Eficiencia.** Segmentando una red se limita el tráfico por segmento, no influyendo el tráfico de un segmento en el de otro
- **Seguridad.** Creando diferentes segmentos de red se pueden definir distintos niveles de seguridad para acceder a cada uno de ellos, siendo no visible por un segmento la información que circula por otro
- **Dispersión.** Cuando la conexión mediante repetidores no es posible debido a la excesiva distancia de separación, los bridges permiten romper esa barrera de distancias

Desventajas de los bridges:

- Son ineficientes en grandes interconexiones de redes, debido a la gran cantidad de tráfico administrativo que se genera
- Pueden surgir problemas de temporización cuando se encadenan varios bridges

Con formato: Derecha: 0.04 cm, Punto de tabulación: 15.56 cm, Derecha + No en 15 cm

Con formato: Centrado, Posición: Horizontal: Centro, Con relación a: Margen

- Pueden aparecer problemas de saturación de las redes por tráfico de difusión

Encaminadores (Routers)

Son dispositivos inteligentes que trabajan en el Nivel de Red del modelo de referencia OSI, por lo que son dependientes del protocolo particular de cada red. Envían paquetes de datos de un protocolo común, desde una red a otra (véase la figura 2.6.8). Convierten los paquetes de información de la red de área local, en paquetes capaces de ser enviados mediante redes de área extensa. Durante el envío, el encaminador examina el paquete buscando la dirección de destino y consultando su propia tabla de direcciones, la cual mantiene actualizada intercambiando direcciones con los demás routers para establecer rutas de enlace a través de las redes que los interconectan.

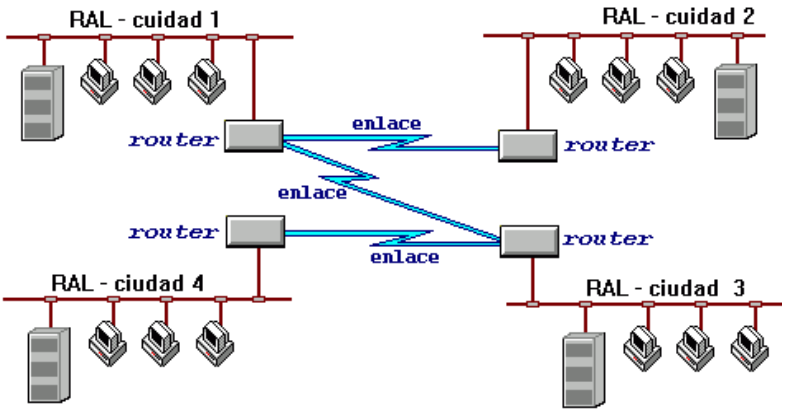


FIGURA 2.6.8 ROUTERS (ENCAMINADORES)

Con formato: Derecha: 0.04 cm, Punto de tabulación: 15.56 cm, Derecha + No en 15 cm

Con formato: Centrado, Posición: Horizontal: Centro, Con relación a: Margen

Los encaminadores se pueden clasificar dependiendo de varios criterios:

En función del área

- **Locales.** Sirven para interconectar dos redes por conexión directa de los medios físicos de ambas al router
- **De área extensa.** Enlazan redes distantes

En función de la forma de actualizar las tablas de encaminamiento (routing)

- **Estáticos.** La actualización de las tablas es manual
- **Dinámicos.** La actualización de las tablas las realiza el propio router automáticamente
- **En función de los protocolos que soportan:** (IPX,TCP/IP ,DECnet ,AppleTalk, XNS, OSI,X.25)

En función del protocolo de encaminamiento que utilicen

- **Routing Information Protocol (RIP).** Permite comunicar diferentes sistemas que pertenezcan a la misma red lógica. Tienen tablas de encaminamiento dinámicas y se intercambian información según la necesitan. Las tablas contienen por dónde ir hacia los diferentes destinos y el número de saltos que se tienen que realizar. Esta técnica permite 14 saltos como máximo
- **Exterior Gateway Protocol (EGP).** Este protocolo permite conectar dos sistemas autónomos que intercambien mensajes de actualización. Se realiza un sondeo entre los diferentes routers para encontrar el destino solicitado. Este protocolo sólo se utiliza para establecer un camino origen-destino; no funciona como el RIP determinando el número de saltos
- **Open Shortest Path First Routing (OSPF)** Está diseñado para minimizar el tráfico de encaminamiento, permitiendo una total autentificación de los mensajes que se envían. Cada encaminador tiene una copia de la topología de la red y todas las copias son idénticas. Cada encaminador distribuye la

Con formato: Derecha: 0.04 cm, Punto de tabulación: 15.56 cm, Derecha + No en 15 cm

Con formato: Centrado, Posición: Horizontal: Centro, Con relación a: Margen

información a su encaminador adyacente. Cada equipo construye un árbol de encaminamiento independientemente

- **IS-IS Encaminamiento OSI según las normas: ISO 9575, ISO 9542 e ISO 10589.** El concepto fundamental es la definición de encaminamiento en un dominio y entre diferentes dominios. Dentro de un mismo dominio el encaminamiento se realiza aplicando la técnica de menor coste. Entre diferentes dominios se consideran otros aspectos como puede ser la seguridad

Ventajas de los encaminadores:

- **Seguridad.** Permiten el aislamiento de tráfico, y los mecanismos de encaminamiento facilitan el proceso de localización de fallos en la red
- **Flexibilidad.** Las redes interconectadas con un encaminador no están limitadas en su topología, siendo estas redes de mayor extensión y más complejas que las redes enlazadas con un puente
- **Soporte de Protocolos**. Son dependientes de los protocolos utilizados, aprovechando de una forma eficiente la información de cabecera de los paquetes de red
- **Relación Precio / Eficiencia.** El coste es superior al de otros dispositivos, en términos de precio de compra, pero no en términos de explotación y mantenimiento para redes de una complejidad mayor
- **Control de Flujo y Encaminamiento.** Utilizan algoritmos de encaminamiento adaptativos (RIP, OSPF, etc), que gestionan la congestión del tráfico con un control de flujo que redirige hacia rutas alternativas menos congestionadas

Desventajas de los routers:

- **Lentitud de proceso** de paquetes respecto a los puentes
- **Necesidad de gestionar** el subdireccionamiento en el Nivel de Enlace
- **Precio superior** a los puentes

Con formato: Derecha: 0.04 cm, Punto de tabulación: 15.56 cm, Derecha + No en 15 cm

Con formato: Centrado, Posición: Horizontal: Centro, Con relación a: Margen

Por su posibilidad de segregar tráfico administrativo y determinar las rutas más eficientes para evitar congestión de red, son una excelente solución para una gran interconexión de redes con múltiples tipos de RALs, MANs, WANs y diferentes protocolos. Es una buena solución en redes de complejidad media, para separar diferentes redes lógicas, por razones de seguridad y optimización de las rutas.

Pasarelas (Gateways): Estos dispositivos están pensados para facilitar el acceso entre sistemas o entornos soportando diferentes protocolos. Operan en los niveles más altos del modelo de referencia OSI (Nivel de Transporte, Sesión, Presentación y Aplicación) y realizan conversión de protocolos para la interconexión de redes con protocolos de alto nivel diferente. Los gateways incluyen los 7 niveles del modelo de referencia OSI, y aunque son más caros que un bridge o un router, se pueden utilizar como dispositivos universales en una red corporativa compuesta por un gran número de redes de diferentes tipos. Los gateways tienen mayores capacidades que los routers y los bridges porque no sólo conectan redes de diferentes tipos, sino que también aseguran que los datos de una red que transportan son compatibles con los de la otra red. Conectan redes de diferentes arquitecturas procesando sus protocolos y permitiendo que los dispositivos de un tipo de red puedan comunicarse con otros dispositivos de otro tipo de red.

Ventajas:

- Simplifican la gestión de red
- Permiten la conversión de protocolos

Desventajas:

- Su gran capacidad se traduce en un alto precio de los equipos
- La función de conversión de protocolos impone una sustancial sobrecarga en el gateway, la cual se traduce en un relativo bajo rendimiento. Debido a esto, un gateway puede ser un cuello de botella potencial si la red no está optimizada para mitigar esta posibilidad

Con formato: Derecha: 0.04 cm, Punto de tabulación: 15.56 cm, Derecha + No en 15 cm

Con formato: Centrado, Posición: Horizontal: Centro, Con relación a: Margen

Conmutadores (Switches)

Los conmutadores tienen la funcionalidad de los concentradores a los que añaden la capacidad principal de dedicar todo el ancho de banda de forma exclusiva a cualquier comunicación entre sus puertos. Esto se consigue debido a que el conmutador no actúa como repetidor multipuerto, sino que únicamente envía paquetes de datos hacia aquella puerta a la que van dirigidos. Esto es posible debido a que los equipos configuran unas tablas de encaminamiento con las direcciones MAC (nivel 2 de OSI) asociadas a cada una de sus puertas. Estos equipos habitualmente trabajan con anchos de banda de 10 y 100 Mbps, pudiendo coexistir puertas con diferentes anchos de banda en el mismo equipo. Las puertas de un conmutador pueden dar servicio tanto a puestos de trabajo personales como a segmentos de red (hubs), siendo por este motivo ampliamente utilizados como elementos de segmentación de redes y de encaminamiento de tráfico.

Esta tecnología permite una serie de facilidades tales como:

- **Filtrado inteligente**. Posibilidad de hacer filtrado de tráfico no sólo basándose en direcciones MAC, sino considerando parámetros adicionales, tales como el tipo de protocolo o la congestión de tráfico dentro del switch o en otros switches de la red
- **Soporte de redes virtuales**. Posibilidad de crear grupos cerrados de usuarios, servidos por el mismo switch o por diferentes switches de la red, que constituyan dominios diferentes a efectos de difusión. De esta forma también se simplifican los procesos de movimientos y cambios, permitiendo a los usuarios ser ubicados o reubicados en red mediante software
- **Integración de routing.** Inclusión de módulos que realizan función de los routers (encaminamiento), de tal forma que se puede realizar la conexión entre varias redes diferentes mediante propios switches(conmutadores)

Con formato: Derecha: 0.04 cm, Punto de tabulación: 15.56 cm, Derecha + No en 15 cm

Con formato: Centrado, Posición: Horizontal: Centro, Con relación a: Margen

Routers multiprotocolo

Estos dispositivos han permitido a los usuarios transportar protocolos diferentes sobre la misma infraestructura de red, lo cual permitiría ahorrar en costes de la infraestructura de transmisión y una potencial mejora de la interoperabilidad.

Interconexión de LAN/WAN bajo switches

Los conmutadores han evolucionado rápidamente dotándose de altas capacidades y velocidad de proceso. Pensados para soportar conmutación ATM (Asynchronous Transfer Mode, Modo de Transferencia Asíncrono) bajo una arquitectura punto a punto, han logrado gran implantación como mecanismo de interconexión de redes de áreas locales heterogéneas, Token Ring y Ethernet en un mismo dominio. Esto se consigue dado que el conmutador permite la segmentación de la red en subredes conectadas a cada uno de sus puertos que puede gestionar de manera independiente.

Capacidad de gestión

Los fabricantes están dotando a sus dispositivos de interconexión con mayores capacidades de gestión que permitan la monitorización de la red mediante estaciones de gestión y control de los dispositivos de la red, enviando comandos por la red desde la estación de gestión hasta el dispositivo de la red para cambiar/inicializar su configuración.

Número de redes que van a ser conectadas y topología de las redes

El conocimiento del número de redes a interconectar y las características específicas de cada uno de ellas, permitirá dimensionar correctamente tanto la estructura de la red final como los elementos necesarios para realizar la interconexión. También se han de analizar las necesidades de adquisición de nuevas redes o infraestructura de red para poder dar soporte a la futura red. Es necesario delimitar claramente el tipo de redes existentes (Ethernet, TokenRing,

Con formato: Derecha: 0.04 cm, Punto de tabulación: 15.56 cm, Derecha + No en 15 cm

Con formato: Centrado, Posición: Horizontal: Centro, Con relación a: Margen

FDDI, etc), su topología (estrella, bus, anillo, etc), su distribución espacial en el entorno de operación (localización y distancias). Es recomendable realizar planos del entorno en cuestión. Algunos de los dispositivos necesitan conexión remota o local de consola, por lo que habrá que tener en cuenta que el dispositivo presente esta característica. Gestión disponible (SNMP , CMIP, CMOT). La complejidad de las redes impone la necesidad de utilizar sistemas de gestión capaces de controlar, administrar y monitorizar las redes y los dispositivos de interconexión. Los routers son dispositivos que necesitan que se realicen funciones de gestión. En los otros dispositivos es recomendable que tengan esta facilidad. Protocolos necesarios del nivel de red soportados (TCP/IP, IPX, DECnet, AppleTalk, SNA, XNS, etc.). Dependiendo de las características de las redes a interconectar, serán necesarios unos u otros protocolos. Cada tipo de red determinará cuales son los protocolos que tendrán que soportar los dispositivos a adquirir. Protocolos de WAN soportados (X.25, SDLC HDLC, Frame Relay, SMDS, TCP/IP, etc). Dependiendo de las características de las redes de área extensa que se desee conectar serán necesarios unos u otros protocolos a decir: Protocolos de filtrado (STP, SRP, SRTP), Protocolos de encaminamiento (RIP, EGP, OSPF, IS-IS).

Factores de seguridad

La interconexión de diferentes redes puede plantear problemas importantes respecto a la seguridad. Pueden ocurrir interferencias (de tipo intencionado o no): modificación de parámetros en algún dispositivo, reconfiguración de un sistema, interrupciones de rutas, controles erróneos, etc. Acceso a información: captura de palabras de paso, monitorización de datos, etc. Ataque a información: virus, corrupción, manipulación de datos estadísticos, etc. Para evitar lo anterior se pueden utilizar técnicas como: Filtros en los dispositivos de red para diferentes accesos y en modo confidencial (Secure mode). Cada dispositivo tiene en su tabla las estaciones que están autorizadas.

Con formato: Derecha: 0.04 cm, Punto de tabulación: 15.56 cm, Derecha + No en 15 cm

Con formato: Centrado, Posición: Horizontal: Centro, Con relación a: Margen

3.1 PROBLEMÁTICA ACTUAL

La inmobiliaria cuenta con diferentes proyectos a lo largo de la república mexicana los cuales duran varios años. La operación de un proyecto durante su período de vida, genera documentación contable, fiscal, técnica, de ingeniería, etc. Toda esta documentación en futuros proyectos es sumamente valiosa por la reutilización de información. Por tal motivo, al finalizar cada proyecto toda su documentación debe ser catalogada, empacada, enviada y almacenada en el archivo general ubicado en la ciudad de México, en dónde se concentra toda la documentación generada durante la vida de dichos proyectos. En un futuro, si un proyecto nuevo necesita consultar la documentación de un proyecto anterior, se consulta el catálogo de documentos almacenados, se selecciona alguno y se solicita este para consultarlo. Descripción a detalle de cada una de las siguientes fases:

- **Envío y almacenamiento de paquetes**, véase figura 3.1.1
- **Consulta de documentos contenidos en un paquete**, véase figura 3.1.2

Envío y almacenamiento de paquetes

Solicitud de envío de paquetes al archivo general; una vez que un proyecto termina su vida de operación, el responsable de la documentación generada durante este período escribe el memo “Solicitud de envío de paquetes”, y envía por mensajería este y 2 copias al responsable del departamento de Administración en las oficinas centrales de la inmobiliaria, con 3 semanas de anticipación para que el responsable del departamento de Administración informe al almacenista que esta por realizarse un envío y que debe prepararse para liberar espacio, eliminar paquetes con fechas de vigencia vencidos o para esperar el envío de los nuevos paquetes. Mientras se cumple el plazo de 3 semanas, quién solicita el envío inicia la catalogación de la documentación a enviar agrupándolos en paquetes. En estos paquetes se coloca el documento “Listado de documentación contenida”, en original y 2 copias. En este documento se escribe la siguiente información:

- Nombre del proyecto
- Responsable del envío
- Fecha del envío

- Fecha de destrucción
- Descripción del contenido del paquete
- Folio (que será colocado por el almacenista)
- Ubicación física del paquete (que será colocado por el almacenista)

Una vez generados todos los documentos para los paquetes a enviar, y sí no se ha recibido rechazo a la solicitud de envío del departamento de Administración a través del memo anterior, el solicitador procede al envío de los paquetes a través de mensajería directamente al archivo general. Si se rechaza la solicitud, se debe esperar el tiempo indicado e iniciar nuevamente la solicitud. Esta parte presenta los siguientes problemas:

- **Por cargas de trabajo** puede perderse el seguimiento del memo de solicitud de envío, provocando la no-notificación al almacenista del próximo envío y de la no-notificación al solicitador del envío de la aceptación o rechazo de este
- **A falta de rechazo de la solicitud de envío**, el solicitador envía los paquetes al almacenista lo que puede provocar un cuello de botella en espacio disponible para almacenamiento o sobre carga para el almacenista al no tener notificación de este envío
- **La sobre carga del almacenista** puede provocar falta de control en el registro de los paquetes entrantes en el archivo general provocando a futuro inconsistencia en la ubicación o perdida de documentos

Autorización o rechazo de la solicitud de envío de paquetes

Una vez recibido el memo y sus 2 copias de la solicitud de envío de paquetes al archivo general, el responsable del departamento de Administración procede a notificar al almacenista, enviándole por mensajería el memo y sus copias, el cuál hará lo necesario para recibir el nuevo envío. Al estar listo envía de regreso el memo firmado notificando que está listo para el nuevo envío. Se queda con la copia que le corresponde. Una vez notificado el responsable del departamento por el almacenista de que todo está listo para el nuevo envío, envía notificación al solicitador de envío de paquetes a través del memo firmado. Se queda con la copia que le corresponde. Una vez notificado el solicitador con el memo firmado, debe proceder a enviar los paquetes al

archivo general. La solicitud es rechazada al no haber espacio disponible para almacenar la totalidad de paquetes, se debe esperar el tiempo indicado e iniciar nuevamente la solicitud. Todo esto no debe exceder el plazo de las 3 semanas indicadas al inicio. Esta parte presenta los siguientes problemas:

- **Si el plazo de 3 semanas se cumple** y el solicitador no ha recibido notificación de rechazo, este procederá a enviar los paquetes al archivo general
- **Al no tener garantizado el espacio para su envío,** el solicitador corre el riesgo de perder sus paquetes o parte de ellos, provocando perdida de documentos de uso futuro

Envío de paquetes al archivo general

Una vez el solicitador del envío ha recibido la notificación de autorización ó se ha cumplido el plazo, este envía por mensajería los paquetes con su correspondiente “Listado de documentación contenida” en original y 2 copias.

Así mismo debe llevar su registro de la cantidad de paquetes enviados al archivo para validar posteriormente que todos han sido almacenados y generar un documento que debe engrapar con el formato de almacenamiento que se le hará llegar posteriormente.

Almacenamiento de los paquetes

Una vez que llegan los paquetes al archivo general, el almacenista procede a colocarlos en la ubicación física asignada para esto. Al colocar el paquete en su ubicación, registra en el original y copias del “Listado de documentación contenida” el folio correspondiente y sus “coordenadas”, deja el original pegado al paquete y retira las copias para enviarlas de regreso al responsable del departamento de Administración. Así procede con todos los paquetes. Una vez terminado el almacenamiento, registra en una carpeta nuevamente la información del folio y ubicación física del paquete para posteriores consultas o aclaraciones. Firma las copias y las envía de regreso al responsable del departamento vía mensajería. Esta parte presenta los siguientes problemas:

- Posible equivocación en el registro de la ubicación de paquetes
- Posible falta de consistencia en el formato de las coordenadas de los paquetes, al existir varios almacenistas

- Al tenerse 2 copias por cada paquete almacenado, se corre el riesgo de perder o traspapelar las copias

Registro del almacenamiento en el departamento de Administración

Una vez que el responsable recibe las copias enviadas por el almacenista, 2 por cada paquete almacenado, procede a archivar una de las copias para futuras consultas o aclaraciones. La otra copia la firma y envía al solicitador del envío. Esta parte presenta el siguiente problema:

- El archivo que se va generando es poco práctico y ágil para llevar a cabo la búsqueda de un documento

Notificación del almacenamiento al solicitador y fin del envío

Una vez que el solicitador recibe la copia de almacenamiento por cada uno de los paquetes enviados, procede a engraparlas en conjunto con el documento que genero y donde registró la cantidad de paquetes enviados. Esta parte presenta el siguiente problema:

- Pueden perderse o traspapelarse las copias provocando un proceso de reclamación, que en este momento no esta estandarizada la solución

Consulta de documentos contenidos en un paquete

Solicitud de consulta de documentos contenidos en un paquete; Una vez se ha iniciado un nuevo proyecto y se desea consultar documentos almacenados, se procede a escribir el memo “Solicitud de préstamo de documentos”, en dónde se especifica:

- El proyecto
- Responsable de la solicitud de préstamo
- El tipo de documento solicitado
- Ubicación del documento (que será colocado por el responsable del departamento de Administración)

Este memo es enviado en original y 2 copias por mensajería al responsable del departamento de Administración, el cual con los datos que contiene el memo y con su archivo, registrará la ubicación de lo solicitado y solicitará al almacenista el documento que más se ajusten a lo solicitado. Esta parte presenta el siguiente problema:

- Puede perderse o traspapelarse la solicitud

Asignación de la ubicación de la documentación solicitada

Una vez que el responsable del departamento de Administración recibe el memo, debe consultar su archivo contenido en una hoja de cálculo, para buscar una descripción que se ajuste a lo solicitado y anotar la ubicación física del documento que cumple con lo especificado. Una vez escrita la ubicación en el memo y sus copias, los firma y son enviados al archivo general para que localicen los documentos y los envíen de regreso. Esta parte presenta el siguiente problema:

- Puede perderse o traspapelarse la solicitud

Localización de documentos y envío para su consulta

Al recibir el almacenista el memo y sus copias, las firma y procede a localizar el documento solicitado en las coordenadas especificadas. Extrae el documento del paquete y en su lugar coloca el original del memo. Entonces procede al envío del documento con las copias anexas, al responsable del departamento de Administración.

Consulta del documento solicitado

Una vez el responsable del departamento recibe el documento, firma las copias quedándose con una de ellas y la otra la envía con el documento al solicitador del préstamo vía mensajería. La consulta del documento puede ser indefinidamente. Esta parte presenta el siguiente problema:

- Al pasar el tiempo, puede perderse el control de quién tiene documentos en préstamo

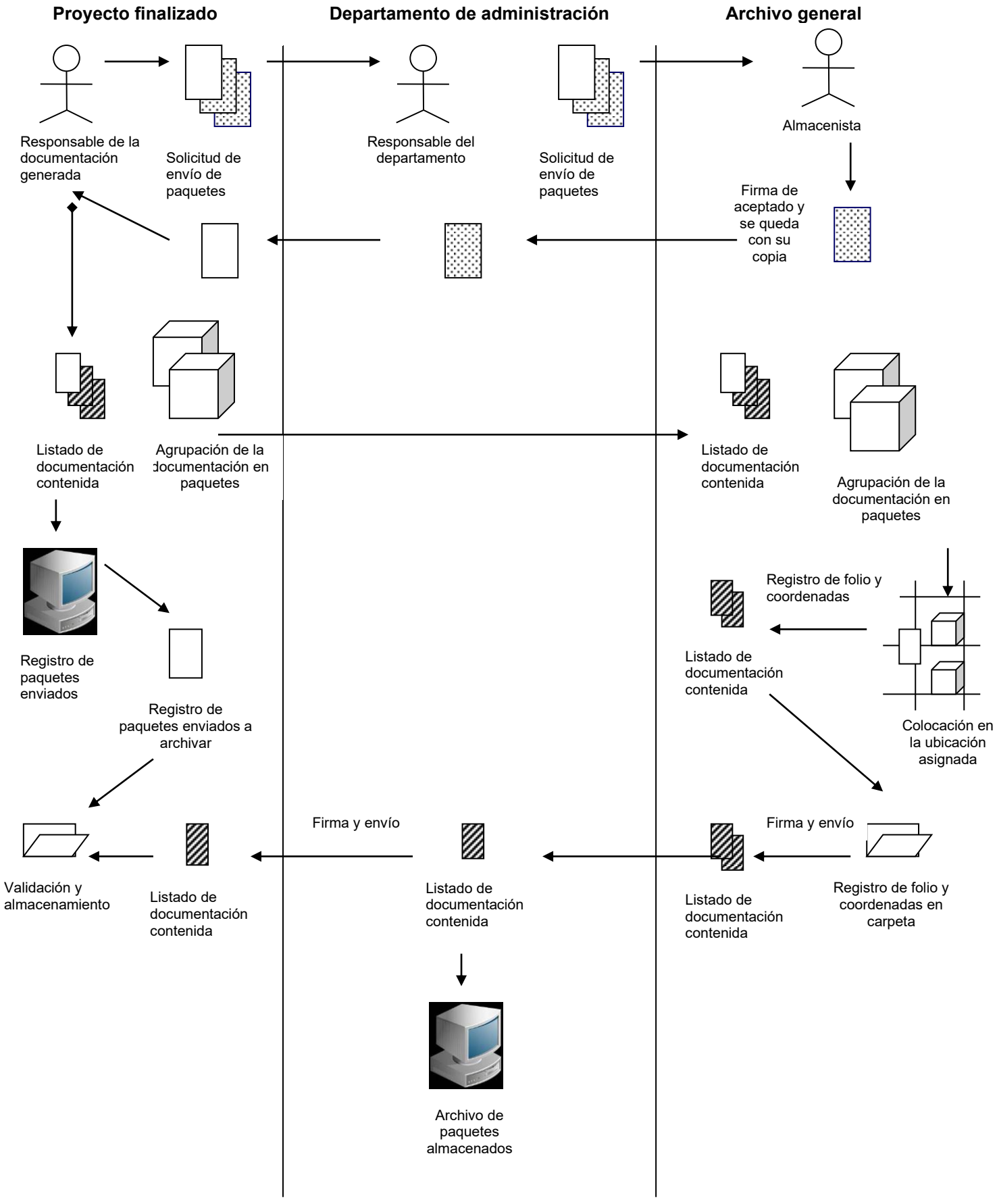


FIGURA 3.1.1 ENVÍO Y ALMACENAMIENTO DE PAQUETES

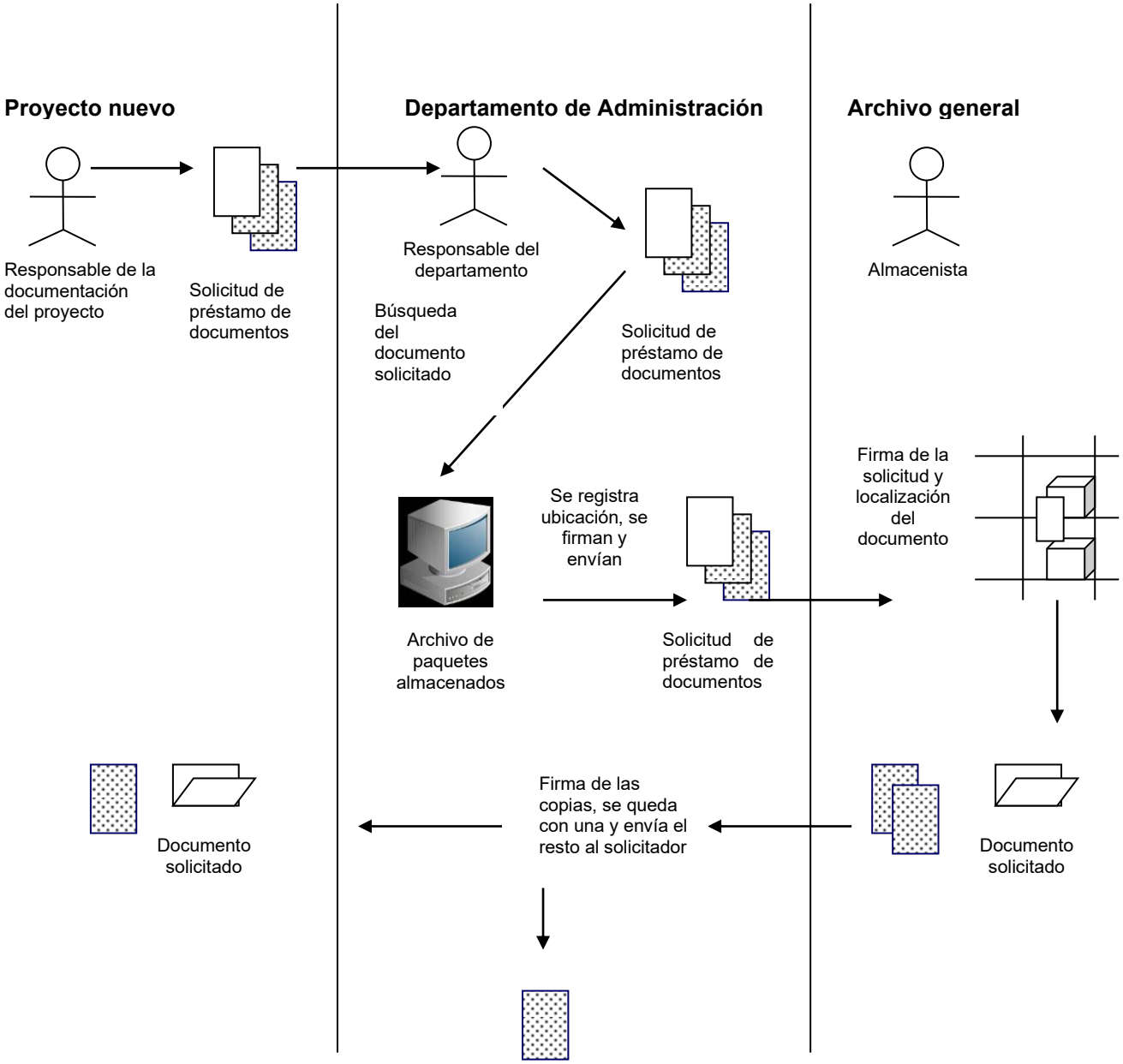


FIGURA 3.1.2 CONSULTA DE DOCUMENTOS CONTENIDOS EN UN PAQUETE

3.2 REQUERIMIENTOS GENERALES Y PARTICULARES

Requerimientos Generales

Los requerimientos generales tienen que ver con características que de una u otra forma pueden limitar al sistema, como por ejemplo, el rendimiento (en tiempo y espacio), interfaces de usuario, fiabilidad (robustez del sistema, disponibilidad de equipo), mantenimiento, seguridad, portabilidad, etc.

Requerimientos de operación:

- El sistema deberá ser desarrollado con las herramientas de que disponga el cliente
- El sistema utilizará una base de datos
- El sistema deberá manejar el sistema operativo Windows 98 en adelante

Requerimientos de rendimiento:

- El sistema no deberá tardar más de 10 segundos en proporcionar un resumen de la información solicitada

Requerimientos de interfaz con el usuario:

- El sistema presentará una interfaz amigable y sencilla de usar por cualquier persona
- El sistema será fácil de usar
- El sistema utilizará un flujo de pantallas sencillo
- El sistema mostrará la información de manera ordenada
- El sistema realizará búsquedas de manera sencilla
- El sistema presentará información general pero representativa como resultado de las búsquedas

Requerimientos de seguridad:

- Se requiere que el sistema cuente con los mecanismos que permitan dotar de la seguridad y administración necesaria que garantice la operación eficiente de la parte del diseño, debe haber un control partiendo de la administración del sistema y continuando con la parte de creación de los usuarios, así como un seguimiento del mismo, requiriéndose de un módulo que permita administrar el sistema para mantenimiento del esquema de los usuarios y seguridad de las formas o pantallas generadas, ya sean de reportes, consultas o de aplicación

- En cuanto a la información que se capture, el sistema deberá contar con catálogos que permitan facilitar la captura de información y con ello reducir el mínimo de error humano
- El sistema deberá proveer de las ayudas y mensajes de error que permitan facilitar al máximo la captura de información, así como la detección de errores
- El sistema deberá contar con claves de acceso únicas para cada usuario autorizado que opere el sistema, con los niveles de seguridad requeridos de acuerdo con la función que desempeñan
- El sistema deberá contar con pistas de auditoria, con el fin de determinar en todo momento el usuario responsable de las operaciones que sobre el sistema se realicen
- En un nivel de funcionamiento deben existir formas para el control de acceso a usuarios. La descripción incluye los requerimientos para los diferentes modos de operación del sistema, capacidades, interfaces, datos, seguridad, privacidad, ambiente de operación, recursos de computo, factores de calidad y otros requerimientos, permitiendo lograr una mayor eficiencia y seguridad en los procesos

Requerimientos de Recursos de Cómputo:

El ambiente requerido es un ambiente que cumpla con todas las expectativas propuestas, dando como resultado un ambiente de trabajo apto para los niveles de tecnología actuales. Es fundamental que cuente con los recursos de hardware y software necesario para la operación eficiente del sistema, para ello se propone una configuración mínima recomendable ya que de no contar con ellos el sistema propuesto no se podrá poner en operación.

- Procesador Pentium II 350 MHZ
- Memoria RAM 64Mb
- Espacio en Disco Duro 120Mb
- Sistema Operativo Windows 98 ó superior

Requerimientos de personal

Es indispensable que el sistema brinde facilidades de operación, ya que las personas involucradas en el sistema deberán familiarizarse rápidamente con el, esto previa capacitación y distribución de los manuales de usuario e instalación correspondiente.

Requerimientos Particulares

Para obtener los requerimientos particulares del sistema, se debe llevar a cabo un proceso de análisis del problema el cual tiene como propósito fundamental obtener una descripción formal de la solución, que proporcione la información suficiente y necesaria para resolver el problema en forma lógica. Por lo que se debe comprender lo que se desea y definir: los datos de entrada, la información que se desea producir (salida) y los métodos para procesar los datos. Es necesario realizar un análisis correcto del problema planteado, antes de hacer cualquier tarea de diseño del sistema. El éxito de un sistema se basa principalmente en qué tan bien diseñado está y para esto es fundamental analizar en forma detallada y crítica la serie de problemas que se plantean. El análisis del problema involucra capturar el máximo de información referente a éste, obtener la visión del mismo por parte de los involucrados y generar un modelo para cada una de estas visiones. Se deben refinar estos modelos hasta obtener una o varias representaciones del problema, los que posibilitan su análisis. Estos modelos que describen las visiones del problema son los que permitirán que se vislumbre alguna solución. Después de un detallado análisis con el cliente, se obtuvieron como resultado los tipos de documentos a manejar, los datos que el sistema deberá registrar y las actividades generales que responden al flujo de cada una de las áreas. A continuación se exponen los resultados obtenidos. En la aplicación aparecerán las siguientes funciones:

Formato

Esta función permitirá al usuario capturar la información que será enviada.

Datos de entrada:

- Datos del que envía
    - Nombre(s)
    - Apellidos

Destrucción de documentos

Fecha en la cual se destruirán los documentos, a partir de la fecha de su ubicación física y su tiempo de vida. Mediante una consulta al contenido del paquete hecha por el usuario, el encargado de los envíos será el responsable de proporcionar dicha información.

Datos de entrada:

- Nombre del proyecto
- Nombre del usuario
- Folio

Ubicación física

Datos de salida:

- Fecha de destrucción
- Identificador del paquete
- Descripción

3.3 BÚSQUEDA Y ANÁLISIS DE LA INFORMACIÓN

Búsqueda. Es hacer gestiones para encontrar algo, información, personas, direcciones, etc. La búsqueda de información tiene diversas formas:

- **Una de ellas es la investigación documental**. Esta se realiza para obtener información orientada a descubrir un conocimiento nuevo, elaborar uno propio, identificar algún conocimiento que se deriva del uso creativo de la información. Se realiza en bibliotecas, hemerotecas, archivos, centros de información
- **Otra forma es la búsqueda en Internet**. La búsqueda se hace para reconocer la información existente, determinar la que hace falta o es limitada o actualizar la que se tiene
- **Otra forma de búsqueda es la que se hace en bases de datos**. Tiene iguales propósitos que la búsqueda en Internet. Cuando hay que buscar información acerca de un asunto, es necesario emplear una estrategia de búsqueda para que el resultado de la búsqueda sea exitoso: más rápido, con mejor y mayor información (objetiva, significativa, pertinente, confiable, actual o vigente)

Una estrategia de búsqueda de información se define con acciones u operaciones lógicas que resuelven cosas como:

- Sobre qué buscar información (se definen necesidades existentes)
- Qué ignorancias tengo y qué sé sobre el tema de búsqueda. (Se definen preguntas que llevan a responder las necesidades de indagación)
- Cuál es el ámbito de relaciones (o de información) del tema principal (se define el tema general, los subtemas, los temas relacionados y los equivalentes)
- Dónde buscar (Se responde a: ¿quién tiene o dónde está la información?)
- Con qué herramientas buscar (Se determina: donde se encuentra la información)
- Cómo hacerlo (Se define: con qué criterios, acotaciones, indicadores, palabras claves)

- En qué puntos o ámbitos temáticos (Se define desde qué otros temas relacionados y subtemas se puede llegar a la información)

En este caso particular nos centraremos en la búsqueda de información ya que para nuestros propósitos es primordial el manejo, presentación, oportunidad y confiabilidad al encontrar información relacionada con los casos de la inmobiliaria en cuestión para lo cual también tenemos que hacer mención de lo que es el análisis de la información.

Análisis de la Información

El análisis de la información puede definirse como la aplicación de técnicas de procesamiento automático del lenguaje natural, de clasificación automática y de representación gráfica (cartografía) del contenido cognitivo (conocimientos) y factual (fecha, lengua, tipo de publicación...) de los datos bibliográficos (o textuales). Esta definición corresponde al análisis asistido por computador. En general, por análisis de la información se entiende la fase de interpretación que el usuario realiza de una manera directa y manual. Los límites de este tipo de análisis son evidentes desde el momento que se trabaja sobre una cantidad importante de datos, y se trata además de incorporar el análisis en un sistema de producción de información elaborada o especializada. El análisis de la información supone que los problemas relativos al almacenamiento de la información y su acceso se encuentran relativamente resueltos. Se parte del postulado de que se dispone de la información, que ésta se encuentra en las bases de datos y que el acceso esta asegurado por sistemas eficaces de búsqueda de la información. A partir de este postulado, el problema que el análisis de la información trata de resolver es el de la explotación de la información relativa a un área determinada (por ejemplo: recursos humanos, marketing, economía y finanzas, ciencia y tecnología). Dentro de nuestra inmobiliaria existe un gran problema con respecto a la búsqueda, análisis y almacenamiento de la información ya que esta se lleva de una forma rudimentaria a través de diversos medios de almacenamientos y captura los cuales para empezar, se recibe información a través de oficios y solicitudes escritas de autorización y entrega de los paquetes los cuales después de llenar algunos formatos, son capturados de manera manual en bitácoras, las cuales

no van normalizadas ni cuentan con un acceso rápido de la información para poder conocer todo lo relacionado con dichos paquetes, no sin antes mencionar que estas capturas las realizan diferentes personas y las cuales tienen diversas formas de cómo levantar los requerimientos de las solicitudes complicando así la pronta búsqueda y localización de los paquetes en cuestión, aparte que estos mismos no cuentan con algo que muestre el estatus en el que se encuentran actualmente ya que pasado algún tiempo y con el poco espacio con el que se cuenta, estos paquetes son destruidos para optimizar costos de almacenamiento.

Este tipo de registro (en bitácoras), es muy lento ya que cuándo se solicita la información sobre algún paquete o sobre el estatus de cualquier solicitud, los encargados tienen que hacer búsquedas exhaustivas, no sin antes mencionar que como existían diversos criterios de captura se aumentaba el tiempo de búsqueda por lo que se demoraba la entrega de la información ocasionándole perdidas o gastos mayores a la empresa. Ante esta problemática, se vio la posibilidad de registrar las solicitudes ya no en bitácoras si no también en hojas de cálculo que permiten un mayor manejo de información, así como una búsqueda más eficaz y la entrega óptima del estatus y de las características de los paquetes en cuestión.

Este tipo de captura resulto ser una solución a medias ya que facilitaba la captura de las solicitudes pero no agilizaba los trámites de recepción, envío y almacenamiento aparte que seguía sin tener un identificador sobre el estatus del paquete entregado, mas sin embargo el procedimiento de captura en las bitácoras nunca se descartó por lo que el usuario independientemente de que empezaba a utilizar las hojas de cálculo, recurría a la captura en papel como se había venido haciendo desde antes para poder llevar un mejor control sobre lo que se estaba registrando en una computadora, por lo que el usuario terminaba haciendo doble trabajo ya que primero era capturar en las bitácoras y después vaciar los datos obtenidos haciendo que se hiciera mas difícil el cambio de un sistema manual de bitácoras a un archivo o medio computacional que agilizara de menos la búsqueda e implementara algunos medios mas para poder hacer fácil la entrega de información así como el estatus de la misma, aparte que la

inmobiliaria en cuestión necesitaba tener un antecedente histórico de lo que había recibido para posibles revisiones o imprevistos que se pudieran suscitar después de haber ya concluido con todos los tramites al entregar un proyecto. Al paso del tiempo y con la captura de la información de las solicitudes recientes, se agilizó en gran medida el proceso de búsquedas minimizando los tiempos de entrega de información y reduciendo algunos costos, aunque se empezaron a ver limitantes y problemas ya que se recurrían a tener los mismos errores de captura por medio de los usuarios que seguían sin tener una normatividad para la captura de los datos mínimos necesarios para levantar una solicitud. Se observo también que era muy necesaria la búsqueda de información en los archivos antiguos los cuales son capturados en hojas de cálculo separadas de la información reciente, lo cual empezaba a ser tedioso y de riesgo ya que se empezaron a crear muchos y diversos archivos según el usuario que capturara las solicitudes de almacenamiento y requisiciones de información las cuales no tenían un estándar o normatividad necesaria para agilizar los procesos volviéndose a complicar la búsqueda, análisis y entrega de información pertinente siendo esto lo que se quería corregir para no presentar nuevamente los mismos errores que se detectaron cuando se utilizaba la captura de solicitudes en las bitácoras de papel. No obstante de que ya hubiese una mejor búsqueda, se necesitaba algunos informes para mostrar los datos de cada paquete y su estatus, por lo cual se menciona como deben ser las características de un informe para que se vea la necesidad de tener alguna herramienta que nos sirva para poder visualizar, entender y verificar los estados de cada parte del proceso que lleva acabo el recibir, almacenar y destruir dichos paquetes.

Generación de Reportes

La generación de reportes consiste en acceder a los datos, formatearlos y presentarlos como información de negocio tanto dentro como fuera de la organización. Esta función constituye la piedra angular en las reglas de negocio al proporcionar los elementos de información mas solicitados de una manera fiable y segura permitiendo:

- Atender cualquier necesidad en cuanto a la generación de informes
- Rentabilizar las inversiones ya realizadas en informática
- Garantizar la disponibilidad de información

Otro contratiempo que se observó es que al realizar la captura manual en las bitácoras, cuando se requería obtener un informe sobre algún tipo de paquete o alguna información relacionada con el almacenamiento estos no llevaban un estándar debido a la captura de datos sin normalización, y que no existe un método o formato en el cual se deban ingresar los datos necesarios para poder entregarse como informe para así poder saber de la situación de un paquete. Cabe mencionar que esto ocasiona que la información se duplique teniendo un atraso en las entregas lo que repercute como perdidas a la organización. Esto se creyó solucionar cuando se implantó el sistema computacional (basado en hojas de calculo), pero los usuarios no podían emitir un reporte con características similares ya que por los diversos criterios de captura se llegan a omitir datos importantes para la presentación de un buen informe que fuera confiable y específico, el cual mostrara la situación real del proceso y el estatus del almacenamiento como tal. En gran medida cuando el informe se tenia realizado se partía ha verificar los estados manualmente para no cometer equivocaciones ya que las ubicaciones de destino final de los paquetes solo eran conocidas por algunas personas y así poder proceder ya sea a regresar el paquete o en definitiva hacer una destrucción de los mismos para poder ocupar su espacio en la bodega.

3.4 ÁREAS INVOLUCRADAS EN EL PROCESO E IDENTIFICACIÓN DEL PROBLEMA

Departamento de Contabilidad y Finanzas

El departamento de Contabilidad y Finanzas da apoyo a los administradores de nivel ejecutivo con el propósito de incrementar el valor de la inversión de los accionistas. Lleva a cabo el desempeño de su tarea a través de:

- El control de costos
- La protección de los activos de la empresa
- La prestación de asesorías sobre inversiones de capital
- El descubrimiento de fuentes de capital corporativo de bajo costo
- El cumplimiento dentro de la administración ejecutiva de la corporación
- Gestión de Información

Departamento de Sistemas de Información

El departamento de sistemas de información proporciona los medios necesarios para integrar las funciones del negocio de la empresa inmobiliaria en su búsqueda por la creación de un valor para los participantes del negocio. Lleva a cabo el desempeño de su tarea a través de:

- El suministro de canales de acceso hacia la información necesaria para impulsar los sistemas de producción, los sistemas de control administrativo, los sistemas de apoyo a las decisiones y los sistemas de información ejecutiva
- La asistencia a las organizaciones funcionales para la creación, estandarización, administración y compartimiento del activo representado por la información corporativa
- La construcción de una infraestructura de información corporativa en conjunción con la administración de nivel ejecutivo y con las organizaciones funcionales

- La promoción del empleo apropiado de la tecnología de la información para apoyar el desempeño de la empresa a través de la educación y el ejemplo
- El suministro de infraestructura para la consecución de informaciones corporativos

Integración y unión de las funciones del negocio

A través de las diversas áreas funcionales de la empresa, se desplaza una línea común que está representada por la necesidad de lograr una integración y una unión global entre las funciones corporativas de la empresa. Los directores ejecutivos de información desempeñan el papel que implica el mayor potencial, debido a su participación directa en la infraestructura. Deben considerar todo proceso de la empresa y toda relación con los participantes externos cada vez que modifiquen la infraestructura.

Tecnología informativa

- Los servicios de procesamiento de datos (capacidades de distribución de información)
- Los servicios de telecomunicaciones (capacidades de acceso a la información)
- Los servicios de información (capacidades relativas a necesidades de información)
- Administración de proyectos (capacidades de sistemas de información y capacidades técnicas de los negocios)
- Consultoría técnica y de negocios (capacidades aplicadas a la tecnología informativa y a la consultoría de negocios)

La organización central del área de tecnología informativa necesita posicionarse a un nivel organizacional lo suficientemente alto que le permita integrar las divisiones del negocio dentro de sistemas que abarquen a toda la empresa cuando ello sea apropiado.

Identificación del problema

La inmobiliaria cuenta con diferentes proyectos a lo largo de la república mexicana los cuales duran varios años. Al finalizar cada proyecto en algún punto de la república, por cuestiones fiscales y por reutilización de información para proyectos futuros, esta se almacena y es enviada al archivo general del Distrito Federal (D.F.), en donde se concentra toda la documentación generada durante la vida de dichos proyectos. En la actualidad no existe una herramienta de administración sistematizada que integre las necesidades de recepción de solicitud de envío, ubicación física en el archivo general y control de destrucción en fechas preestablecidas. Al finalizar cada proyecto, se procede a empacar la documentación, colocando en esta un registro de su contenido con la descripción pertinente. Una vez que se ha empacado toda la documentación se elabora una solicitud de almacenamiento, a través de correo electrónico. Acto seguido se envían al archivo general del D. F. por mensajería aún cuando no se ha recibido notificación de aceptación, provocando esto problemas de logística y espacio disponible en el archivo general.

El departamento de Contabilidad y Finanzas debe primero validar con el responsable del archivo general el cupo del número de paquetes de la solicitud, si hay cupo el departamento manda notificación de aceptación del envío, si no hay espacio se solicita al responsable del archivo verificar si existen paquetes que puedan ser destruidos para liberar espacio. El registro y control de los paquetes en el archivo se lleva en bitácoras a papel y lápiz ó en hojas de cálculo, que el almacenista debe validar periódicamente para evitar su eliminación accidental, debido a movimiento ó reubicación no autorizada de expedientes, ya que la destrucción por error provoca una enorme pérdida de información que genera retrasos en tiempos de entrega. El responsable del archivo valida periódicamente la existencia de paquetes vencidos a ser destruidos para dar cupo a futuros envíos. Para esto primero revisa sus registros y posteriormente valida visualmente la existencia de estos, y su contenido para corroborar la concordancia con sus registros; una vez identificados debe pedir permiso al departamento correspondiente para iniciar la destrucción de los mismos. Con respecto a su ingreso y almacenamiento,

una vez colocados los paquetes en su ubicación física, deben ser registrados por el responsable, que en cada caso es diferente y como consecuencia cada uno de ellos los registra de diferente forma en papel sin ninguna normalización.

Al terminar el resguardo de todos los paquetes recibidos, se elabora un reporte, detallando al proyecto al que pertenece, su cantidad y ubicación física. Este reporte lo entrega al departamento responsable, quien lo guarda en una carpeta para posteriores consultas y ya pasado el tiempo estimado proceder a su destrucción. Véase figura 3.4.1 Situación Actual

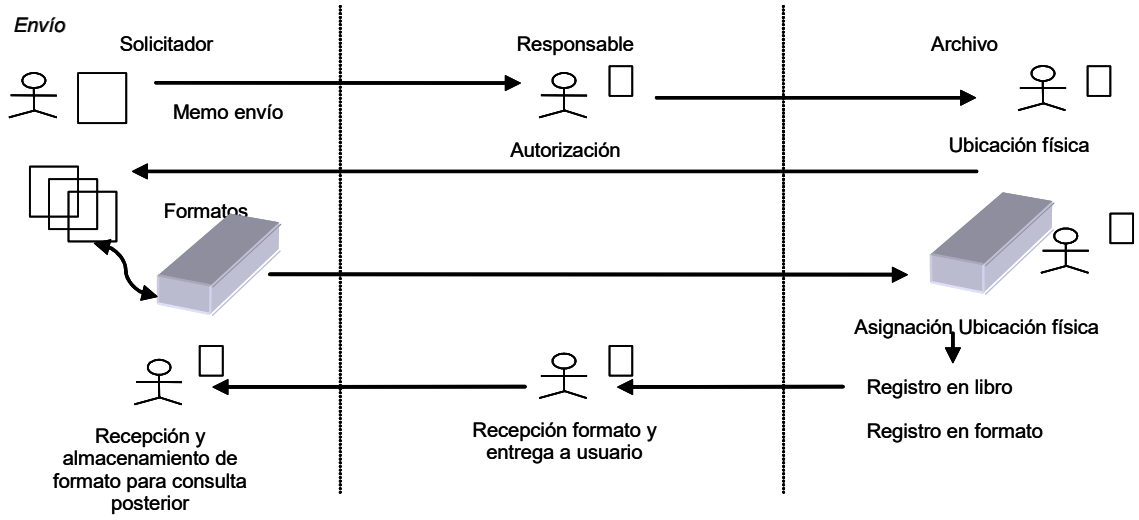


FIGURA 3.4.1 SITUACIÓN ACTUAL

Utilizando el ambiente cooperativo de la empresa (intranet), se dará acceso a los roles identificados (Solicitador y Responsable y Almacenista), a una aplicación que permita controlar de manera centralizada la solicitud de envío de paquetes, autorización, seguimiento de su ubicación física así como su estatus. Diseño y construcción de un sistema de gestión de información para el archivo de empresas inmobiliarias, este ayudará a optimizar el proceso de solicitud, control, envío y almacenamiento de la información generada en esta, y su posterior localización física para consulta del contenido de la misma. Además poseerá un control para su eliminación cuando deje de ser relevante para la empresa. La motivación del desarrollo de dicho sistema surge, por la necesidad que tiene la empresa en agilizar el almacenamiento y la rápida consulta de la

información contenida en el archivo, ya que se lleva actualmente el registro de información en bitácoras, las cuales tienen que ser capturadas manualmente o en hojas de cálculo por el responsable de esta función, el cual no lleva un control eficaz para después recuperar dicha información con la premura que se requiere en cada caso. Esto ocasiona que el trabajo se vaya retrasando debido a este tipo de contingencia. Este sistema deberá permitir el almacenamiento, registro y ubicación en forma organizada, de tal forma que cuando se requiera de esta pueda ser fácilmente localizada y consultada. Por lo cual el sistema propuesto debe cubrir lo siguiente: recepción automatizada de la solicitud de envío al archivo general, seguimiento del envío, registro de la ubicación física del material para la consulta posterior, además de proveer reportes de la ubicación, contenido y el control de destrucción en las fechas establecidas por el usuario. El sistema deberá ser construido con las herramientas de software que posea el usuario para evitar un gasto adicional. Así mismo deberá poseer una base del tipo relacional que contenga toda la información antes mencionada, y que posea un front-end (interfaz de usuario) que cualquier usuario que haya manejado un ambiente gráfico pueda operar sin ningún problema. Para ello deberá contar con todos los catálogos para que el operador elija de estos.

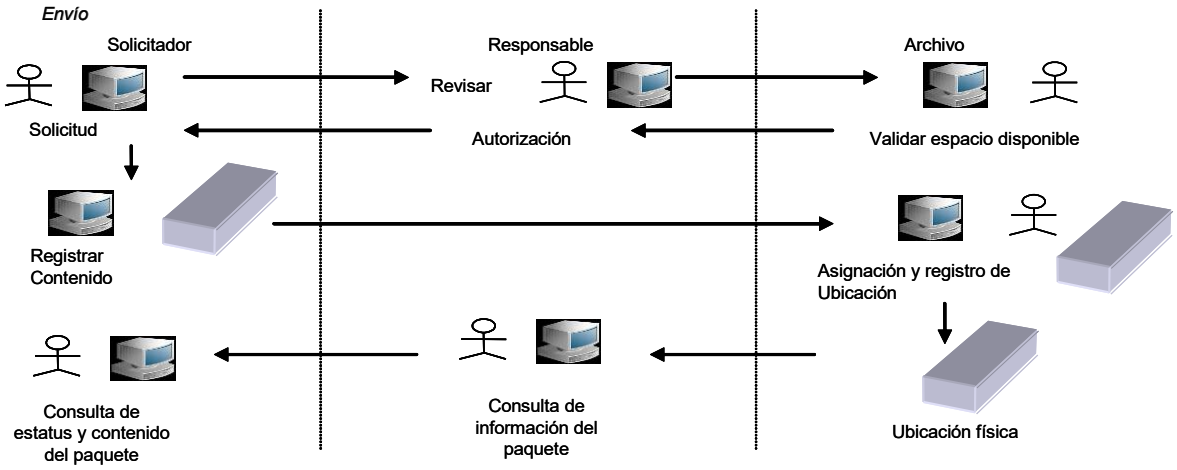


FIGURA 3.4.2 SITUACIÓN PROPUESTA

Resultados esperados

- Acceso a la aplicación desde cualquier equipo PC conectado a una red
- Control centralizado del movimiento de paquetes
- Generación de la aplicación sin costo alto para la empresa
- Colocación de la información directamente en la base de datos
- Acceso a la información dependiendo de los privilegios otorgados por los administradores de la base
- Control efectivo del almacenamiento y recuperación de la información cuando esta se requiera
- Búsqueda y consulta ágil de la información de cada proyecto
- Veracidad y consistencia en el inventario de paquetes
- Que corra en la plataforma informática que posee la empresa
- Optimización de los recursos de la empresa

3.5 OPCIONES DE SOLUCIÓN Y ELECCIÓN DE LA ÓPTIMA

Pruebas de rendimiento del Servidor de Base de Datos

Buscar un rendimiento de datos óptimo como ayuda para elegir entre las distintas tecnologías es tan importante como la creación en si de las estructuras de datos. En pruebas de rendimiento realizadas sobre los manejadores de bases de datos (Obtenido de www.eweek.com artículo Server Database Clash por Timothy Dyck), se encontraron los siguientes resultados que permiten tomar la decisión del manejador de Base de Datos a utilizar para el desarrollo del sistema que nos ocupa. Se realizaron pruebas sobre cinco manejadores de bases de datos, de las cuales elegimos los de SQL Server, MySQL y Oracle. Se seleccionaron estas Bases de Datos debido a que se requiere poner sobre la balanza el manejador con el que cuenta la empresa que es SQL Server, un manejador bien posicionado en el mercado que es Oracle y un manejador de código libre (open source) MySQL.

Las pruebas se enfocaron en obtener los mejores resultados al utilizar un servidor de aplicaciones Java. De las evaluaciones realizadas, todas las Bases de Datos tuvieron un rendimiento estándar cuando se intento dar servicio a 250 usuarios. De ahí en adelante SQL Server tuvo un desempeño pobre con respecto a MySQL y Oracle, al responder solo a un poco mas de 200 peticiones por segundo cuando el número de usuarios se incrementó a mas de 300 usuarios. Este resultado de SQL Server se suscitó por importantes problemas con su manejador JDBC (Java Database Connectivity). Véase Figura 3.5.1 Rendimiento dependiendo de la concurrencia de usuarios. Los manejadores (drivers), puesta a punto de la memoria (memory tuning) y el diseño de la base de datos fueron los tres factores con mayor impacto en el desarrollo de las pruebas, La puesta a punto en forma manual hace gran diferencia en el desempeño de las bases de datos en general, y en la prueba los resultados fueron doblemente rápidos cuando se realizó una puesta a punto correcta de la base de datos. Los manejadores de Oracle y MySQL tuvieron la mejor combinación de características y estabilidad de JDBC.



FIGURA 3.5.1 RENDIMIENTO DEPENDIENDO DE LA CONCURRENCIA DE USUARIOS

SQL Server y MySQL son los manejadores más sencillos en la puesta a punto de memoria, mientras que Oracle es el más complejo. Oracle además requirió la mayor memoria RAM por cada conexión concurrente a la B.D. comparado com MySQL y SQL Server que requieren por conexión menos del 15% de la memoria utilizada por Oracle (400 Kb vs 50 Kb). Debido a esto, los datos y caché de los planes de ejecución deben ser mas pequeños que en las otros manejadores debido a la memoria utilizada por las conexiones de usuario. El buen desempeño de MySQL fue en gran medida debido a la generación de resultados de queries en memoria caché, pues cuando se deshabilitó este caché, los resultados cayeron sustancialmente, es por eso muy importante conocer los procedimientos para optimización de memoria de cada manejador. Véase Figura 3.5.2

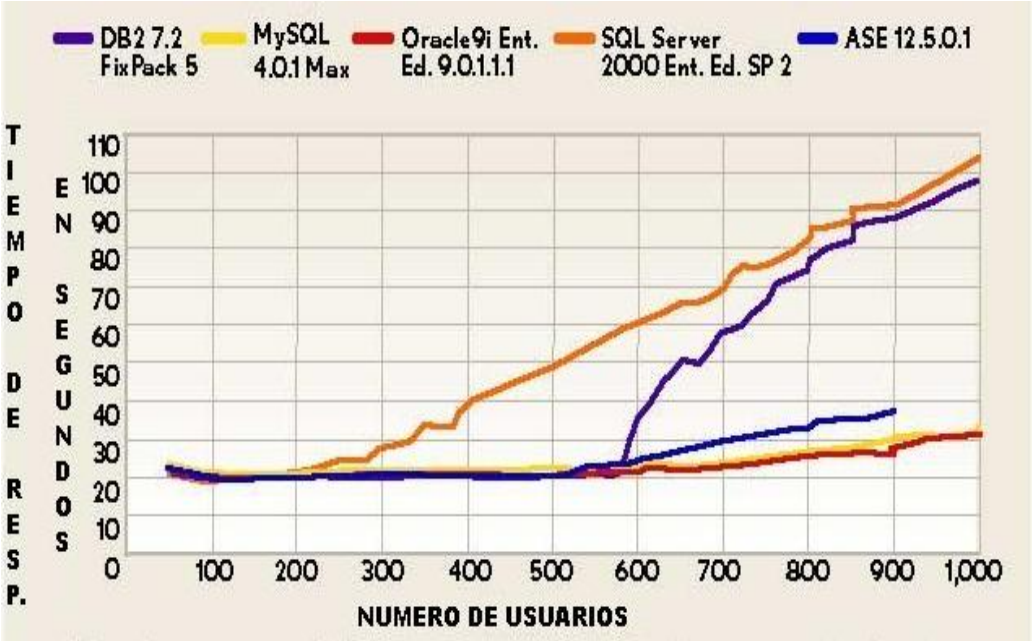


FIGURA 3.5.2 RESPUESTA DEPENDIENDO DE LA CONCURRENCIA DE LOS USUARIOS

Finalmente, al ajustar el diseño de base de datos por si mismo (agregando índices y almacenando los registros de la tabla en el mejor orden físico acorde a los queries ejecutados) permitió una mejora en el rendimiento, aunque fue menos significante que los efectos de los manejadores y la puesta a punto de memoria de la base de datos.

Conclusiones de la prueba

Para las pruebas se utilizó la misma plataforma con cada uno de los servidores de base de datos, utilizando el mismo sistema operativo. Se utilizó una aplicación Web para generar la carga a la base de datos y las pruebas de estrés de la misma con registro de 50 hasta 1000 usuarios concurrentes. El servidor de aplicaciones fue el mismo para todos. Cada prueba corrió alrededor de 1 hora y se generaron unas 50,000 órdenes con entre 150,000 y 200,000 items asociados a las mismas.

Prueba de SQL Server utilizando plataforma .NET

Como prueba extra, se rescribieron las pruebas utilizando Asp.NET, probando solamente SQL Server. Los resultados obtenidos no son equiparables ya que

se utilizaron distinto servidor Web (IIS), diferente motor de la aplicación (ASP.NET) y diferente driver de base de datos (OLE DB). Aún así los resultados indican que esta combinación de productos Microsoft puede producir excelente rendimiento obteniendo unas 870 páginas por segundo.



3.5.3 RENDIMIENTO DE SQL SERVER CON PLATAFORMA .NET



3.5.3 RESPUESTA DE SQL SERVER CON PLATAFORMA .NET

Tabla comparativa sobre funcionalidad del manejador de Base de Datos

La tabla 3.5.1 muestra algunas de las diferencias entre los manejadores de base de datos MySQL, SQL Server y Oracle. Existen muchas otras diferencias

de forma entre los mismos, pero para el proyecto actual, estas no son determinantes para la selección del manejador de Base de Datos.

Descripción	MySQL	SQL Server	Oracle
Longitud de nombre de columna	64	128	30
Longitud de nombre de índice	64	128	30
Longitud de nombre de tabla	64	128	30
Conexiones simultáneas (default)	101	1,000	138
Columnas en una tabla	2,599	1,000	138
Máxima longitud de registro en una tabla (no incluye blobs)	65,534	8,036	255,000
Máxima longitud de registro en una tabla con nulos (no incluye blobs)	65,502	8,036	255,000
Tamaño de Query	10,48,574	16,777,216	16,777,216

TABLA 3.5.1 DIFERENCIAS ENTRE MANEJADORES DE BASE DE DATOS

Conclusiones de las pruebas de rendimiento y funcionalidad de bases de datos

Dados los elementos de análisis de solución disponibles, y considerando el presupuesto y la tecnología informática con la que cuenta actualmente la empresa, se puede identificar como la mejor opción para el desarrollo del proyecto la utilización de una base de datos relacional SQL Server de Microsoft, utilizando como herramienta de programación el lenguaje Java para un desarrollo Intraweb que estará disponible para cada computadora conectada a la red de datos de la empresa, esto a pesar de que las pruebas de rendimiento no fueron satisfactorias, se pudo constatar que el rendimiento del servidor con menos de 250 usuarios recurrentes es muy parecido al rendimiento de MySQL y Oracle, y tomando en cuenta que el sistema no

llegará a dichos niveles es perfectamente entendible la decisión tomada. Otro parámetro de comparación son las pruebas de funcionalidad, en las que no se encontraron elementos suficientes para seleccionar un manejador de base de datos específico, con lo que SQL Server se mantiene como la mejor opción.

Selección de Base de datos

Las ventajas de utilizar una base de datos Microsoft SQL Server como RDBMS se enumeran a continuación:

- **Conocimiento de la herramienta**. Actualmente la mayoría de los sistemas informáticos de la empresa trabajan con este mismo manejador de BD, por lo que el equipo de desarrollo no requiere capacitación sobre la herramienta, lo que además optimiza el tiempo de desarrollo
- **No es necesario adquirir la herramienta**. La empresa cuenta con el manejador de bases de datos, con lo que los gastos son mínimos, aunque existe la posibilidad de requerir mayor número de licencias dependiendo del incremento en el acceso al RDBMS con nuevo sistema
- **No se requiere contratar personal**. Debido a la experiencia adquirida por el personal de la empresa administrando la BD y desarrollando con la misma, no es necesario contratar personal extra
- **No se requiere equipamiento**. El volumen de transacciones calculadas para la nueva aplicación permite establecer que el servidor en el que será instalada la B.D. no requiere crecer en procesamiento o memoria. Las únicas tareas que deberán realizarse son las de puesta a punto del servidor previo a la liberación del sistema

La utilización de un RDBMS distinto a SQL Server, por ejemplo Oracle provocaría retrasos en el desarrollo ya que se tendría que capacitar a los desarrolladores y administradores en la herramienta, además de que el producto y las licencias de usuario aumentarían en forma drástica el costo del desarrollo. El utilizar este producto complicaría además la integración de la información de los distintos sistemas, que en la empresa operan utilizando SQL Server. A pesar de que Oracle es reconocido como una BD muy competitiva, los gastos derivados de su implementación para el sistema no justifican su

utilización, en comparación con SQL Server, en el que la empresa ha invertido tiempo, esfuerzo y dinero. En caso de utilizar una B.D. de código libre como MySQL tampoco justifican su utilización, ya que, a pesar de ser gratuito, el equipo de desarrollo tendría que invertir dinero y tiempo en la capacitación para una adecuada administración de base de datos, o tendría que contratar personal con la experiencia en la utilización de este manejador para poder entregar resultados inmediatos.

Selección de la plataforma de programación

El desarrollo de aplicaciones empresariales, la colaboración tanto entre departamentos como entre empresas, y el desarrollo de aplicaciones web han sufrido un auge muy importante durante los últimos años. Frente a esta nueva demanda surgen dos plataformas distintas para el desarrollo de este tipo de aplicaciones: J2EE de Sun y .NET de Microsoft.

Microsoft .NET

.NET es una plataforma llena de servicios para construir aplicaciones basadas en web y desarrollar experiencias interactivas para los usuarios y sus sistemas.

La arquitectura de la plataforma .NET se puede dividir principalmente en:

- **.NET framework**. Es la parte más importante de la plataforma .NET. Incluye COM+, un entorno de ejecución común, un compilador JIT, y un conjunto de librerías de sistema que dan acceso a un amplio conjunto de servicios
- **Servidores .NET.** Son un conjunto de aplicaciones que pueden usarse en conjunción con el .NET framework para facilitar el desarrollo de aplicaciones empresariales. Como por ejemplo SQL Server 2000, Exchange 2000 server o BizTalk Server 2000

Java 2 Entreprise Edition (J2EE)

J2EE define un estándar para el desarrollo de aplicaciones empresariales multicapa. J2EE simplifica las aplicaciones empresariales basándolas en

componentes modulares y estandarizados, proveyendo un completo conjunto de servicios a estos componentes, y manejando muchos de las funciones de la aplicación de forma automática, sin necesidad de una programación compleja. A diferencia de Microsoft .NET que es un producto, J2EE es un estándar. Por lo tanto, no es posible descargarte J2EE sino que es necesario adquirir alguna de las versiones de plataformas de desarrollo basadas en J2EE que existen en el mercado como IBM WebSphere, BEA Weblogic, Oracle9iAS o Sun ONE. Cada una estas versiones proporcionan servicios añadidos a los propuestos en el estándar.

La arquitectura de J2EE se puede dividir en cinco partes:

- El lenguaje de programación java
- El modelo de programación del cliente
- La infraestructura de la capa de middleware
- La API de negocios para los programadores
- La API no visible para los programadores

Analogías y diferencias entre J2EE y .NET

Veamos de forma resumida las similitudes y diferencias tanto tecnológicas como de servicios entre Microsoft .NET y J2EE. Veremos que todos los conceptos aparecen en ambas arquitecturas. Tabla 3.5.2

CARACTERÍSTICA	.NET	J2EE
Tipo de tecnología	Producto	Estándar
Empresas que lo ofrecen	Microsoft	Más de 30
Librerías de desarrollo	.NET Framework SDK	Java core API
Intérprete	CLR	JRE
Páginas dinámicas	ASP.NET	Servlets, JSP
Componentes	.NET Managed Components	EJB
Acceso a bases de datos	ADO.NET	JDBC, SQL/J
Servicios Web	SOAP, WDSL, UDDI	SOAP, WDSL, UDDI
Interfaces gráficas	Win Forms y Web Forms	Java Swing
Herramienta de programación	Visual Studio .NET	Depende del fabricante
Transacciones distribuidas	MS-DTC	JTS
Servicios de directorios	ADSI	JNDI
Librería de encolado de mensajes	MSMQ	JMS 1.0
Lenguajes utilizados	C#, Visual Basic, C++, otros	Java
Lenguaje intermedio	IL	Bytecodes

TABLA 3.5.2. SIMILITUDES Y DIFERENCIAS ENTRE MICROSOFT .NET Y J2EE

Los parecidos entre las dos plataformas en los aspectos principales son:

Un lenguaje de programación que unido a una librería de clases, compila a un código intermedio independiente de la máquina ("Intermediate Language" en el caso de Microsoft .NET y "Bytecodes" en el caso de Java). Este código se ejecutará en máquina virtual, que proporciona un "entorno de ejecución" que transformará el lenguaje intermedio a código propio de la máquina en la que se corre la aplicación ("Common Language Runtime (CLR)" en Microsoft .NET y "Java Runtime Environment (JRE)" en J2EE).

Las dos plataformas disponen de multitud de servicios que facilitan la labor del programador, entre ellos, el acceso a bases de datos, los servicios de directorio, las transacciones distribuidas.

Lenguajes de programación

El único lenguaje que soporta J2EE es Java y es el que se tendrá que utilizar para desarrollo de todos los componentes. Existen sólo dos formas oficiales para acceder a la plataforma J2EE con otros lenguajes, la primera es a través de JNI (Java Native Interface) y la segunda es a través de la interoperabilidad que ofrece CORBA. Por otra parte, Microsoft .NET ofrece soporte oficial para Visual Basic.NET, C++.NET y un nuevo lenguaje C# que es equivalente (con la excepción de portabilidad) a Java. Microsoft .NET va más allá de soportar estos lenguajes sino que también ofrece plena interoperabilidad entre ellos, por lo que es posible construir un componente en un lenguaje, introducirlo en una aplicación escrita en otro distinto e incluso heredarlo y añadir nuevas características en un tercero.

Respecto a esta capacidad de Microsoft .NET de trabajar con varios lenguajes existen dos posturas enfrentadas:

- **Por un lado** están lo que están a favor de esta capacidad los cuales alegan que permite una migración más sencilla para antiguos programadores, reduciendo el tiempo de formación. Además, trabajar con un lenguaje conocido proporciona gran productividad individual
- **Por otro lado,** los detractores (al menos con algunas limitaciones), explican que la existencia de varios lenguajes de programación en una única empresa acarrea efectos negativos:
    - **La sencillez de mantenimiento se reduce.** Si una aplicación está realizada en varios lenguajes se necesitan expertos en varios lenguajes para entenderla y mantenerla, aumentando los costes considerablemente

- **La productividad del grupo decrece.** Si los programadores utilizan lenguajes diferentes no pueden comunicar fácilmente sus conocimientos de unos a otros
- **Transferencia de conocimientos**. En el caso de que un desarrollador o grupo de desarrolladores especializados en un lenguaje dejen un proyecto, se necesitan otros que conozcan el mismo lenguaje para continuar con el desarrollo. Por tanto será necesario entrenar a los existentes para que construyan aplicaciones útiles y en poco tiempo

Conclusiones

Tomando como referencia las plataformas de desarrollo analizadas, podemos concluir que las dos tienen todos los elementos requeridos para el desarrollo del sistema, por lo que la decisión se centra en los costos, el conocimiento actual y la rapidez en el desarrollo. Para el proyecto decidimos utilizar la plataforma J2EE con lenguaje de programación Java por las siguientes razones:

- **Conocimiento de la herramienta**. El equipo de desarrollo de la empresa ha adquirido experiencia en la utilización de la misma, por lo que no se tendrán gastos en capacitación
- **No es necesario adquirir la herramienta**. Este tipo de herramienta es de utilización libre, además de que existen varias herramientas o entornos de programación gratuitos
- **Reutilización de librerías**. El equipo de desarrollo cuenta con librerías propias que permitirán reutilizar código, y por lo tanto optimizar el tiempo de construcción del sistema
- **No se requiere contratar personal**. El personal de desarrollo tiene suficiente experiencia en el manejo del entorno de programación

4.1 APLICACIÓN DE LA METODOLOGÍA ELEGIDA (BACK-END PARA CADA MÓDULO UML)

4.1.1 DIAGRAMA DE CONTEXTO Y DIAGRAMA DE CASOS DE USO

Diagrama de contexto

Un diagrama de contexto indica las interacciones del sistema con entidades externas a este, como lo muestra la figura 4.1.1.1

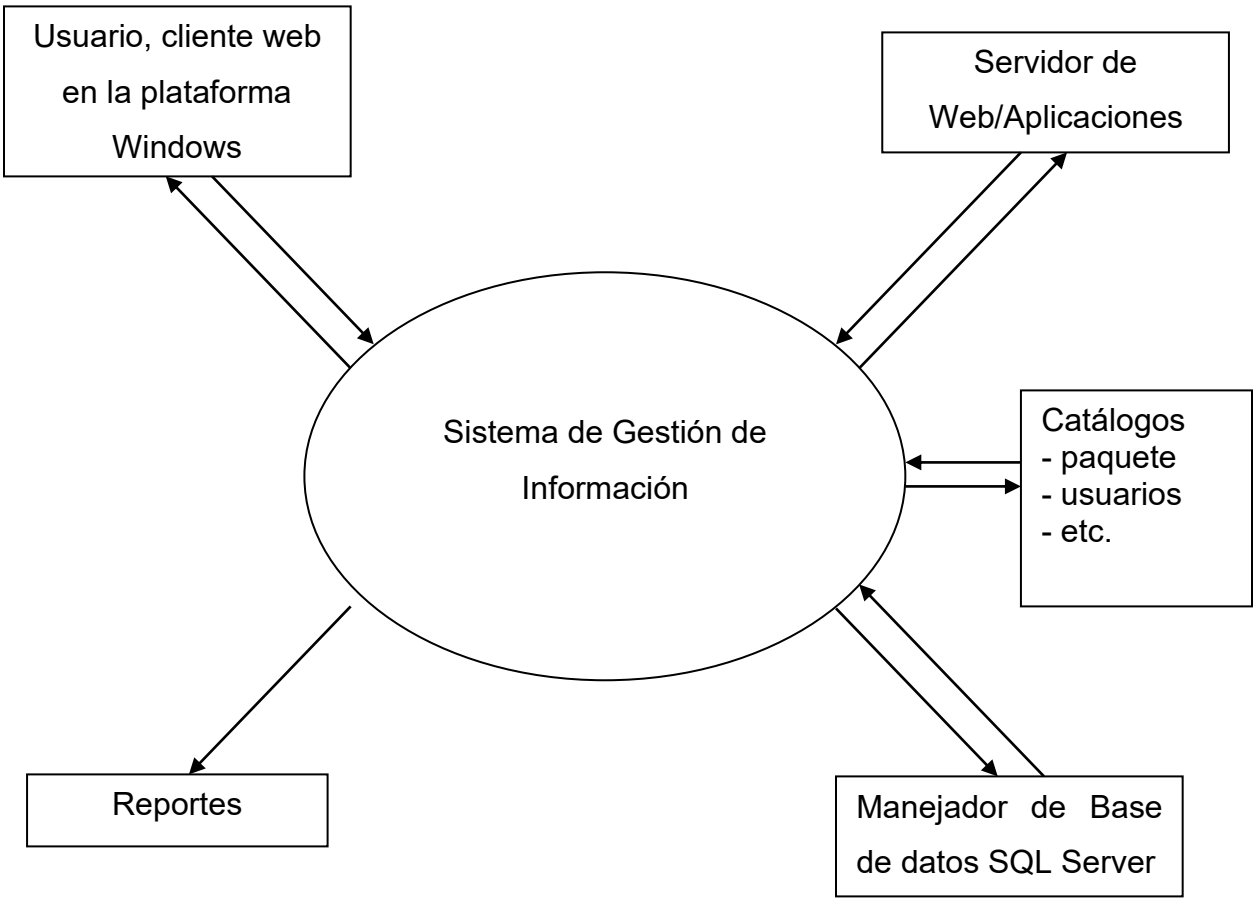


FIGURA 4.1.1.1 DIAGRAMA DE CONTEXTO

Diagrama de casos de uso

Un caso de uso es la funcionalidad que se espera sea satisfecha por el sistema y que ha sido solicitada por un rol de este. Analizando los requerimientos solicitados, se tiene lo siguiente:

Roles identificados:

- Solicitador del envío o préstamo
- Responsable del departamento de Administración
- Almacenista

Tareas que los usuarios del sistema deben realizar con la ayuda de este:

- Solicitar autorización de envío de paquete
- Verificar correcto llenado de la solicitud
- Modificar la solicitud
- Confirmar espacio para el envío
- Autorizar envío
- Consultar el estatus del envío
- Registrar el contenido del paquete
- Notificar envío del paquete
- Registrar la ubicación del paquete
- Notificar fin del envío
- Consultar el contenido de los paquetes
- Solicitar el préstamo de un paquete
- Consultar el estatus del paquete
- Autorizar préstamo
- Consultar la ubicación del paquete
- Notificar fin del préstamo
- Consultar la fecha de destrucción de los paquetes
- Administración de catálogos (Usuario, Proyecto)

Todo esto está modelado en los diagramas de casos de uso de las figuras 4.1.1.2 a la 4.1.1.41

Diagramas de Casos de Uso

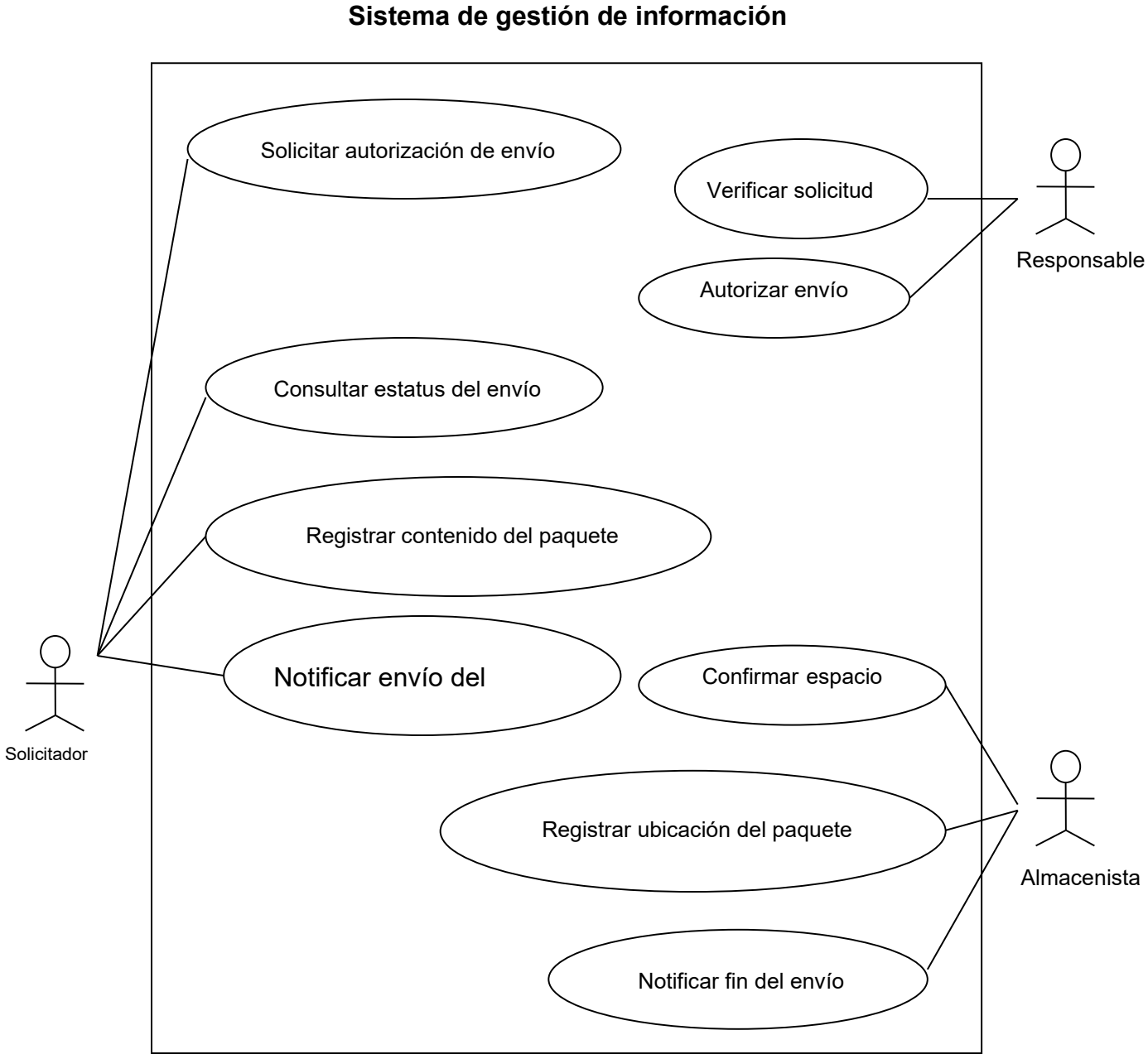


FIGURA 4.1.1.2 DIAGRAMA DE CASOS DE USO DEL ENVÍO

Nivel 2 de casos de uso del envío

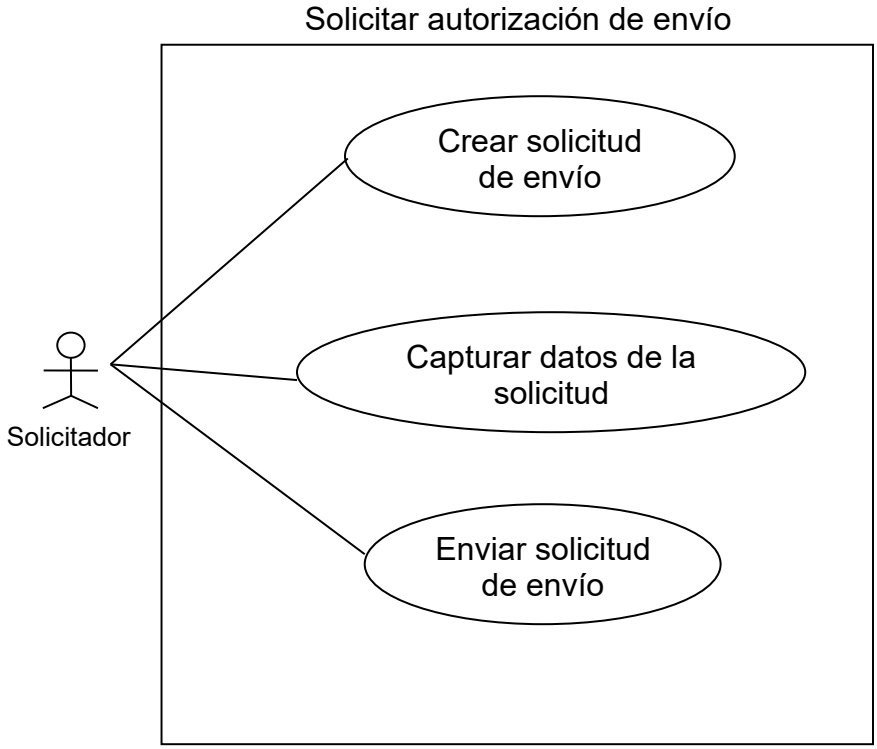


FIGURA 4.1.1.3 DIAGRAMA DE CASOS DE USO DEL ENVÍO

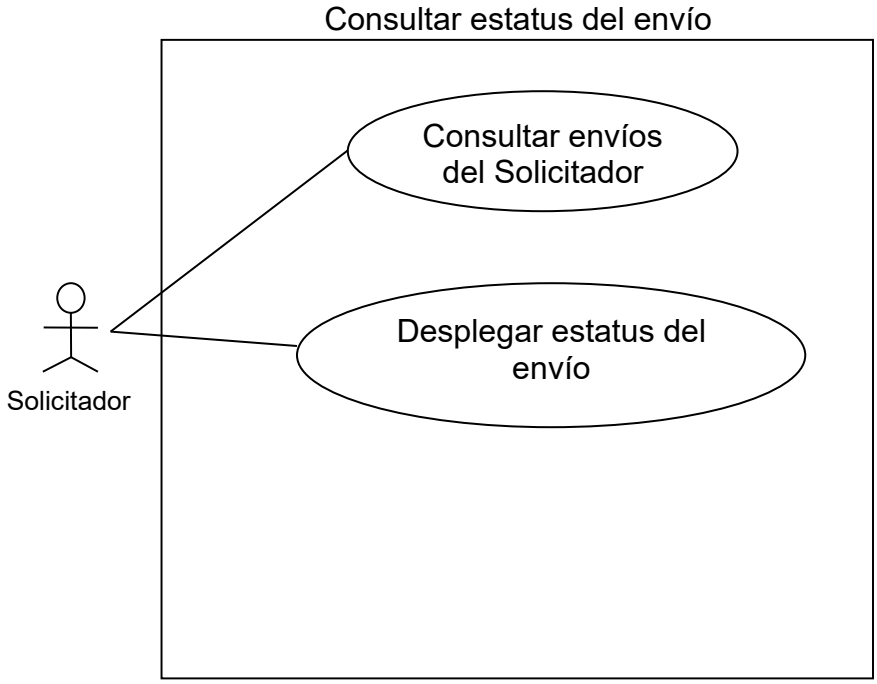


FIGURA 4.1.1.4 DIAGRAMA DE CASOS DE USO DEL ENVÍO

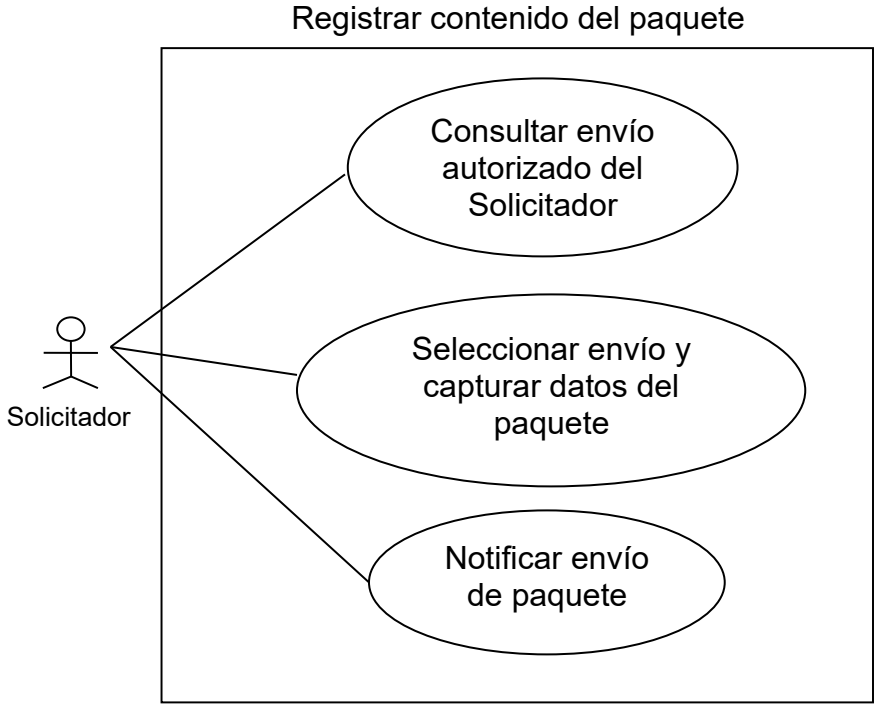


FIGURA 4.1.1.5 DIAGRAMA DE CASOS DE USO DEL ENVÍO

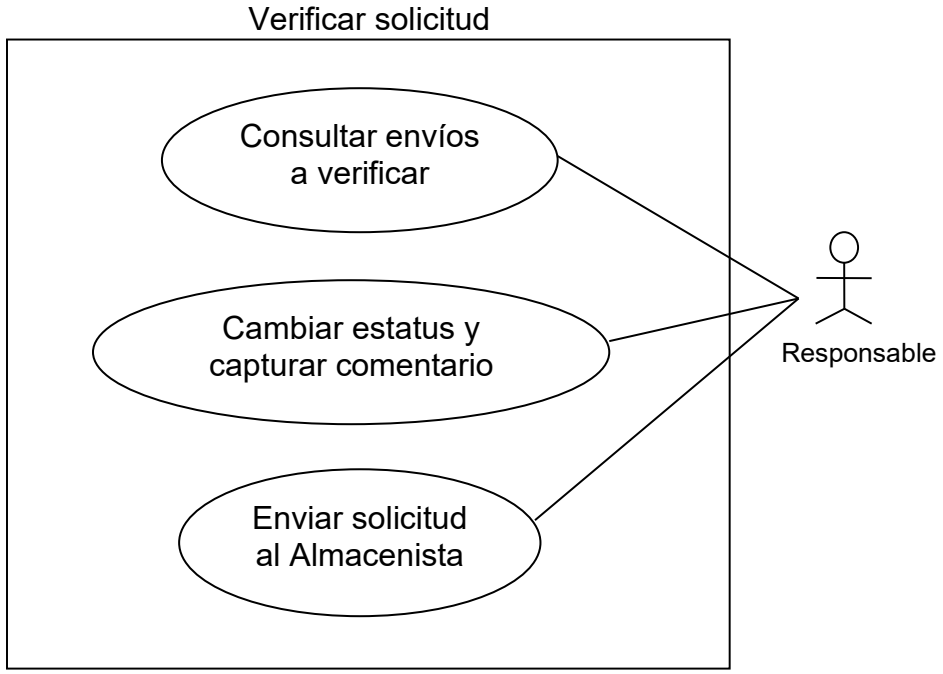


FIGURA 4.1.1.6 DIAGRAMA DE CASOS DE USO DEL ENVÍO

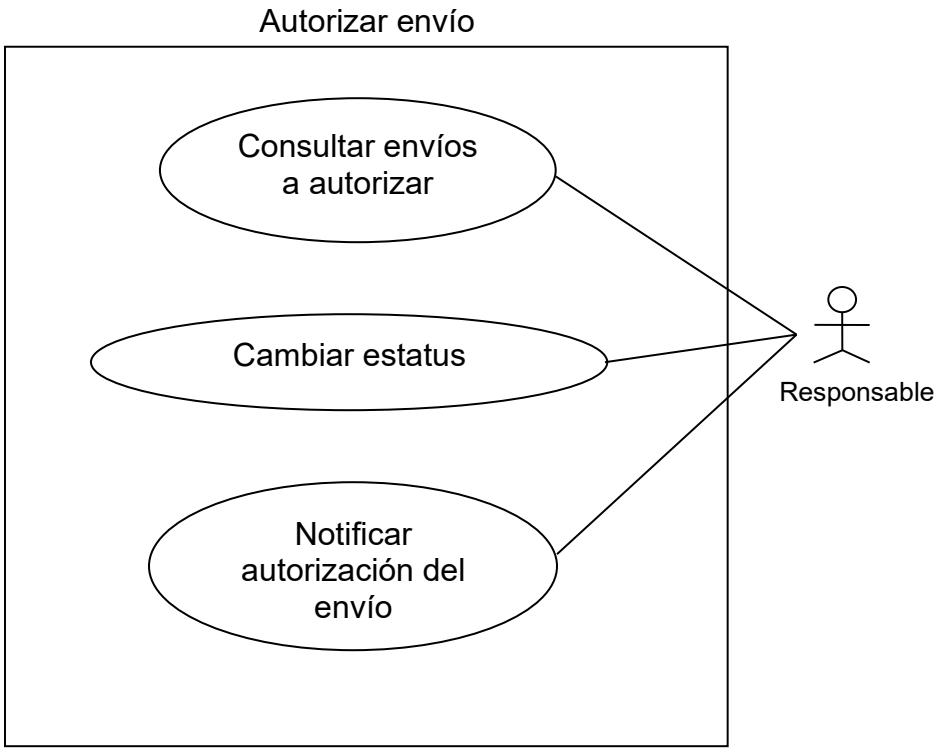


FIGURA 4.1.1.7 DIAGRAMA DE CASOS DE USO DEL ENVÍO

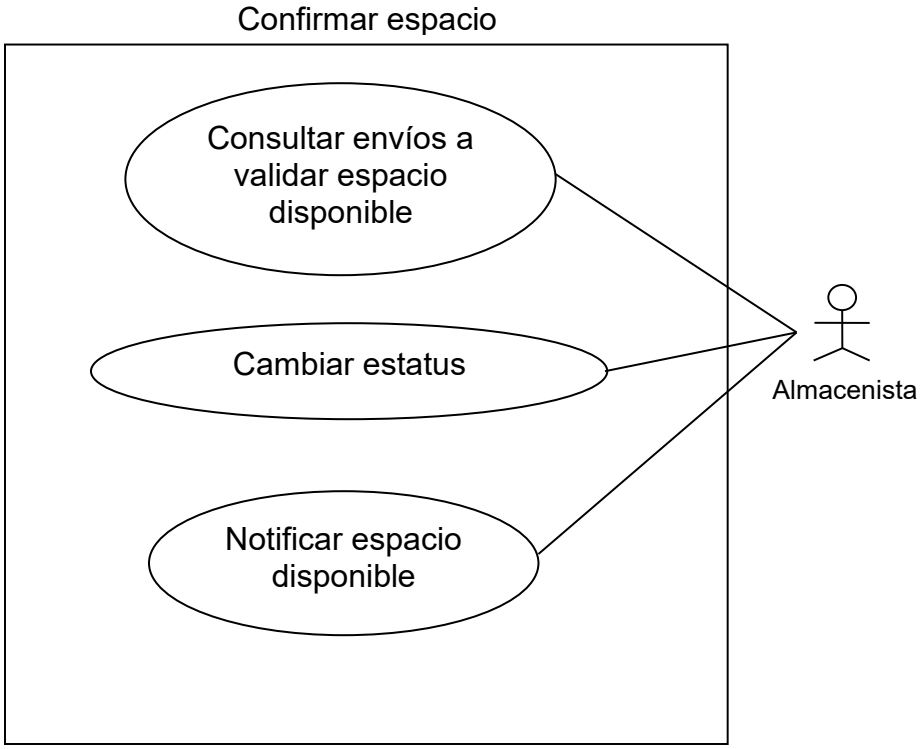


FIGURA 4.1.1.8 DIAGRAMA DE CASOS DE USO DEL ENVÍO

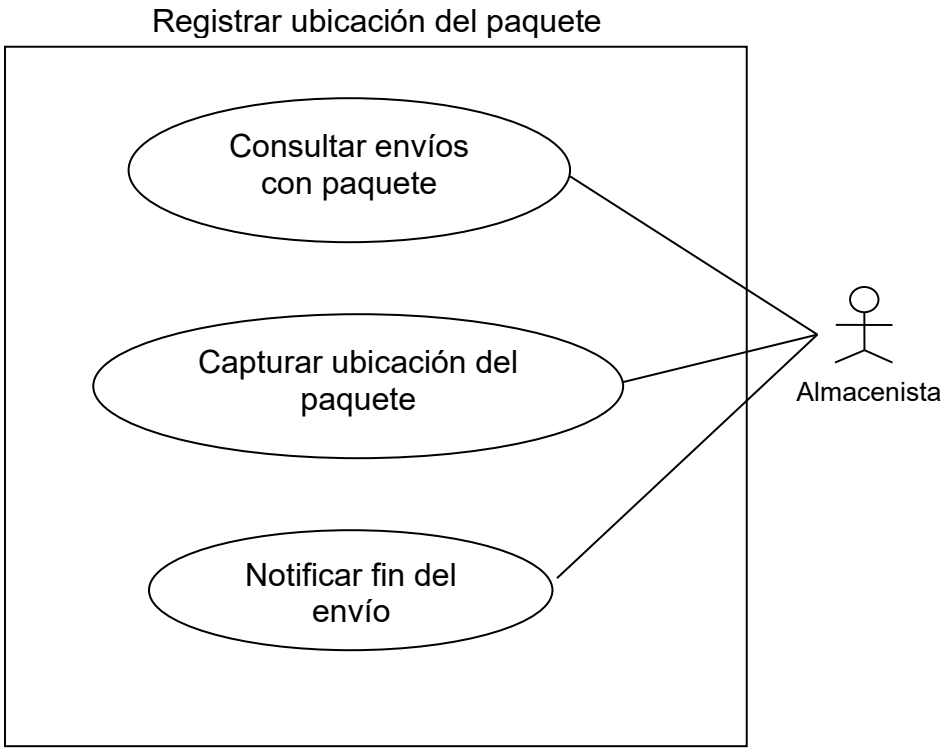


FIGURA 4.1.1.9 DIAGRAMA DE CASOS DE USO DEL ENVÍO

Nivel 3 de casos de uso del envío

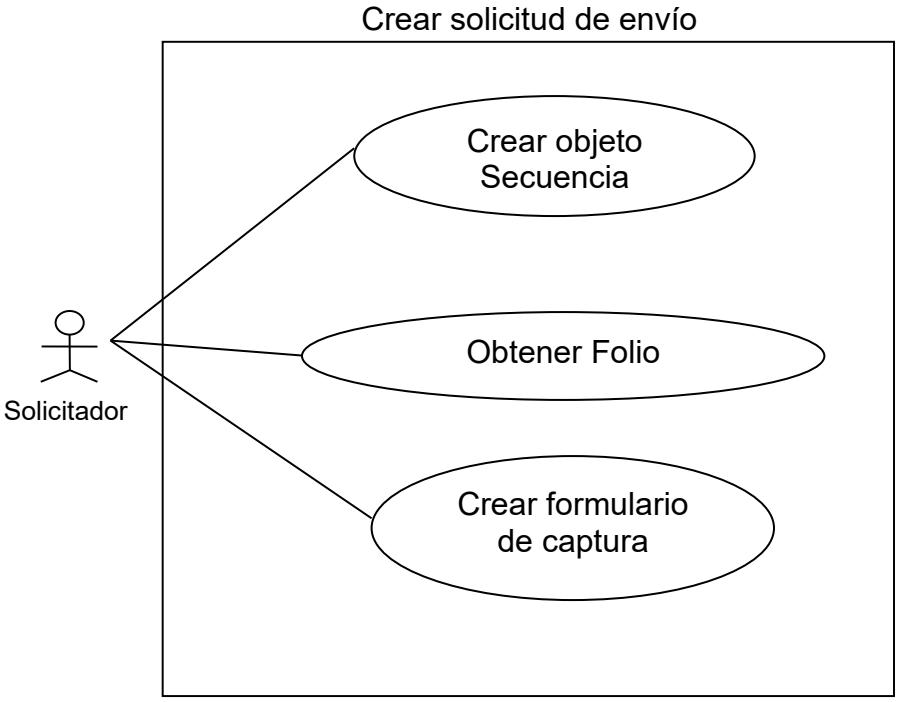


FIGURA 4.1.1.10 DIAGRAMA DE CASOS DE USO DEL ENVÍO

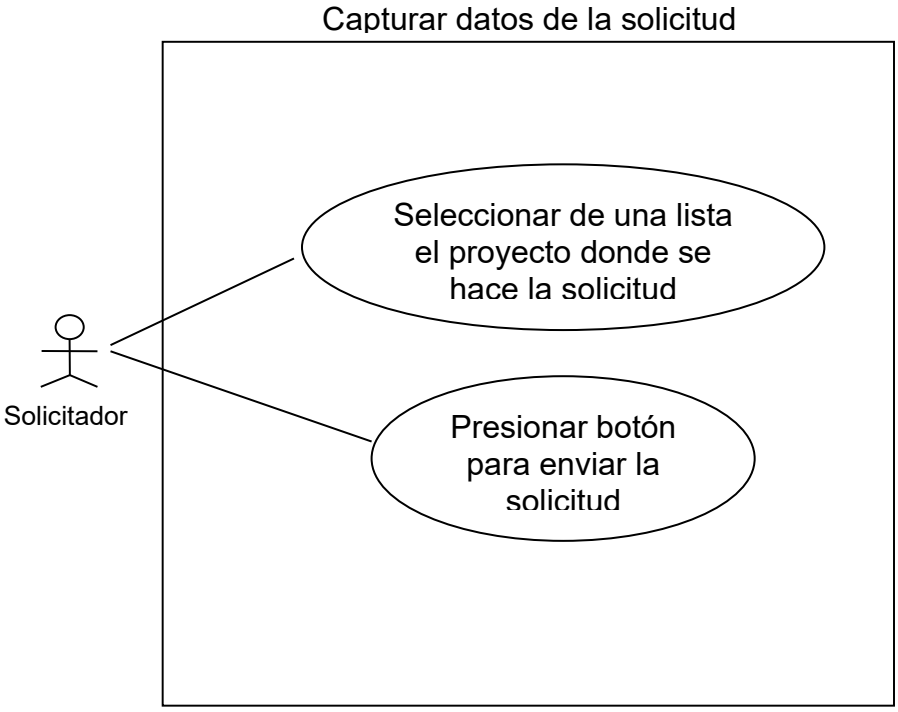


FIGURA 4.1.1.11 DIAGRAMA DE CASOS DE USO DEL ENVÍO

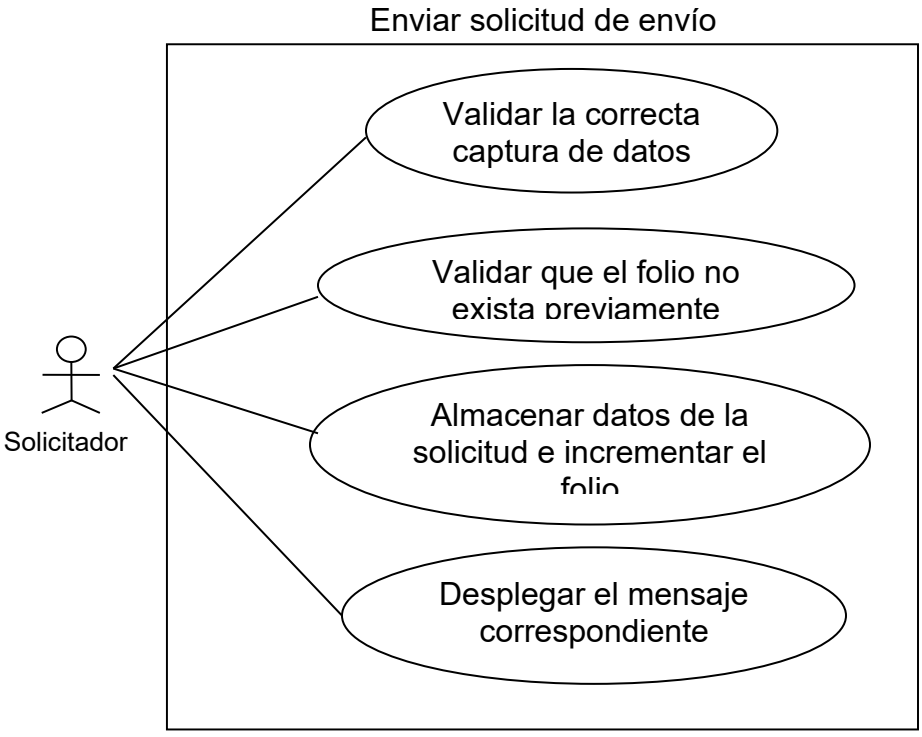


FIGURA 4.1.1.12 DIAGRAMA DE CASOS DE USO DEL ENVÍO

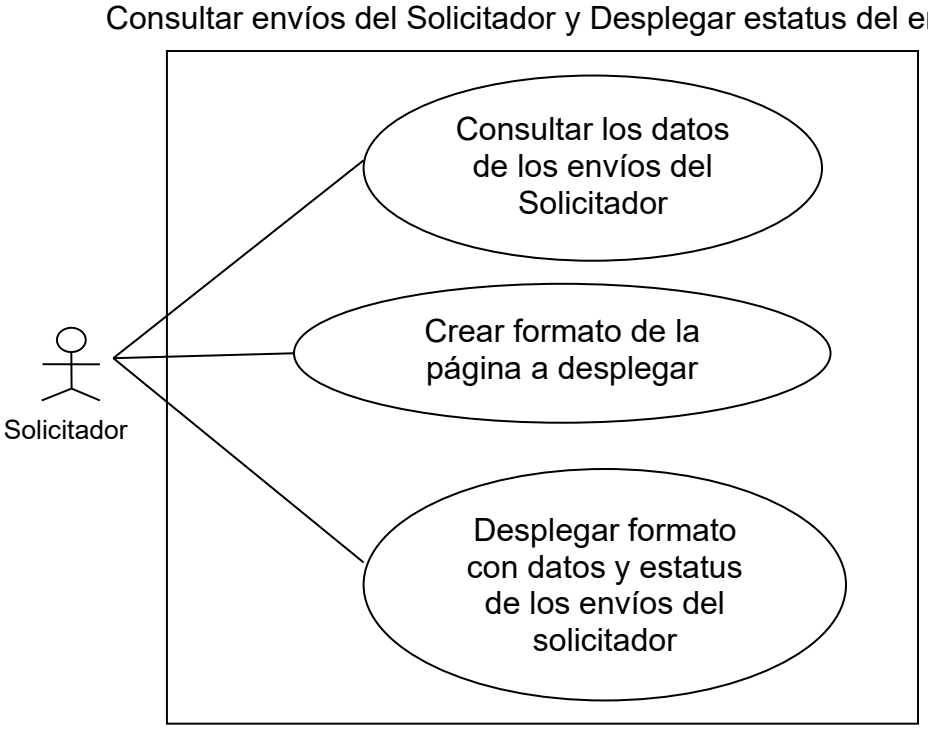


FIGURA 4.1.1.13 DIAGRAMA DE CASOS DE USO DEL ENVÍO

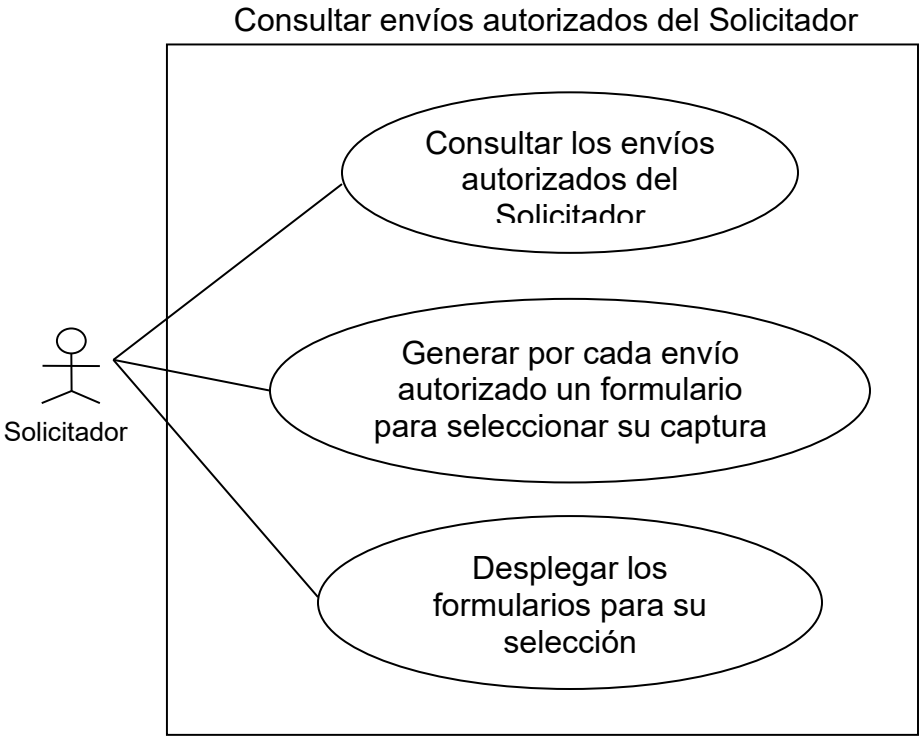


FIGURA 4.1.1.14 DIAGRAMA DE CASOS DE USO DEL ENVÍO

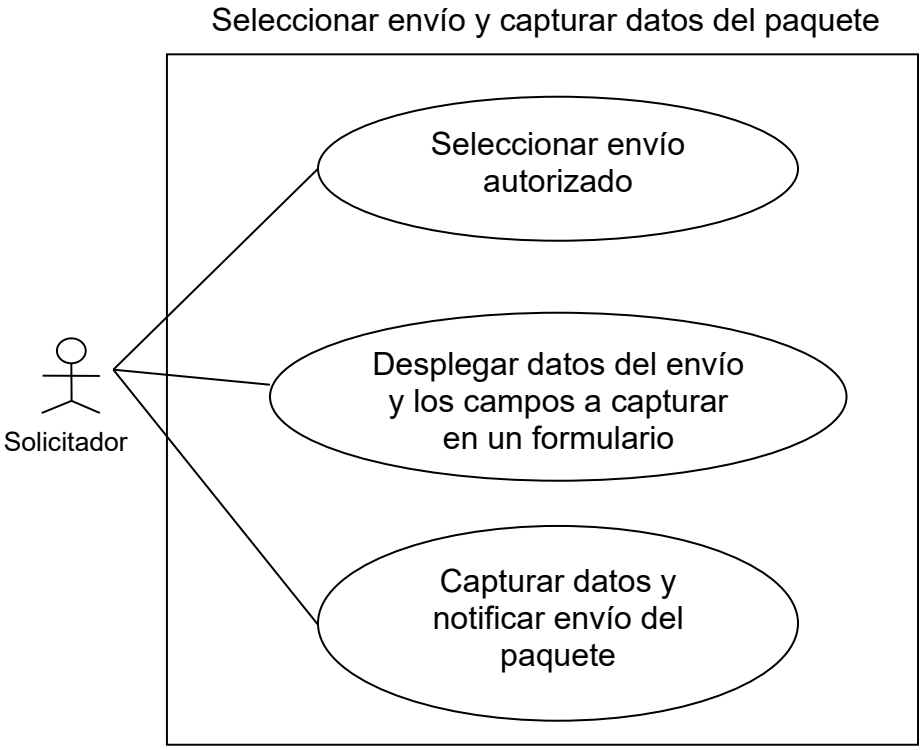


FIGURA 4.1.1.15 DIAGRAMA DE CASOS DE USO DEL ENVÍO

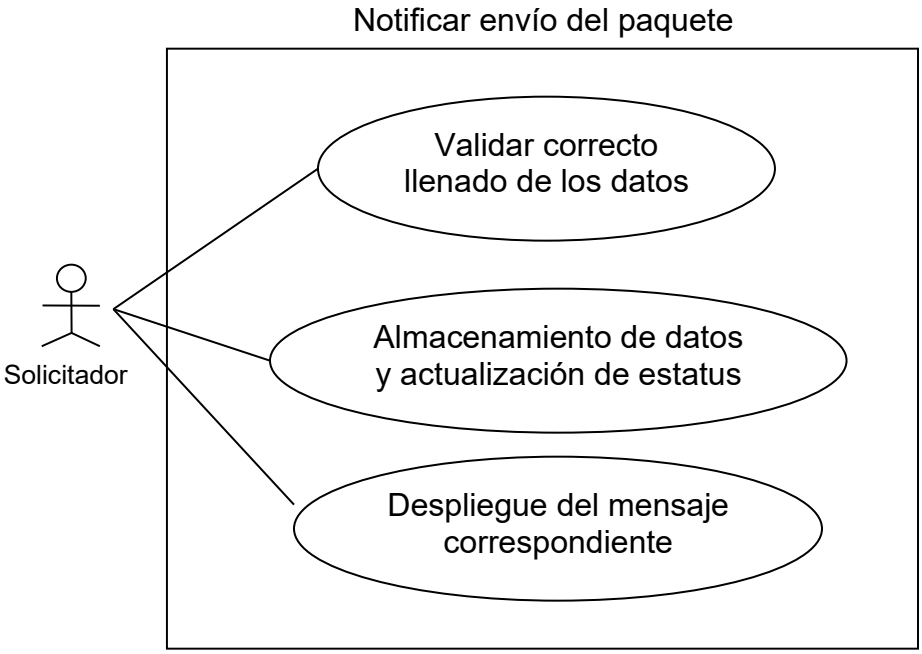


FIGURA 4.1.1.16 DIAGRAMA DE CASOS DE USO DEL ENVÍO

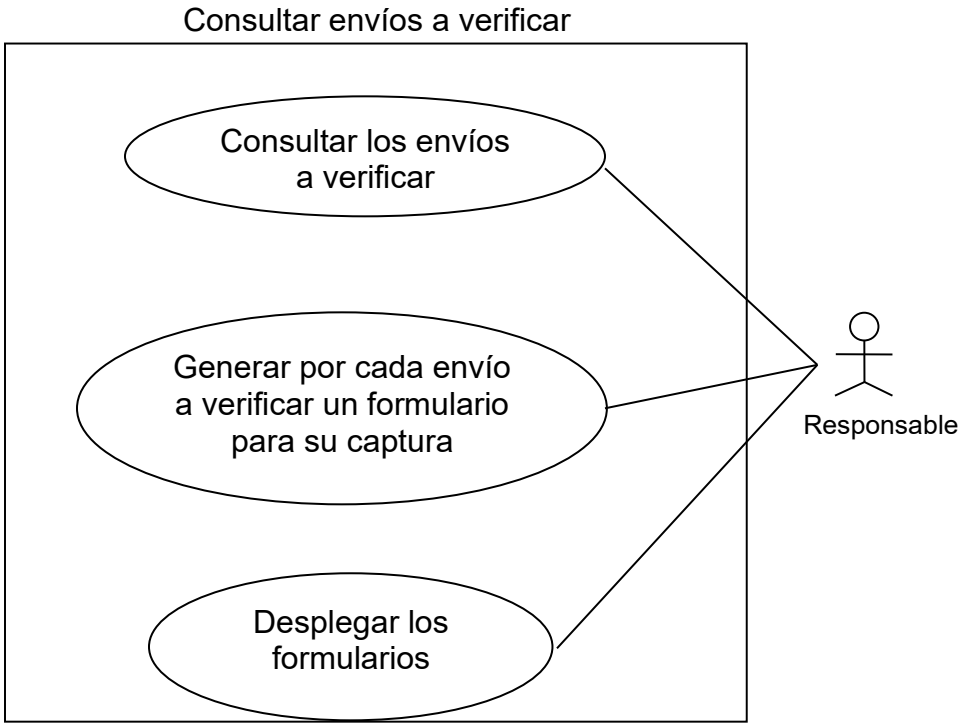


FIGURA 4.1.1.17 DIAGRAMA DE CASOS DE USO DEL ENVÍO

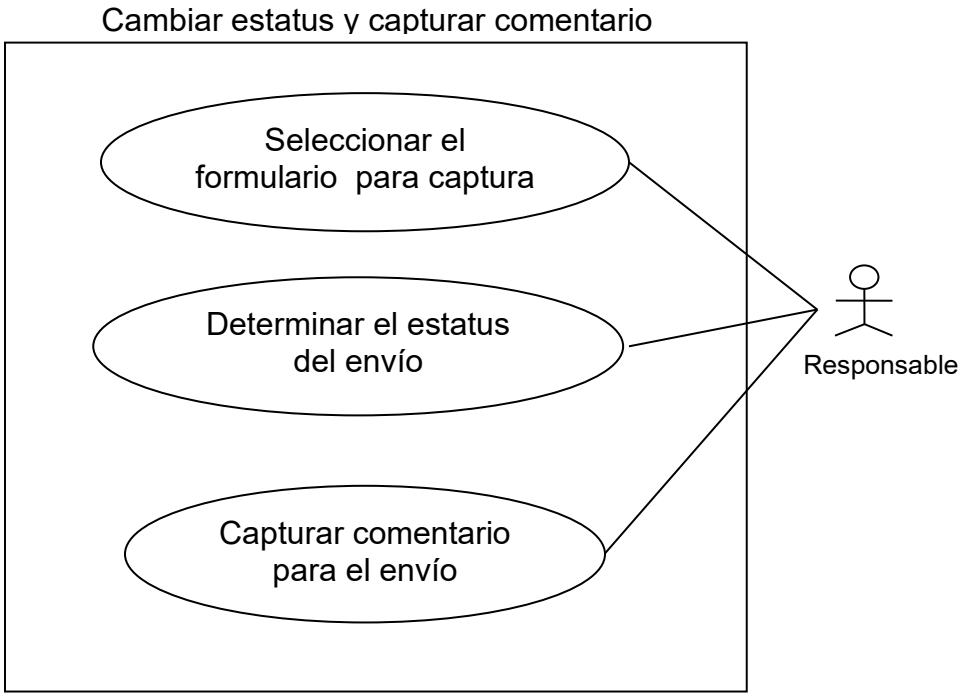


FIGURA 4.1.1.18 DIAGRAMA DE CASOS DE USO DEL ENVÍO

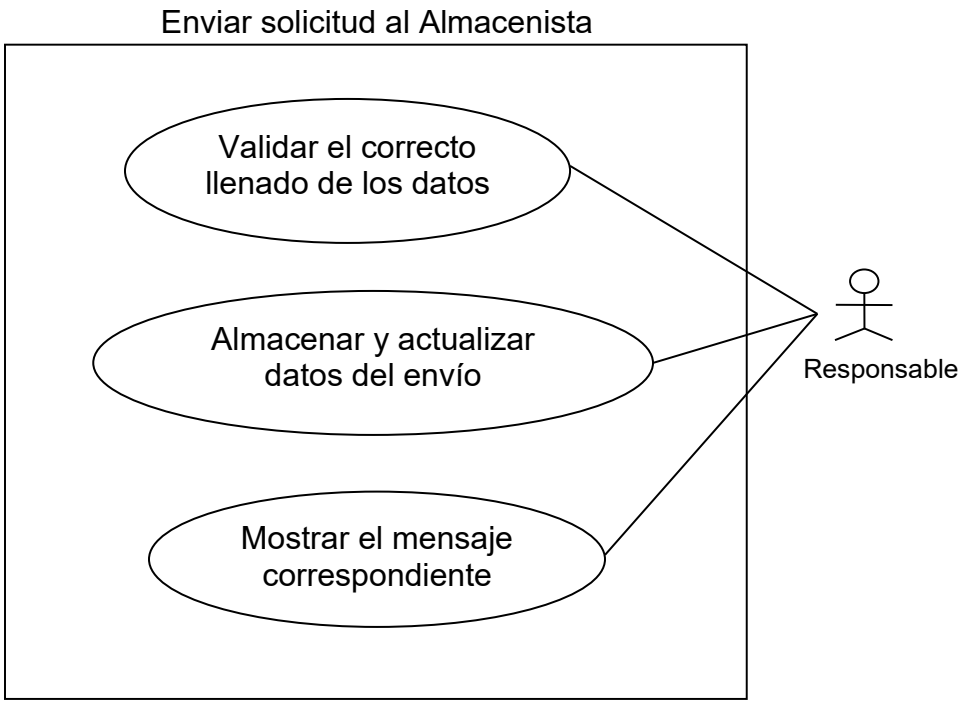


FIGURA 4.1.1.19 DIAGRAMA DE CASOS DE USO DEL ENVÍO

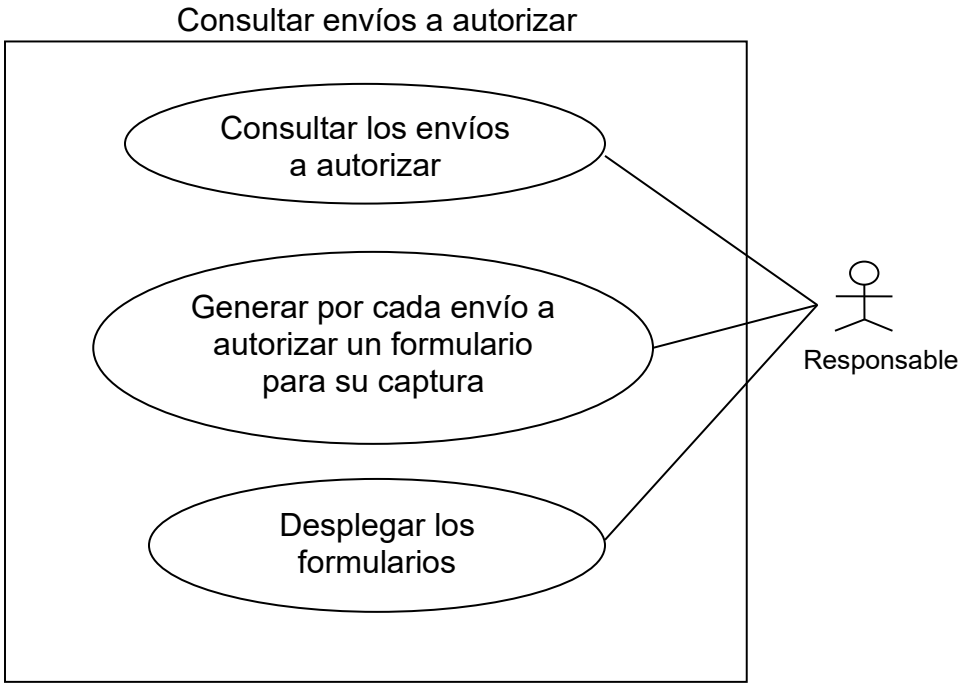


FIGURA 4.1.1.20 DIAGRAMA DE CASOS DE USO DEL ENVÍO

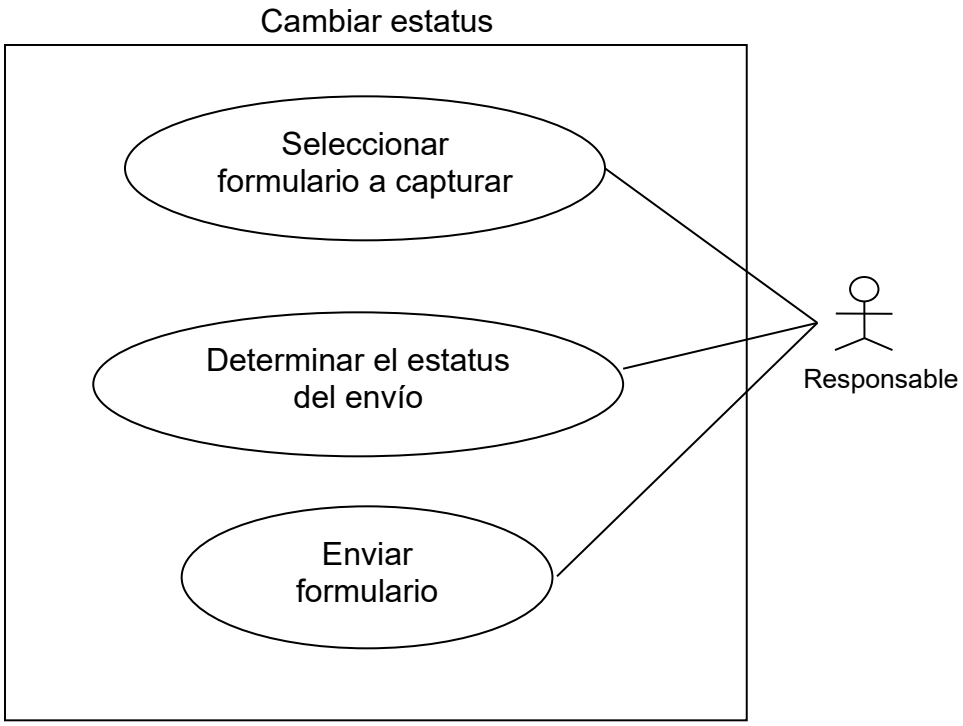


FIGURA 4.1.1.21 DIAGRAMA DE CASOS DE USO DEL ENVÍO

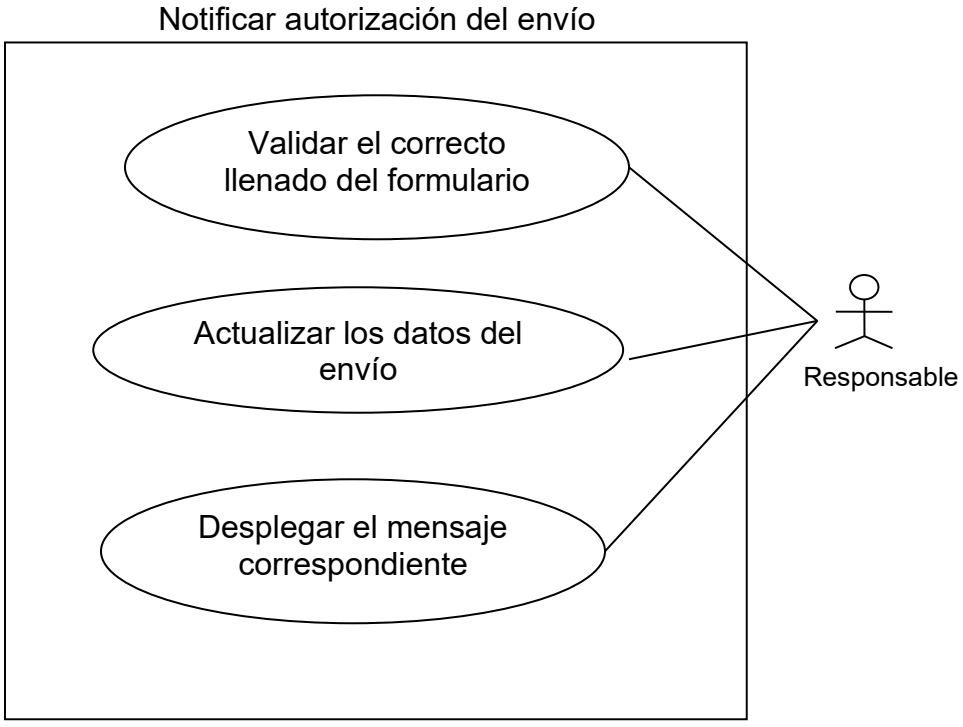


FIGURA 4.1.1.22 DIAGRAMA DE CASOS DE USO DEL ENVÍO

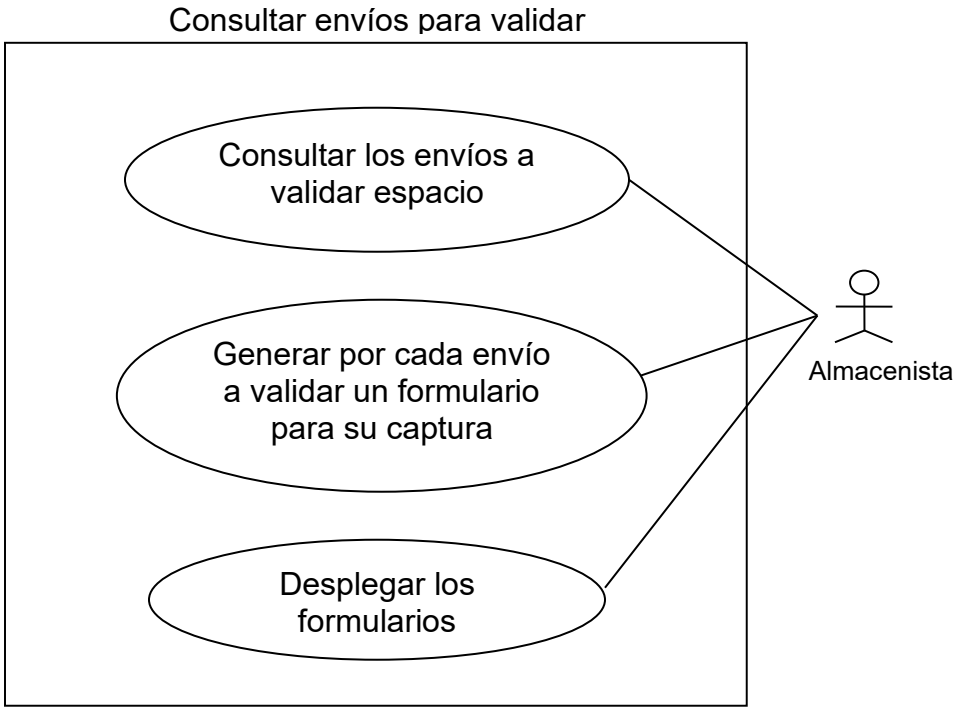


FIGURA 4.1.1.23 DIAGRAMA DE CASOS DE USO DEL ENVÍO

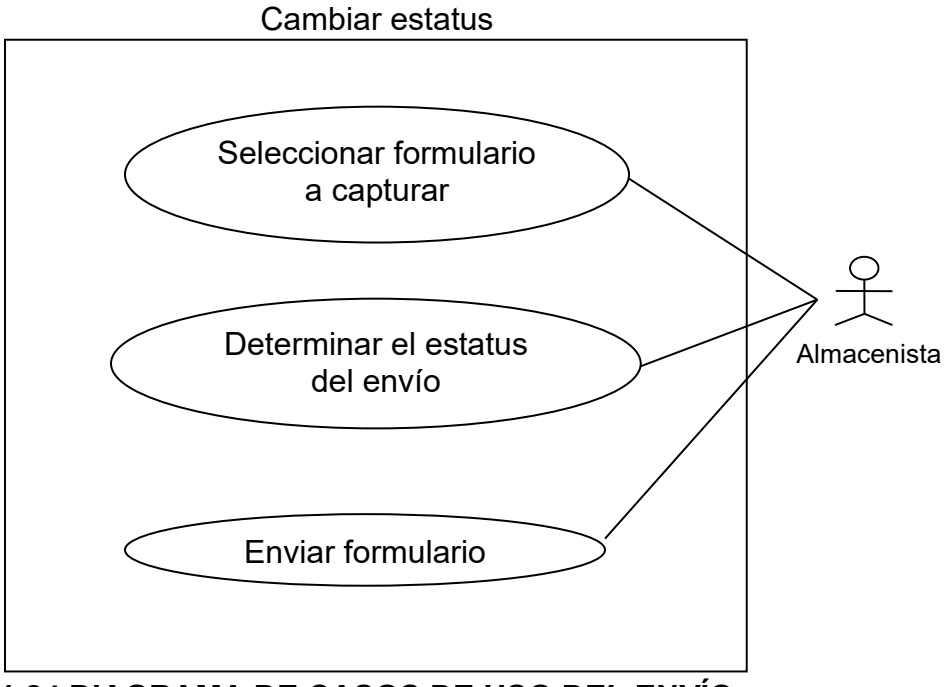


FIGURA 4.1.1.24 DIAGRAMA DE CASOS DE USO DEL ENVÍO

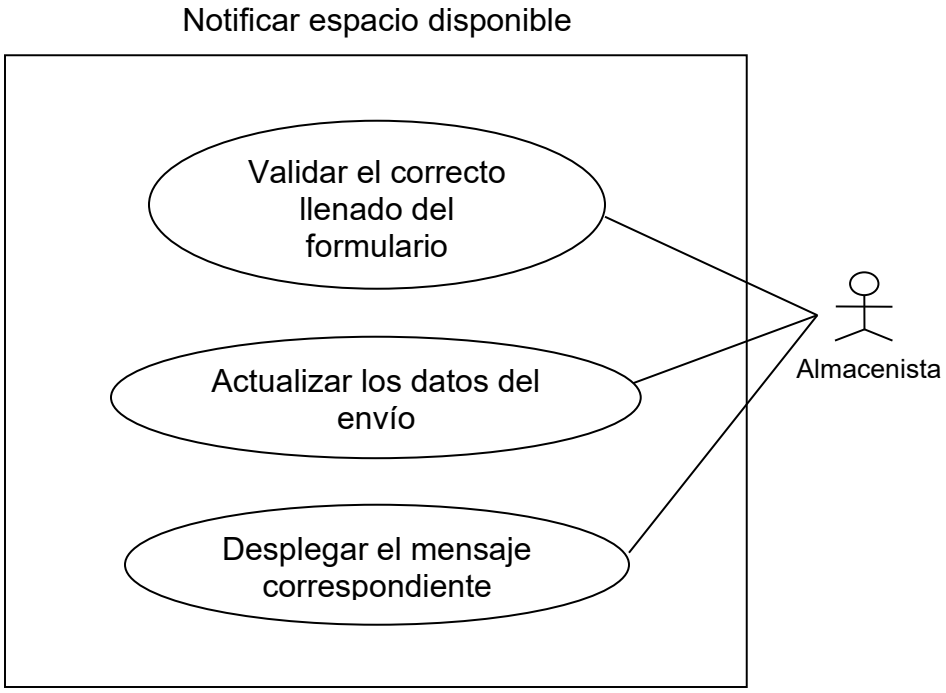


FIGURA 4.1.1.25 DIAGRAMA DE CASOS DE USO DEL ENVÍO

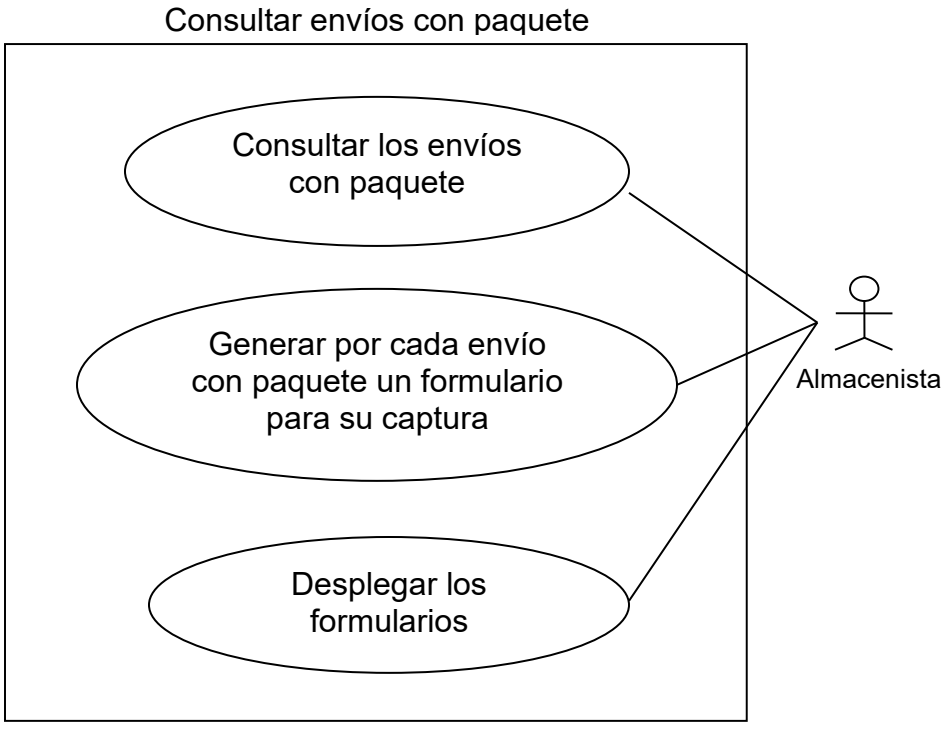


FIGURA 4.1.1.26 DIAGRAMA DE CASOS DE USO DEL ENVÍO

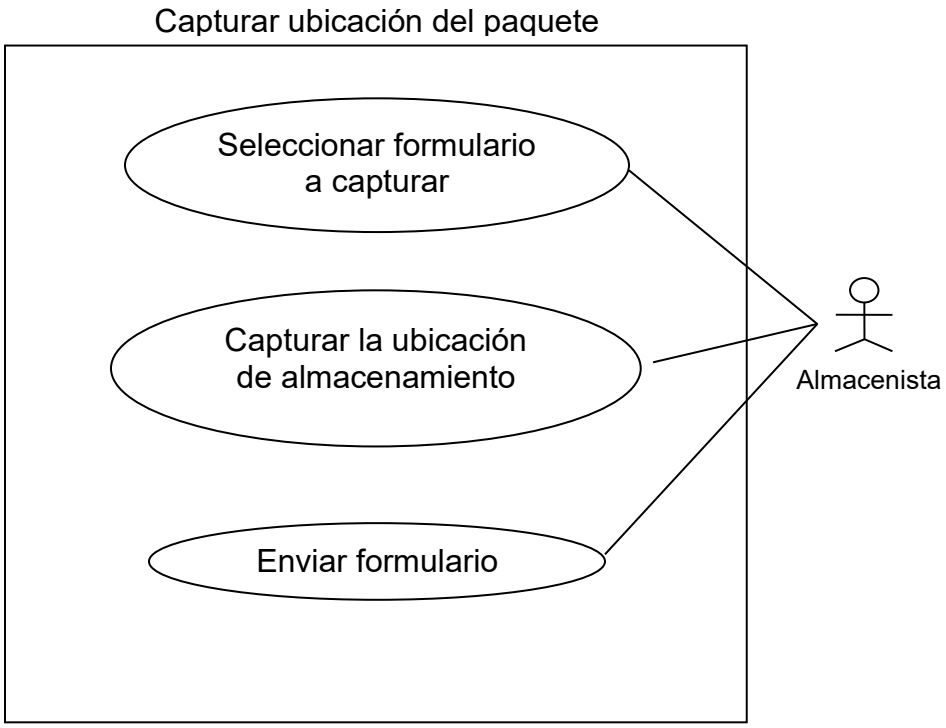


FIGURA 4.1.1.27 DIAGRAMA DE CASOS DE USO DEL ENVÍO

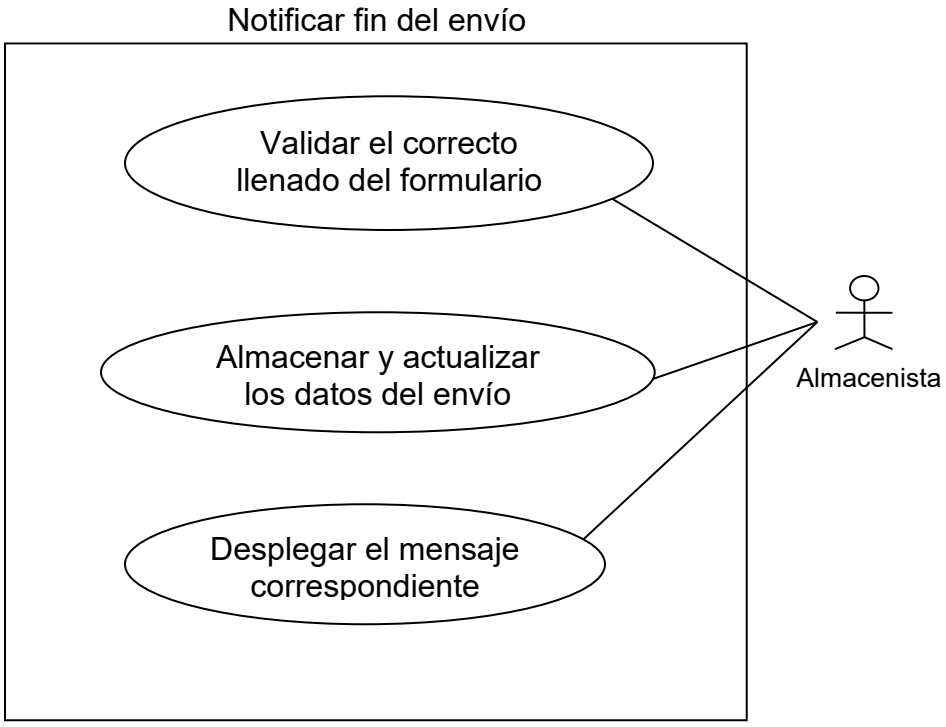


FIGURA 4.1.1.28 DIAGRAMA DE CASOS DE USO DEL ENVÍO

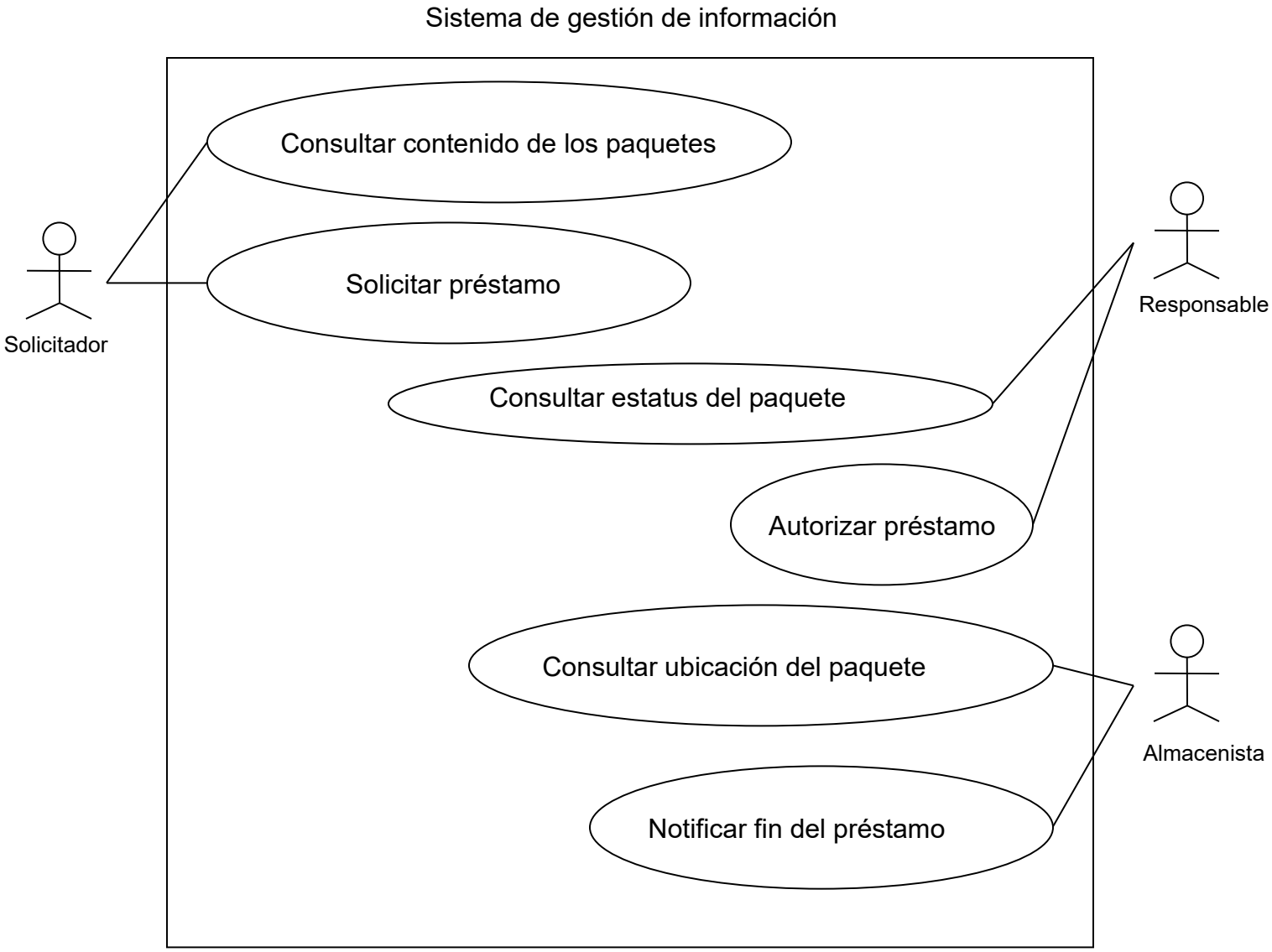


FIGURA 4.1.1.29 DIAGRAMA DE CASOS DE USO DEL PRÉSTAMO

Nivel 2 de casos de uso del préstamo

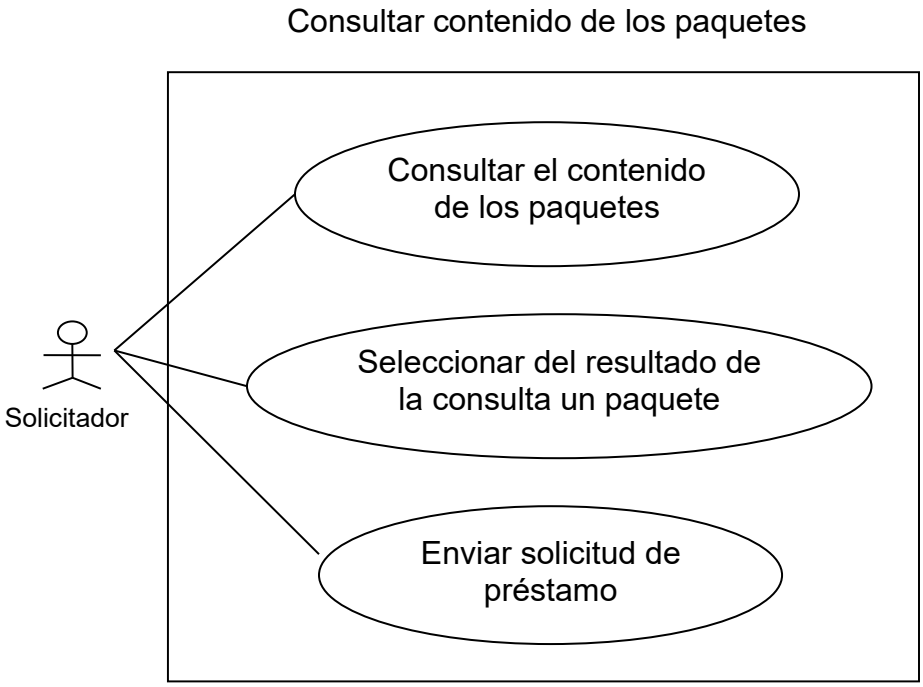


FIGURA 4.1.1.30 DIAGRAMA DE CASOS DE USO DEL PRÉSTAMO

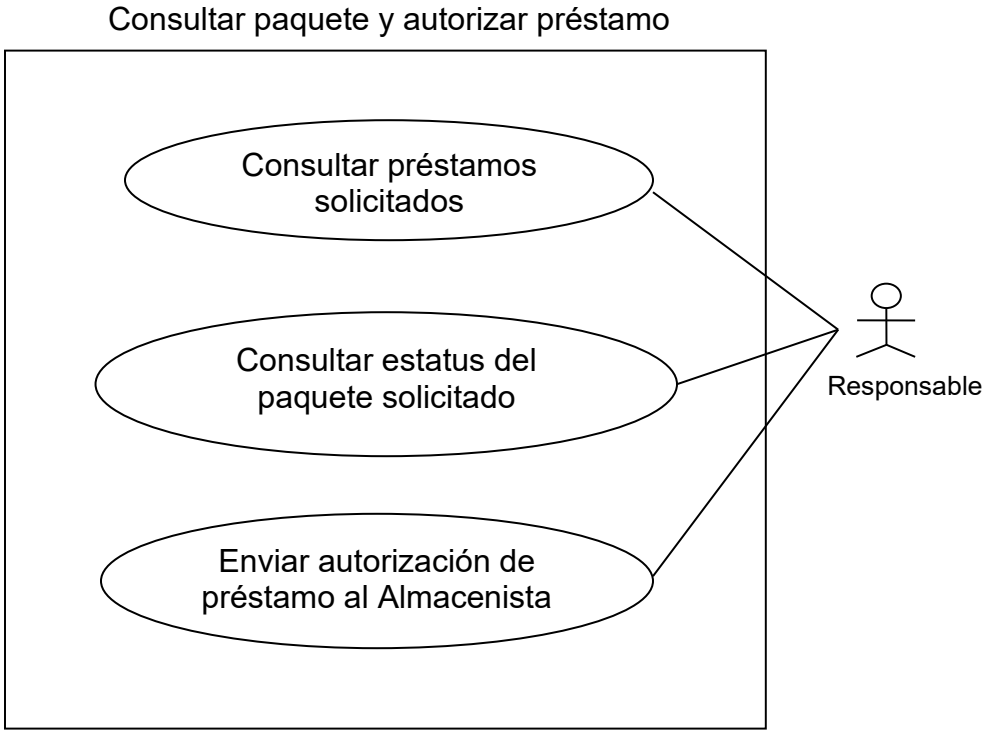


FIGURA 4.1.1.31 DIAGRAMA DE CASOS DE USO DEL PRÉSTAMO

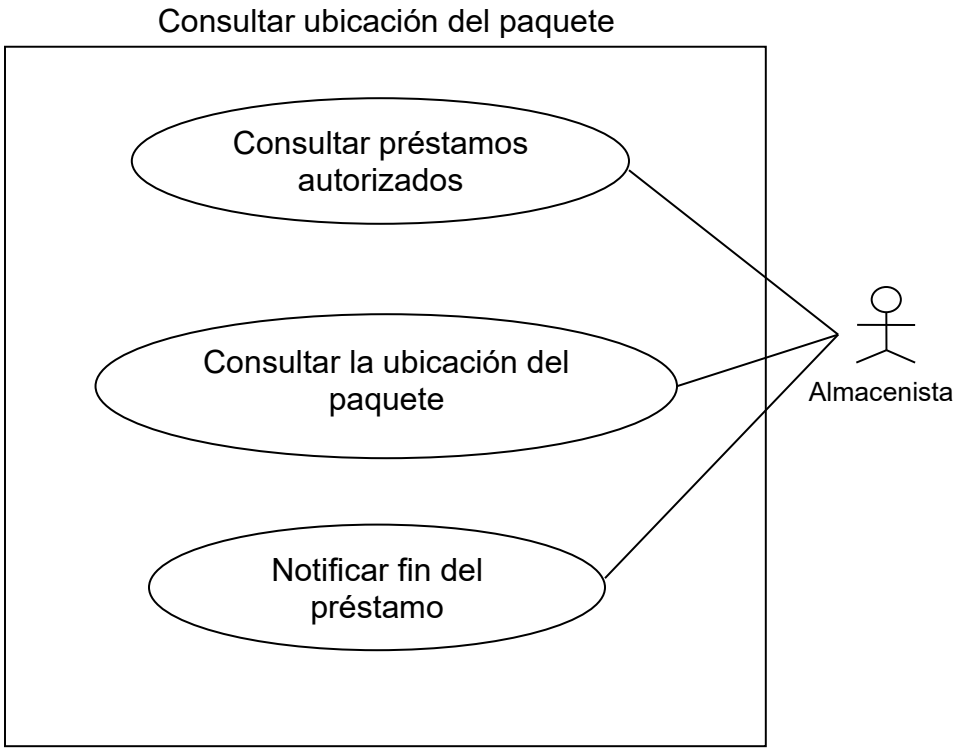


FIGURA 4.1.1.32 DIAGRAMA DE CASOS DE USO DEL PRÉSTAMO

Nivel 3 de casos de uso del préstamo

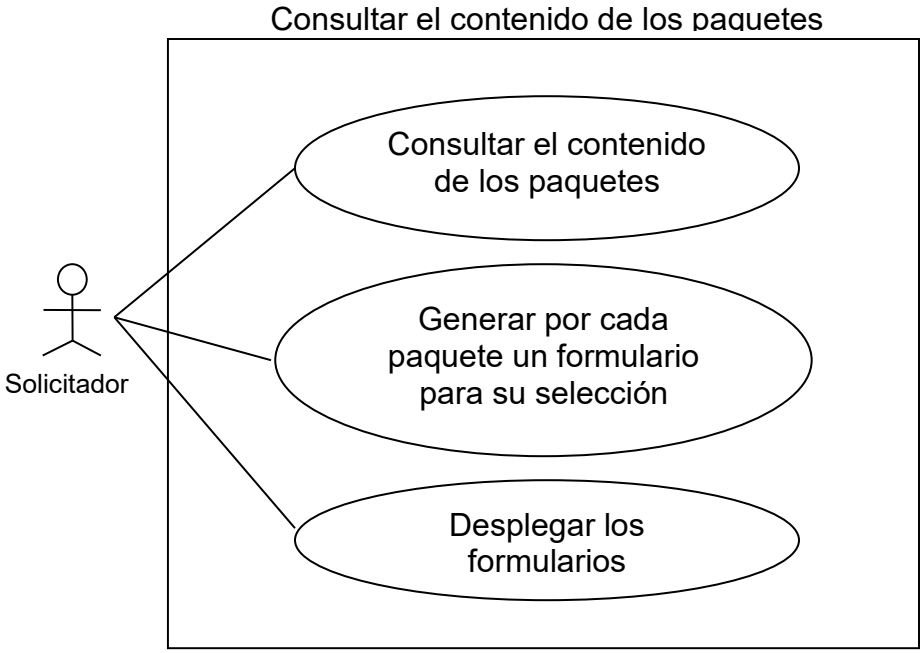


FIGURA 4.1.1.33 DIAGRAMA DE CASOS DE USO DEL PRÉSTAMO

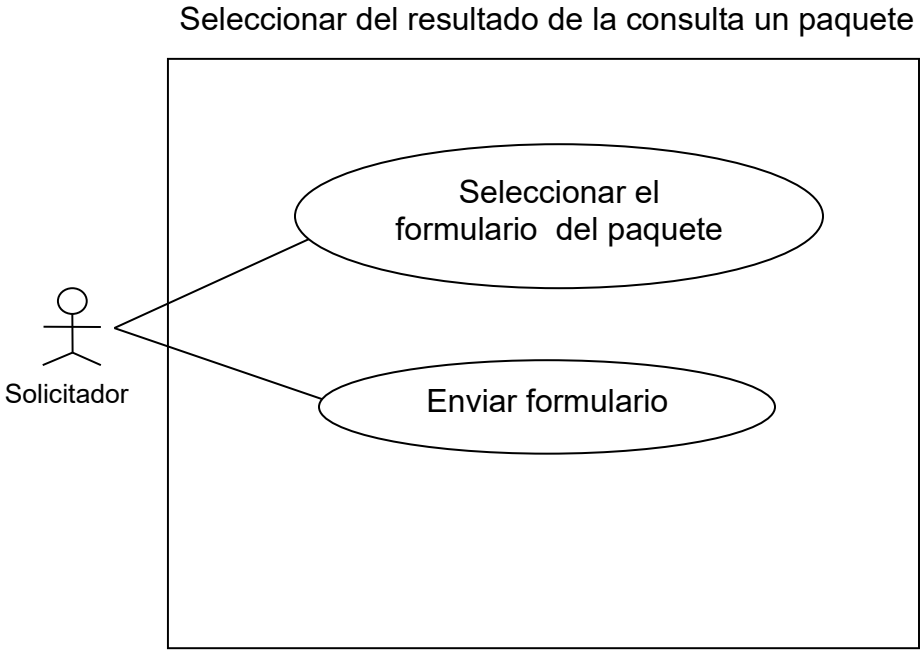


FIGURA 4.1.1.34 DIAGRAMA DE CASOS DE USO DEL PRÉSTAMO

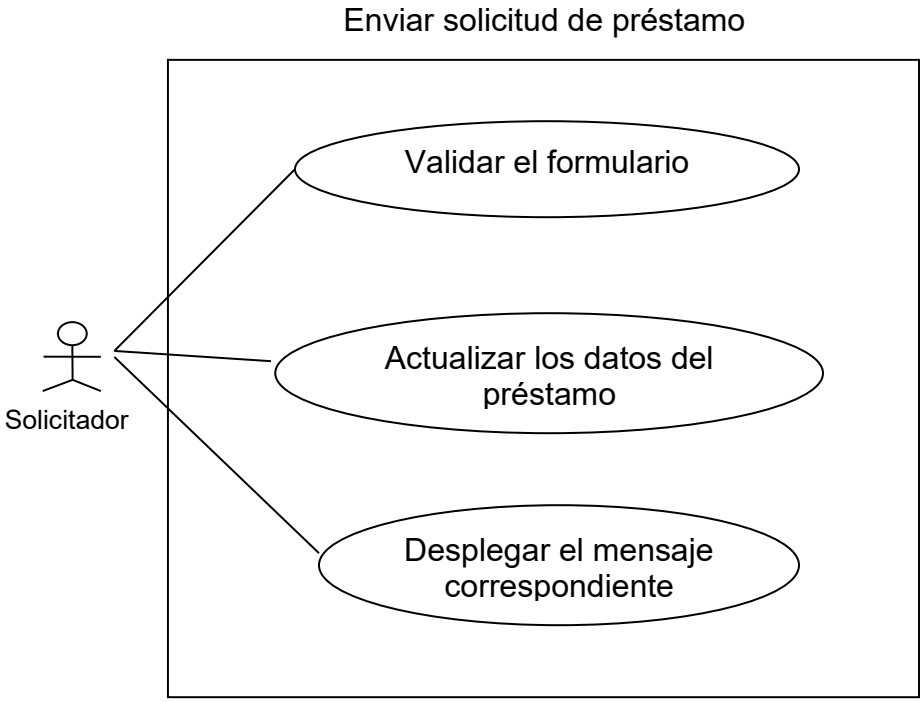


FIGURA 4.1.1.35 DIAGRAMA DE CASOS DE USO DEL PRÉSTAMO

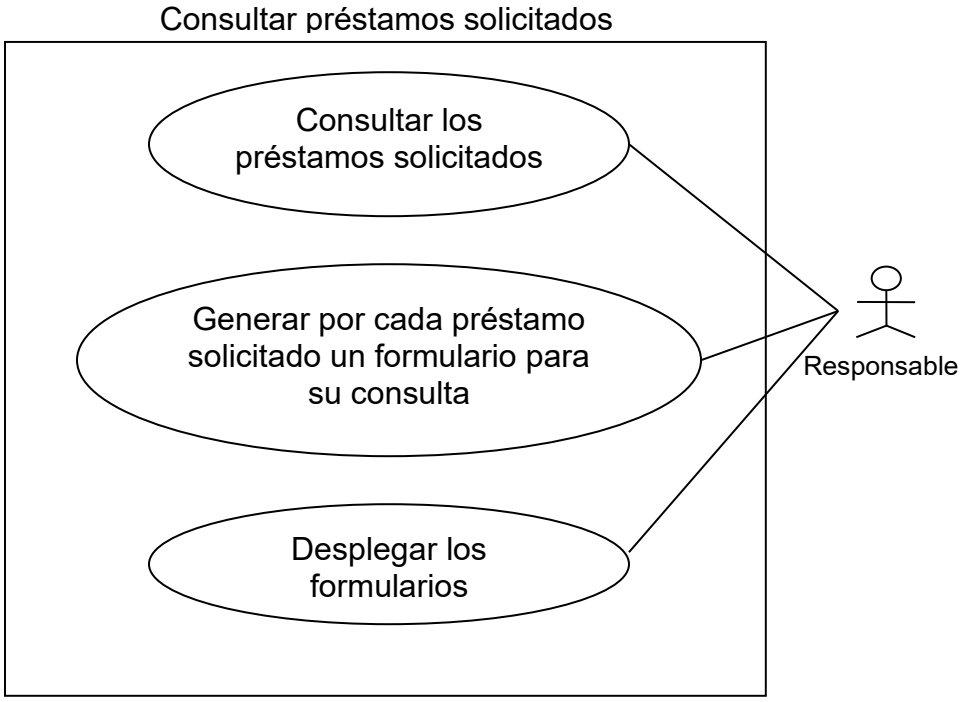


FIGURA 4.1.1.36 DIAGRAMA DE CASOS DE USO DEL PRÉSTAMO

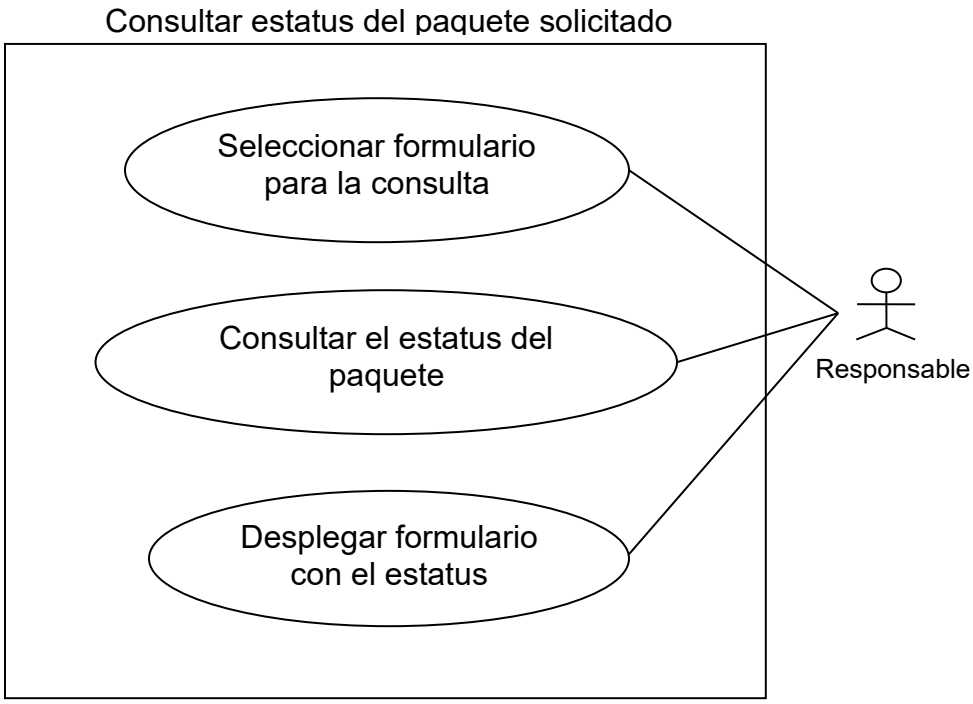


Figura 4.1.1.37 Diagrama de casos de uso del préstamo

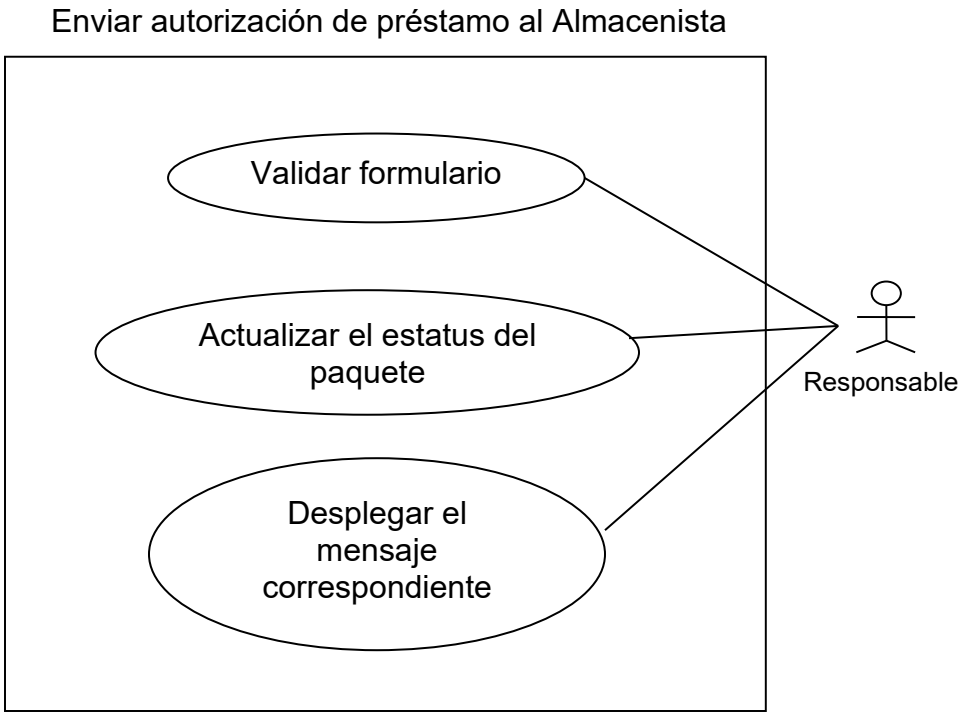


FIGURA 4.1.1.38 DIAGRAMA DE CASOS DE USO DEL PRÉSTAMO

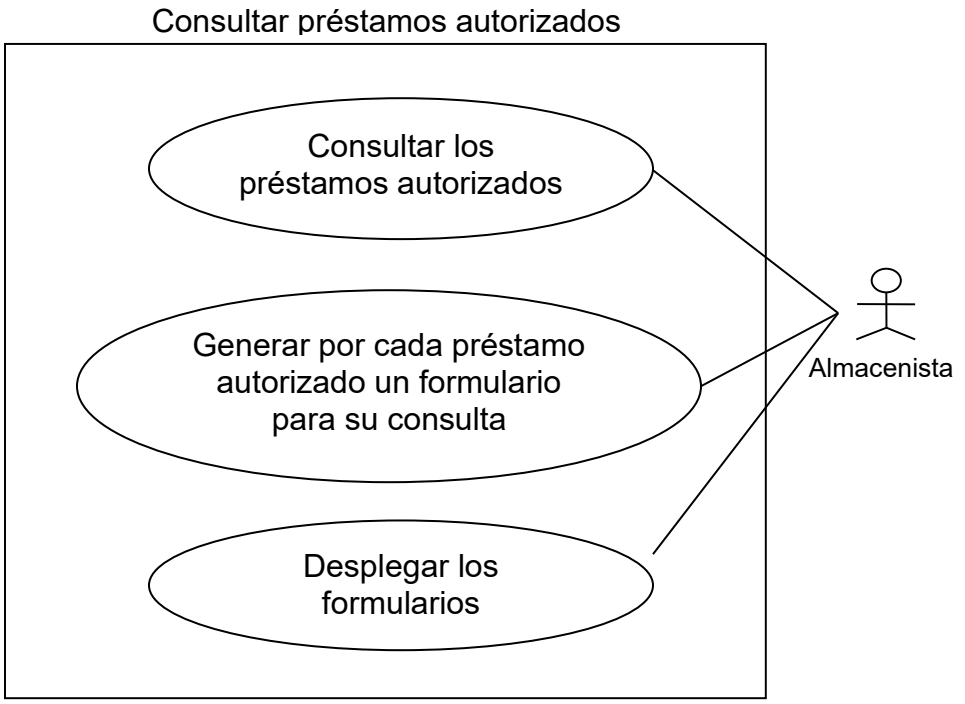


FIGURA 4.1.1.39 DIAGRAMA DE CASOS DE USO DEL PRÉSTAMO

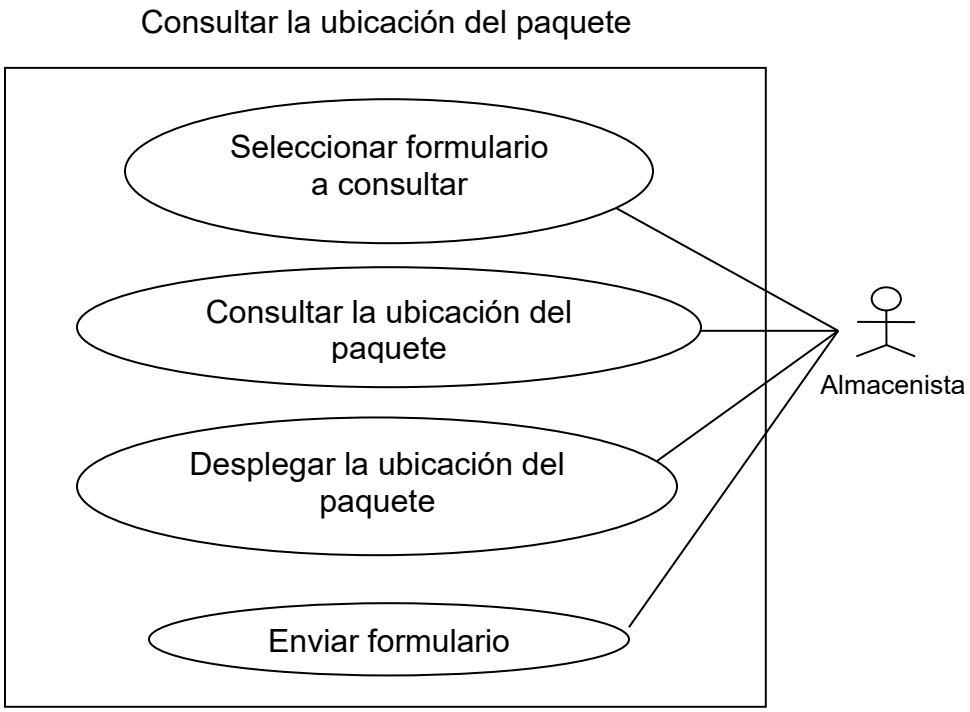


FIGURA 4.1.1.40 DIAGRAMA DE CASOS DE USO DEL PRÉSTAMO

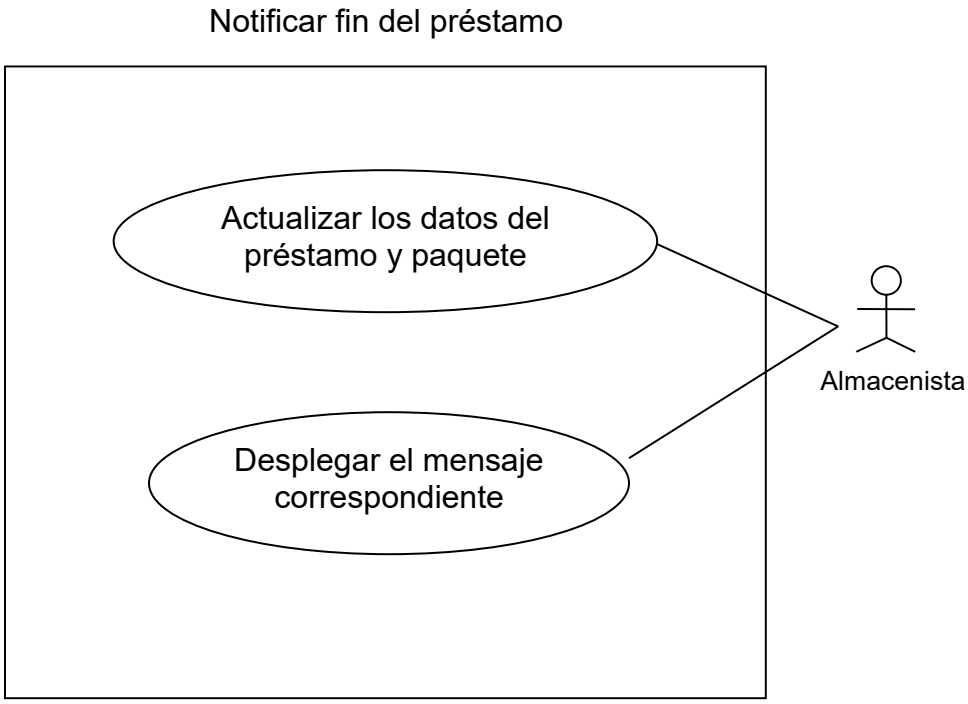


FIGURA 4.1.1.41 DIAGRAMA DE CASOS DE USO DEL PRÉSTAMO

4.1.2 DIAGRAMA DE SECUENCIA Y DIAGRAMA DE ACTIVIDADES

Diagrama de secuencia

Un diagrama de secuencia muestra los objetos y actores que participan en la realización de un caso de uso, así como la dinámica de sus partes para llevarlo a cabo. Véase los diagramas de secuencia de las figuras 4.1.2.1 al 4.1.2.9

Diagrama de secuencia [Caso de uso: Solicitar autorización de envío]

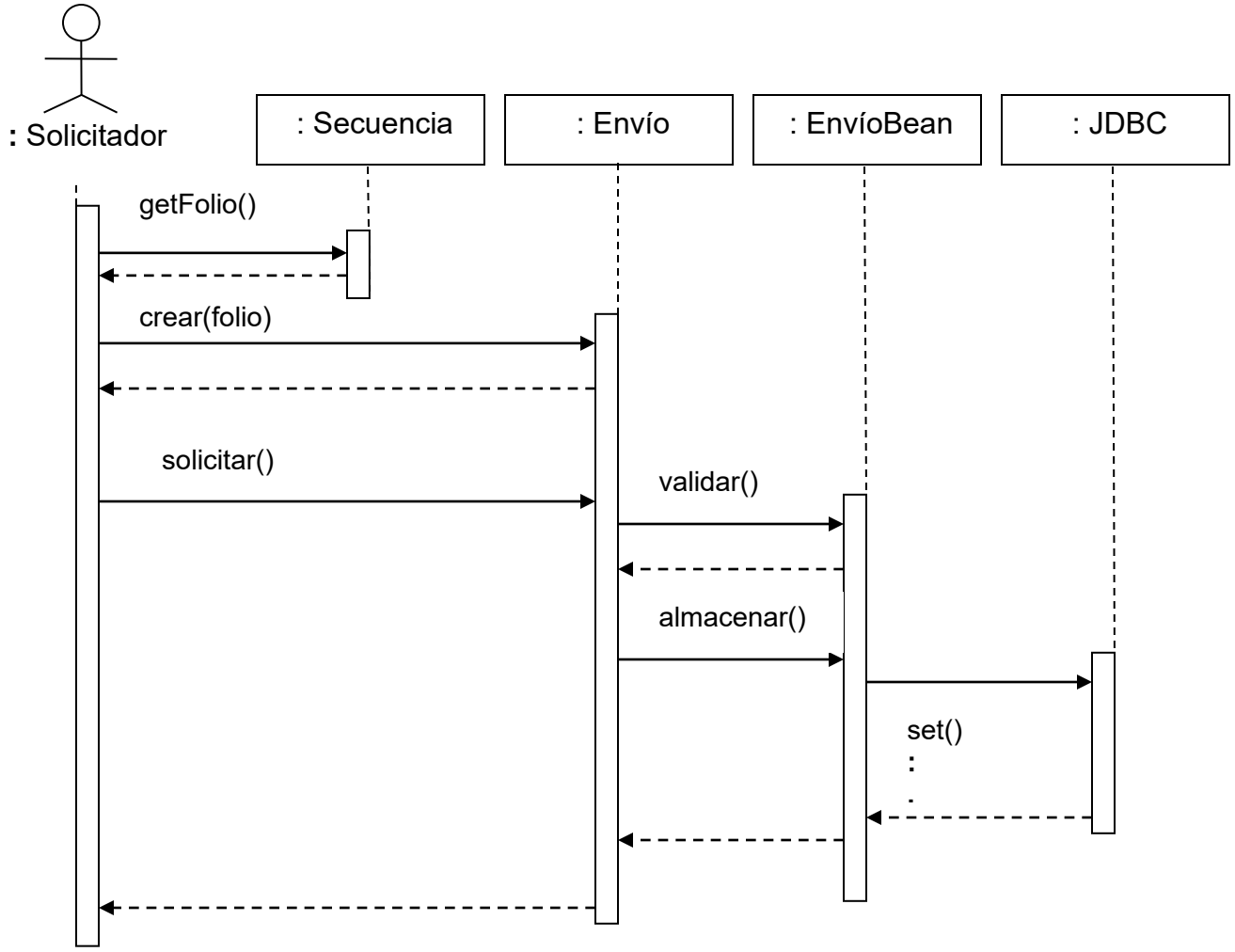


FIGURA 4.1.2.1 DIAGRAMA DE SECUENCIA DEL CASO DE USO "SOLICITAR AUTORIZACIÓN DE ENVÍO"

Diagrama de secuencia [Caso de uso: Verificar solicitud]

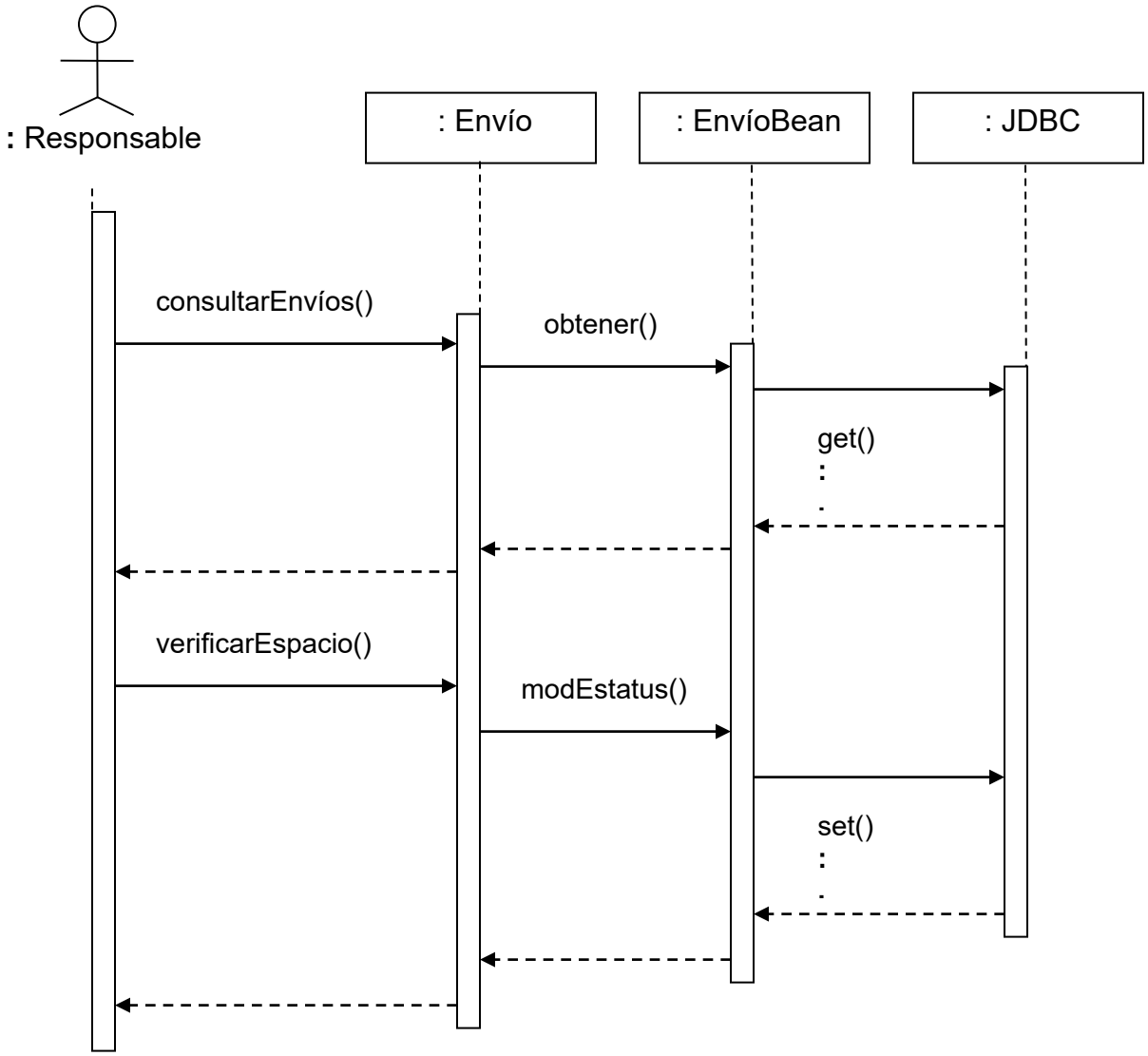


FIGURA 4.1.2.2 DIAGRAMA DE SECUENCIA DEL CASO DE USO "VERIFICAR SOLICITUD"

Diagrama de secuencia [Caso de uso: Confirmar espacio]

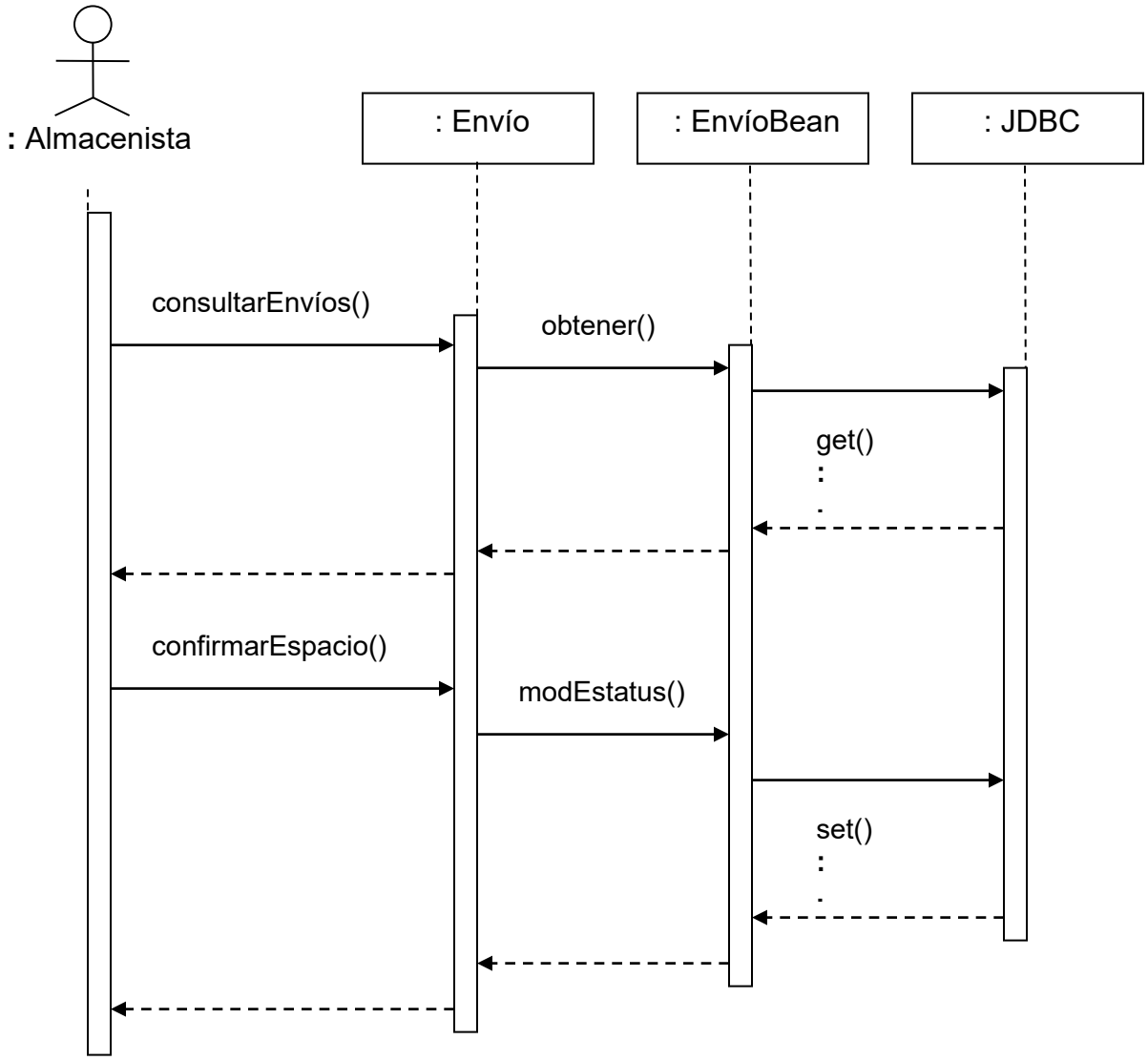


FIGURA 4.1.2.3 DIAGRAMA DE SECUENCIA DEL CASO DE USO "CONFIRMAR ESPACIO"

Diagrama de secuencia [Caso de uso: Autorizar envío]

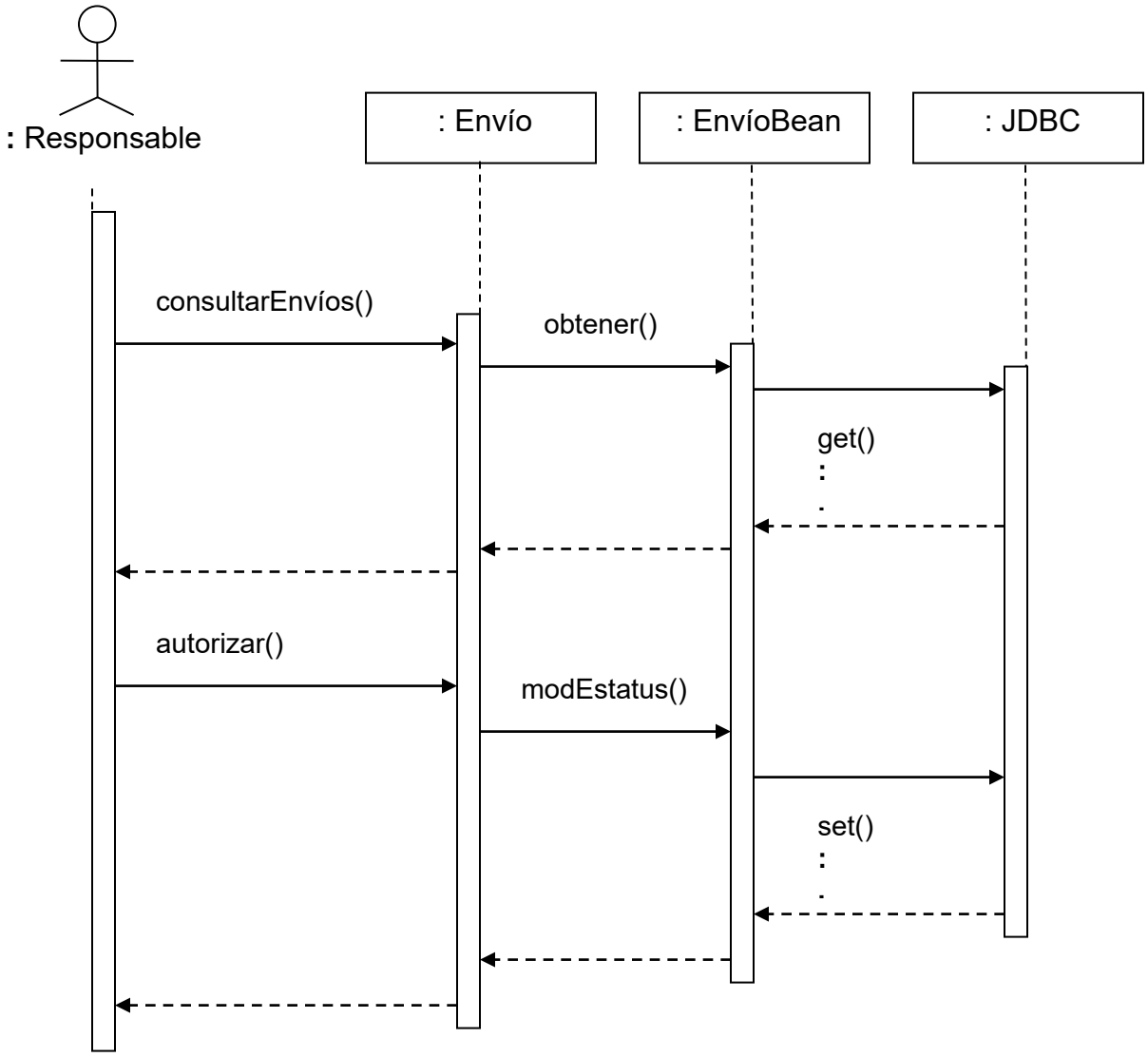


FIGURA 4.1.2.4 DIAGRAMA DE SECUENCIA DEL CASO DE USO "AUTORIZAR ENVÍO"

Diagrama de secuencia [Casos de uso: Consultar estatus, Registrar paquetes y Notificar envío]

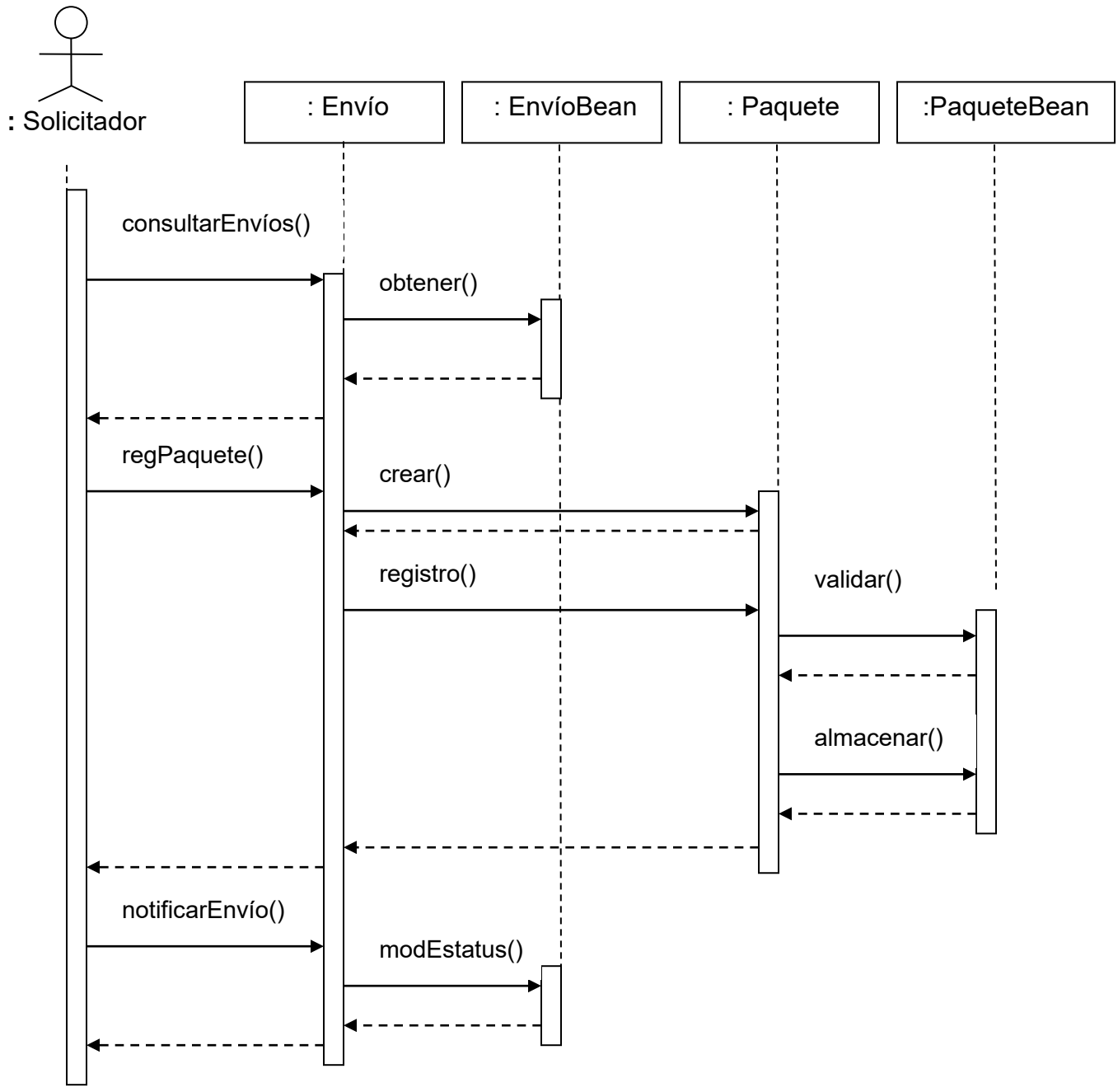


FIGURA 4.1.2.5 DIAGRAMA DE SECUENCIA DE LOS CASOS DE USO "CONSULTAR ESTATUS, REGISTRAR PAQUETES Y NOTIFICAR ENVÍO"

Diagrama de secuencia [Casos de uso: Registrar ubicación de los paquetes y Notificar fin del envío]

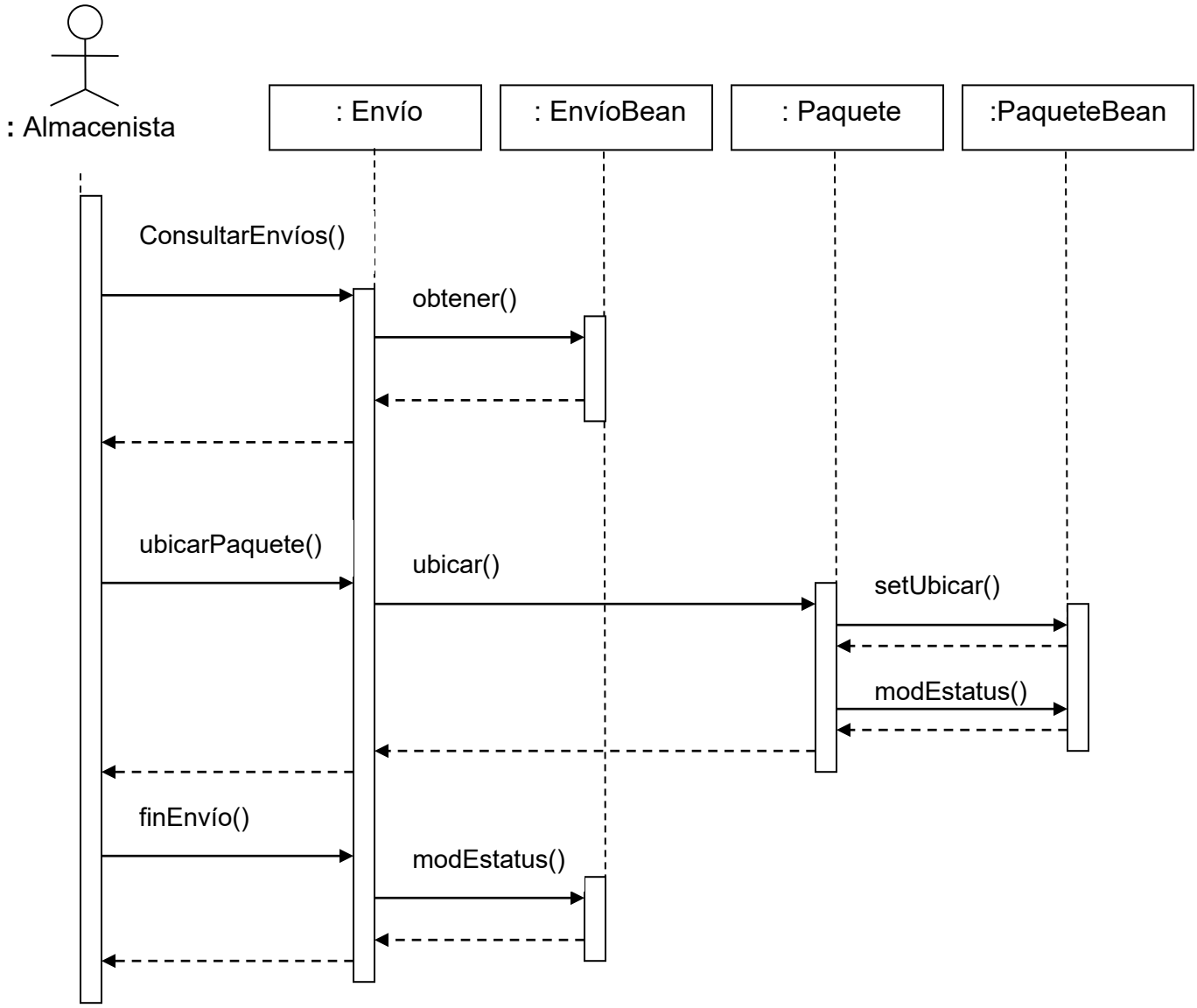


FIGURA 4.1.2.6 DIAGRAMA DE SECUENCIA DE LOS CASOS DE USO "REGISTRAR UBICACIÓN DE LOS PAQUETES Y NOTIFICAR FIN DEL ENVÍO"

Diagrama de secuencia [Casos de uso: Consultar contenido de los paquetes y Solicitar préstamo]

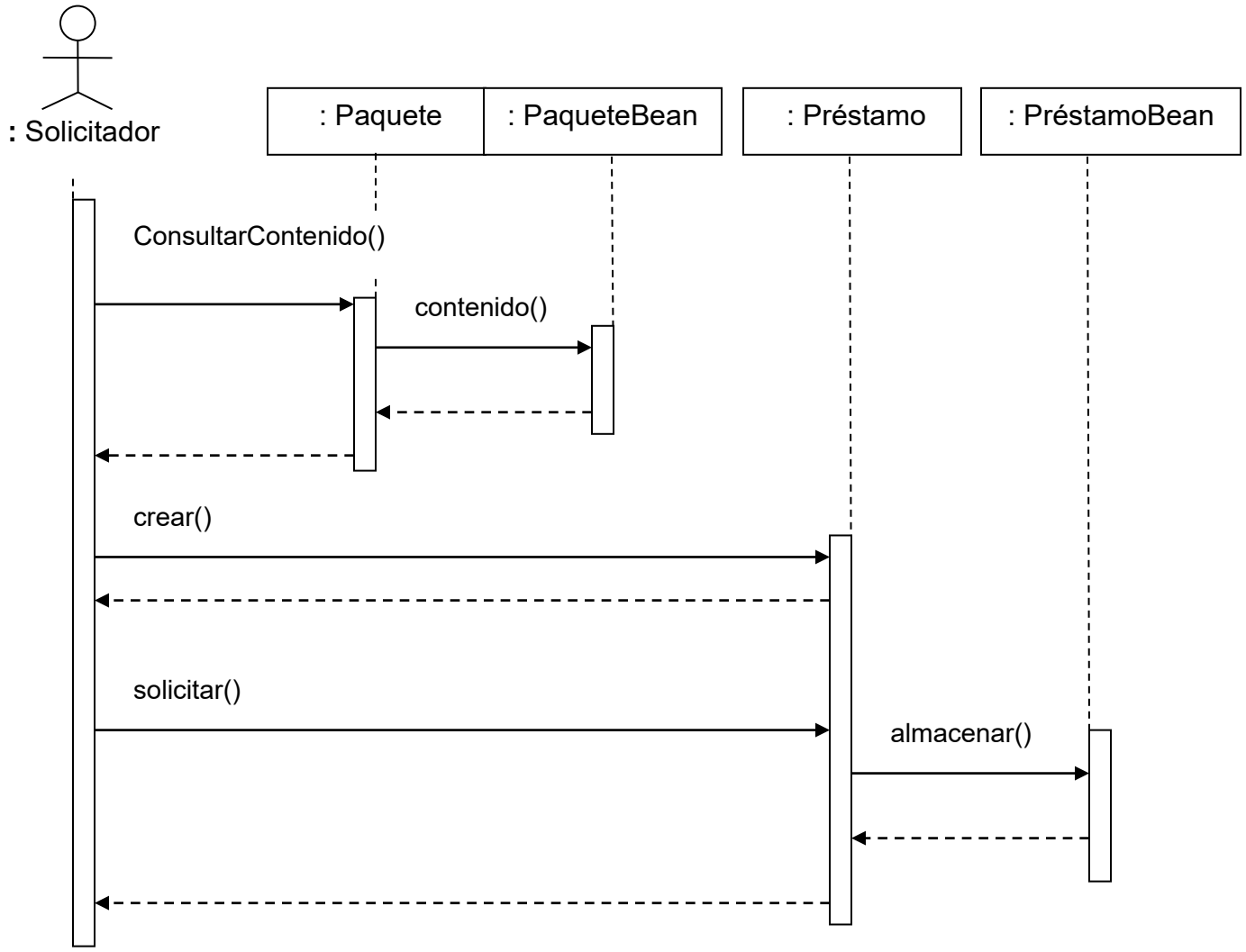


FIGURA 4.1.2.7 DIAGRAMA DE SECUENCIA DE LOS CASOS DE USO "CONSULTAR CONTENIDO DE LOS PAQUETES Y SOLICITAR PRÉSTAMO"

Diagrama de secuencia [Casos de uso: Consultar estatus del paquete y Autorizar préstamo]

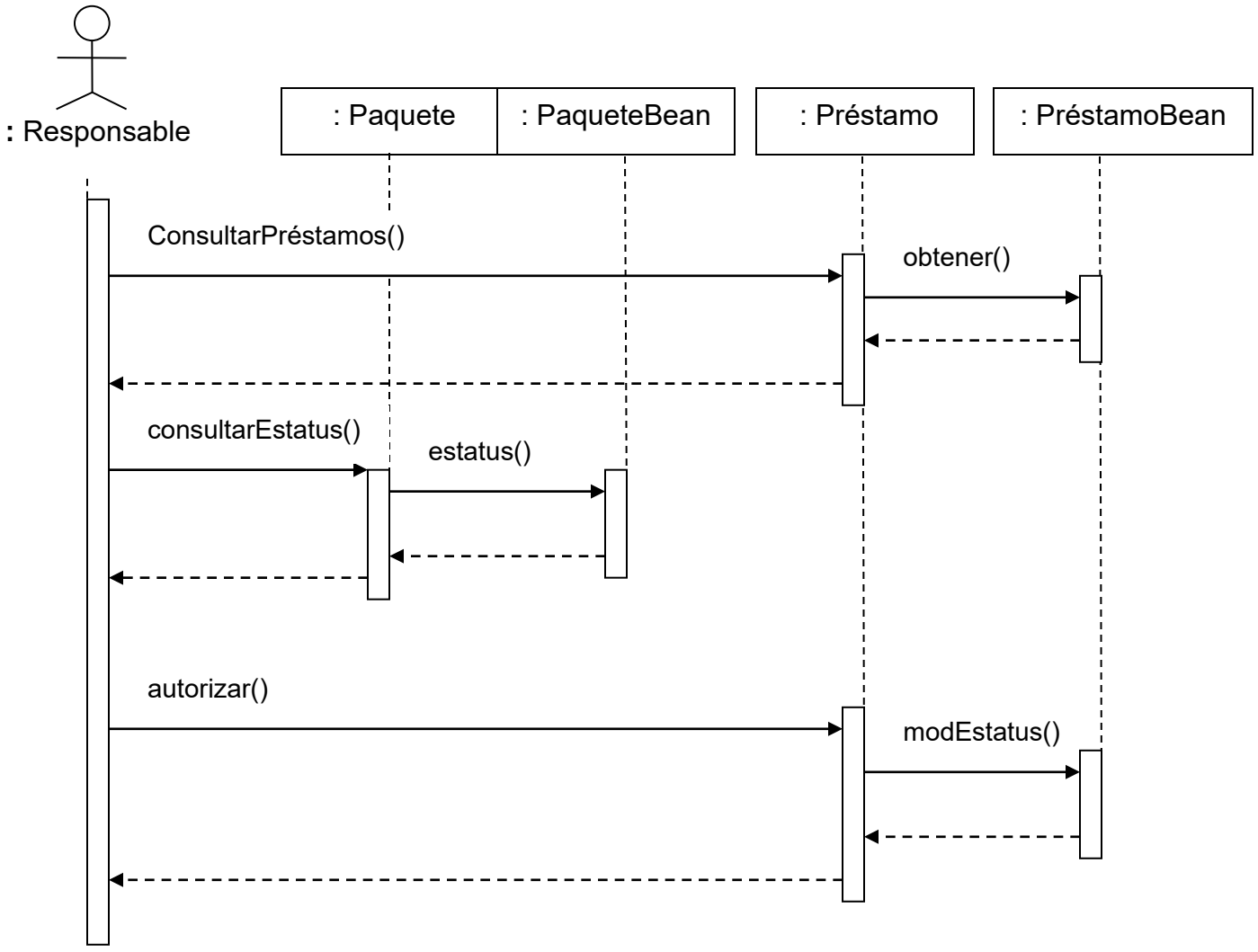


FIGURA 4.1.2.8 DIAGRAMA DE SECUENCIA DE LOS CASOS DE USO "CONSULTAR ESTATUS DEL PAQUETE Y AUTORIZAR PRÉSTAMO"

Diagrama de secuencia [Casos de uso: Consultar ubicación del paquete y Notificar fin del préstamo]

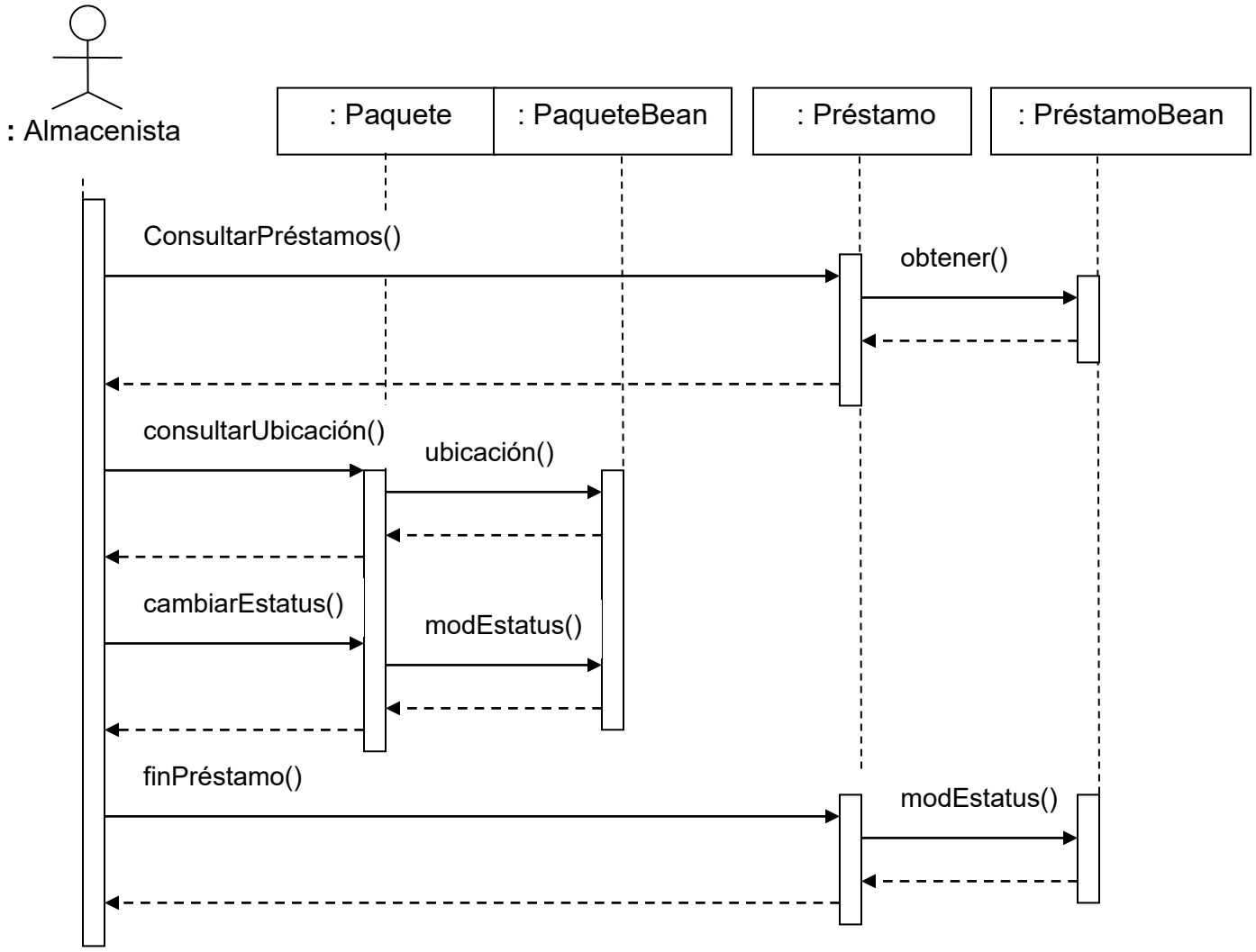


FIGURA 4.1.2.9 DIAGRAMA DE SECUENCIA DE LOS CASOS DE USO "CONSULTAR UBICACIÓN DEL PAQUETE Y NOTIFICAR FIN DEL PRÉSTAMO"

Diagrama de actividades

Un diagrama de actividades muestra cómo fluye el control de unas clases a otras con la finalidad de culminar con un flujo de control total, que corresponde con la consecución de un proceso más complejo hasta llegar al alcance total del sistema.

Los diagramas de actividades del envío y préstamo de paquetes se pueden observar en las figuras 4.1.2.10 y 4.1.2.11 respectivamente.

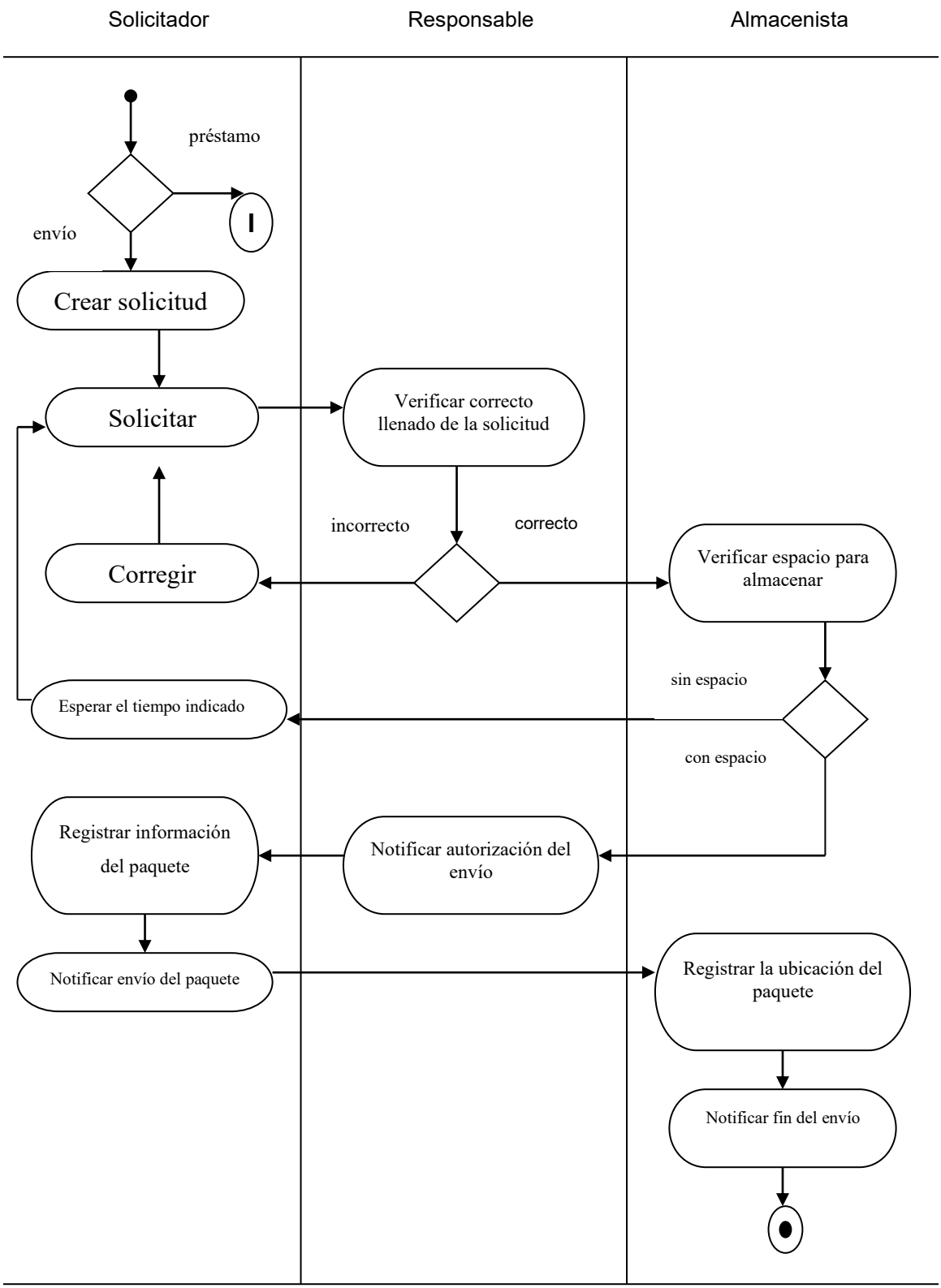


FIGURA 4.1.2.10 DIAGRAMA DE ACTIVIDADES DEL ENVÍO

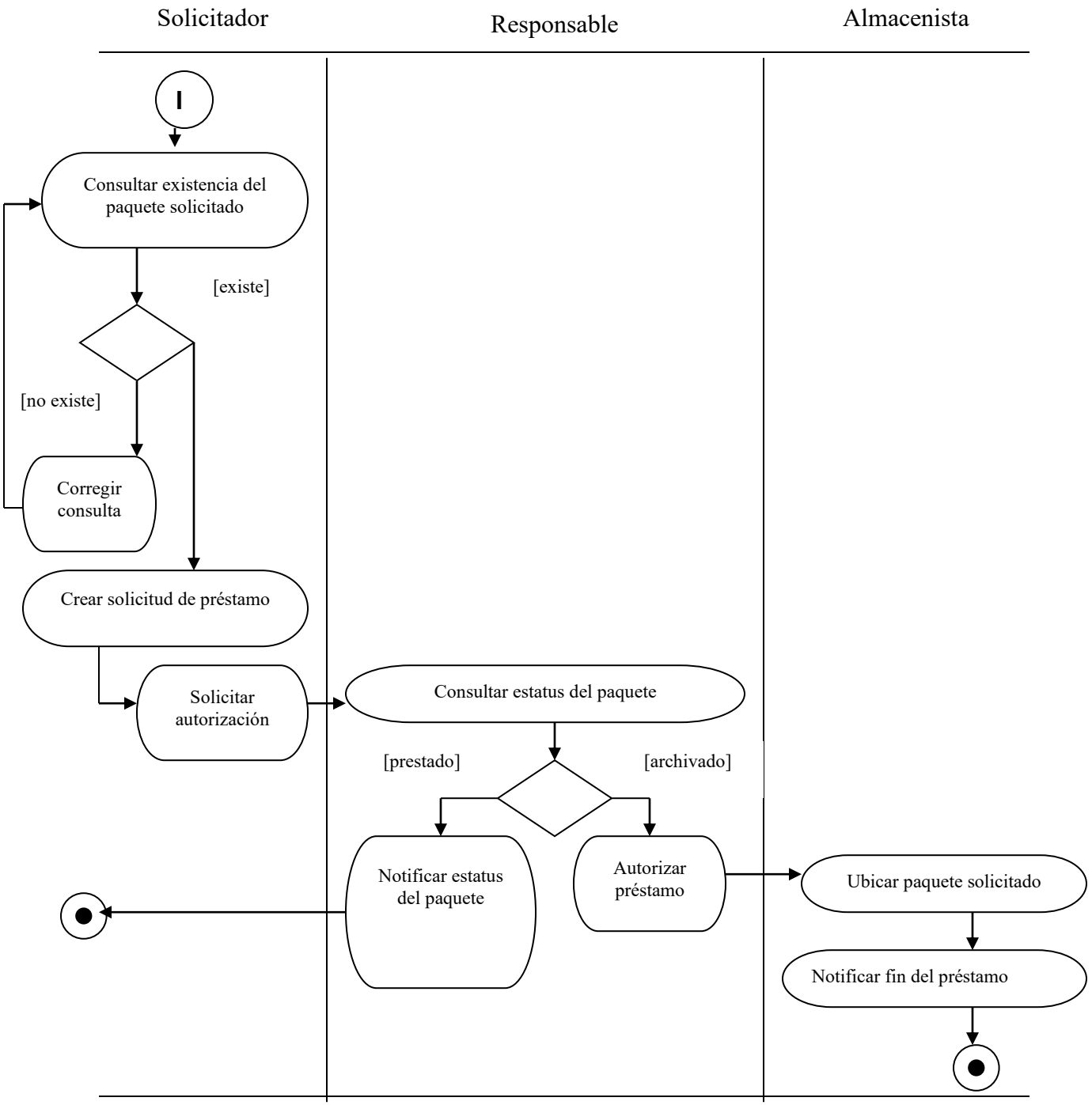


FIGURA 4.1.2.11 DIAGRAMA DE ACTIVIDADES DEL PRÉSTAMO

4.1.3 DICCIONARIO DE DATOS

El diccionario de datos contiene las características lógicas de los sitios donde se almacenan los datos del sistema, incluyendo nombre, descripción, acrónimo, llave, tipo y longitud de campo. Identifica los procesos donde se emplean los datos y los sitios donde se necesita el acceso inmediato a la información. Se desarrolla durante en análisis de flujo de datos.

Usuario

Nombre	Acrónimo	Tipo y Longitud	Llave	Descripción	Tabla
Identificador del usuario.	Id	int	primaria	Identificador único para el usuario	Usuario
Usuario	Usuario	Varchar(20)	no	Usuario para acceder al sistema	Usuario
Clave	Clave	Varchar(20)	no	Contraseña para acceder al sistema	Usuario
Rol	Rol	Varchar(50)	Foránea	Función o actividad que desempeña un usuario	Usuario
Nombre del usuario	Nombre	Varchar(30)	no	Nombre del usuario	Usuario
Apellido del Usuario	Apellidos	Varchar(70)	no	Apellidos del usuario	Usuario

TABLA 4.1.3.1 DICCIONARIO DE DATOS PARA EL USUARIO

Envío

Nombre	Acrónimo	Tipo y Longitud	Llave	Descripción	Tabla
Folio	Folio	int	Primaria	Número de la captura el cual permitirá al usuario identificar la información	Envío
Comentario	Comentario	Varchar (350)	no	Comentario referente al motivo por el cual se coloca cierto estatus.	Envío
Identificador del Proyecto	IdProyecto	int	Foránea	Identificador único del proyecto	Envío
Identificador del Solicitador	IdSolicitador	int	Foránea	Identificador del usuario que solicita el envío	Envío
Estado del paquete	Estatus	Varchar(20)	no	Es el estado actual en que esta el paquete que puede ser: terminado, inconcluso, por validar, validado o rechazado.	Envío

TABLA 4.1.3.2 DICCIONARIO DE DATOS PARA ENVÍOS (CONTINUACIÓN)

Contenido

Nombre	Acrónimo	Tipo y Longitud	Llave	Descripción	Tabla
Identificador	Id	Int	Primaria	Identificador	Contenido
Identificador del paquete	IdPaquete	int	Foránea	Identificador para dicho paquete	Contenido
Descripción	Descripcion	Varchar(1500)	no	Breve explicación del contenido del paquete enviado	Contenido

TABLA 4.1.3.3 DICCIONARIO DE DATOS PARA EL CONTENIDO DEL PAQUETE

Paquete

Nombre	Acrónimo	Tipo y Longitud	Llave	Descripción	Tabla
Identificador del paquete	Id	int	primaria	Identificador único que hace referencia a cierto paquete	Paquete
Folio	Folio	int	Foránea	Número de la captura el cual permitirá al usuario identificar la información	Paquete
Ubicación	Ubicación	Varchar(50)	no	Localización física del o los paquetes en el archivo general	Paquete
Fecha de destrucción	FechaDestruccionDia	Char(2)	no	Fecha Día en que un paquete en archivo general, cumple tiempo máximo de almacenamiento y puede ser destruido	Paquete
Fecha de destrucción	FechaDestruccionMes	Char(10)	no	Fecha Mes en que un paquete en archivo general, cumple tiempo máximo de almacenamiento y puede ser destruido	Paquete
Fecha de destrucción	FechaDestruccionAnio	Char(4)	no	Fecha Año en que un paquete en archivo general, cumple tiempo máximo de almacenamiento y puede ser destruido	Paquete
Estado del paquete	Estatus	Varchar(12)	no	Es el estado actual en que esta el paquete que puede ser: terminado, inconcluso, por validar, validado o rechazado.	Paquete

TABLA 4.1.3.4 DICCIONARIO DE DATOS PARA PAQUETES

Proyecto

Nombre	Acrónimo	Tipo y Longitud	Llave	Descripción	Tabla
Identificador del proyecto	Id	int	primaria	Identificador único de un proyecto	Proyecto
Nombre del Proyecto	Nombre	Varchar(70)	no	Descripción corta e los proyectos actuales de la empresa	Proyecto
Contable	CContable	Varchar(15)	no	Contable	Proyecto

TABLA 4.1.3.5 DICCIONARIO DE DATOS PARA PROYECTOS

Secuencia

Nombre	Acrónimo	Tipo y Longitud	Llave	Descripción	Tabla
Folio	Folio	int	Primaria	Número de la captura el cual permitirá al usuario identificar la información	Secuencia

TABLA 4.1.3.6 DICCIONARIO DE DATOS PARA SECUENCIA

Préstamo

Nombre	Acrónimo	Tipo y Longitud	Llave	Descripción	Tabla
Identificador del prestamo	Id	int	primaria	Identificador único para un préstamo	Préstamo
Identificador del paquete	IdPaquete	int	Foránea	Identificador único que hace referencia a cierto paquete	Préstamo
Usuario que solicita	IdSolicitador	int	Foránea	Persona que esta solicitando un préstamo	Préstamo
Identificador del proyecto	Id Proyecto	int	Foránea	Identificador único de un proyecto	Préstamo
Estado del paquete	Estatus	Varchar(11)	no	Es el estado actual en que esta el paquete que puede ser: terminado, inconcluso, por validar, validado o rechazado.	Préstamo
Comentario	Comentario	Varchar(250)	no	Comentario referente al motivo por el cual se coloca cierto estatus.	Préstamo

TABLA 4.1.3.7 DICCIONARIO DE DATOS PARA PRÉSTAMOS

4.1.4 DIAGRAMA DE ENTIDAD-RELACIÓN Y DIAGRAMA DE CLASES

Diagrama de Entidad-Relación

Este diagrama representa a la realidad a través de un esquema gráfico empleando la terminología de entidades, que son objetos que existen y son los elementos principales que se identifican en el problema a resolver y se distinguen por sus características particulares denominadas atributos, el enlace que rige la unión de las entidades esta representada por la relación del modelo.

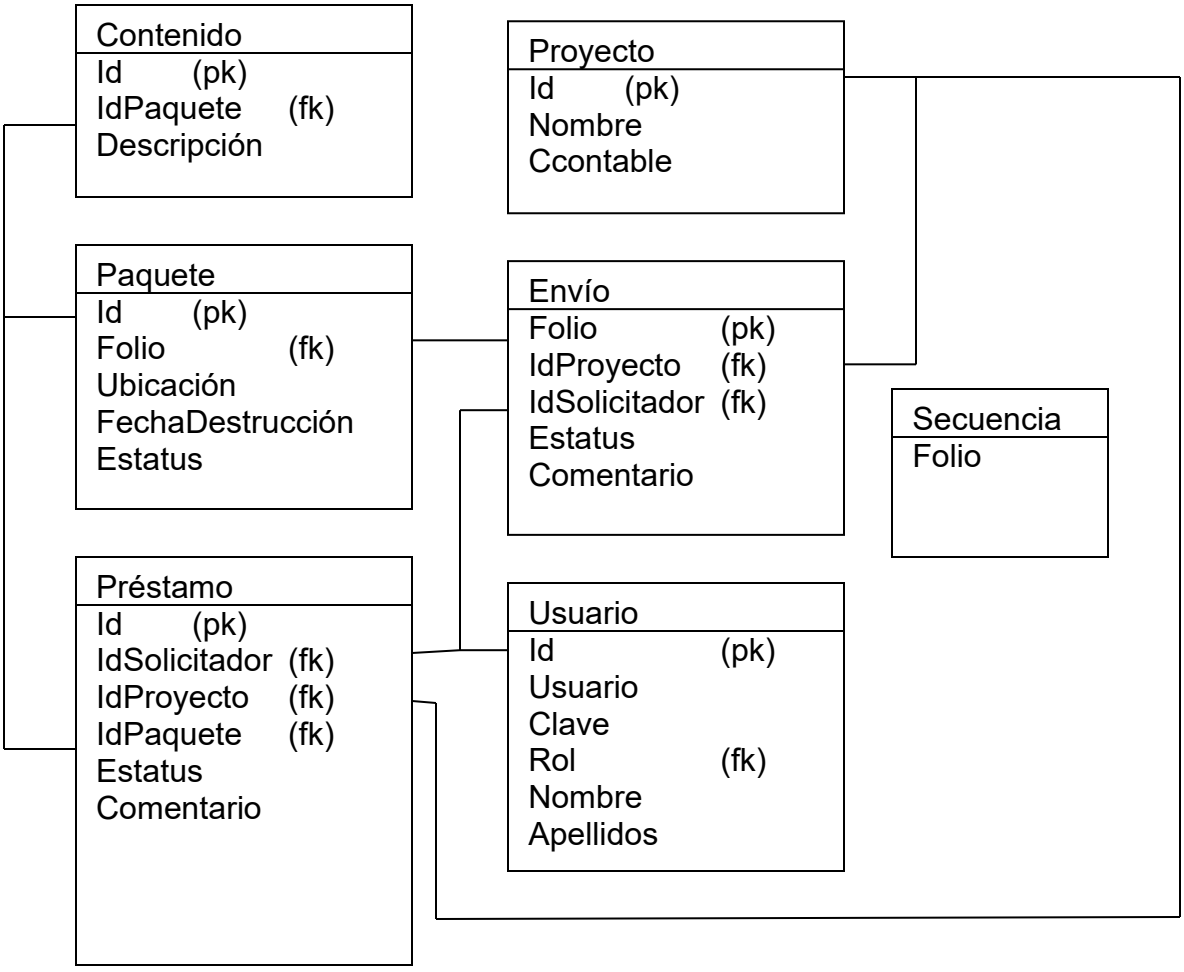


FIGURA 4.1.4.1 DIAGRAMA ENTIDAD - RELACIÓN

Diagrama de clases

El Diagrama de Clases es el diagrama principal para el análisis y diseño. Un diagrama de clases presenta las clases del sistema con sus relaciones estructurales y de herencia. La definición de clase incluye definiciones para atributos y operaciones.

El diagrama de clases se puede observar el la figura 4.1.4.2

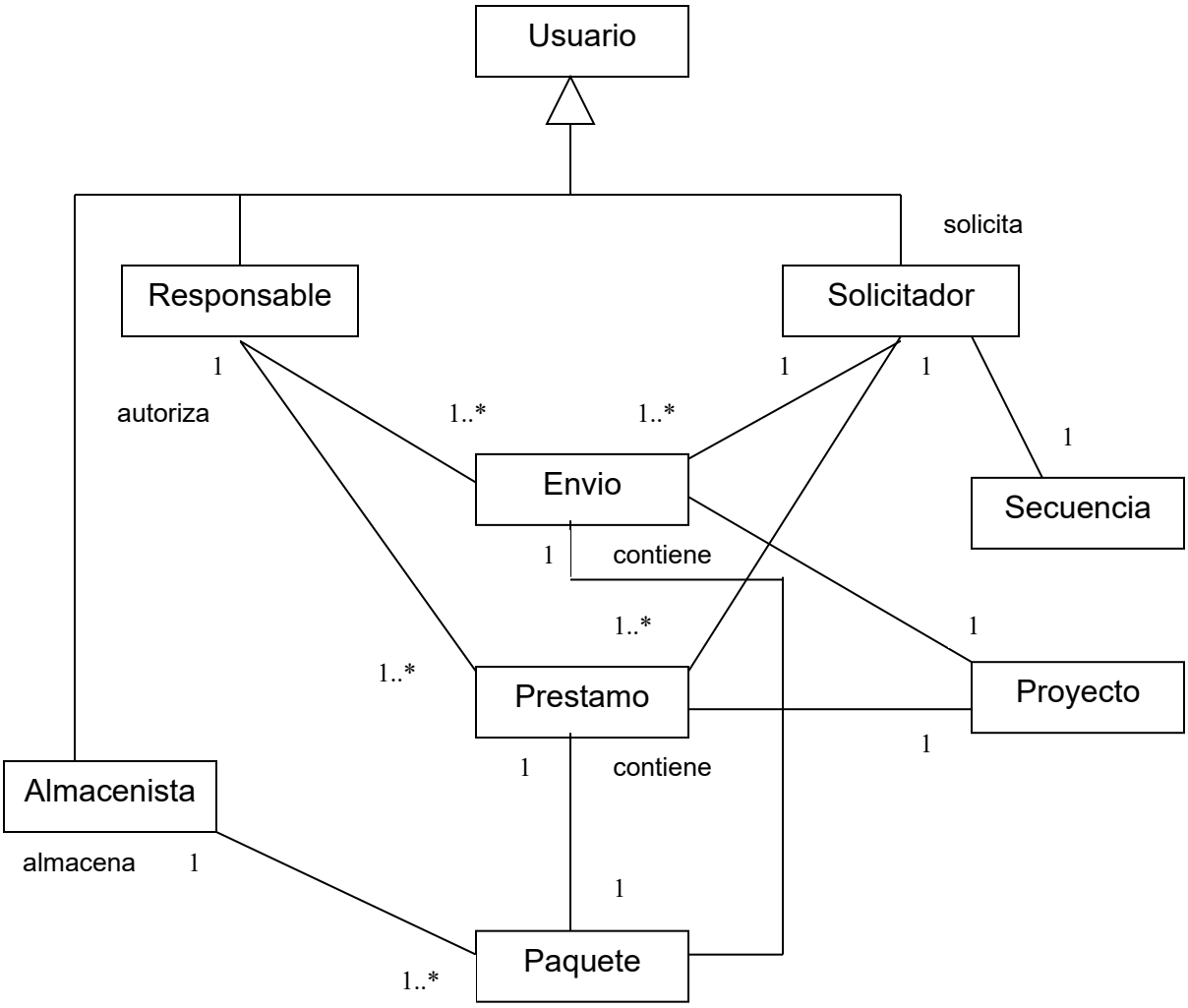


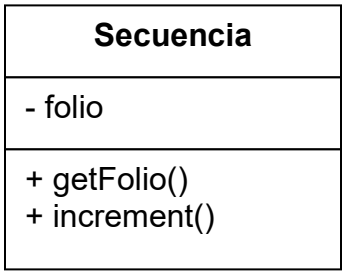
FIGURA 4.1.4.2 DIAGRAMA DE CLASES

Descripción de las clases

Atributos y Operaciones

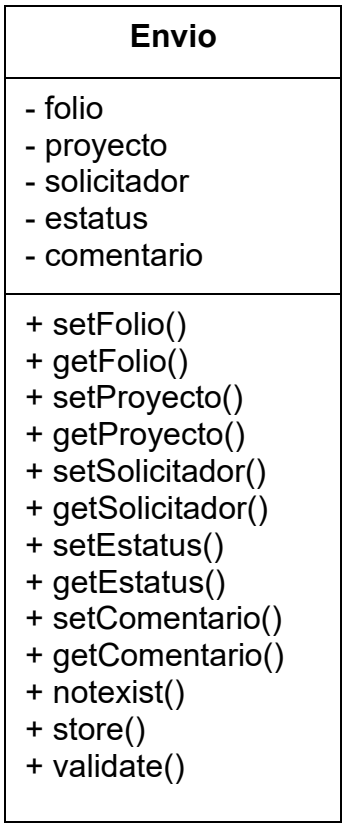
Secuencia
- folio

+ getFolio()
+ increment()



Envio
- folio
- proyecto
- solicitador
- estatus
- comentario

+ setFolio()
+ getFolio()
+ setProyecto()
+ getProyecto()
+ setSolicitador()
+ getSolicitador()
+ setEstatus()
+ getEstatus()
+ setComentario()
+ getComentario()
+ notexist()
+ store()
+ validate()

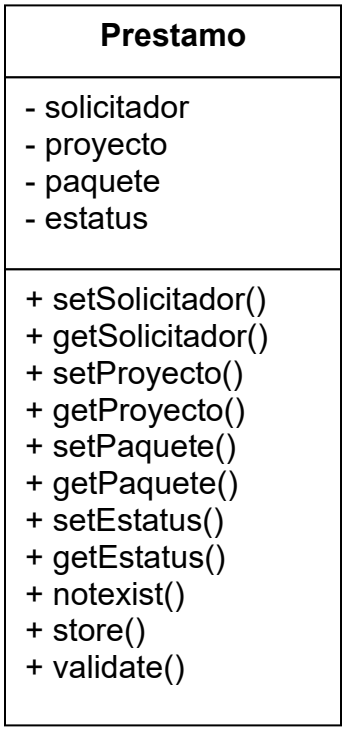


Atributos y Operaciones

Prestamo

- solicitador
- proyecto
- paquete
- estatus

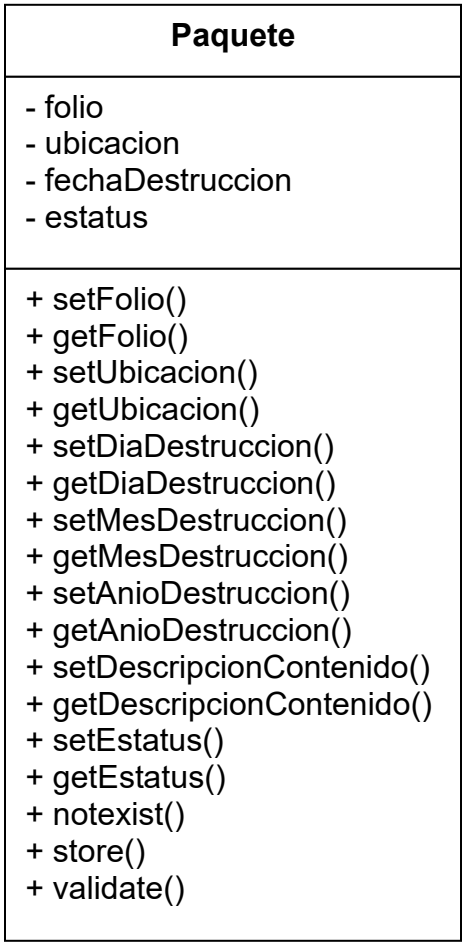
+ setSolicitador()
+ getSolicitador()
+ setProyecto()
+ getProyecto()
+ setPaquete()
+ getPaquete()
+ setEstatus()
+ getEstatus()
+ notexist()
+ store()
+ validate()



Atributos y Operaciones

Paquete
- folio
- ubicacion
- fechaDestruccion
- estatus

+ setFolio()
+ getFolio()
+ setUbicacion()
+ getUbicacion()
+ setDiaDestruccion()
+ getDiaDestruccion()
+ setMesDestruccion()
+ getMesDestruccion()
+ setAnioDestruccion()
+ getAnioDestruccion()
+ setDescripcionContenido()
+ getDescripcionContenido()
+ setEstatus()
+ getEstatus()
+ notexist()
+ store()
+ validate()

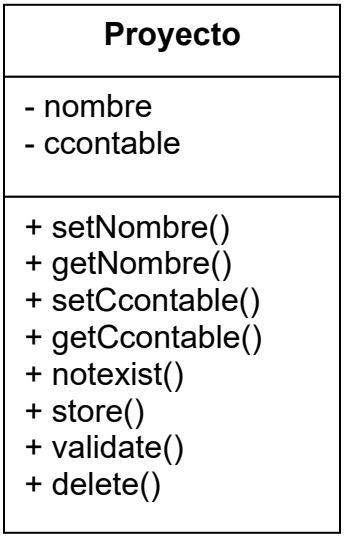


Atributos y Operaciones

Proyecto

- nombre
- ccontable

+ setNombre()
+ getNombre()
+ setCcontable()
+ getCcontable()
+ notexist()
+ store()
+ validate()
+ delete()

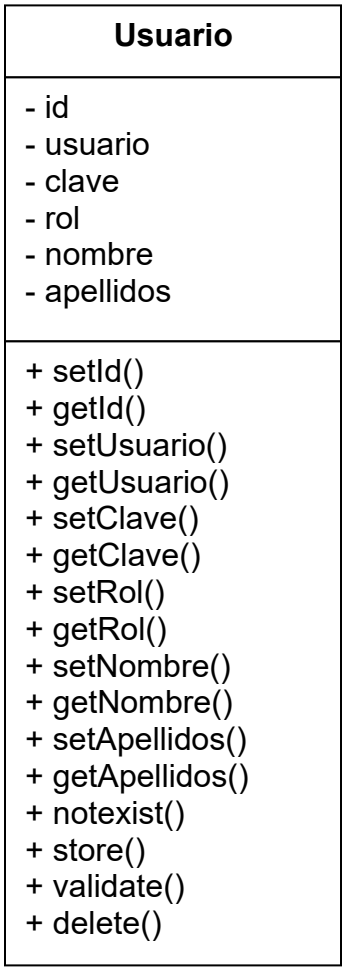


Atributos y Operaciones

Usuario

- id
- usuario
- clave
- rol
- nombre
- apellidos

+ setId()
+ getId()
+ setUsuario()
+ getUsuario()
+ setClave()
+ getClave()
+ setRol()
+ getRol()
+ setNombre()
+ getNombre()
+ setApellidos()
+ getApellidos()
+ notexist()
+ store()
+ validate()
+ delete()



Tarjetas CRC (Clase-Responsabilidad-Colaboración)

Las tarjetas CRC ayudan a:

- Identificar las clases que participan del diseño del sistema
- Obtener las responsabilidades que deben cumplir cada clase
- Establecer cómo colabora una clase con otras clases para cumplir con sus responsabilidades

Nombre de clase: **Envio**	
Responsabilidades	Colaboradores
• Atender la petición de creación de un nuevo envío. • Mantener el estatus del envío.	• Paquete • Solicitador • Responsable • Almacenista

Nombre de clase: **Prestamo**	
Responsabilidades	Colaboradores
• Atender la petición de préstamo de un paquete. • Mantener el estatus del préstamo.	• Paquete • Solicitador • Responsable • Almacenista

Nombre de clase: **Paquete**	
Responsabilidades	Colaboradores
• Mantener la descripción de su contenido. • Mantener su fecha de destrucción. • Mantener su ubicación física dentro del archivo general. • Mantener el estatus del paquete.	• Envio • Prestamo • Solicitador • Responsable • Almacenista

Nombre de clase: **Almacenista**	
Responsabilidades	Colaboradores
• Confirmar la existencia de espacio para almacenar el envío. • Registrar la ubicación de los paquetes.	• Envio • Paquete

Nombre de clase: **Responsable**	
Responsabilidades	Colaboradores
• Validar que los datos de la solicitud sean correctos. • Autorizar envío. • Consultar datos de los paquetes. • Autorizar préstamo de paquetes.	• Envio • Prestamo • Paquete

Nombre de clase: **Solicitador**	
Responsabilidades	Colaboradores
• Solicitar autorización de un nuevo envío. • Registrar contenido de los paquetes del envío. • Registrar el envío de paquete. • Solicitar autorización de préstamo de paquete.	• Envío • Paquete

Nombre de clase: **Proyecto**	
Responsabilidades	Colaboradores
• Creación, modificación y borrado de datos de un proyecto.	• Responsable

4.1.5 NORMALIZACIÓN

Para generar el diagrama entidad-relación del sistema se aplicaron las tres primeras formas normales (FN1, FN2, FN3).

- **Primera forma normal (FN1).** No se tienen nombres repetidos. Para almacenar los paquetes de un envío utilizamos un renglón por cada paquete. Para el registro de esta información se utiliza la tabla ENVIO, que tiene la información específica para identificar el envío de información, y que utiliza referencia a las llaves primarias del proyecto al que pertenece dicho envío así como a la persona que lo solicita, su estatus actual y algún comentario referente a dicho estatus. La tabla de PAQUETE permite describir el paquete que será enviado, y tiene como llave primaria un identificador de paquete, en dicha tabla se hace referencia al envío mediante el atributo folio (no existiendo en la misma información extra que corresponda al envío), la información de ubicación , fecha de destrucción y estatus específico del paquete que será almacenado:

ENVÍO

Folio(PK)	IdProyecto	IdSolicitador	Estatus	Comentario
15	1	test2	ACEPTADO	
21	5	test4	EN_TRANSITO	
32	4	test2	EN_ARCHIVO	

PAQUETE

ID(PK)	Folio	Ubicación	FecDestruccion	Estatus
55	15	A25B15	10/08/2010	ACEPTADO
74	21	A32B19	05/10/2011	EN_TRANSITO
89	32	A57B01	06/09]/2012	EN_ARCHIVO

- **Segunda forma normal (FN2).** Los atributos dependen de su llave primaria completa. Para cada una de las tablas se incluye información específica de la misma, sin información redundante que pertenezca a otras entidades del sistema. La información está agrupada lógicamente y las referencias se hacen utilizando llaves foráneas sin incluir información

extra de las mismas. De esta forma, para identificar el PROYECTO o los datos del mismo en un envío, se deberá utilizar el valor del atributo **ENVIO.IdProyecto.** Para el caso del envío con Folio 15, los datos del proyecto al que pertenece son:

ENVÍO

Folio(PK)	IdProyecto	IdSolicitador	Estatus	Comentario
15	1	test2	ACEPTADO	

PROYECTO

Id(PK)	Nombre	Ccontable
1	RESIDENCIAL LOS CEDROS-CANCUN	PROY-001-2005

- **Tercera forma normal (FN3).** No debe haber atributos que dependan de otros que no son llave. Esta forma se cumple debido a que, en primer lugar, no existe información redundante dentro del sistema, y por otro lado, no existen llaves compuestas dentro de las entidades del mismo

4.2 DISEÑO Y CONSTRUCCIÓN DEL BACK-END

Microsoft Sql Server de Microsoft se esta convirtiendo en un serio competidor de las aplicaciones reales de bases de datos. Esta base de datos es en cierta forma menos costosa que el resto de los productos de base de datos. La explosiva popularidad de SQL es una de las tendencias más importantes de la industria de la informática actual. En los últimos años, SQL se ha convertido en el lenguaje estándar de bases de datos. Alrededor de 100 productos de base de datos ofrecen SQL, ejecutándose en un rango de sistemas informáticos que van desde computadoras personales a los sistemas basados en una computadora central. SQL juega un papel principal en la arquitectura de base de datos de los principales proveedores de computadoras, y esta en el núcleo de la estrategia de base de datos de Microsoft. SQL es una herramienta para organizar, gestionar y recuperar datos almacenados en una base de datos informática. El nombre SQL es una abreviación de Structured Query Languaje(Lenguaje Estructurado de Consultas). Por razones históricas, SQL se pronuncia generalmente sequel, pero también se utiliza para interactuar con una base de datos. De hecho SQL funciona con un tipo específico de base de datos, llamado base de datos relacional.

Creación de la base de datos

Después de tener instalado el software correctamente, entramos al SQL Server Enterpise Manager el cual nos permite crear nuestra propia base de datos a través de un asistente, el cual se muestra a continuación en la siguiente pantalla. La figura 4.2.1 muestra todos los elementos necesarios para tener una base de datos, un diagrama, procesos, vistas, etc. No sin antes mencionar que nos da la posibilidad de administrar todo esto de una manera muy sencilla. En la figura 4.2.2 se muestra como el asistente de creación de base de datos nos guía para definir y crear una nueva base de datos.

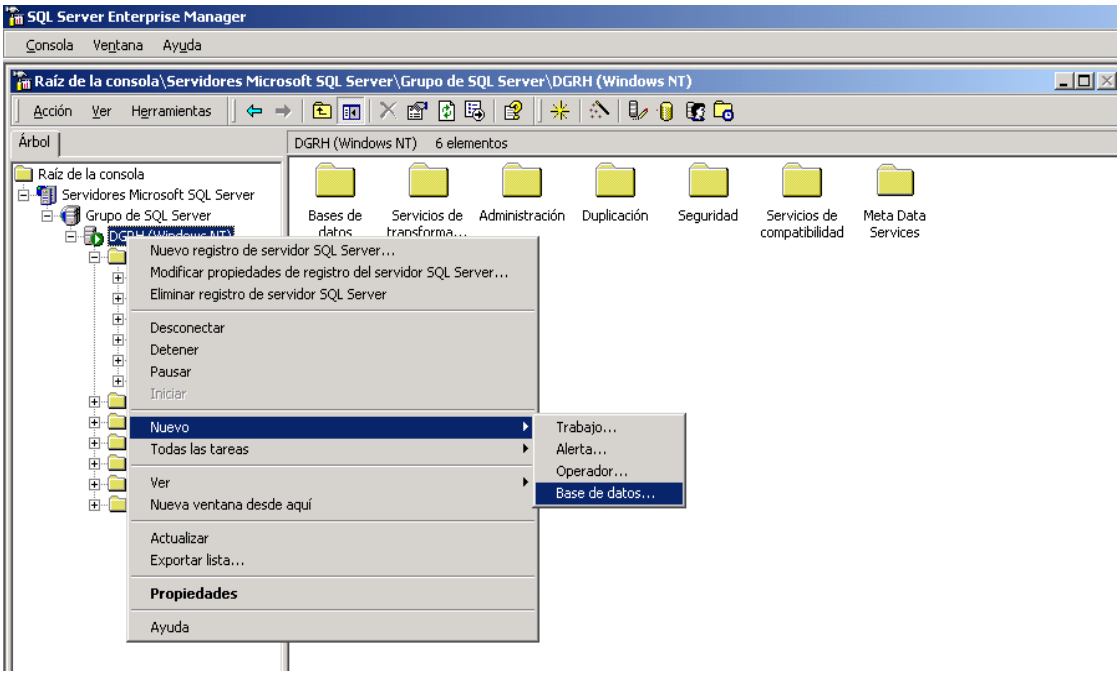


FIGURA 4.2.1 PANTALLA PRINCIPAL SQL SERVER ENTERPRISE MANAGER



FIGURA 4.2.2 ASISTENTE PARA LA CREACIÓN DE UNA BASE DE DATOS

Cuando uno esta creando la base de datos, se le tiene que definir un nombre a la instancia, se definen también los archivos en los cuales va a ser contenida la base de datos además del tamaño que tendrán estos archivos para poder contener dicha instancia, se definen también la forma en que va a estar limitada la base de datos en su crecimiento el cual es muy recomendable que sea en forma automática y que nada mas la limite el espacio de disco en el que se encuentre como lo muestra la figura 4.2.3.

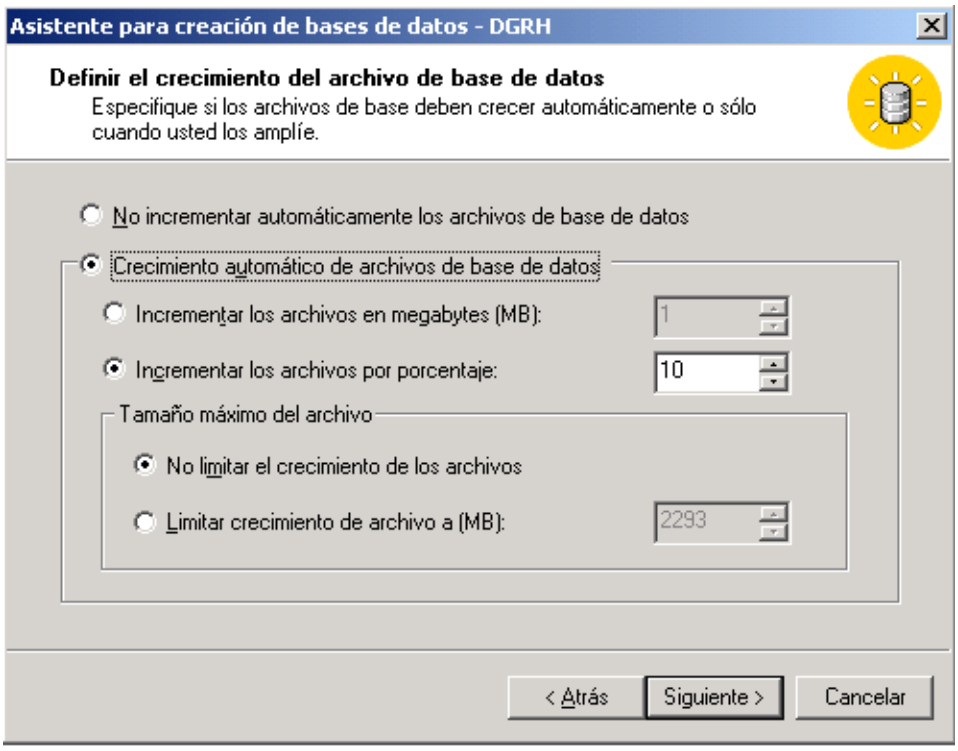


FIGURA 4.2.3 DEFINICIÓN DE ARCHIVOS DE CRECIMIENTO DE LA BASE DE DATOS

Cabe mencionar que también se deben tener en cuenta donde van a ser almacenados los datos y todas las transacciones que se puedan realizar en la base de datos ya que estos son muy importantes para poder hacer los respaldos parciales o totales de la instancia o poder generar o regenerar en alguna contingencia la base de datos tanto parcialmente como totalmente.

En la figura 4.2.4 se definen los archivos en donde se van a registrar todas las transacciones que se dan el la base de datos.

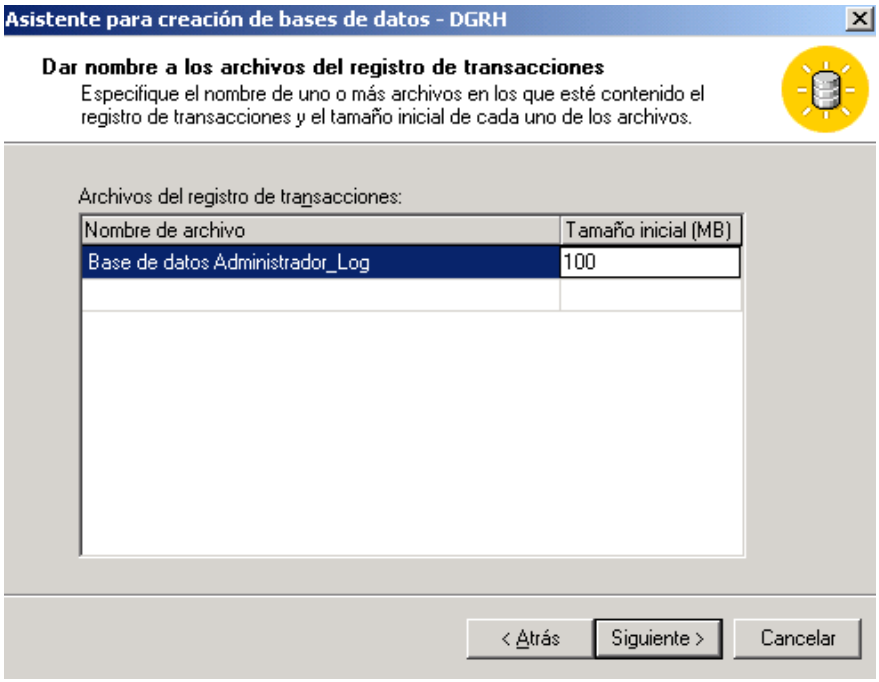


FIGURA 4.2.4 DEFINICIÓN DE ARCHIVOS DE TRANSACCIONES DE LA BASE DE DATOS

Con esta información, el asistente termina de configurar la base de datos como lo muestra en la figura 4.2.5.



FIGURA 4.2.5 FINALIZACIÓN DE LA CREACIÓN DE UNA BASE DE DATOS

Este software cuenta con la gran ventaja de que podemos recuperar la información o la base de datos a través de un asistente el cual nos permite dar un mantenimiento correcto además de poder respaldar la base de datos en un Backup el cual se muestra en la figura 4.2.6

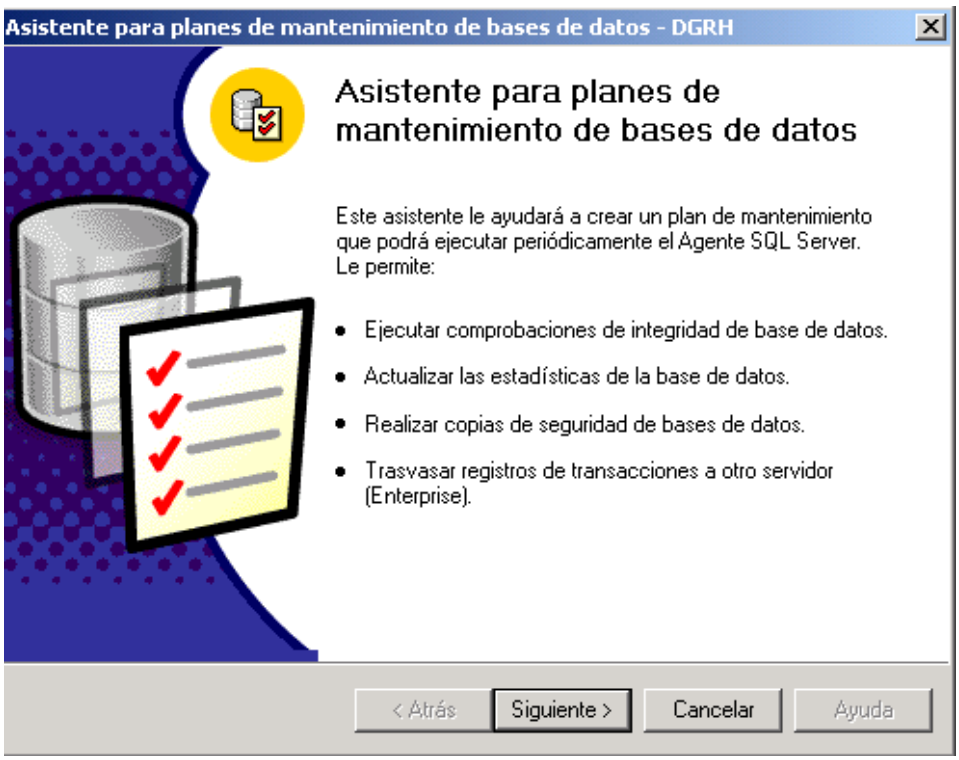


FIGURA 4.2.6 ASISTENTE DE MANTENIMIENTO DE LA BASE DE DATOS

Esto es muy importante ya que desde aquí se pueden prever futuros problemas ya que con esto se esta monitoreando el flujo de información a la base de datos y los posibles problemas que se pudieran encontrar en alguna inconsistencia del flujo de información.

El software nos permite administrar los índices de nuestra aplicación, nos permite comprobar la integridad de la base de datos así como de la información existente en ella para hacer un respaldo confiable de lo que es toda la instancia y poderla tener disponible en caso de recuperación en alguna contingencia sin perder nada relacionado con la instancia mencionada como lo muestra la figura 4.2.7.

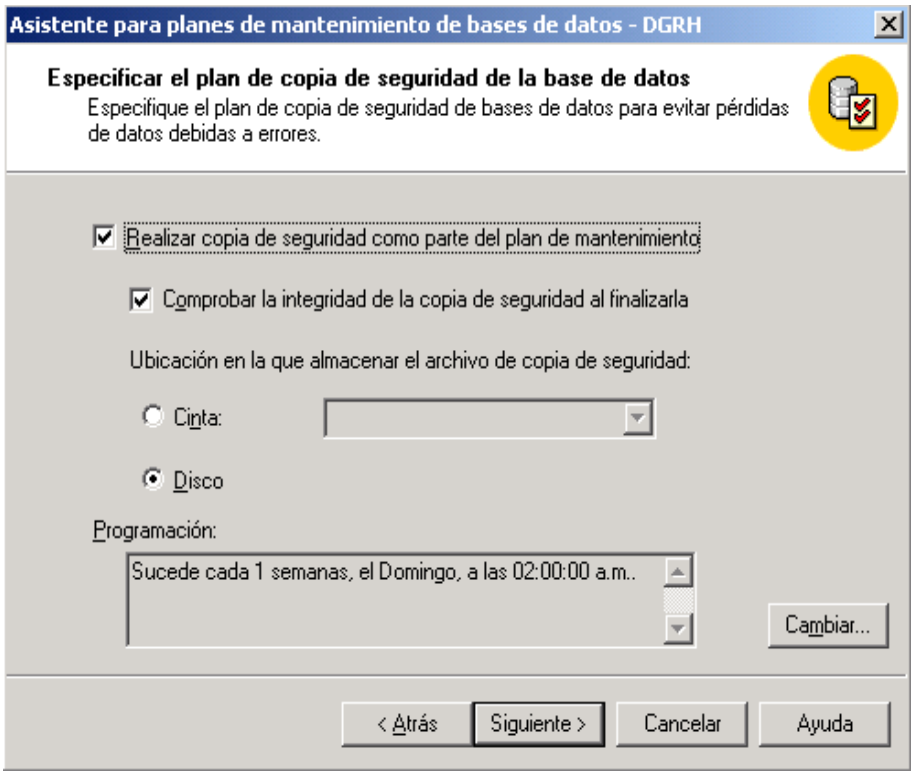


FIGURA 4.2.7 PANTALLA DONDE SE DEFINE EL BACKUP DE LA BASE DE DATOS

Cabe mencionar que también se realiza una administración y una definición de donde se almacenaran los archivos de registro de las transacciones que se hacen a la base de datos para una mejor recuperación de la información y la base de datos figura 4.2.8.

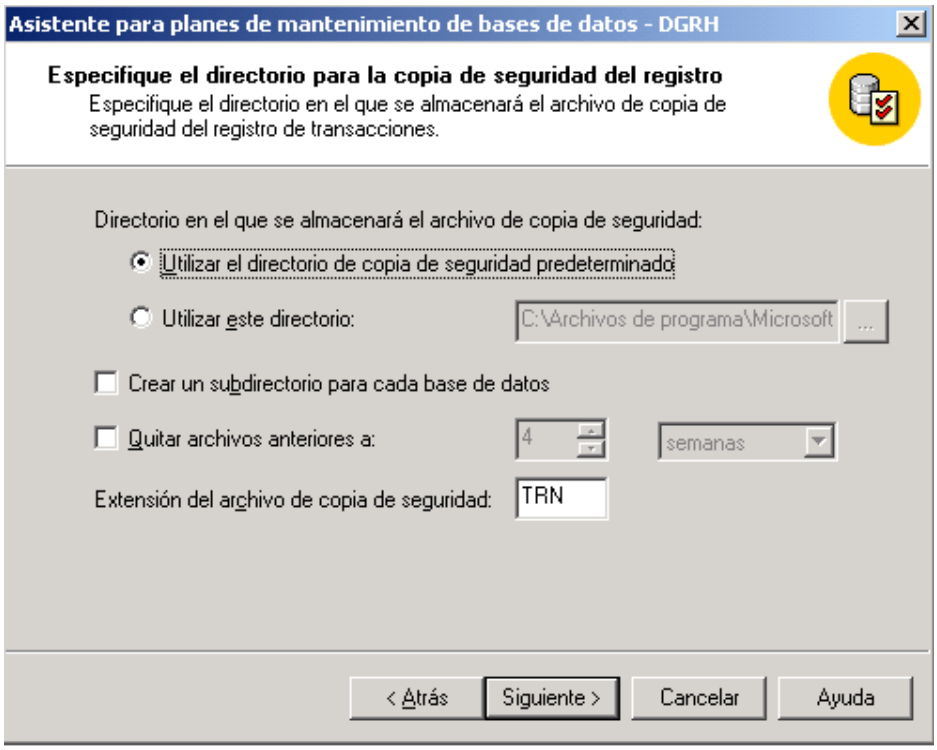


FIGURA 4.2.8 COPIA DE SEGURIDAD DE LAS TRANSACCIONES

Dando así por terminado el poder hacer el backup o mantenimiento de la base de datos como lo muestra la figura 4.2.9.

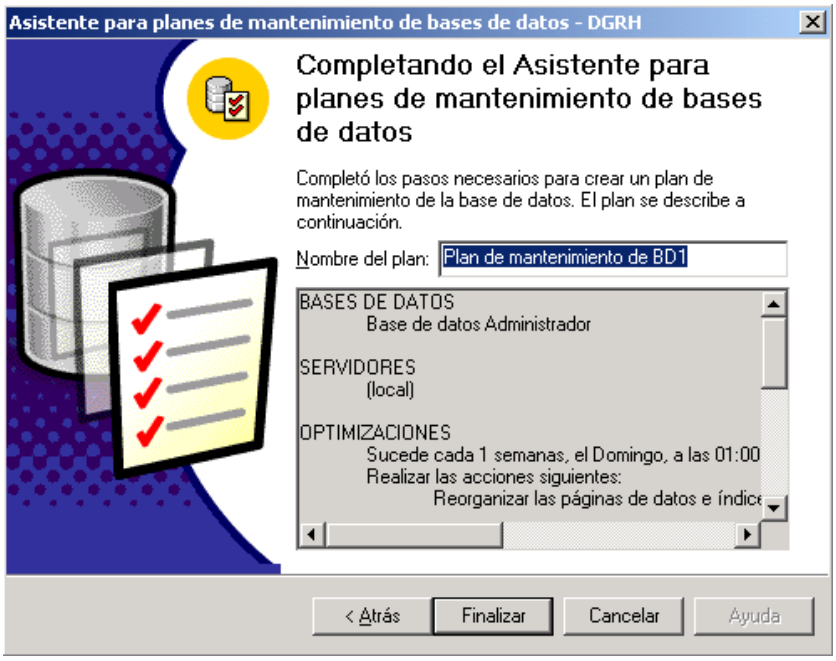


FIGURA 4.2.9 FINALIZACIÓN DEL ASISTENTE DE MANTENIMIENTO Y BACKUP

Definición y creación de las tablas de la base de datos

Dentro de nuestra base de datos, es muy importante la definición y creación de las tablas que necesitaremos para nuestra aplicación estas son:

- **Tabla de Contenido.** Esta tabla asigna un único valor al paquete, el cual también tiene una descripción de lo que tiene el paquete
- **Tabla Paquete.** esta tabla contiene los datos de registro, un ID característico que lo diferencia de los demás, un folio y una ubicación necesaria para su búsqueda y registro
- **Tabla Proyecto.** Esta tabla contiene los datos de un identificador único, así como los datos de nombre y cuenta contable a la que se va para afectar presupuesto
- **Tabla Envío.** esta contiene todos los datos generales del paquete, siendo el detalle ya del envío como numero de cajas, tipo de información, fechas de envío, estatus y comentarios
- **Tabla de usuarios.** Esta como su nombre lo dice tiene a las personas que pueden acceder a la información de la base de datos y poder así manipular la información correspondiente a los permisos que se le otorguen

Las tablas dentro de la aplicación fueron creadas bajo un mismo estándar por lo cual solo se explicara la creación de una a través del asistente. En la pantalla del SQL Server Enterprise Manager, se da clic con el botón derecho sobre la base de datos específica y se le da nueva tabla, apareciendo una pantalla como la figura 4.2.10.

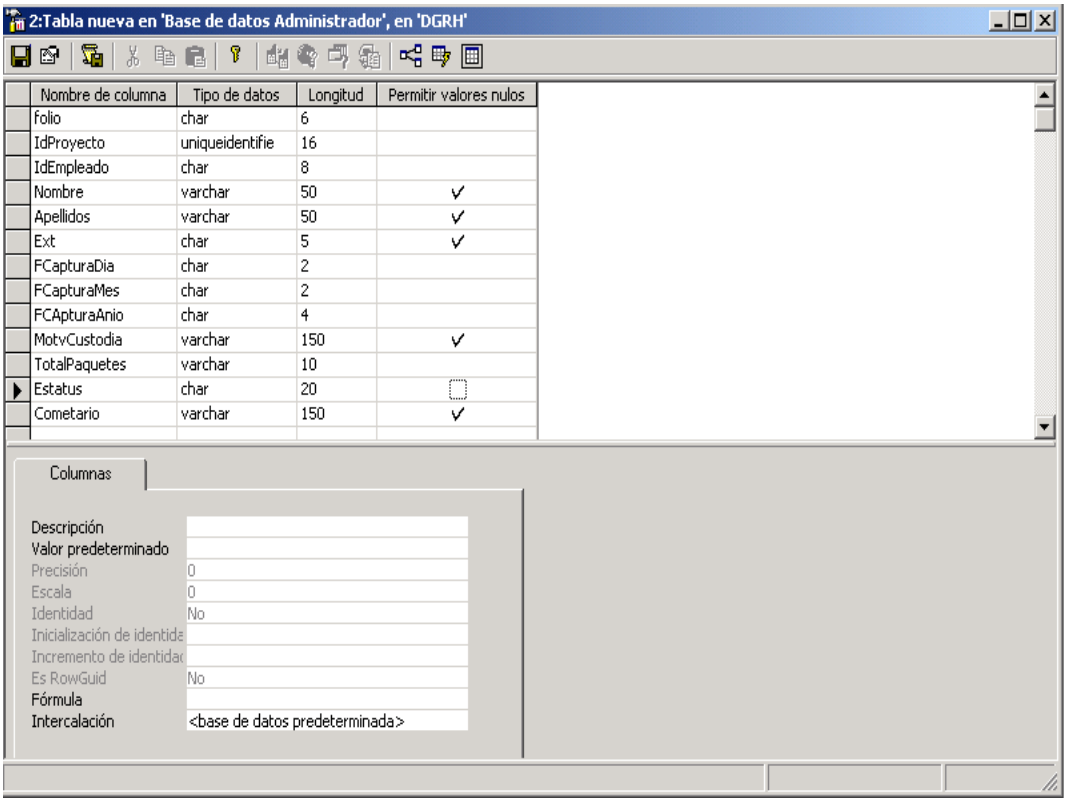


FIGURA 4.2.10 PANTALLA DE DEFINICIÓN DE UNA TABLA EN SQL SERVER 2000

En esta pantalla se definen los campos que va a tener la tabla, se elige el tipo de dato para cada campo, su longitud, si acepta o no valores nulos, si es o no una llave primaria, etc. todo esto de una manera gráfica y muy sencilla, esta forma grafica también genera un código con el cual también se puede ver como están creadas las tablas y se muestra a continuación.

```
CREATE TABLE Paquete (
 IdPaquete                    uniqueidentifier NOT NULL,
 Folio                        char (6) NOT NULL,
 Ubicación                    char (8) NOT NULL,
FechaDestruccionDia           varchar (20) NOT NULL,
FechaDestruccionMes           varchar (20) NOT NULL,
FechaDestruccionAnio          varchar (20) NOT NULL,
)
Go
ALTER TABLE EDocumento
      ADD PRIMARY KEY NONCLUSTERED (IdEDocumento)
Go
```

Creación de los Diagramas del Modelo Entidad Relación

El modelo entidad relación (E-R) se basa en una percepción del mundo real, la cual esta formada por objetos básicos llamados entidades y las relaciones entre estos objetos así como las características de estos objetos llamados atributos. En nuestro caso particular existen 6 entidades con atributos distintos que reflejan la problemática que tratamos de resolver. Estas entidades se crearon como tablas las cuales se relacionaron así mediante el asistente de creación que a continuación se muestra en las figura 4.2.11 y 4.2.12.

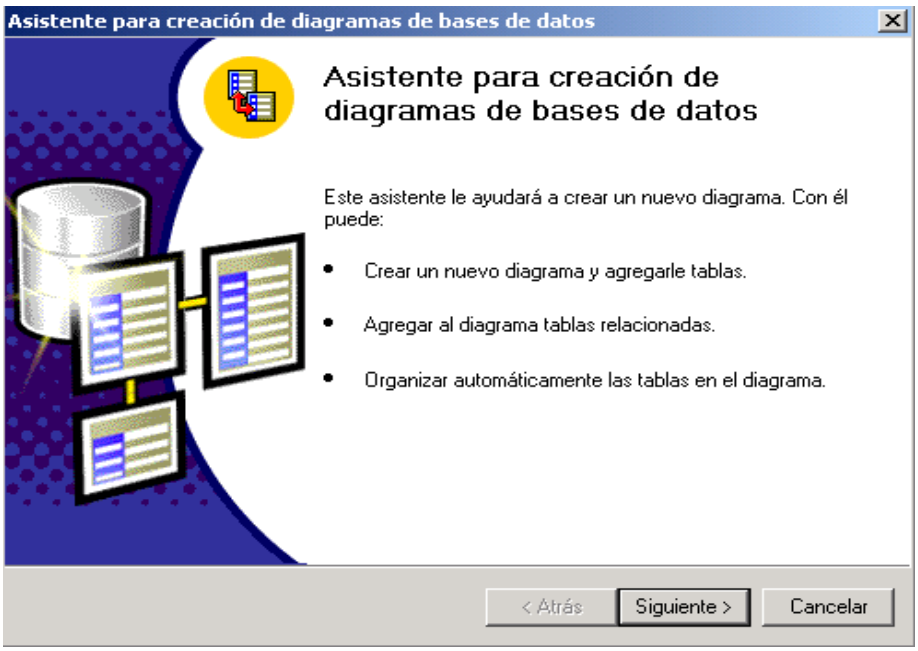


FIGURA 4.2.11 ASISTENTE DE CREACIÓN DE RELACIONES

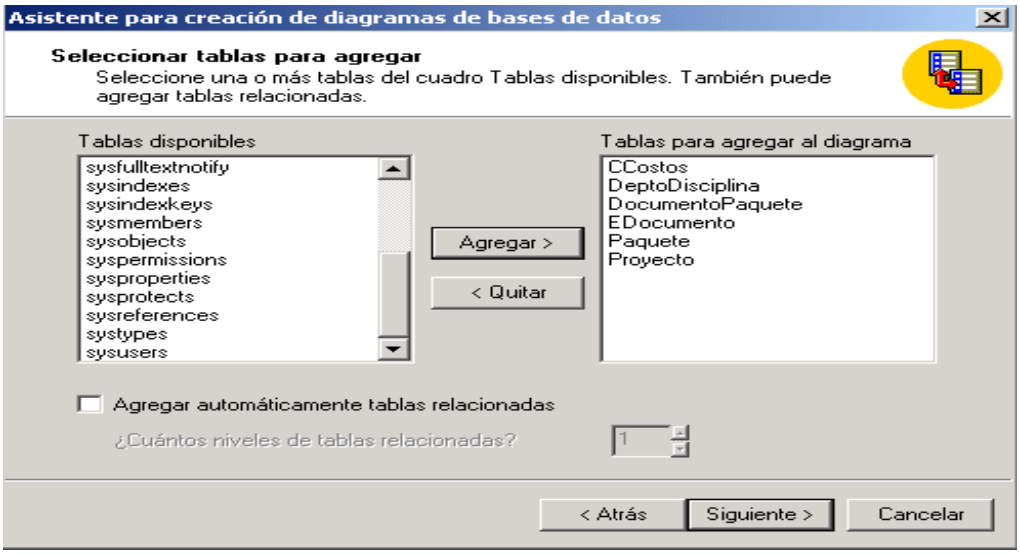


FIGURA 4.2.12 SELECCIÓN DE ENTIDADES

En la figura 4.213 se muestra como se crean las relaciones existentes entre las diversas entidades que están en nuestra problemática y como se unen mediante diversas llaves primarias y que nos ayudan a hacer referencia a las demás entidades que definen un objeto en particular.

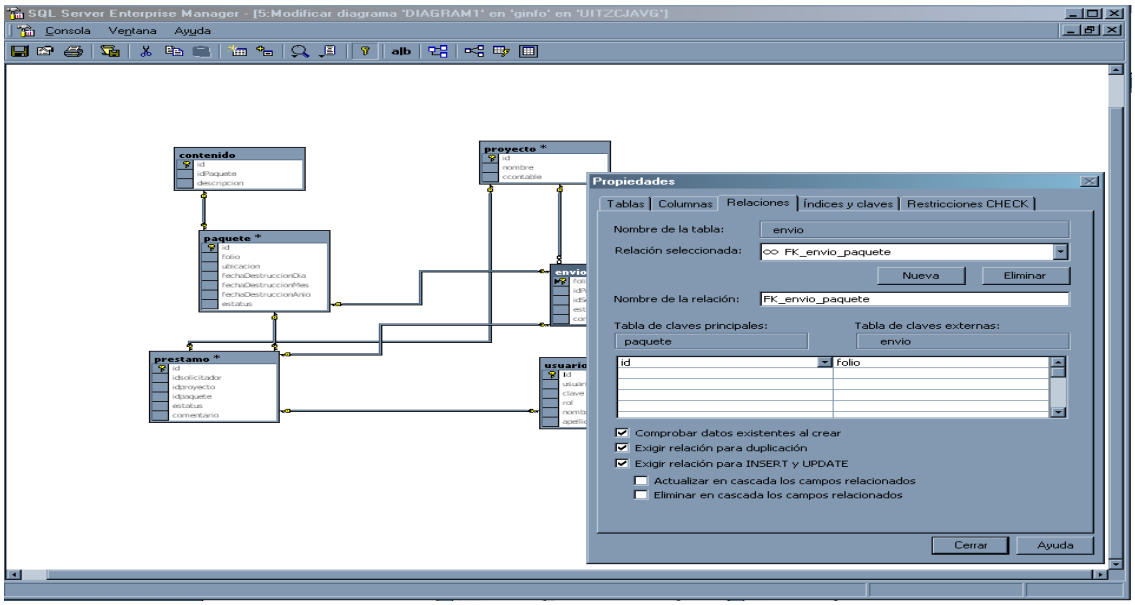


FIGURA 4.2.13 CREACIÓN DE RELACIONES ENTRE ENTIDADES

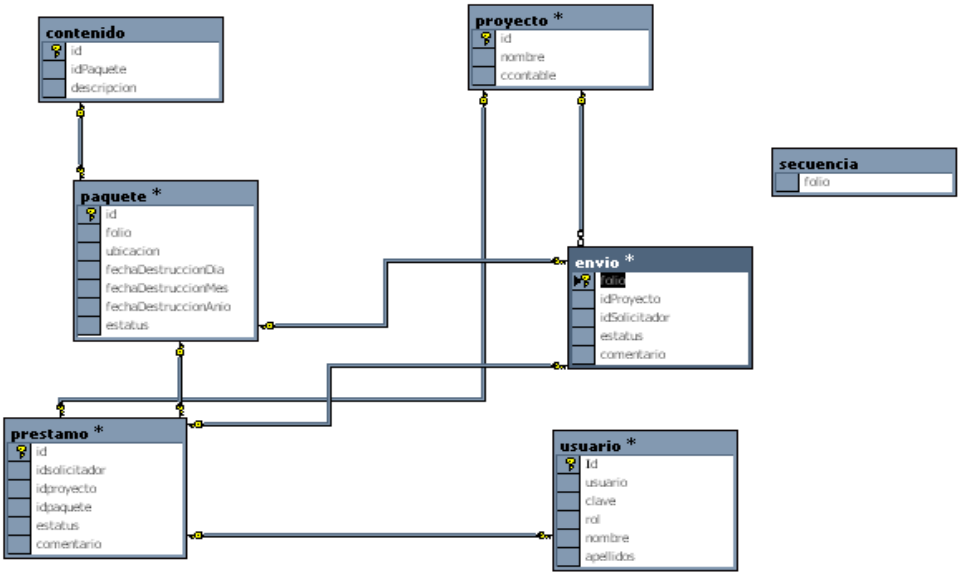


FIGURA 4.2.14 DIAGRAMA ENTIDAD RELACIÓN

El modelo entidad relación final (figura 4.2.14) muestra como las diferentes entidades se relacionan para definir la problemática que tenemos y como se accede a la información necesaria para poder definir una entidad con todos sus atributos mediante una consulta sencilla de información. Las relaciones en este caso particular se hicieron de uno a muchos ya que el acceso a ella parte de un numero único seleccionado de un catalogo el cual da información de una sola entidad, el que relaciona ese mismo con un detalle que tiene varios registros por proyecto, el cual se relaciona de uno a muchos ya que el documento que se registra puede tener mas de un registro de captura, teniendo así la información necesaria para solucionar la búsqueda de información.

JDBC

Hoy en día se puede observar la cantidad de usuarios en el World Wide Web, porque dentro de este ambiente se puede encontrar gran cantidad de información. Es por eso que muchas compañías están usando bases de datos relacionales para manejar la información en sus sitios Web. Para la gente del mundo Windows, JDBC es para Java lo que **ODBC** es para Windows. Windows en general no sabe nada acerca de las bases de datos, pero define el estándar ODBC consistente en un conjunto de primitivas que cualquier driver o fuente

ODBC debe ser capaz de entender y manipular. Los programadores que a su vez deseen escribir programas para manejar bases de datos genéricas en Windows utilizan las llamadas ODBC.

Con JDBC ocurre exactamente lo mismo: JDBC es una especificación de un conjunto de clases y métodos de operación que permiten a cualquier programa Java acceder a sistemas de bases de datos de forma homogénea. Lógicamente, al igual que ODBC, la aplicación de Java debe tener acceso a un driver JDBC adecuado. Este driver es el que implementa la funcionalidad de todas las clases de acceso a datos y proporciona la comunicación entre el API JDBC y la base de datos real. La necesidad de JDBC, a pesar de la existencia de ODBC, viene dada porque ODBC es un interfaz escrito en lenguaje C, que al no ser un lenguaje portable, haría que las aplicaciones Java también perdiesen la portabilidad. Y además, ODBC tiene el inconveniente de que se ha de instalar manualmente en cada máquina; al contrario que los drivers JDBC, que al estar escritos en Java son automáticamente instalables, portables y seguros.

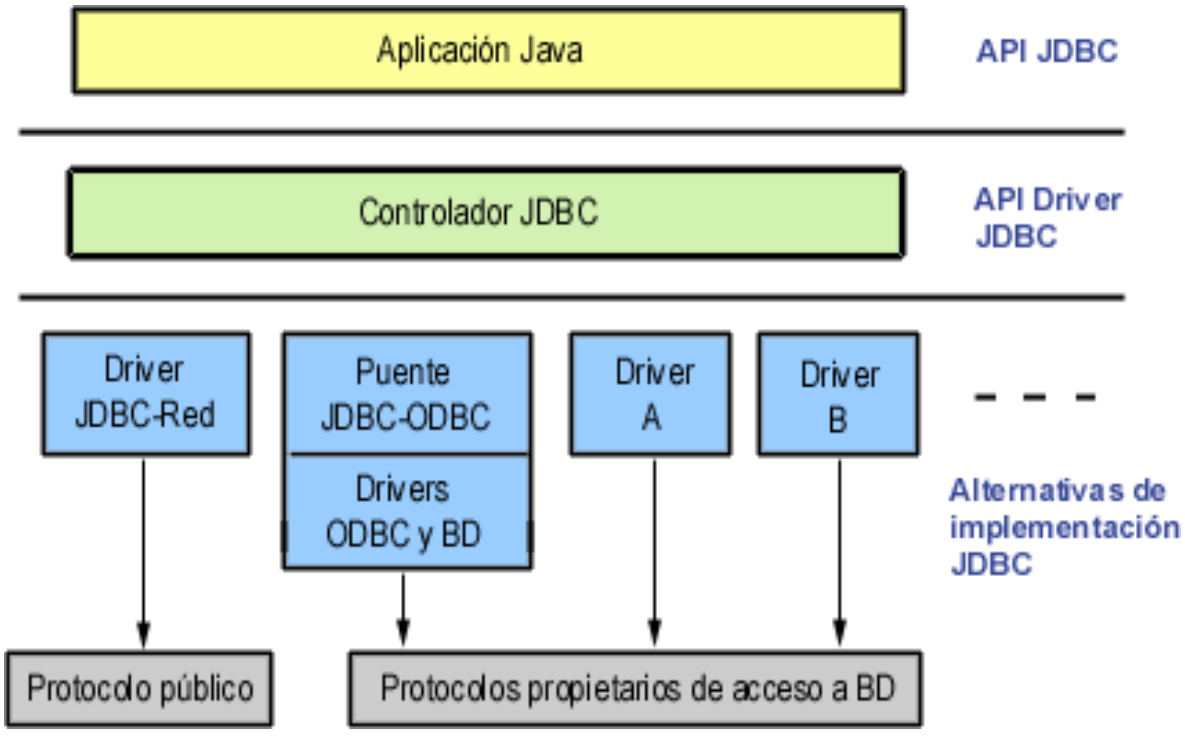


FIGURA 4.2.15 ACCESO A BASES DE DATOS A TRAVÉS DE JDBC

Controladores JDBC-ODBC

Necesitamos acceder a un origen de datos ODBC pero contamos con una API que usa el estándar JDBC. Para solventar este problema las empresas realizan drivers que traducen el ODBC a JDBC. Hay varios tipos de Driver, pero para nuestro ejemplo usaremos los llamados puentes JDBC-ODBC.

Para realizar la conexión a una base de datos JDBC-ODBC necesitaremos crear un perfil DSN desde el panel de control y posteriormente accederemos a la base de datos a partir del nombre del perfil. En el perfil DSN lo que se hace es indicar el driver a utilizar, así como el archivo o archivos del origen de datos. Estos son los pasos a llevar a cabo para configurar un perfil DSN.

1.- Iremos a **Panel de Control**. Ello se hace desde Inicio->Configuración o desde Mi PC.

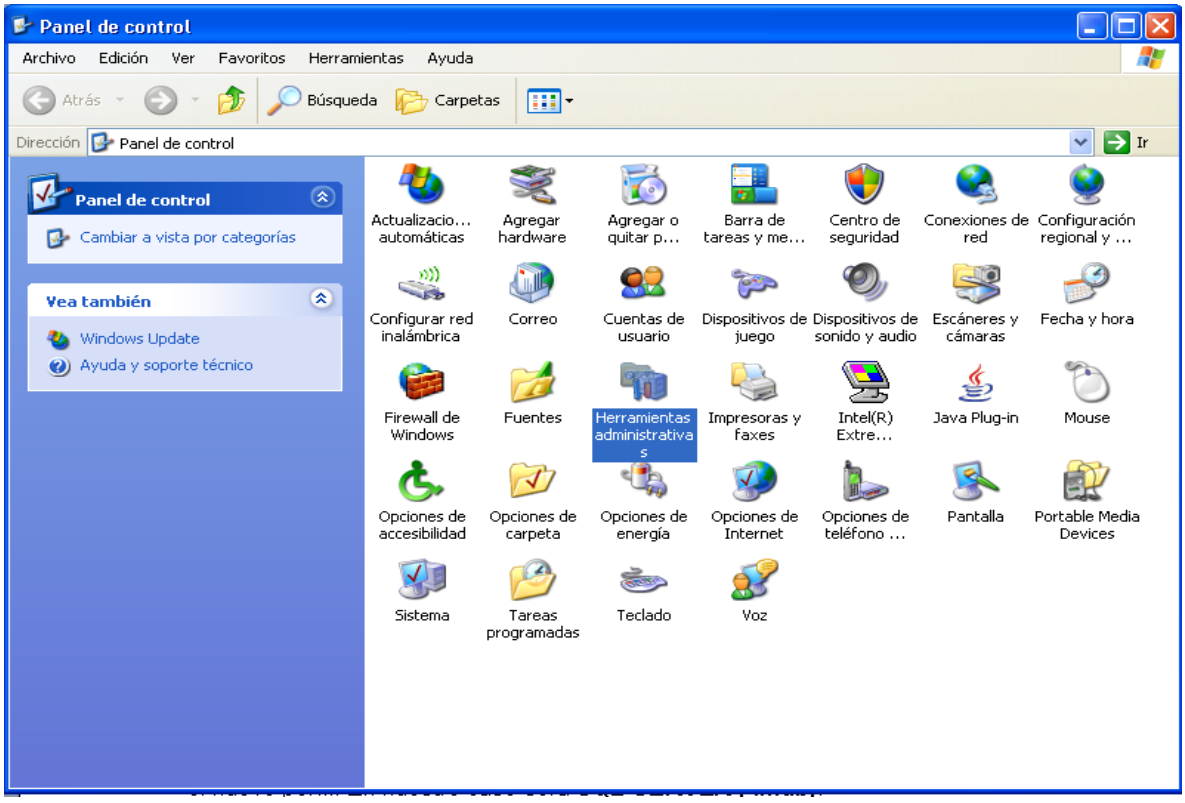


FIGURA 4.2.16 CREACIÓN DEL JDBC

2.- Ahora hacemos doble-click en el icono de **Fuentes de datos ODBC.**

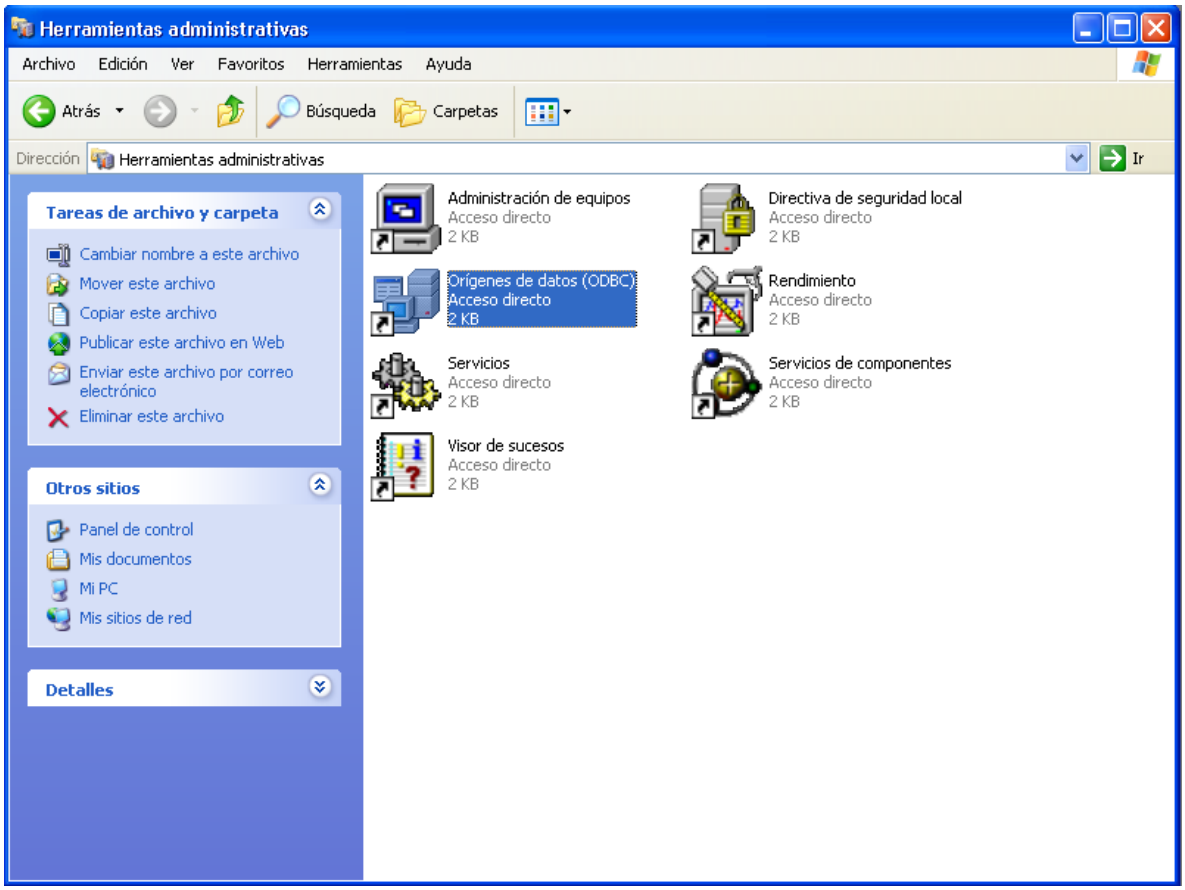


FIGURA 4.2.17 ELECCIÓN DEL JDBC

3.- En nuestra pantalla aparecerá ahora la pestaña **DSN usuario** seleccionado. Para crear un nuevo perfil haremos click en **Agregar...**

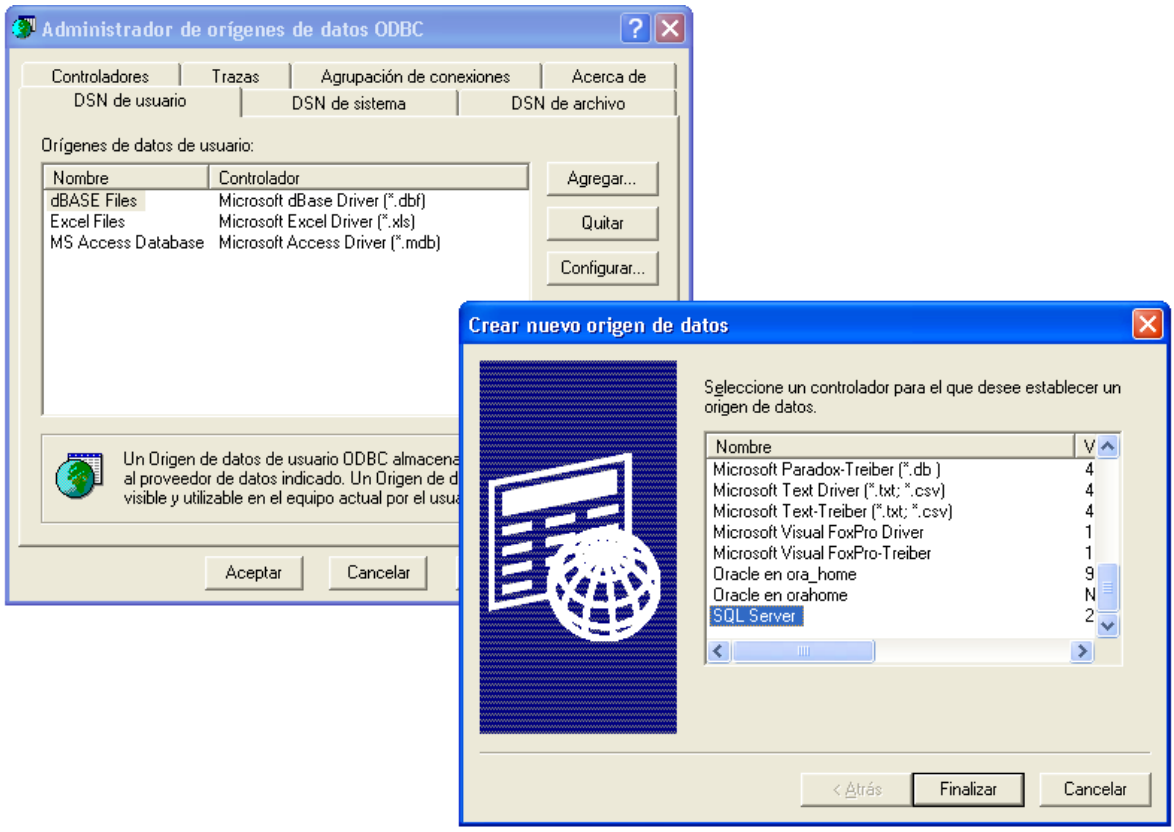


FIGURA 4.2.15 CONFIGURACIÓN DEL JDBC

4.- A continuación se nos pide que ingresemos el controlador que vamos a usar en el nuevo perfil. En nuestro caso será **SQL SERVER (*.mdb)**.

5.- Una vez aquí sólo nos queda dar un nombre al origen de datos y especificar el archivo .mdb de origen. Tras aceptar la ventana ya tenemos creado un perfil con lo que ya podemos comenzar a programar y realizar la conexión a la base de datos que utilizará la aplicación.

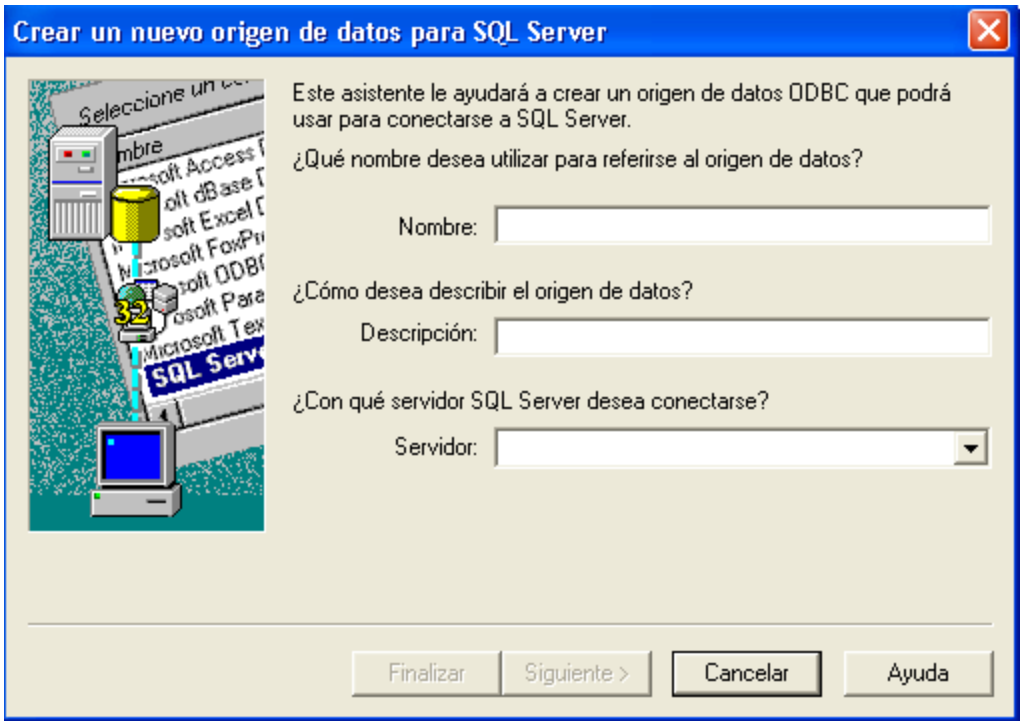


FIGURA 4.2.15 ACCESO ELECCIÓN DEL SERVIDOR Y LA BASE DE DATOS PARA EL JDBC

Lo que necesitamos para hacer nuestro programa es la API JDBC incluida en la última versión del JDK. El paquete a utilizar y el cual deberemos importar es el paquete java.sql.

Las primeras líneas de código suelen ser rutinarias ya que siempre serán muy similares.

Lo primero es hacer una llamada al Driver JDBC-ODBC para cargarlo. Eso se consigue con las siguientes líneas de código:

```
try{
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");                    }
catch(Exception e){ System.out.println("No se ha podido cargar el Driver
JDBC-ODBC"); }
```

Con esto ya tenemos cargado el Driver. Ahora básicamente trabajaremos con tres objetos. Estos objetos son: **Connection**, **Statement** y **ResultSet**. El objeto Connection se obtiene al realizar la conexión a la base de datos. El objeto Statement se crea a partir del anterior y nos permite ejecutar SQL para hacer consultas o modificaciones en la base de datos. En caso de hacer una consulta (SELECT... FROM...) se nos devolverá un objeto que representa los datos que deseamos consultar; este objeto es un objeto ResultSet (Hoja de resultados).

Debemos realizar la conexión a nuestro origen de datos. Para ello debemos crear un objeto Connection de la siguiente forma:

Connection con =
DriverManager.getConnection("jdbc:odbc:Nombre_Perfil_DSN",
"Nombre_Usuario", "Contraseña");

Los dos últimos parámetros pueden ser cadenas vacías "" a no ser que la base de datos las requiera. Con esto ya hemos realizado una conexión a nuestra base de datos. Pero esto todavía no es suficiente. Ahora vamos a crear un objeto Statement con el que podremos ejecutar y hacer consultas SQL. Si hasta ahora todo ha sido rutinario a partir de ahora vamos a poder crear código más adaptado a las necesidades de nuestro programa los cuales se mostraran mas adelante.

4.3 DISEÑO Y CONSTRUCCION DE LA INTERFAZ DEL USUARIO (FRONT END)

Para el diseño y construcción de la interfaz del usuario final (Front End) del sistema de Gestión de Información se han utilizado las siguientes herramientas de desarrollo. Antes de continuar se da a conocer el siguiente término.

Software libre. El cual se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. De modo más preciso se refiere a cuatro libertades de los usuarios del software.

- Libertad 1: La libertad de usar el programa, por cualquier persona con cualquier propósito
- Libertad 2: La libertad de estudiar cómo funciona el programa, y adaptarlo a sus necesidades. El acceso al código fuente es una condición previa para esto
- Libertad 3: La libertad de distribuir copias, con lo que se puede ayudar a el vecino. No existe el concepto de piratería
- Libertad 4: La libertad de mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie. El acceso al código fuente es un requisito previo para esto

Navegador Web

Herramienta de navegación Web de software libre la cual se obtiene en la siguiente liga:

Firefox – navegador WEB
http://www.mozila.org/products/firefox

Paquete de herramientas de desarrollo de Java

Se descargo la herramienta “SE v 1.4.2_08 SDK includes the JVM technology”

http://java.sun.com/j2ee/index.jsp
http://java.sun.com/j2ee/1.4/download.html

Compilador denominado Java Virtual Machine la cual esta contenida en la herramienta Java SDK (Software Development Kit).

Esta se obtuvo a través de la liga siguiente:

http://java.sun.com/j2se/1.4.2/download.html

Un editor de desarrollo que da el soporte al lenguaje Java; el cual se obtuvo de la siguiente liga:

JCreator

http://www.jcreator.com/download.php?productid=215952

Este se sugiere por no incluir herramientas de ayuda extras que saturan la memoria de la computadora y es muy ligero.

Tomcat

Servidor de Web/Aplicaciones de desarrollo
http://apache.mirror99.com/jakarta/tomcat-5/v5.0.28/bin/jakarta-tomcat-5.0.28.exe

SQL Server 2000

Esta herramienta es la base de datos y la conexión es a través de JDBC-ODBC en el cliente JSP (Java) y en el servidor Servlets (Java)

La aplicación está elaborada en Java en una arquitectura de dos capas que utiliza un servidor de Web (en servidores Web) Tomcat utilizado como tal y un navegador de Web como cliente.

Diseño de la interfaz

Una vez que se cuenta con el análisis del sistema y la lista de requerimientos, el siguiente paso es la creación de la interfaz de usuario. Esta es una etapa importante en el desarrollo de aplicaciones, la interfaz de usuario es con lo que va a interactuar el usuario final.

En ocasiones los programadores subestiman la importancia de la interfaz de usuario y concentran sus esfuerzos en la funcionalidad del sistema y en optimizar su desempeño. Sin embargo, si la interfaz de usuario es pobre, el usuario no se sentirá cómodo al usar la aplicación o, peor aún, dejará de usarla.

Una buena interfaz de usuario debe de ser diseñada según ciertos lineamientos:

- **Consistencia.** Todas las pantallas de la aplicación Web deben de tener una distribución consistente de imágenes, texto y controles gráficos. Ante acciones de usuario comunes (elegir un elemento de una lista desplegable, dejar un campo en blanco donde no es permitido) el sistema debe de mostrar mensajes con la misma estructura sin importar en que parte del sistema ocurra el evento. La única manera de lograr una interfaz de usuario consistente es la implementación de estándares de interfaz. Con ello, los grupos de desarrollo sabrán como acomodar los controles gráficos, dónde poner

las etiquetas, que lineamientos de justificación de texto seguir, entre otras cosas

- **Dar soporte a diferentes niveles de usuarios.** Una aplicación pueda ser usada tanto por usuarios neófitos como por expertos, bajo este tenor, la aplicación debe de ser configurable para mostrar toda la gama de opciones que un usuario experto puede requerir o solo mostrar las indispensables para que un usuario neófito pueda hacer su trabajo sin verse agobiado por una gran cantidad de opciones
- **Flujo de pantallas.** El paso de una pantalla a otra debe de ser coherente con el trabajo que intente realizar el usuario. De esta manera, no es nada conveniente hacer que el usuario tenga que pasar por una pantalla de configuración de colores en pantalla para llegar a la pantalla de configuración de márgenes de impresión
- **No sobrepoblar las pantallas.** Al presentar gran cantidad de controles gráficos en una misma pantalla dificultan la comprensión de la misma. Si para realizar una tarea determinada se requiere de la obtención de mucha información por parte del usuario, es muy recomendable dividir en diferentes pantallas sucesivas la captura de información y/o establecimiento de opciones. A esto último se le conoce como asistente o ``wizard''
- **Agrupar elementos relacionados.** Una buena práctica en la creación de interfaces de usuario es la agrupación de elementos que estén relacionados entre sí. Por ejemplo, podemos delimitar por un rectángulo o algún otro tipo de imagen los datos generales de un empleado tales como nombre, dirección, edad, etc.; En otro cuadro podemos agrupar otros datos tales como: Número de cuenta bancaria, fecha de expiración, saldo, etc.

Los anteriores puntos son aplicables a todos los tipos de aplicaciones, ya sea Web o de escritorio. Sin embargo, existen lineamientos que son específicos al desarrollo de aplicaciones Web, una buena referencia es ``Los siete pecados

mortales de un sitio Web (y por qué deben de evitarse a toda costa)" de Jesse Berst. Los grandes pecados o errores que se pueden cometer en el diseño de un sitio Web son:

- **Navegación inconsistente.** A veces se hace click en una barra de menú lateral, otras es un menú despegable. La ubicación de los títulos, gráficos, vínculos, etc. debe de ser consistente en todas las páginas del sitio
- **Vínculos rotos**
- **Sitios que requieren de un navegador específico.** Es muy molesto toparse con sitios que están diseñados para un navegador específico, así como los que requieren que se baje un **plug-in** determinado sin ofrecer una versión del documento que no requiera de dicho plug-in
- **No poner información de contacto.** Muchos diseñadores de sitios de empresas importantes olvidan poner la dirección de la empresa, números telefónicos, números de atención a clientes, etc.
- **Uso de marcos.** También conocidos como frames, traen consigo problemas con el uso de los botones Anterior y Siguiente de los navegadores, muchas veces son creados de un tamaño fijo y la información contenida en ellos no es visible completamente, etc.
- **Sitios que abren otras ventanas de navegación.** Es molesto para el navegador ingresar a un sitio y que éste abra nuevas ventanas de navegación con vínculos a otros sitios
- **Símbolos de ``En construcción".** La mayoría de los sitios en el Web están en constante evolución, Es poco profesional indicar que el sitio está en construcción

Creación de un prototipo

Un prototipo es una implementación básica de la interfaz de usuario que incluye todos los componentes visuales de la aplicación así como el código necesario para permitir al usuario desplazarse a través de todas las pantallas. Como el prototipo sólo es un modelo de prueba, no tiene integrada ninguna de las reglas

de negocios ni tiene acceso a la base de datos. El prototipo tiene la finalidad de presentar a los usuarios finales las pantallas con las que contará el sistema final para que éstos puedan determinar si el sistema cumple con sus necesidades.

La elaboración del prototipo es un proceso iterativo descrito a continuación:

- **Determinar las necesidades.** En esta etapa se define qué es lo que los usuarios esperan realizar con el sistema, qué reportes esperan que arroje, etc.
- **Construcción del prototipo.** Para la elaboración del prototipo es muy recomendable el uso de herramientas de alto nivel, tales como un editor de documentos HTML ya sea uno estándar o uno ``Lo que ves es lo que obtienes" (WYSIWYG, What You See Is What You Get, por sus siglas en inglés). Con el editor HTML se crean las diferentes pantallas de las que se compondrá el sistema estableciendo los vínculos entre ellas. En esta etapa es conveniente la creación de un diagrama de flujo de interfaz. Este diagrama muestra en un diagrama de bloques las pantallas de las que se compone el sistema y como se puede llegar a ellas. Con este diagrama se puede tener una mejor comprensión del sistema y se puede verificar fácilmente que el flujo de pantallas sea coherente
- **Evaluar el prototipo.** Una vez que se tiene una versión del prototipo, el siguiente paso es evaluarla con los usuarios para determinar si se cumple con sus expectativas. Es importante determinar lo que es correcto, lo que está mal y lo que falta. Si se descubre que hay errores en el prototipo o que éste tiene características faltantes tenemos que evaluar nuevamente desde el inicio
- **Determinar si el prototipo está terminado.** Una vez que en el paso anterior no se determinan nuevos requerimientos o estos son de poca importancia

En la parte de las pantallas para la elaboración de un nuevo envío se muestra que, para realizar la tarea es necesario pasar por varias ventanas, en la primera

se capturan los datos del remitente, en la segunda los datos del destinatario y en la tercera los datos pertinentes al paquete que se enviará, por último se muestra una pantalla que contiene un resumen con los datos previamente capturados. Este flujo de pantallas que piden información para realizar una tarea específica es conocido como Asistente. Se puede observar que en las pantallas para la captura de un envío, las flechas apuntan en ambas direcciones, lo que indica que el usuario puede avanzar por las pantallas y retroceder, en caso de que hubiese cometido algún error. El paso siguiente es crear el esqueleto del prototipo en HTML. Se deben de crear todas las pantallas de las que se compondrá el sistema, incluidos los reportes. En las figuras siguientes se muestran algunos ejemplos de las pantallas creadas.

Diseño de la interfaz

Para el sistema se crearon diferentes pantallas que dan la funcionalidad deseada.

Esquema general de la aplicación

En este inciso se muestra una aplicación completa desarrollada bajo los lineamientos descritos en éste documento.

Directorio de la aplicación. Es el directorio bajo el cual residirán los módulos de la aplicación.
Directorio corus: Contiene los módulos de autenticación, despachador de mensajes, manejo de errores, etc.
Directorio template: Contiene las plantillas html que definen la presentación de las pantallas.

En la figura 4.3.1 se muestra al navegador de Web FIRE FOX; la cual tiene la particularidad de navegación por ceja (pestaña) que lo hace más amigable con respecto a otros navegadores.

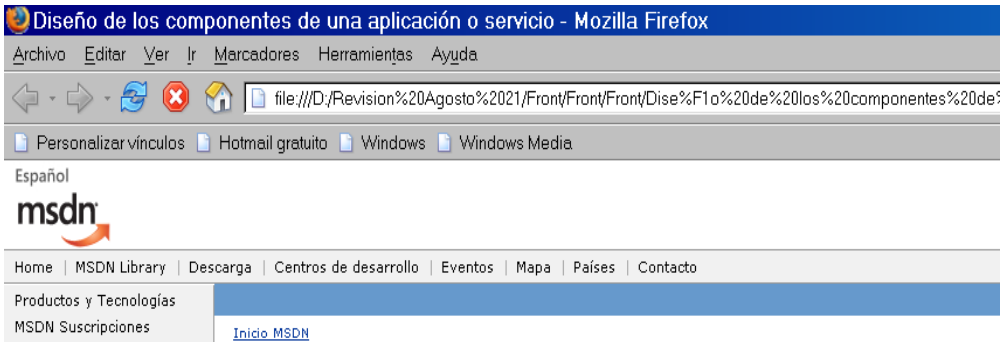


FIGURA 4.3.1 NAVEGADOR FIRE FOX

La aplicación del Sistema de Gestión de Información se puede acceder desde la Intranet de la siguiente manera:

Acceso a diversas pantallas por medio de las pestañas lo cual hace más fácil la navegación al usuario, véase figura 4.3.2

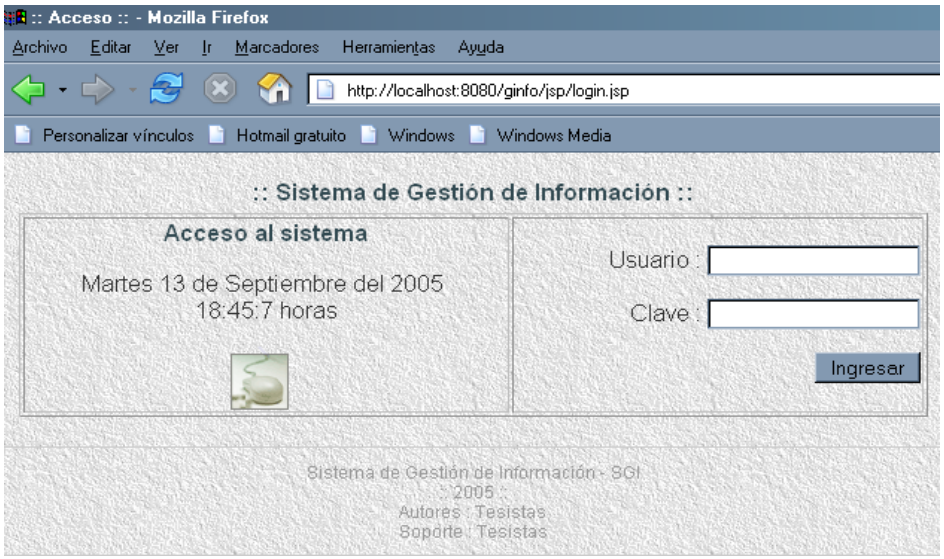


FIGURA 4.3.2 NAVEGACIÓN CON PESTAÑAS

La aplicación es de ventanas al estilo de la plataforma Windows y con los apuntadores de arrastre tradicionales. Véase figura 4.3.3

Se presenta menú tipo selección



FIGURA 4.3.3 VENTANAS TIPO WINDOWS

Las flechas de navegación nos permiten desplazamiento hacia atrás y hacia adelante. Véase figura 4.3.4.

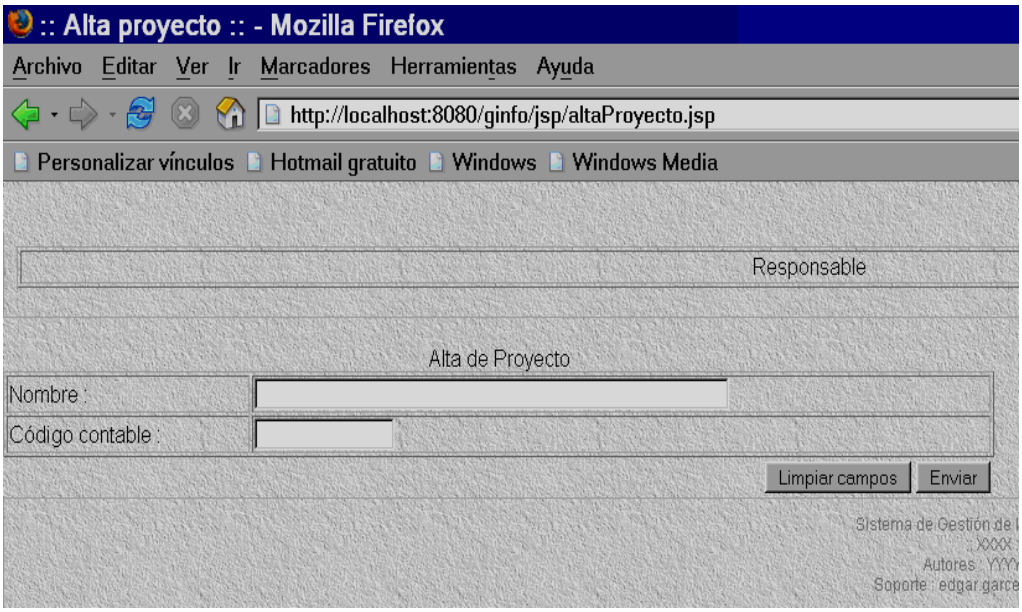


FIGURA 4.3.4 FLECHAS DE NAVEGACIÓN

Incluye campos de limpieza

Pantalla de actualización de información de proyectos. Vea figura 4.3.5

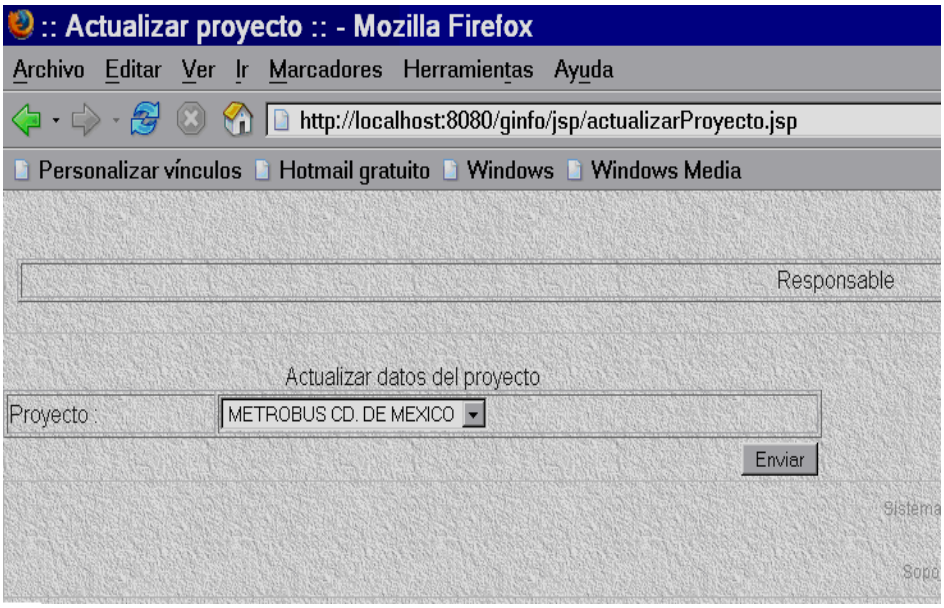


FIGURA 4.3.5 ACTUALIZACIÓN DE INFORMACIÓN

La pantalla incluye menú de selección de proyecto a ser actualizado

Pantalla Menú Alta de Usuario. Vea figura 4.3.6

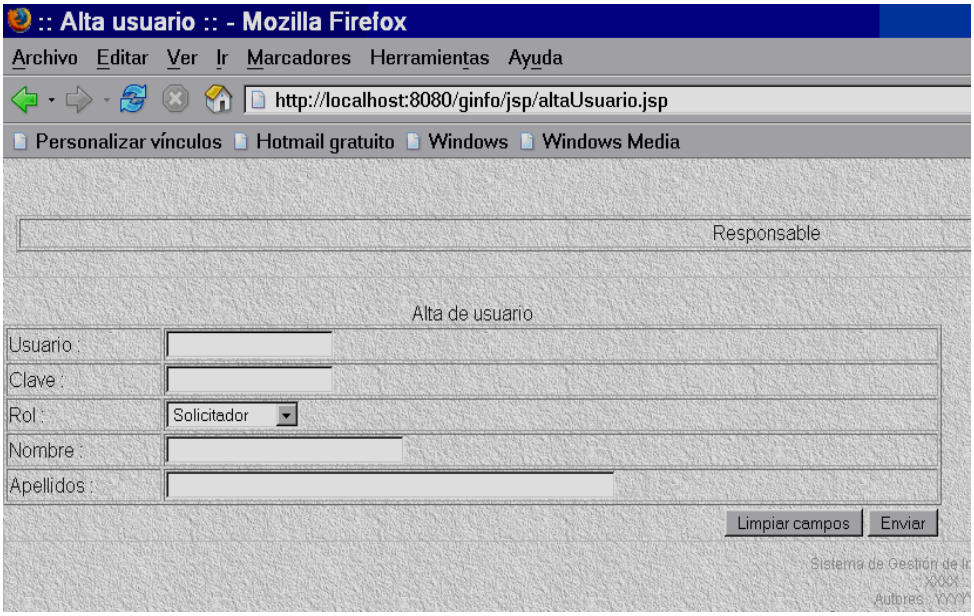


FIGURA 4.3.6 ALTA DE USUARIO

Alta de usuario de acuerdo a su rol del sistema sea solicitador, administrador o almacenista

Pantalla de Bajas de Usuario. Vea figura 4.3.7

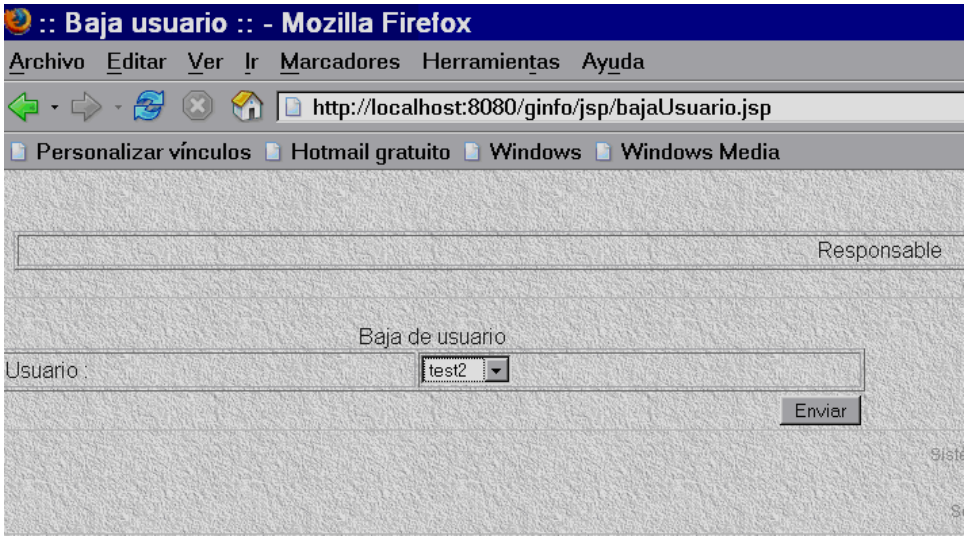


FIGURA 4.3.7 BAJAS DE USUARIO

Basta con incluir la cuenta del usuario y eliminar al usuario al accionar el botón **Enviar** para que sea eliminado y actualizada la información.

Menú rol de solicitador. Véase figura 4.3.8

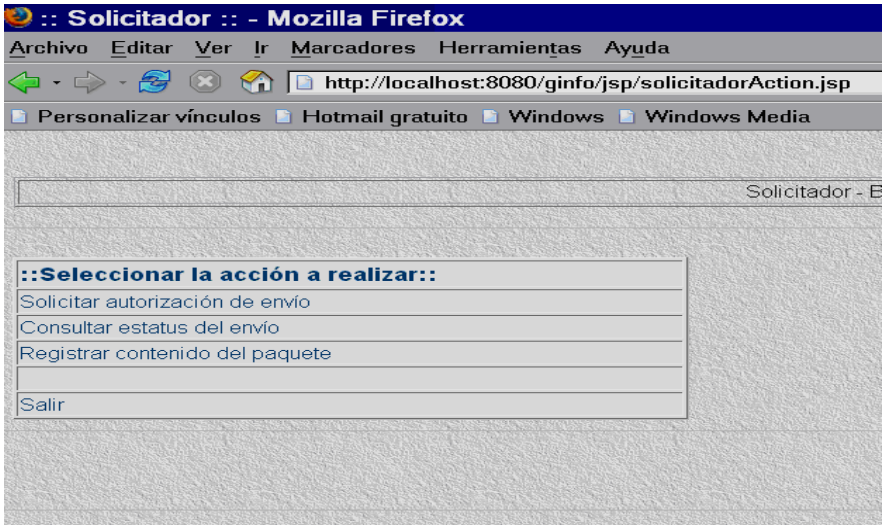


FIGURA 4.3.8 SOLICITADOR

Describe sus acciones principales, solicitud, consulta y registro.

4.4 PRUEBAS E INTEGRACIÓN DEL SISTEMA

Una de las últimas fases del ciclo de vida del proceso de desarrollo antes de entregar un programa para su explotación, es la fase de pruebas.

Una de las sorpresas con las que se suelen encontrar los programadores es la enorme cantidad de tiempo y esfuerzo que requiere esta fase. Se estima que la mitad del esfuerzo de desarrollo de un programa (tanto en tiempo como en gastos) se va en esta fase. Si hablamos de programas que involucran vidas humanas (medicina, equipos nucleares, etc) el costo de la fase de pruebas puede fácilmente superar el 80%.

El objetivo específico de la fase de pruebas es encontrar cuantos más errores, mejor. Es frecuente encontrarse con el error de afirmar que el objetivo de esta fase es convencerse de que el programa funciona bien. En realidad ese es el objetivo propio de las fases anteriores. Cumplido ese objetivo, lo mejor posible, se pasa a pruebas. Esto basta para reconocer que el objetivo último de todo el proceso de fabricación de programas sea hacer programas que funcionen bien, pero cada fase tiene su objetivo específico, y el de las pruebas es destapar errores.
Probar un programa es ejercitarlo con la peor intención a fin de encontrarle fallos. En lo que sigue intentaremos la siguiente estructura para la presentación:

Fases de prueba

- **Unidades**
  Planteamientos:
  - Caja blanca
    Cobertura:
    - de segmentos
    - de ramas
    - de condición/decisión
    - de bucles

    - Caja negra

      Cobertura de requisitos
- Integración
- Aceptación

La prueba de unidades se plantea a pequeña escala, y consiste en ir probando uno a uno los diferentes módulos que constituyen una aplicación.
Las pruebas de integración y de aceptación son pruebas a mayor escala, que puede llegar a dimensiones industriales cuando el número de módulos es muy elevado, o la funcionalidad que se espera del programa es muy compleja.

Las pruebas de integración se centran en probar la coherencia semántica entre los diferentes módulos, tanto de semántica estática (se importan los módulos adecuados, se llama correctamente a los procedimientos proporcionados por cada módulo), como de semántica dinámica (un módulo recibe de otro lo que esperaba). Normalmente estas pruebas se van realizando por etapas, englobando progresivamente más y más módulos en cada prueba.

Las pruebas de integración se pueden empezar en cuanto tenemos unos pocos módulos, aunque no terminarán hasta disponer de la totalidad. En un diseño descendente (top-down) se empieza a probar por los módulos más generales, mientras que en un diseño ascendente (down-up) se empieza a probar por los módulos de base.

El planteamiento descendente tiene la ventaja de estar siempre pensando en términos de la funcionalidad global, pero también tiene el inconveniente de que para cada prueba hay que inventarse algo sencillo (pero fiable) que simule el papel de los módulos inferiores, que aún no están disponibles.

El planteamiento ascendente evita tener que escribir módulos ficticios, pues vamos construyendo pirámides más y más altas con lo que vamos teniendo. Su

desventaja es que se centra más en el desarrollo que en las espectativas finales del cliente.

Estas clasificaciones no son las únicas posibles. Por ejemplo, en sistemas con mucha interacción con el usuario es frecuente codificar sólo las partes de cada módulo que hacen falta para una cierta funcionalidad. Una vez probada, se añade otra funcionalidad y así hasta el final. Esto da lugar a un planteamiento más vertical de las pruebas. A veces se conoce como codificación incremental.

Las pruebas de aceptación son las que se plantea el cliente final, que decide qué pruebas va a aplicarle al producto antes de darlo por bueno y pagarlo. De nuevo, el objetivo del que prueba es encontrar los fallos lo antes posible, en todo caso antes de pagarlo y antes de poner el programa en producción.

Prueba de Unidades

Normalmente cabe distinguir una fase informal antes de entrar en la fase de pruebas propiamente dicha. La fase informal la lleva a cabo el propio codificador en su despacho, y consiste en ir ejecutando el código para convencerse de que "básicamente, funciona". Esta fase suele consistir en pequeños ejemplos que se intentan ejecutar. Si el módulo falla, se suele utilizar un depurador para observar la evolución dinámica del sistema, localizar el fallo, y repararlo. En lenguajes antiguos, poco rigurosos en la sintaxis y/o en la semántica de los programas, esta fase informal llega a ser muy dura, laboriosa, y susceptible de dejar pasar grandes errores sin que se note. En lenguajes modernos, con reglas estrictas, hay herramientas que permiten análisis exhaustivos de los aspectos estáticos de la semántica de los programas: tipado de las variables, ámbitos de visibilidad, parámetros de llamada a procedimientos, etc. Hay asimismo herramientas más sofisticadas capaces de emitir "opiniones" sobre un programa y alertar de construcciones arriesgadas, de expresiones muy complicadas (que se prestan a equivocaciones), etc. A veces pueden prevenir sobre variables que pueden usarse antes de tomar algún valor (no inicializadas), variables que se cargan

pero luego no se usan, y otras posibilidades que, sin ser necesariamente errores en sí mismas, sí suelen apuntar a errores de verdad. Más adelante, cuando el módulo parece presentable, se entra en una fase de prueba sistemática. En esta etapa se empieza a buscar fallos siguiendo algún criterio para que "no se escape nada". Los criterios más habituales son los denominados de caja negra y de caja blanca.

Se dice que una prueba es de **caja negra** cuando prescinde de los detalles del código y se limita a lo que se ve desde el exterior. Intenta descubrir casos y circunstancias en los que el módulo no hace lo que se espera de él. Por oposición al término "caja negra" se suele denominar "caja blanca" al caso contrario, es decir, cuando lo que se mira con lupa es el código que está ahí escrito y se intenta que falle. Quizás sea más propia la denominación de "pruebas de caja transparente".

Caja blanca

Sinónimos:

- pruebas estructurales
- pruebas de caja transparente

En estas pruebas estamos siempre observando el código, que las pruebas se dedican a ejecutar con ánimo de "probarlo todo". Esta noción de prueba total se formaliza en lo que se llama "cobertura" y no es sino una medida porcentual de ¿cuánto código hemos cubierto? Hay diferentes posibilidades de definir la cobertura. Todas ellas intentan sobrevivir al hecho de que el número posible de ejecuciones de cualquier programa no trivial es (a todos los efectos prácticos) infinito. Pero si el 100% de cobertura es infinito, ningún conjunto real de pruebas pasaría de un infinitésimo de cobertura.

Cobertura de segmentos

A veces también denominada "cobertura de sentencias". Por segmento se entiende una secuencia de sentencias sin puntos de decisión. Como el

ordenador está obligado a ejecutarlas una tras otra, es lo mismo decir que se han ejecutado todas las sentencias o todos los segmentos. El número de sentencias de un programa es finito. Basta tomar el código fuente e ir contando. Se puede diseñar un plan de pruebas que vaya ejercitando más y más sentencias, hasta que hayamos pasado por todas, o por una inmensa mayoría.
En la práctica, el proceso de pruebas termina antes de llegar al 100%, pues puede ser excesivamente laborioso y costoso provocar el paso por todas y cada una de las sentencias. A la hora de decidir el punto de corte antes de llegar al 100% de cobertura hay que ser precavido y tomar en consideración algo más que el índice conseguido. En efecto, ocurre con frecuencia que los programas contienen código muerto o inalcanzable. Puede ser que este trozo del programa, simplemente "sobre" y se pueda prescindir de él, pero a veces significa que una cierta funcionalidad, necesaria, es inalcanzable, esto es un error y hay que corregirlo.

Cobertura de ramas

La cobertura de segmentos es engañosa en presencia de segmentos opcionales. Por ejemplo:

```
IF Condicion THEN

EjecutaEsto

ELSE ENDIF
```

Desde el punto de vista de cobertura de segmentos, basta ejecutar una vez, con éxito la condición, para cubrir todas las sentencias posibles. Sin embargo, desde el punto de vista de la lógica del programa, también debe ser importante el caso de que la condición falle (si no lo fuera, sobra el IF). Sin embargo, como en la rama ELSE no hay sentencias, con 0 ejecuciones tenemos el 100%. Para afrontar estos casos, se plantea un refinamiento de la cobertura de segmentos consistente en recorrer todas las posibles salidas de los puntos de decisión.

Para el ejemplo de arriba, para conseguir una cobertura de ramas del 100% hay que ejecutar (al menos) 2 veces, una satisfaciendo la condición, y otra no.
Estos criterios se extienden a las construcciones que suponen elegir 1 de entre varias ramas. Por ejemplo, el CASE. Nótese que si lográramos una cobertura de ramas del 100%, esto llevaría implícita una cobertura del 100% de los segmentos, pues todo segmento está en alguna rama. Esto es cierto salvo en programas triviales que carecen de condiciones (a cambio, basta 1 sola prueba para cubrirlo desde todos los puntos de vista). El criterio también debe refinarse en lenguajes que admiten excepciones. En estos casos, hay que añadir pruebas para provocar la ejecución de todas y cada una de las excepciones que pueden dispararse.

Cobertura de condición/decisión

La cobertura de ramas resulta a su vez engañosa cuando las expresiones booleanas que usamos para decidir por qué rama ir son complejas. Por ejemplo:

IF Condicion1 OR Condicion2 THEN HazEsto ELSE ENDIF

Las condiciones 1 y 2 pueden tomar 2 valores cada una, dando lugar a 4 posibles combinaciones. No obstante sólo hay dos posibles ramas y bastan 2 pruebas para cubrirlas. Pero con este criterio podemos estar cerrando los ojos a otras combinaciones de las condiciones.
Consideremos sobre el caso anterior las siguientes pruebas:

Prueba 1: Condicion1 = TRUE y Condicion2 = FALSE
Prueba 2: Condicion1 = FALSE y Condicion2 = TRUE
Prueba 3: Condicion1 = FALSE y Condicion2 = FALSE
Prueba 4: Condicion1 = TRUE y Condicion2 = TRUE

Bastan las pruebas 2 y 3 para tener cubiertas todas las ramas. Pero con ellos sólo hemos probado una posibilidad para la Condición1.

Para afrontar esta problemática se define un criterio de cobertura de condición/decisión que troza las expresiones booleanas complejas en sus componentes e intenta cubrir todos los posibles valores de cada uno de ellos.
Nótese que no basta con cubrir cada una de las condiciones componentes, si no que además hay que cuidar de sus posibles combinaciones de forma que se logre siempre probar todas y cada una de las ramas. Así, en el ejemplo anterior no basta con ejecutar las pruebas 1 y 2, pues aun cuando cubrimos perfectamente cada posibilidad de cada condición por separado, lo que no hemos logrado es recorrer las dos posibles ramas de la decisión combinada. Para ello es necesario añadir la prueba 3. El conjunto mínimo de pruebas para cubrir todos los aspectos es el formado por las pruebas 3 y 4. Aún así, nótese que no hemos probado todo lo posible. Por ejemplo, si en el programa ponemos AND donde queríamos poner OR (o viceversa), este conjunto de pruebas no lo detecta. Sólo queremos decir que la cobertura es un criterio útil y práctico, pero no es prueba exhaustiva.

Cobertura de bucles

Los bucles no son más que segmentos controlados por decisiones. Así, la cobertura de ramas cubre plenamente la esencia de los bucles. Pero eso es simplemente la teoría, pues la práctica descubre que los bucles son una fuente inagotable de errores, todos triviales, algunos mortales. Un bucle se ejecuta un cierto número de veces, pero ese número de veces debe ser muy preciso, y lo más normal es que ejecutarlo una vez de menos o una vez de más tenga consecuencias indeseables. Y, sin embargo, es extremadamente fácil equivocarse y redactar un bucle que se ejecuta 1 vez de más o de menos.
Para un bucle de tipo WHILE hay que pasar 3 pruebas:

- 0 ejecuciones
- 1 ejecución
- más de 1 ejecución

Para un bucle de tipo REPEAT hay que pasar 2 pruebas:

- 1 ejecución
- más de 1 ejecución

Los bucles FOR, en cambio, son muy seguros, pues en su cabecera está definido el número de veces que se va a ejecutar. Ni una más, ni una menos, y el compilador se encarga de garantizarlo. Basta pues con ejecutarlos 1 vez.
No obstante, conviene no engañarse con los bucles FOR y examinar su contenido. Si dentro del bucle se altera la variable de control, o el valor de alguna variable que se utilice en el cálculo del incremento o del límite de iteración, entonces eso es un bucle FOR con trampa. También tiene "trampa" si contiene sentencias del tipo EXIT (que algunos lenguajes denominan BREAK) o del tipo RETURN. Todas ellas provocan terminaciones anticipadas del bucle. Si el programa contiene bucles LOOP, o simplemente bucles con trampa, la única cobertura aplicable es la de ramas. El riesgo de error es muy alto, pero no se conocen técnicas sistemáticas de abordarlo, salvo reescribir el código. En la práctica de cada día, se suele procura alcanzar una cobertura cercana al 100% de segmentos. Es muy recomendable (aunque cuesta más) conseguir una buena cobertura de ramas. En cambio, no suele hacer falta ir por una cobertura de decisiones atomizadas.

Caja negra

Sinónimos:

- pruebas de caja opaca
- pruebas funcionales
- pruebas de entrada/salida
- pruebas inducidas por los datos

Las pruebas de caja negra se centran en lo que se espera de un módulo, es decir, intentan encontrar casos en que el módulo no se atiene a su especificación. Por ello se denominan pruebas funcionales, y el probador se limita a suministrarle datos como entrada y estudiar la salida, sin preocuparse de

lo que pueda estar haciendo el módulo por dentro. Las pruebas de caja negra están especialmente indicadas en aquellos módulos que van a ser interfaz con el usuario (en sentido general: teclado, pantalla, ficheros, canales de comunicaciones, etc.). Las pruebas de caja negra se apoyan en la especificación de requisitos del módulo. De hecho, se habla de "cobertura de especificación" para dar una medida del número de requisitos que se han probado. Es fácil obtener coberturas del 100% en módulos internos, aunque puede ser más laborioso en módulos con interfaz al exterior. En cualquier caso, es muy recomendable conseguir una alta cobertura en esta línea. El problema con las pruebas de caja negra no suele estar en el número de funciones proporcionadas por el módulo (que siempre es un número muy limitado en diseños razonables), sino en los datos que se le pasan a estas funciones. El conjunto de datos posibles suele ser muy amplio (por ejemplo, un entero). A la vista de los requisitos de un módulo, se sigue una técnica algebraica conocida como "clases de equivalencia". Esta técnica trata cada parámetro como un modelo algebraico donde unos datos son equivalentes a otros. Si logramos partir un rango excesivamente amplio de posibles valores reales a un conjunto reducido de clases de equivalencia, entonces es suficiente probar un caso de cada clase, pues los demás datos de la misma clase son equivalentes. El problema está pues en identificar clases de equivalencia, tarea para la que no existe una regla de aplicación universal, pero hay recetas para la mayor parte de los casos prácticos:

- sí un parámetro de entrada debe estar comprendido en un cierto rango, aparecen 3 clases de equivalencia: por debajo, en y por encima del rango.
- sí una entrada requiere un valor concreto, aparecen 3 clases de equivalencia: por debajo, en y por encima del rango.
- sí una entrada requiere un valor de entre los de un conjunto, aparecen 2 clases de equivalencia: en el conjunto o fuera de él.
- sí una entrada es booleana, hay 2 clases: si o no.

- los mismos criterios se aplican a las salidas esperadas: hay que intentar generar resultados en todas y cada una de las clases.

Ejemplo: utilizamos un entero para identificar el día del mes. Los valores posibles están en el rango [1..31]. Así, hay 3 clases:

- números menores que 1
- números entre 1 y 31
- números mayores que 31

Durante la lectura de los requisitos del sistema, nos encontraremos con una serie de valores singulares, que marcan diferencias de comportamiento. Estos valores son claros candidatos a marcar clases de equivalencia: por abajo y por arriba. Una vez identificadas las clases de equivalencia significativas en nuestro módulo, se procede a tomar un valor de cada clase, que no esté justamente al límite de la clase. Este valor aleatorio, hará las veces de cualquier valor normal que se le pueda pasar en la ejecución real. La experiencia muestra que un buen número de errores aparecen en torno a los puntos de cambio de clase de equivalencia. Hay una serie de valores denominados "frontera" (o valores límite) que conviene probar, además de los elegidos en el párrafo anterior. Usualmente se necesitan 2 valores por frontera, uno justo abajo y otro justo encima.

Pruebas de Integración

Las pruebas de integración se llevan a cabo durante la construcción del sistema, involucran a un número creciente de módulos y terminan probando el sistema como conjunto. Estas pruebas se pueden plantear desde un punto de vista estructural o funcional. Las pruebas estructurales de integración son similares a las pruebas de caja blanca, pero trabajan a un nivel conceptual superior. En lugar de referirnos a sentencias del lenguaje, nos referiremos a llamadas entre módulos. Se trata pues de identificar todos los posibles esquemas de llamadas y ejercitarlos para lograr una buena cobertura de segmentos o de ramas.

Las pruebas funcionales de integración son similares a las pruebas de caja negra. Aquí trataremos de encontrar fallos en la respuesta de un módulo cuando su operación depende de los servicios prestados por otro(s) módulo(s). Según nos vamos acercando al sistema total, estas pruebas se van basando más y más en la especificación de requisitos del usuario. Las pruebas finales de integración cubren todo el sistema y pretenden cubrir plenamente la especificación de requisitos del usuario. Además, a estas alturas ya suele estar disponible el manual de usuario, que también se utiliza para realizar pruebas hasta lograr una cobertura aceptable. En todas estas pruebas funcionales se siguen utilizando las técnicas de partición en clases de equivalencia y análisis de casos límite (fronteras).

Pruebas de Aceptación

Estas pruebas las realiza el cliente. Son básicamente pruebas funcionales, sobre el sistema completo, y buscan una cobertura de la especificación de requisitos y del manual del usuario. Estas pruebas no se realizan durante el desarrollo, pues sería impresentable de caras al cliente, sino una vez pasadas todas las pruebas de integración por parte del desarrollador. Por estas razones, muchos desarrolladores ejercitan unas técnicas denominadas "pruebas alfa" y "pruebas beta". Las pruebas alfa consisten en invitar al cliente a que venga al entorno de desarrollo a probar el sistema. Se trabaja en un entorno controlado y el cliente siempre tiene un experto a mano para ayudarle a usar el sistema y para analizar los resultados. Las pruebas beta vienen después de las pruebas alfa, y se desarrollan en el entorno del cliente, un entorno que está fuera de control. Aquí el cliente se queda a solas con el producto y trata de encontrarle fallos (reales o imaginarios) de los que informa al desarrollador. Las pruebas alfa y beta son habituales en productos que se van a vender a muchos clientes. Algunos de los potenciales compradores se prestan a estas pruebas bien por ir entrenando a su personal con tiempo, bien a cambio de alguna ventaja económica (mejor precio sobre el producto final, derecho a mantenimiento gratuito, a nuevas versiones, etc.). La experiencia muestra que estas prácticas son muy eficaces.

Pruebas de Caja blanca y Negra al Sistema de Gestión de Información desarrollado

Caja blanca

Se probará el siguiente código involucrado en la funcionalidad de ingreso de usuarios al sistema, el cuál está constituido por tres módulos:

- login.jsp : módulo de interfaz al usuario y dónde se captura los campos Usuario y Clave. Los campos Usuario y Clave son del tipo texto.
- ValidateUserServlet.java : módulo que válida el contenido de los campos capturados. Dependiendo de la captura, se muestra al usuario interfaces que indican un incorrecto llenado de los campos o que el Usuario/Clave no es válido para ingresar al sistema.
- RedirectServlet.java : módulo que, al haber sido previamente validado por el módulo ValidateUserServlet, selecciona la interfaz a desplegar en función del rol del usuario que ingresa al sistema, esto es, la interfaz para el rol de Solicitador o Responsable o Almacenista.

Los lenguajes utilizados son HTML y Java. El sistema ha sido desarrollado para funcionar en una Intranet.

login.jsp

```
<!-- Código del formulario -->
<form action="/ginfo/ValidateUserServlet" method="post">
        <p align="right">Usuario : <input type="text" name="usuario" maxlength="20"
        value=""></p>
        <p align="right">Clave : <input type="password" name="clave" maxlength="20"
        value=""></p>
        <p align="right"><input type="submit" value="Ingresar"></p>
</form>
<!-- Fin código formulario -->
```

En el código anterior identificamos lo siguiente:

- La etiqueta FORM tiene el atributo METHOD=POST, lo cuál es correcto cuando se está enviando información a través de un formulario

- Se permite la captura de texto para los campos Usuario y Clave, con una longitud de 20 caracteres alfanuméricos.
- El botón “Ingresar” simplemente dirige la captura al módulo ValidateUserServlet.java, que la válida y que corrobora que sea un usuario del sistema.

ValidateUserServlet.java

```
package library;

import java.util.*;
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;

public class ValidateUserServlet extends HttpServlet {

public void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException
{
        processRequest(request, response);
}

public void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException
{
        processRequest(request, response);
}

protected void processRequest(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException
{
//se inicia sesión
HttpSession session = request.getSession(false);

String usuario = request.getParameter("usuario");
String clave = request.getParameter("clave");

session.setAttribute("usuario",usuario);
session.setAttribute("clave",clave);

response.setContentType("text/html");
PrintWriter out = response.getWriter();

out.println("<html>");
out.println("<head>");
```

```
out.println("     <title>:: Valida usuario ::</title>");
out.println("<link rel=stylesheet href=/ginfo/jsp/style.css type=text/css>");
out.println("</head>");
out.println("<body>");
out.println("<p> </p>");

if ((usuario.equals(null)) || (usuario.length()==0)){
        out.println("<hr></hr><h2>Campo 'Usuario' vacío ...</h2><hr></hr>");
        out.println("<br>Por favor intentarlo de <a href=/ginfo/jsp/login.jsp>nuevo</a>");
}
else if ((clave.equals(null)) || (clave.length()==0)){
        out.println("<hr></hr><h2>Campo 'Clave' vacío ...</h2><hr></hr>");
        out.println("<br>Por favor intentarlo de <a href=/ginfo/jsp/login.jsp>nuevo</a>");
}
else {
        try{
        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver").newInstance();

        Connection conn =
        DriverManager.getConnection("jdbc:odbc:ginfo","ginfo","ginfo");
        Statement stmt = conn.createStatement();

        //Se verifica existencia del usuario
        ResultSet rs = stmt.executeQuery("SELECT id,rol,nombre,apellidos FROM
        usuario WHERE usuario='" + usuario + "' and clave='" + clave + "'");

        if (rs.next()){
                session.setAttribute("resultSet",rs);

                getServletConfig().getServletContext().getRequestDispatcher("/RedirectS
                ervlet").forward(request,response);
        }else{
                response.sendRedirect("/ginfo/jsp/errorPage.jsp");
        }
        conn.close();

        }catch(Exception ex){
                out.println("<h2>Error detectado : </h2>");
                out.println("<hr></hr>");
                out.println(ex.toString());
                out.println("<hr></hr>");
        }
}

out.println("</body>");
out.println("</html>");
}
}
```

En el código anterior identificamos lo siguiente:

- En caso de que en el formulario del módulo anterior se utilizara METHOD=GET, este módulo contempla la alternativa al redireccionar la petición al método protegido “processRequest()”, esto es adecuado pues contempla las dos opciones de envío del formulario, esto es GET y POST.
- El método “processRequest()”, válida el contenido de los campos “Usuario” y “Clave” a través de ramas condicionales del tipo:

IF (Condicion1 OR Condicion2) THEN

Esta condición es utilizada para comprobar que no hay texto en estos campos, sí es así se despliega un mensaje que indica que se capture texto en el campo correspondiente, sí hay texto se corrobora que corresponda a un usuario válido. Consideremos el comportamiento de la condición ante las siguientes pruebas:

Prueba 1: Condicion1 = true y Condicion2 = false
Prueba 2: Condicion1 = false y Condicion2 = true
Prueba 3: Condicion1 = false y Condicion2 = false
Prueba 4: Condicion1 = true y Condicion2 = true

Para la primera condición “if ((usuario.equals(null)) || (usuario.length()==0)){“, tenemos que se verifica sí el campo “usuario” es nulo o vacío, pero no se verifica sí este campo está capturado con espacios. Esto implica que es necesario revisar la parte de alta de usuarios para garantizar que el uso de espacios no es válido e incluir la validación de la inexistencia de espacios.

Aplicando las pruebas anteriores tenemos:

Usuario.equals(null) OR usuario.length==0

Prueba1:	true	OR	false	= true
Prueba2:	false	OR	true	= true
Prueba3:	false	OR	false	= false
Prueba4:	true	OR	true	= true

Que cumple con el objetivo de validar que no hay texto. Sólo hay que incluir el caso de uso de espacios.

Lo mismo se aplica a la condición “if ((clave.equals(null)) || (clave.length()==0)){”, para el caso del campo “clave”.

- Una vez comprobado que los campos contienen texto, se corrobora que el “usuario” y su “clave” son validos para ingresar al sistema. Para esto se lleva a cabo una consulta del tipo "SELECT id,rol,nombre,apellidos FROM usuario WHERE usuario='" + usuario + "' and clave='" + clave + "'". Sí la consulta genera una excepción, como en el caso de que el servidor de base de datos este fuera de servicio, se presenta un mensaje indicando el problema, sin “tirar” el sistema. Al obtener el resultado de la consulta, se presentan los casos de que sea un resultado “vacío” o un resultado “con datos”. Están contemplados ambos casos.

errorPage.jsp

```
<html>
<head>
<title>Página de Error</title>
<link rel="stylesheet" href="style.css" type="text/css">
</head>
<body>
<br>
<br>
<hr></hr>
        <table width="100%">
        <tr>
        <td align="center">
        <h1>
        Usuario/Clave incorrecto...!<br>
        Por favor intentarlo de <a href="/ginfo/jsp/login.jsp">nuevo</a>.
        </h1>
        </td>
        </tr>
```

```
        </table>
<hr></hr>

<br>
<jsp:include page="info.jsp"/>

</body>
</html>
```

El código anterior es el que se presenta cuando el Usuario y Clave no son válidos para un usuario del sistema.

RedirectServlet,java

```
package library;

import java.util.*;
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;

public class RedirectServlet extends HttpServlet {

public void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException
{
        processRequest(request, response);
}

public void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException
{
        processRequest(request, response);
}


protected void processRequest(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException
{
        try{
                HttpSession session = request.getSession(false);

                ResultSet rs = (ResultSet) session.getAttribute("resultSet");

                String id = rs.getString("id");
                String rol = rs.getString("rol");
                String nombre = rs.getString("nombre");
```

```
                String apellidos = rs.getString("apellidos");

                session.setAttribute("id",id);
                session.setAttribute("rol",rol);
                session.setAttribute("nombre",nombre);
                session.setAttribute("apellidos",apellidos);

                if (rol.equals("solicitador")){
                        response.sendRedirect("/ginfo/jsp/solicitadorAction.jsp");
                } else if (rol.equals("responsable")){
                        response.sendRedirect("/ginfo/jsp/responsableAction.jsp");
                } else if (rol.equals("almacenista")){
                        response.sendRedirect("/ginfo/jsp/almacenistaAction.jsp");
                }
        } catch (Exception ex){
        }
}
}
```

En el código anterior identificamos lo siguiente:

- Dada la información contenida en un objeto “session”, se obtiene el rol del usuario que ingresa al sistema y en función a este despliega la página correspondiente. Se observa que no está contemplado que sucede sí se presenta un rol no especificado y sí se presenta una excepción al obtener el objeto “sessión”.

NOTA: Los errores identificados fueron resueltos en el código correspondiente.

Caja negra

Sean las siguientes interfaces al usuario que están involucradas en la funcionalidad de ingreso de un usuario al sistema. La funcionalidad esperada es sí el usuario captura información válida en la pantalla de acceso, se permite el ingreso a la página correspondiente dependiendo del rol del usuario que ingresa, en caso contrario indica las razones por las que no se puede ingresar. La Figura 4.4.1 muestra la pantalla de acceso donde se capturan los datos del usuario que quiere ingresar. Sí el usuario no captura el campo ‘usuario’ se muestra la Figura 4.4.2, y si no captura el campo ‘clave’ se muestra la Figura 4.4.3

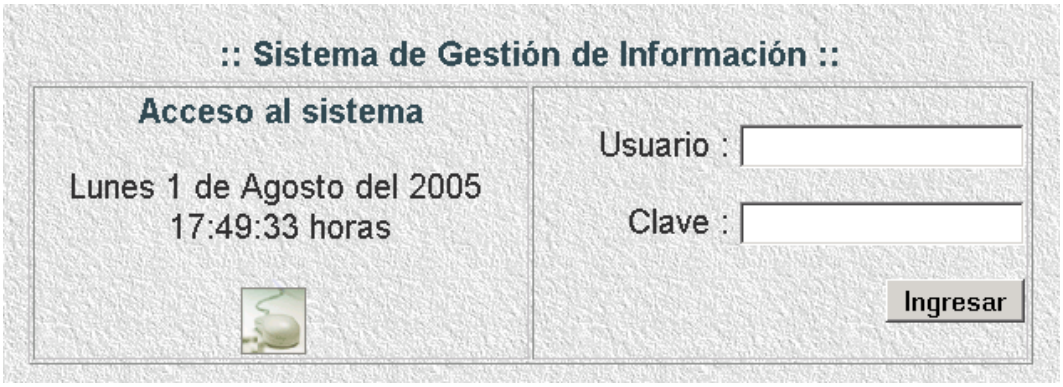


FIGURA 4.4.1 PANTALLA DE ACCESO AL SISTEMA

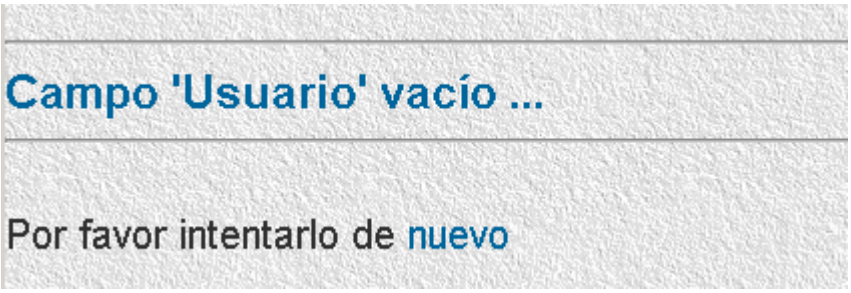


FIGURA 4.4.2 PANTALLA DE CAMPO 'USUARIO' VACÍO

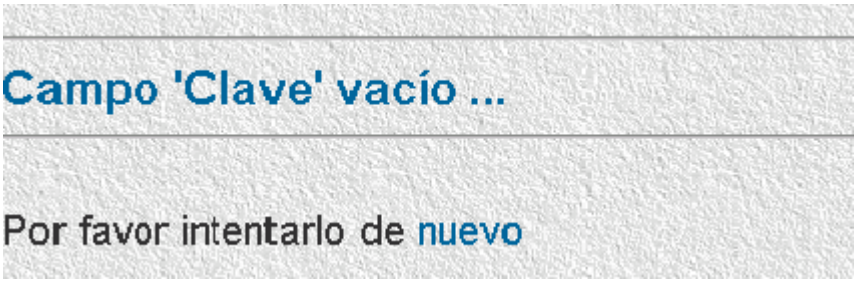


FIGURA 4.4.3 PANTALLA DE CAMPO 'CLAVE' VACÍO

Cuando el usuario captura los campos usuario/clave con datos que no son validos, la pantalla de la Figura 4.4.4 se presenta, indicando que que se intente de nuevo la captura.

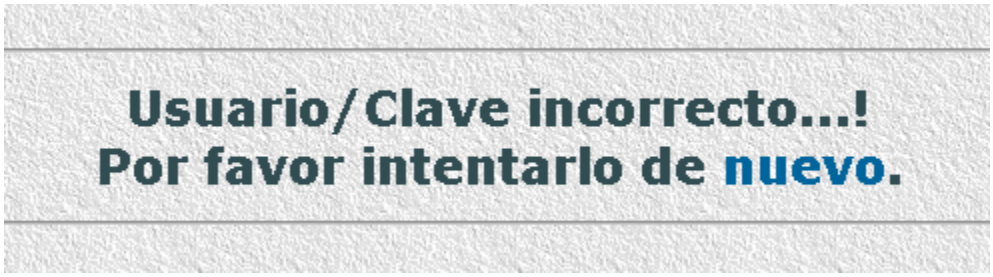


FIGURA 4.4.4 PANTALLA DE USUARIO/CLAVE INCORRECTO

Cuando el usuario captura los campos usuario/clave con datos validos, se presentan las pantallas de las Figuras 4.4.5, 4.4.6 y 4.4.7, que corresponden a los roles Solicitador, Responsable y Almacenista.

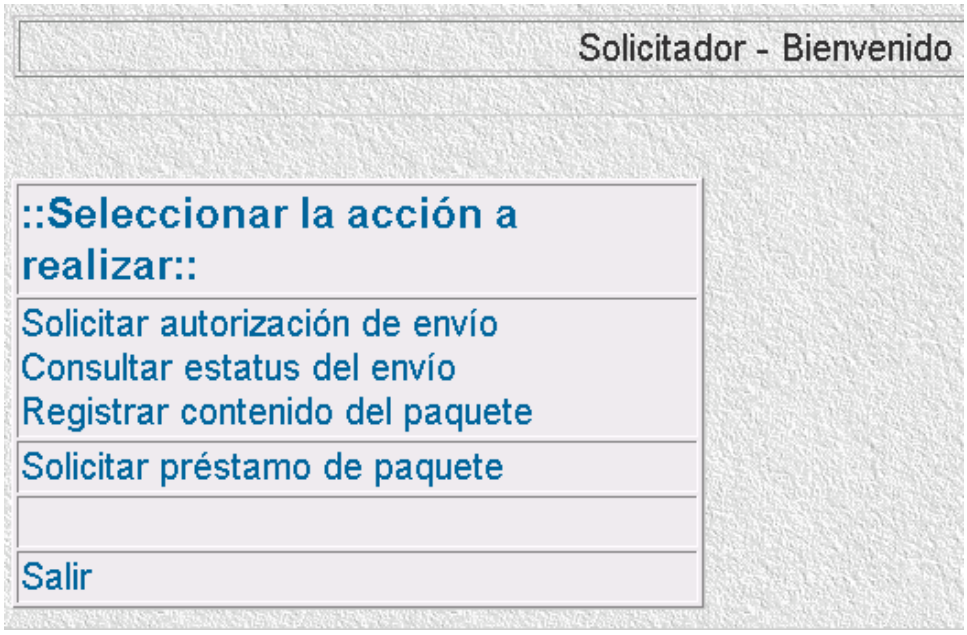


FIGURA 4.4.5 PANTALLA DEL ROL SOLICITADOR

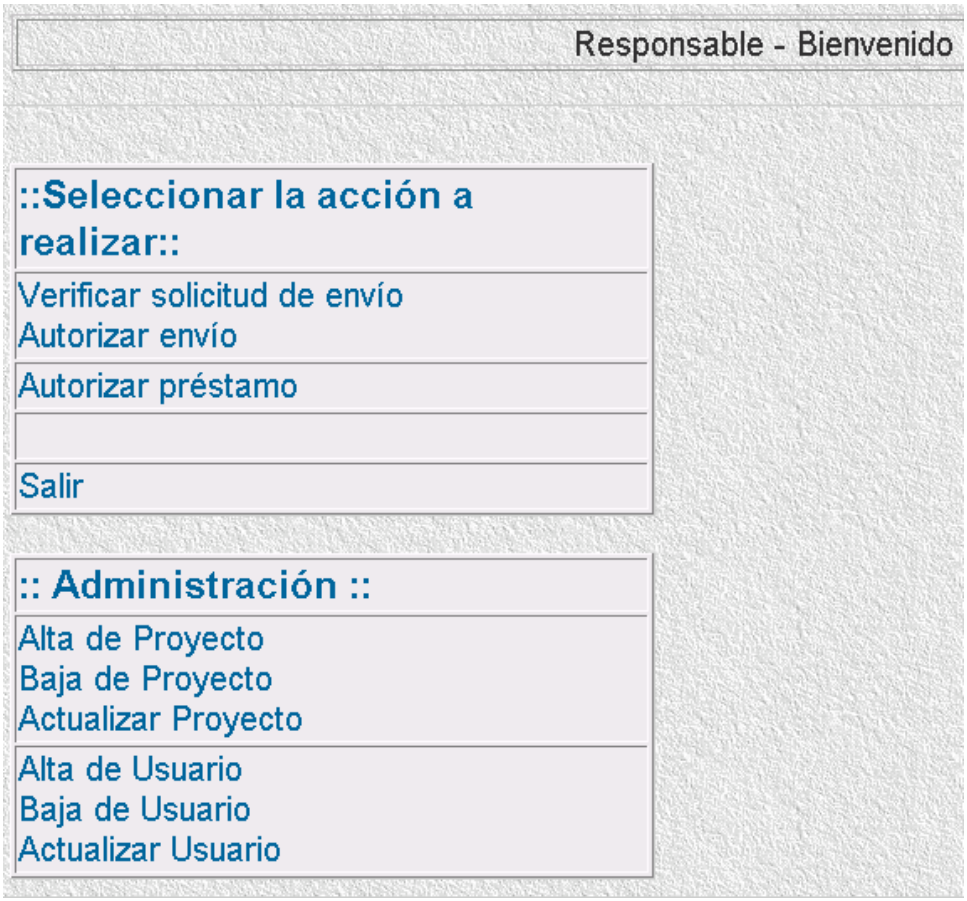


FIGURA 4.4.6 PANTALLA DEL ROL RESPONSABLE

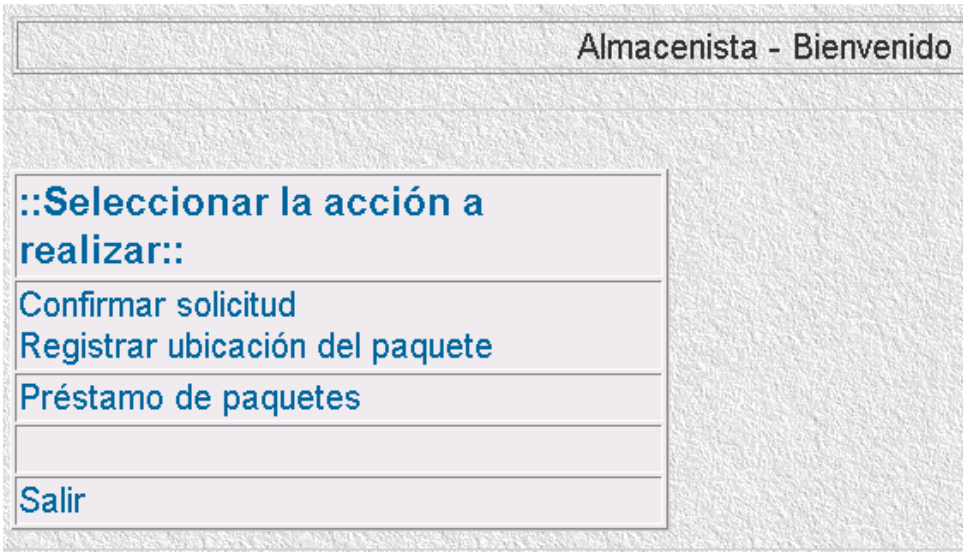


FIGURA 4.4.7 PANTALLA DEL ROL ALMACENISTA

4.5 GENERACIÓN DE REPORTES PARA LA TOMA DE DECISIONES

Todo sistema de registro debe contar con reportes o salidas hacia la pantalla, impresora o ambos. Los reportes que se incluyen en el sistema permiten realizar consulta de información enfocada a apoyar a la operación mediante la presentación detallada de proyectos, fechas de destrucción de documentación de proyectos, ubicación de paquetes y contenido de los mismos, así como la información de estatus de envíos para detectar retrasos en las fases por las que transitan estos por parte de alguna de las áreas involucradas en la operación del sistema. Los reportes o consultas del sistema permiten a cada uno de los actores del mismo, contar con las herramientas que les permitan optimizar los tiempos de respuesta así como el llevar un control más preciso del almacén. Mediante un registro correcto de los paquetes de documentación que entran o salen del mismo, así como de los espacios disponibles, se tendrán mejores resultados en la operación, tanto en tiempo como en seguimiento de los envíos.

Pantalla de selección de reporte y parámetros del mismo

Para que el usuario pueda establecer los filtros de la información requerida en la consulta, el sistema está provisto de una pantalla de selección de parámetros de reporte, misma que posteriormente permite acceder a una segunda pantalla en la que se delimita con mayor precisión el filtro que utilizara el sistema para recuperar la información. Para acceder a la pantalla de selección de parámetros para consulta, existe una liga llamada Consulta dentro del menú de acciones a realizar. (Figura 4.5.1).

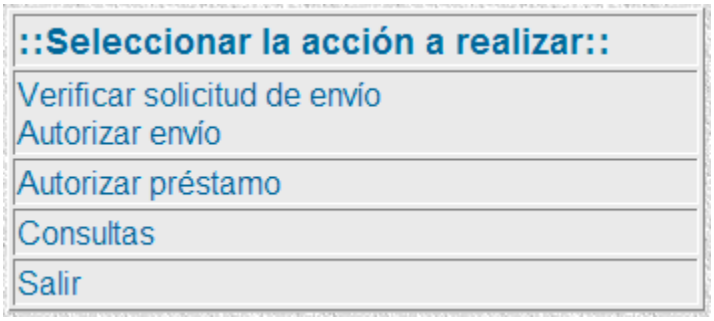


FIGURA 4.5.1 PANTALLA DE ACCESO A CONSULTAS (SE MUESTRA EL MENÚ DE RESPONSABLE)

La pantalla de Consultas permite seleccionar el tipo de parámetro por el que se desea realizar la recuperación de registros para visualizar el reporte. (Figura 4.5.2)

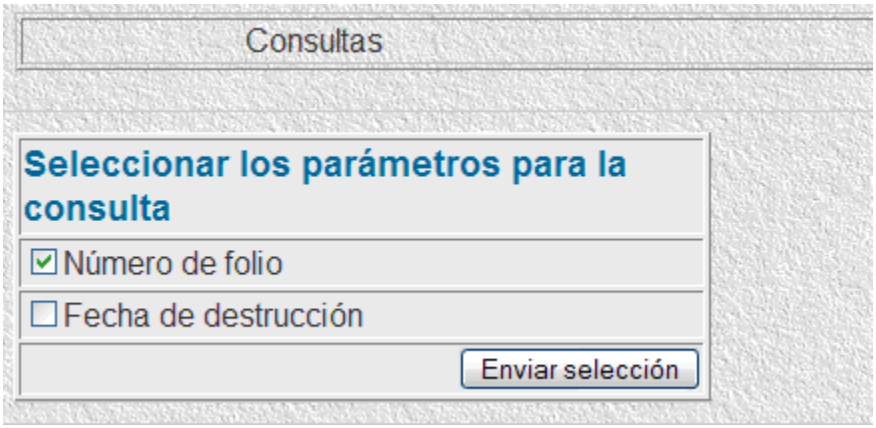


FIGURA 5.4.2 PANTALLA DE CONSULTAS

El siguiente paso es seleccionar, de acuerdo con el tipo de parámetro seleccionado, el filtro con el que el sistema generará la consulta. (Figura 4.5.3)

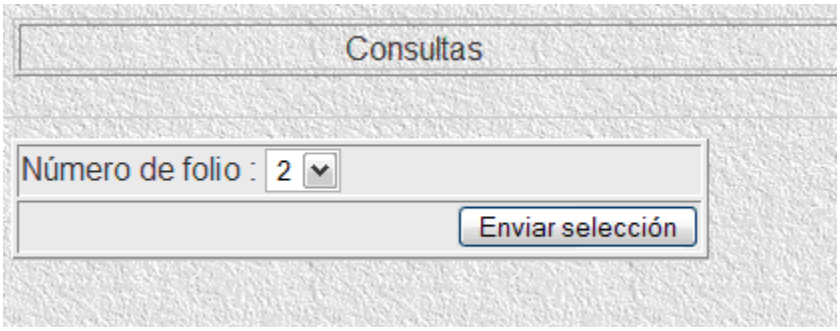


FIGURA 4.5.2 DE ACUERDO CON EL TIPO DE PARÁMETRO SELECCIONADO, LA PANTALLA MUESTRA LAS OPCIONES PARA FILTRAR LA INFORMACIÓN

Finalmente la pantalla con el Resultado de la consulta (Figura 4.5.4) muestra cuatro secciones:

- La primera contiene la información correspondiente al envío.
- La segunda es para la identificación del proyecto.
- La tercer parte corresponde al paquete e incluye su ubicación, fecha de destrucción, su estatus y su contenido.
- La última parte indica el usuario responsable del envío.

Resultado de la consulta

Envio
Folio: 2
Estatus: archivado
Comentario:

Proyecto
Nombre: norte sonora
Código contable: proy000123

Paquete
Ubicación: c3p1a5r3l8a
Fecha de destrucción (dd/mm/yy): 2/Abril/2011
Estatus: en_archivo
Descripción de su contenido: paquete que contiene información contable, fiscal, de ingeniería, de la propuesta y se da una extensa explicación de la problematica que se presentó con el área de estimados y procuración en cuento a todo el material utilizado en la construcción.

Usuario
Nombre: edgar garces corella

FIGURA 4.5.4 RESULTADO DE LA CONSULTA

El reporte del que disponen los usuarios del sistema permite tener control operativo sobre los proyectos, envíos y espacios del almacén, ya que se definió así en esta primera etapa del desarrollo.

Para siguientes etapas o versiones del sistema puede incluirse, por ejemplo, el registro preciso de fechas en el envío, autorización y archivado de los documentos, el almacenamiento de paquetes de documentos relacionados entre si y separados físicamente dentro del envío, el préstamo de paquetes de proyectos con fechas de retorno de dicha información, lo que permitiría generar reportes estadísticos sobre los tiempos de respuesta en la gestión de los envíos, así como la frecuencia en los préstamos de los proyectos, con el fin de llevar un control y tener un conocimiento mayor del movimiento de paquetes dentro del almacén general de la empresa.

4.6 FACTIBILIDAD TÉCNICA Y OPERATIVA

Factibilidad Técnica

La factibilidad técnica se basó en la evaluación del equipo con el que se cuenta, así como las herramientas de software necesarias para que pueda desarrollarse el sistema, para una aplicación Web, utilizando Sql Server 2000, Tomcat servidor Web, J2EE SDK y JCreator para la parte de desarrollo además de Firefox - navegador web

En la tabla 4.6.1 se presentan los requerimientos mínimos necesarios para que SQL Server 2000 edición estándar pueda ejecutarse.

Procesador	Procesador de la familia Pentium con 133Mhz de velocidad o mayor.
Sistema Operativo	• Windows Server 2003, Estándar Edition, Enterprise Edition o Datacenter Edition. • Windows® 2000 Server, Advanced o Datacenter Server. • Windows NT® Server version 4.0 Standard o Enterprise Edition whit Service Pack 5 (SP5) o mayor.
Memoria RAM	64 MB
Espacio en Disco Duro	270 MB
Unidades	CD-ROM

TABLA 4.6.1 REQUERIMIENTOS PARA MICROSOFT SERVER 2000

La tabla 4.6.2 se muestran los requerimientos necesarios para JCreator - Editor de desarrollo.

Procesador	Pentium II o equivalente
Memoria RAM	64 MB
Sistemas Operativos	• Windows 95 • Windows 98 • Windows ME • Windows 2000 • Windows NT 4.0 • Windows XP (recomendado)
Internet	Internet Explorer 4 or later

TABLA 4.6.2 REQUERIMIENTOS PARA JCREATOR

En la tabla 4.6.3 se muestran los requerimientos necesarios para Firefox

Procesador	Pentium 233 MHz
Sistemas Operativos	• Windows 98 • Windows 98 ME • Windows ME • Windows NT 4.0 • Windows 2000 • Windows XP (*Recommended*) • Windows Server 2003
Memoria RAM	• 64 MB RAM • 52 MB en espacio de disco duro
Unidades	CD-ROM

TABLA 4.6.3 REQUERIMIENTOS PARA FIREFOX

En la tabla 4.6.4 se muestran los requerimientos necesarios para J2EE SDK - Kit de desarrollo de Java.

Procesador	INTEL Pentium II 300MHz o superior
Memoria RAM	128 MB
Sistemas Operativos	• Tamino X-Bridge • Windows NT 4.0 • Windows 2000 Professional • Windows 2000 Server. • JRE 1.3

TABLA 4.6.4 REQUERIMIENTOS NECESARIOS PARA J2EE SDK

Para la operación óptima del sistema y de la base de datos, se deben cubrir los requerimientos en un nivel mayor al mínimo requerido por los elementos y herramientas de software que servirán de soporte y plataforma de desarrollo para el sistema.

Se cuenta con personal con conocimientos necesarios de administración, así como personal necesario para el mantenimiento de la aplicación y atención a nuevos requerimientos y atenciones del mismo.

Factibilidad Operativa

La operación del sistema deberá satisfacer los siguientes puntos de operación

- **Desempeño**. De acuerdo a las plataformas sobre las que se instalará el sistema, el rendimiento es adecuado, tomando en cuenta que el volumen de la información es acorde a los recursos con los que cuenta el servidor
- **Economía**. Debido a la existencia de licencias para el software empleado, al igual que de personal que tiene los conocimientos sobre las herramientas, se espera contar con tiempos de atención cortos, además de no requerir personal adicional

- **Eficiencia**. Con este sistema se disminuyen los tiempos de respuesta a requerimientos en otras áreas, debido a la automatización de muchos de los proceso, que con anterioridad tomaban un tiempo mucho mayor para su ejecución
- **Control**. Toda la información del departamento estará almacenada en una base de datos segura, que evita la pérdida de control de la información por el uso de múltiples archivos de texto que puedan contener información duplicada o errónea por falta de validaciones
- **Información**. Permitirá tener un recurso centralizado donde toda esta información sea almacenada, podrá ser consultada y actualizada por el administrador que opere el sistema, afectando solo una fuente de datos

Como conclusión del estudio, es posible plantear que las aplicaciones de software y sistemas operativos sobre los que se desarrollo y soportan la aplicación, tienen la madurez y seguridad óptimas, y de acuerdo a las características de hardware con las que se cuenta se lograrán niveles de rendimiento y capacidades adecuados para la prestación de un buen servicio. Por lo anterior se considera factible la implantación del sistema de gestión de información.

Tipos de Mantenimiento

El mantenimiento es un aspecto importante en el diseño y desarrollo de sistemas. Una vez terminada la fase de desarrollo e implantación del sistema, es necesario garantizar el mantenimiento del mismo.

Los programas deben ser modificados con el tiempo ya que las necesidades de la empresa varían, además es necesario adaptarse al entorno o cambios tecnológicos.

Existen diferentes tipos de mantenimiento aplicables al sistema, entre ellos mencionaremos: Mantenimiento Preventivo, Correctivo, Adaptativo y Perfectivo.

- **Mantenimiento preventivo**. Es la actividad en la cual se realizan ajustes a la aplicación para evitar el mantenimiento futuro, la estabilidad y confiabilidad en la operación. También es útil para proporcionar bases seguras sobre las que podrán implementarse mejoras posteriores
- **Mantenimiento correctivo**. Conjunto de actividades dedicadas a corregir defectos en el hardware o en el software detectados por el usuario cuando ocurre un error en la operación normal del sistema
- **Mantenimiento adaptativo**. Conjunto de actividades para adaptar el sistema a los cambios (Hardware o Software) en su entorno tecnológico. Se presenta cuando se generan cambios en los requerimientos de tal manera que la especificación sea adaptada a estos nuevos requerimientos verificando que la nueva implementación cumpla con ellos, además que este opere correctamente en un nuevo hardware.
- **Mantenimiento perfectivo**. Conjunto de actividades para mejorar o añadir nuevas funcionalidades requeridas por el usuario. Se realiza cuando existe la necesidad de optimización de procesos, sin que cambien forzosamente los requerimientos. La especificación permite entender claramente el impacto de los cambios de manera que éstos se implanten confiadamente

CONCLUSIONES

De la presente tesis se concluye el haber obtenido satisfactoriamente los requerimientos del cliente que se planteo en la problemática de la Inmobiliaria; por lo que se logro crear una aplicación en términos de:

- Calidad porque satisface los requerimientos del usuario
- Economía porque utiliza los recursos propios que permitieron maximizar óptimamente las cuestiones económicas
- Versatilidad en la aplicación que le permite trabajar sin mayores problemas con otras aplicaciones
- Multiplataforma porque corre en los diversas plataformas de la Inmobiliaria

El sistema de gestión de información controla la organización, almacenamiento, recuperación, seguridad e integridad de la base de datos de la información documental de todos los proyectos que sean llevados a cabo por cualquier empresa inmobiliaria de manera eficaz.

Utilizando la metodología RUP, que tiene por objetivo usar lenguaje natural durante los procesos iniciales del desarrollo como lo es el análisis de requerimientos, permite a los usuarios del sistema, junto con los desarrolladores, delimitar los alcances del mismo, así como validar su funcionalidad antes del inicio de su construcción. Estas expectativas se cumplieron, ya que al concluir el desarrollo del sistema se entregó la funcionalidad del mismo de acuerdo con el análisis de las etapas iniciales, realizando solo pequeños ajustes que no impactaron en los tiempos de desarrollo.

Partimos de un problema de almacenamiento de paquetes de documentos en los que no existían o no estaban estandarizados los formatos de registro de entrada/salida de almacén así como de ubicación de paquetes y consulta de información de los mismos, entregando un sistema en el que se puede llevar la historia de archivo de la documentación de un proyecto partiendo de la solicitud

de envío, el registro de su contenido hasta la entrega y almacenamiento dentro del archivo general.

La utilización de una metodología de diseño orientado a objetos, permite tener un marco de referencia para basar un adecuado análisis y diseño del sistema a desarrollar sin importar el lenguaje de programación en el que se implementará, esto es, el esfuerzo se enfoca a identificar las necesidades solicitadas y a definir la interacción de las partes componentes para satisfacerlas.

La adecuada identificación de los artefactos "Casos de Uso", nos permitió definir bien los requerimientos del sistema a desarrollar, así como definir el acotamiento de lo que se iba a desarrollar.

Una vez elaborado el análisis y el diseño del sistema a través de los casos de uso, diagrama de clases, diagrama de actividades y diagrama de secuencia, la adecuada selección de la arquitectura en la que se basará la implementación permite tener una visión de como se integran los componentes desarrollados para alcanzar la funcionalidad requerida por el sistema.

La utilización de herramientas de software libre, permite reducir costos en la implantación y la puesta en funcionamiento del sistema desarrollado, esto cumpliendo con unos de los requerimientos indicados de utilizar las herramientas disponibles y reducir al mínimo el costo del desarrollo.

A partir de los resultados obtenidos durante el desarrollo e implantación del sistema de gestión de información, se llegan a las siguientes conclusiones:

La experiencia adquirida en el ámbito profesional nos permite valorar la calidad de la educación que me brindo la universidad, desde el aspecto humanístico hasta el técnico.

El Programa de Apoyo a la Titulación nos brindo la ayuda y orientación necesaria para poder finalizar un ciclo y tener la oportunidad de comenzar otro.

MANUAL TÉCNICO

Introducción

La habilidad de la empresa para saber cómo coordinar y explotar sus recursos humanos y el saber integrar sus prácticas de gestión con dichos recursos mejor que la competencia, o el aprender a hacerlo de formas más rápida que los demás puede ser la clave en su éxito competitivo. Esto no resulta ajeno para las Empresas, ya que a pesar de no competir directamente con otras instituciones al realizar las funciones asignadas, debe saber integrar sus recursos para poder lograr el éxito requerido a costos menores. Resulta difícil concebir que esto se pueda lograr si no es apoyados en un Sistema Automatizado Moderno, ya que este debe ayudar a la Gestión de los procesos que se requieren de una forma sencilla e integral, esto es: que evite la duplicidad de actividades e información, que permita compartir la información y distribuir las cargas de trabajo a través de las áreas encargadas de la Gestión, que sea una herramienta flexible, amigable, de fácil entendimiento y operación, que considere el envío, recepción y almacenaje de los paquetes que son enviados a oficinas centrales por cada uno de los proyectos que se presentan en la Empresa.

Descripción del Sistema

El sistema de gestión de información cubre ampliamente el Manejo de los recursos de captura envío, recepción y almacenamiento de paquetes, pudiendo implantar de manera separada o integral cada uno de estos módulos. Su diseño guarda una homogeneidad total en el aspecto técnico y funcional en sus diferentes módulos, lo que permite eliminar la duplicidad de información y funciones, aspectos que a la larga provocan inconsistencias o diferencias. Esto mismo, permite compartir de manera eficiente la información a los usuarios y facilita su mantenimiento.

Objetivo del Sistema

Herramienta de Apoyo para el Manejo de los recursos de la Empresa, ubicándose como el medio y no el fin de sus funciones sustantivas, que ayuda a simplificar y ubicar en sus ámbitos los procesos, a aprovechar eficientemente las fuentes de información, y que proporciona elementos de análisis que permiten su mejora continua.

Arquitectura del Sistema

El sistema de gestión de información es una aplicación gráfica con arquitectura Web, que utiliza un servidor donde se encuentra la base de datos relacional, cuenta también con un servidor de paginas Web que hace la conexión con el usuario mediante la red y un explorador ya que se maneja código html para la aplicación como lo muestra la figura siguiente.

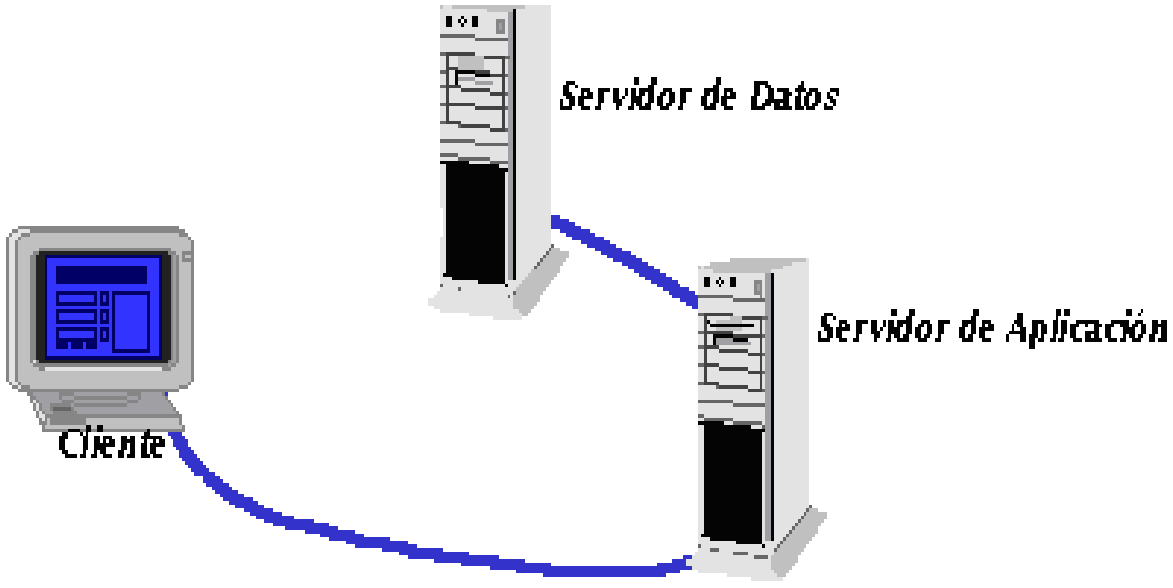


FIGURA: ASISTENTE DE MANTENIMIENTO DE LA BASE DE DATOS

El Sistema de Gestión de Información está desarrollado totalmente en la plataforma Java, para la aplicación y SQL Server 2000 para la base de datos, conectándose directamente al cliente a través de la configuración de un JDBC el

cual sirve como puente para hacer la conexión al servidor Web y así poder acceder a la base de datos.

Requerimientos de Software

- Base de Datos SQL Server 2000
- JAVA
- Sistema Operativo Windows 98 o superior en equipos clientes
- Maquina Virtual de Java (opcional)

Requerimientos de Hardware

- Servidor RISC o Pentium 4 a 1 ghz o superior, Espacio en Disco de 40 GB, Memoria RAM de 1 GB
- Clientes Pentium 3 a 500 ghz o superior, Espacio en Disco de 800 MB, Memoria RAM de 128 MB
- 2 Impresoras Láser de Alta Velocidad
- 2 Impresoras de Matriz o Cadena de Alta Velocidad
- 1 Impresora de Color de Alta Resolución
- 1 unidad de respaldo (CD, Cartucho, DVD)

Como Asegurar la Disponibilidad de los Datos en SQL SERVER 2000

¿Cómo nos aseguramos que nuestra base de datos esté disponible?; no podemos garantizar que nuestra base de datos va a estar ahí cuando la necesitemos pero lo que si podemos garantizar es que si falla o algo sale mal tengamos la capacidad de restaurar todo. Pero, ¿Cómo lograr esto?, creando un buen plan de recuperación y utilizando las operaciones de copia de seguridad y restauración disponibles en SQL 2000. Los elementos básicos de una estrategia de disponibilidad de datos son:

- Planear el futuro
- Comprender los registros de transacciones y saber como utilizarlos para restaurar datos
- Realizar copias de seguridad de la base de datos y restaurar los datos

Creación de una estrategia de copia de seguridad de base de datos

Las copias de seguridad de base de datos son una parte fundamental en la creación de esta estrategia, sin una estrategia de copias de seguridad efectiva podríamos encontrarnos en una situación en que tengamos una base de datos corrupta pero no las suficientes copias de seguridad para restaurarlo. Los tipos de fallas que podrían ocurrir, entre ellas:

- Datos inválidos del usuario.
- Fallo en disco duro.
- Falla en el servidor.

Para evitar perderlo todo en una falla siga las siguientes recomendaciones:

- Realice copias de seguridad con frecuencia (esto depende del uso de la base de datos).
- Mantenga copias de seguridad completas fuera del sitio.
- Realice comprobaciones de consistencia con cierta frecuencia.
- Administre sus copias de seguridad con efectividad.

Uso de registros de transacciones en la copia de seguridad y en la restauración

Un registro de transacciones es utilizada internamente por SQL Server para llevar el registro de las actualizaciones, inserciones y eliminaciones que ocurren en la base de datos. El registro de transacciones junto con la copia de seguridad de la base de datos puede ser utilizado para restaurar una base de datos y devolverla a un estado consistente.

Modelos de recuperación

SQL Server 2000 soporta tres modelos de recuperación:

- **Completa.** El más completo, si falla el disco duro; le permite recuperarse hasta el momento justo de falla o en cualquier momento del tiempo. Para poder lograr esto todas las operaciones se registran lo que hace que el registro crezca demasiado ya que las operaciones masivas también se registran. Esta es una característica muy poderosa cuando se tiene una base de datos 24*7, nos permite asegurar que se pierda la menor cantidad de modificaciones posible
- **Registro masivo.** es una copia de seguridad completa. No obstante, si falla el disco duro, puede recuperarse con el modelo de copia masiva, pero no le permite recuperarse hasta cualquier momento del tiempo
- **Simple.** Es la más sencilla de todas, ocupa el menor espacio en disco, y es la que ocupa menos recurso del sistema, pero también lo expone a mayores pérdidas de datos, este modelo no nos permite recuperar hasta cualquier momento en el tiempo ni hasta el momento de falla

Todos estos modelos tienen ventajas y desventajas, determinar cual es el mejor depende de sus requerimientos individuales. Por ejemplo, una base de datos que tenga muchas transacciones y necesite recuperarse completamente lo mas pronto posible se beneficiaría con el modelo de recuperación completa, Una base de datos que haya tenido muchas actualizaciones masivas y no necesite recuperar las transacciones individuales de los usuarios podría utilizar el modelo de registro masivo, y por último el modelo simple se utiliza para aplicaciones que no sean cruciales o para aplicaciones en desarrollo.

Comprobación del modelo de recuperación

El modelo de recuperación se establece a la hora de crear la base y se basa en la base de datos model. Para saber cual modelo esta utilizando la base de datos se utiliza la función DATABASEPROPERTYEX de la siguiente forma:

1: SELECT DATABASEPROPERTYEX (‘nombre base de datos’,’recovery’)

Cambiar el modelo de recuperación

Para cambiar el modelo de recuperación, utilice la siguiente instrucción ALTER DATABASE

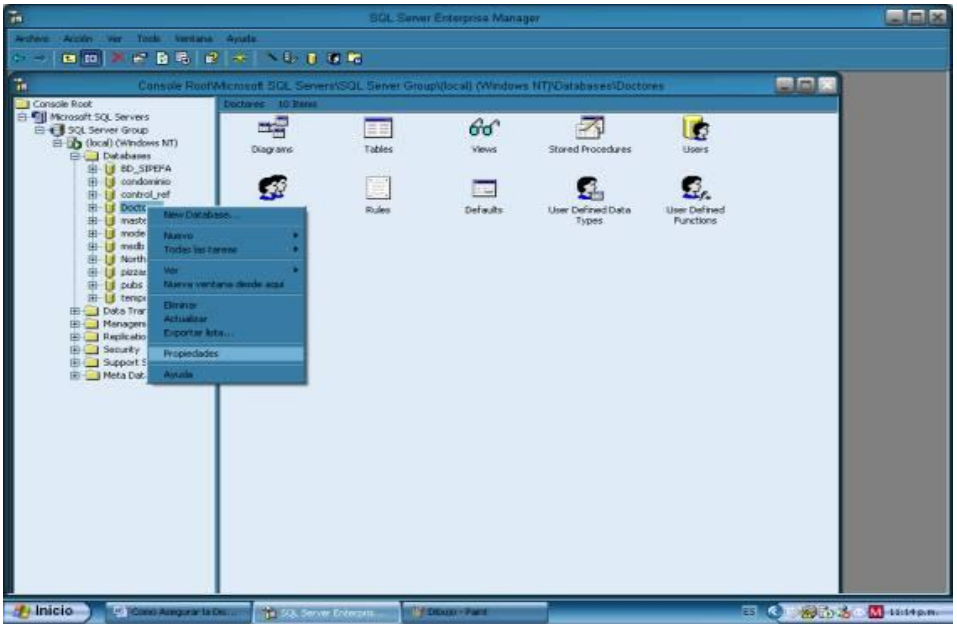
1: ALTER DATABASE *nombre_base_datos* SET RECOVERY [FULL | SIMPLE | BULK_LOGGED]

Full: Completa, Simple: simple y Bulk_logged: Registro masivo

Esta acción de cambiar el modelo de recuperación, también se puede hacer de manera visual desde el Administrador Corporativo de SQL Server:

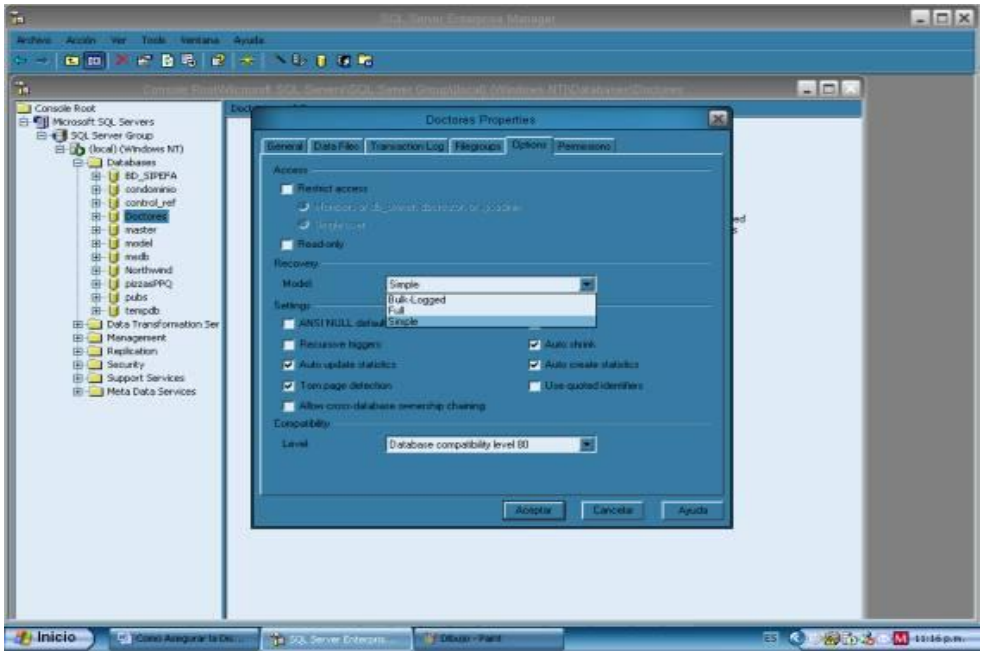
Pasos

- Abra el administrador corporativo
- Seleccione la base de datos en la que desea realizar el cambio
- Una vez ubicada, haga clic derecho y seleccione propiedades como lo muestra la figura siguiente



RECUPERACIÓN DE UNA BASE DE DATOS

Posteriormente, haga clic en la pestaña OPTIONS y seleccione, del combo, los diferentes modelos de recuperación como se muestra en la figura siguiente:



OPCIONES DE RECUPERACIÓN DE UNA BASE DE DATOS

Tipos de copias de seguridad

Aunque, se puede especificar el modelo de recuperación deseado, también se puede especificar que porciones de la base de datos incluiremos en las copias de seguridad:

- **Base de datos completa.** Es copia incluye todos los objetos de la base de datos (tables, views, stored procedure)
- **Base de datos diferencial.** Solo copia las modificaciones que se le hallan hecho a la base de datos (DML, DDL) desde la última copia de seguridad completa
- **Registro de transacciones.** Copia el registro de transacciones, todas las transacciones de los usuarios (INSERT, UPDATE, DELETE), como se tienen todas las transacciones se pueden rehacer todo el trabajo que los usuarios haya hecho
- **Archivo y grupo de archivos.** Nos permite realizar una copia de seguridad de una porción de la base de datos a la vez. Esta se debe de usar cuando la base de datos es muy extensa, también se deben realizar copia del registro de transacción para poder recuperarse

Existen dos formas de realizar la copia de la base de datos, por medio del **Enterprise Manager** o del **Query Analyzer**:

Realizar una copia de seguridad por medio del Administrador Corporativo (Enterprise Manager)

Inicie el **Enterprise Manager**

- Seleccione la base de datos que desea realizar copia de seguridad.
- Haga clic derecho y seleccione todas las tareas y posteriormente haga clic sobre Backup Database como lo muestra la siguiente figura:

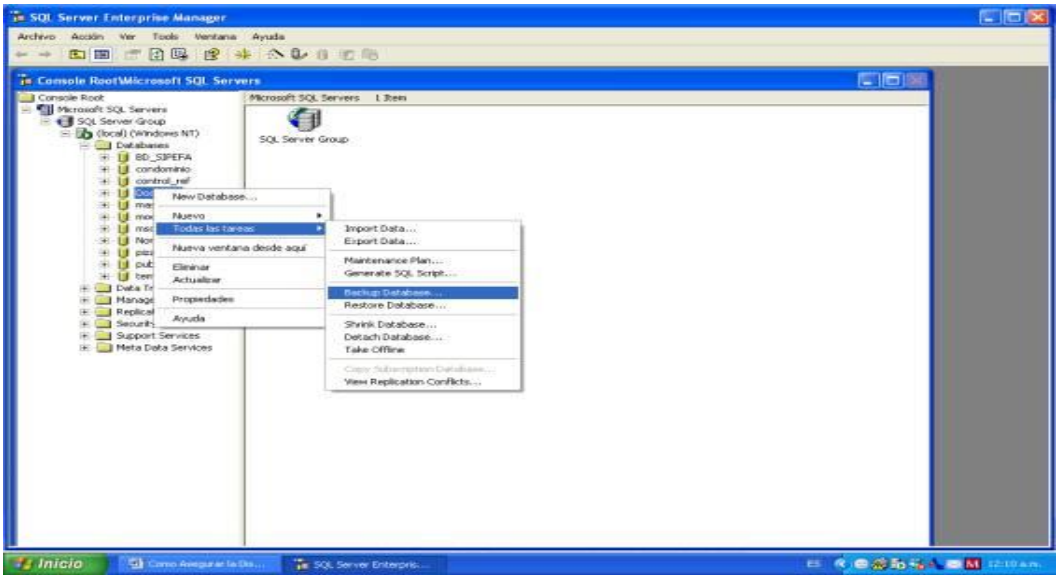


FIGURA: COPIA DE SEGURIDAD MEDIANTE EL ADMINISTRADOR CORPORATIVO

Al hacer los pasos anteriores aparece una pantalla como la siguiente figura que se muestra a continuación:

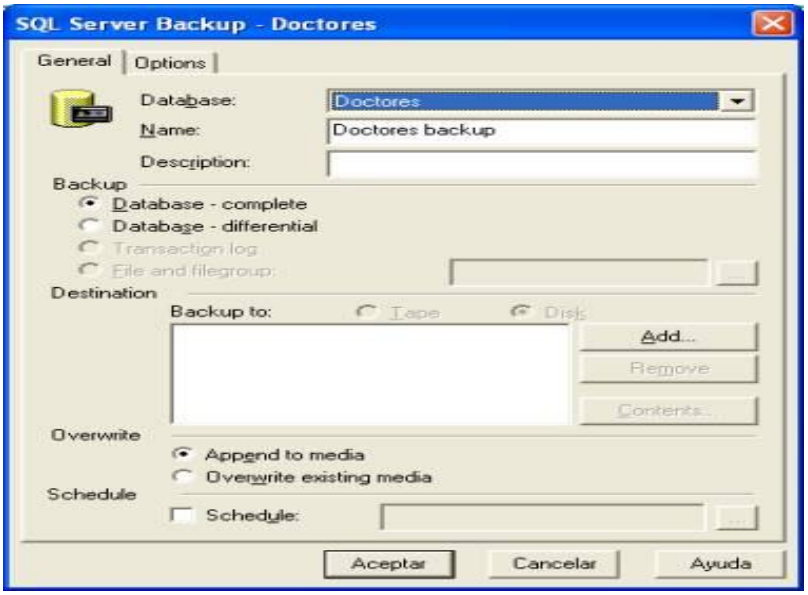


FIGURA: RESPALDO DE UNA BASE DE DATOS

Esta pantalla le permite configurar algunas cosas: lo primero es escoger cual base de datos se desea respaldar, seguidamente está el campo Name, debe establecerse un nombre para el respaldo y anotarlo en ese campo; el campo description no es necesario pero seria bueno especificarle una descripción como por ejemplo “Doctores Backup, 01/03/04, Completa”. En la sección backup debe seleccionar el tipo de copia que desea realizar, según lo vimos anteriormente.

En la sección Destination debe seleccionar el destino de la copia de seguridad, si da clic en el botón Add aparecerá la siguiente pantalla la siguiente figura.



FIGURA: RUTA DE ALMACENAMIENTO DE LOS RESPALDOS DE LA BASE DE DATOS

En la pantalla anterior puede poner la dirección del archivo o escogerlo dando clic en el botón…

El botón Remove es para remover el una copia de la base de datos de la lista.

El botón Contents muestra el número de copias de seguridad actuales.

La opcion Overwrite le permite sobrescribir las copias de seguridad anteriores.

Por último la opción Schedule, es para programar la copia a cualquier hora del día.

Una vez seleccionada todas las opciones se le da clic en botón OK y a continuación aparecerá este mensaje de progreso y al final aparece la confirmación de la copia de seguridad como se muestra a continuación en la figura:



FIGURA: PROGRESO EN LA REALIZACIÓN DE UN BACKUP

Realizar una copia de seguridad de la base de datos por medio del Analizador de Consultas

El código respectivo para realizar dicha operación es la siguiente:

```
BACKUP DATABASE Doctores

TO DISK = 'C:\Archivos de programa\Microsoft
SQLServer\MSSQL$MISQLSERVER\BACKUP\DoctoresBackup.bak'

 WITH NOINIT, NOSKIP, STATS = 10
```

Doctores: nombre de la copia

TO DISK…. Esta instrucción asigna la ruta donde va a estar la copia de seguridad.

WITH: significa que siguen algunas características.

NOINIT: que el conjunto de la copia se debe indexar y no crear uno nuevo

NOSKIP: debe verificarse el nombre del medio y sus detalles.

STATS=10: que muestre el avance del proyecto en números.

Realizar una restauración de la base de datos doctores

Es importante no solo tener el instrumento sino también utilizarlo.

Por medio del administrador corporativo

Debe iniciar el administrador dar clic derecho sobre la base de datos que se desea restaurar, seguidamente se muestra la siguiente figura.

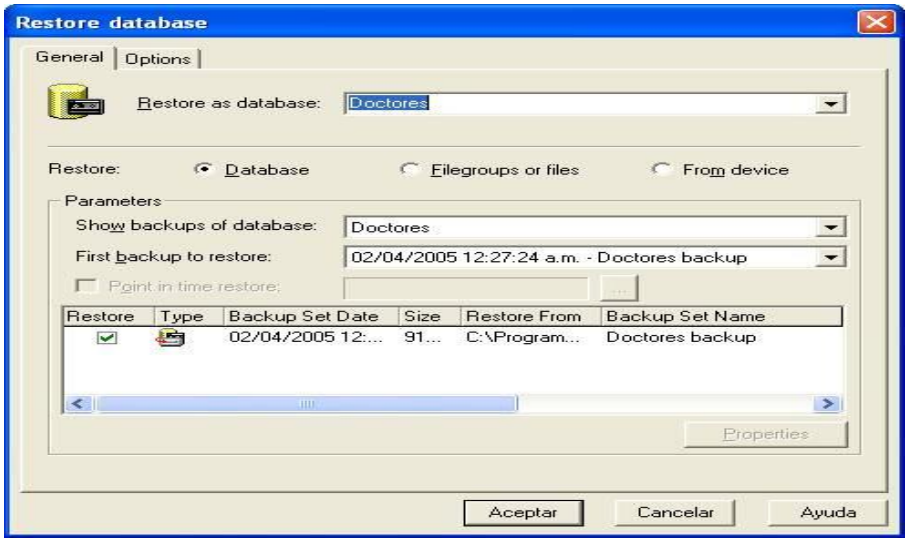


FIGURA: RESTAURACIÓN DE LA BASE DE DATOS MEDIANTE EL ADMINISTRADOR CORPORATIVO

En esta pantalla se configuran todas las opciones de restauración:

En la seccion restore pueden seleccionar el tipo de Restore.

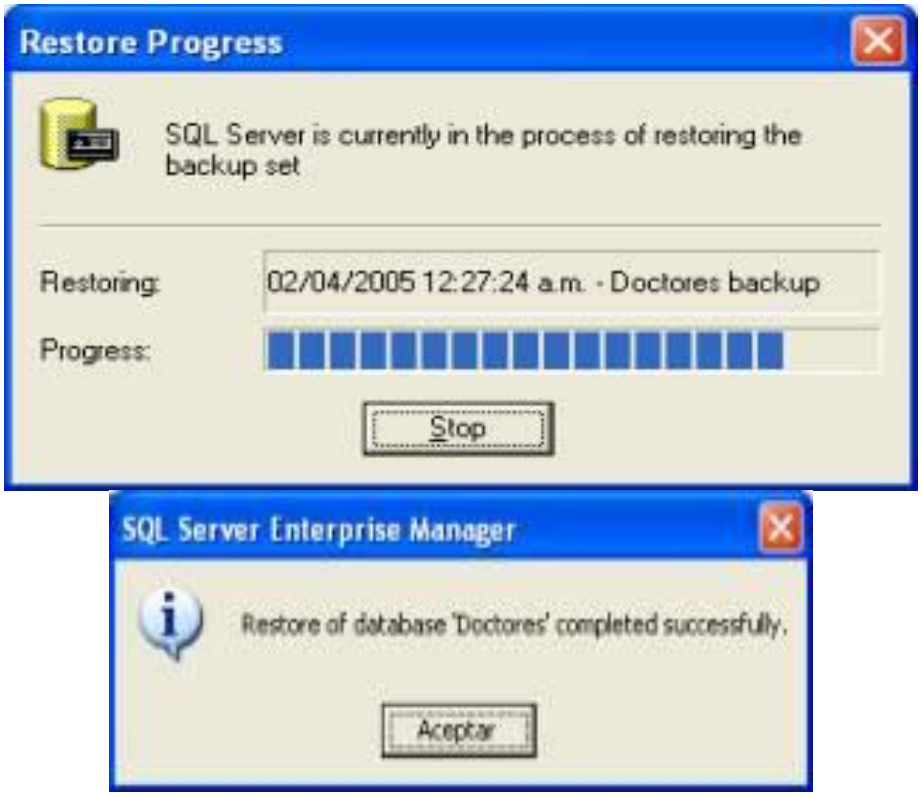
Database: nos permite restaurar la base de datos completa.

Filesgroups or Files: me permite restaurar de un archivo o grupo de archivos.

From Device: Nos permite restaurar desde un dispositivo.

El combo de más abajo en la sección de PARAMETERS contiene información sobre los respaldos mas recuentes de la base de datos.

Una vez terminado esto se le da clic en ACEPTAR y comienza con el proceso de restauración Figura 4.7.9.



4.7.9 PROGRESO EN LA RESTAURACIÓN DE UNA BASE DE DATOS

Restauración de la base de datos por medio del analizador de consultas

Para realizar la restauración, ejecute el siguiente código:

```
RESTORE DATABASE doctores

 FROM DISK = 'C:\Archivos de programa\Microsoft SQL
Server\MSSQL$MISQLSERVER\BACKUP\doctores.bak'

 WITH STATS = 10
```

Eliminar una tabla

En determinadas ocasiones, necesitará eliminar una tabla (por ejemplo, cuando desee implementar un nuevo diseño o liberar espacio en la base de datos). Cuando elimine una tabla, la definición de su estructura, sus datos, índices de texto, restricciones e índices se eliminan permanentemente de la base de datos y el espacio anteriormente utilizado para almacenar la tabla y sus índices queda disponible para otras tablas. Puede quitar explícitamente una tabla temporal si no desea esperar a que se quite automáticamente. Si necesita eliminar tablas que están relacionadas mediante restricciones FOREIGN KEY, UNIQUE o PRIMARY KEY, antes deberá eliminar las tablas con restricciones FOREIGN KEY. Si necesita eliminar una tabla a la que se hace referencia en una restricción FOREIGN KEY, pero no puede eliminar toda la tabla de clave externa, debe eliminar la restricción FOREIGN KEY.

Para eliminar una tabla

Existen distintas formas de eliminar una tabla o depurarlas para ahorrar espacio o simplemente para mantener un mejor control sobre los objetos (tablas y procedimientos) que contiene nuestra baser de datos.

Transact - SQL

- **DROP TABLE**

Quita una definición de tabla y todos los datos, índices, desencadenadores, restricciones y especificaciones de permisos de la tabla. Las vistas o procedimientos almacenados que hagan referencia a la tabla quitada se deben quitar explícitamente con la instrucción DROP VIEW o DROP PROCEDURE.

Sintaxis

DROP TABLE table_name

Argumentos

table_name

Es el nombre de la tabla que se va a quitar.

Observaciones

No se puede utilizar DROP TABLE para quitar una tabla a la que se haga referencia con una restricción FOREIGN KEY. Primero se debe quitar la restricción FOREIGN KEY o la tabla de referencia. El propietario de una tabla puede quitar la tabla de cualquier base de datos. Cuando se quita la tabla, las reglas o valores predeterminados de la misma pierden sus enlaces y se quitan automáticamente las restricciones o desencadenadores asociados con ella. Si vuelve a crear una tabla, debe volver a enlazar las reglas y valores predeterminados apropiados, volver a crear los desencadenadores y agregar todas las restricciones necesarias. No puede utilizar la instrucción DROP TABLE sobre las tablas del sistema. Si elimina todas las filas de una tabla (DELETE tablename) o utiliza la instrucción TRUNCATE TABLE, la tabla existe hasta que se quite.

Permisos

Los permisos para utilizar DROP TABLE pertenecen de manera predeterminada al propietario de la tabla y no se pueden transferir. Sin embargo, los miembros de la función fija de servidor **sysadmin** o de las funciones fijas de base de datos **db_owner** y **db_ddladmin** pueden quitar cualquier objeto si especifican el propietario en la instrucción DROP TABLE.

Ejemplos

- **Quitar una tabla de la base de datos actual**

Este ejemplo quita la tabla **titles1**, y sus datos e índices de la base de datos actual.

DROP TABLE titles1

- **B. Quitar una tabla de otra base de datos**

Este ejemplo quita la tabla **authors2** de la base de datos **pubs**. Se puede ejecutar desde cualquier base de datos.

DROP TABLE pubs.dbo.authors2

Eliminar una tabla de un diagrama de base de datos y de la base de datos a través del Administrador Corporativo

Elimine una tabla del diagrama de base de datos cuando desee realizar todas las operaciones siguientes:

- Quitar la tabla de su diagrama.
- Quitar la tabla de cualquier otro diagrama en el que aparezca.
- Marcar la tabla para su eliminación de la base de datos.

Una tabla que se marca para su eliminación se elimina permanentemente de la base de datos cuando guarde el diagrama. Continúa existiendo una referencia a esa tabla en la memoria hasta que guarde el diagrama. Si cierra el diagrama sin guardarlo, la tabla continuará existiendo en la base de datos y aparecerá en su diagrama y en cualquier otro diagrama en el que aparecía antes de que la marcara para su eliminación.

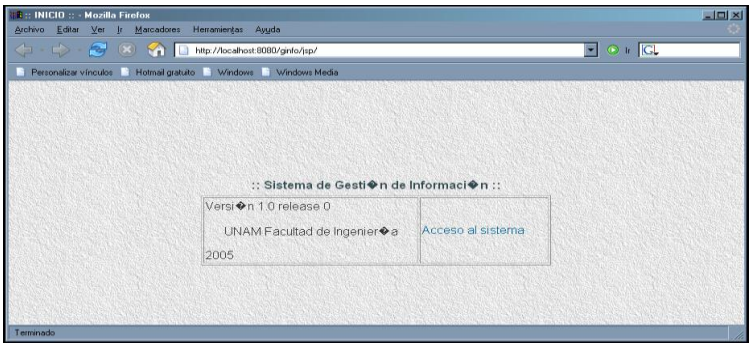
Para eliminar una tabla de la base de datos

- En el diagrama de base de datos, seleccione la tabla que desee eliminar
- Haga clic con el botón secundario del *mouse* (ratón) en la tabla y elija **Eliminar tabla de la base de datos** en el menú contextual
- Un cuadro de mensajes pregunta si desea confirmar la eliminación. Elija **Sí**

Nota: Cuando se elimina una tabla, se quitan automáticamente las relaciones que existen con ella.

Manual del Usuario del Sistema de Gestión de Información

La ubicación de esta aplicación se encuentra en la Intranet de la inmobiliaria; se tendrá que pulsar en “Contabilidad y Finanzas” aparecerán varias secciones, en la sección de “Contabilidad OM y Sistema Contable” se tendrá que pulsar la liga que dice Archivo Sistema de Gestión de Información.



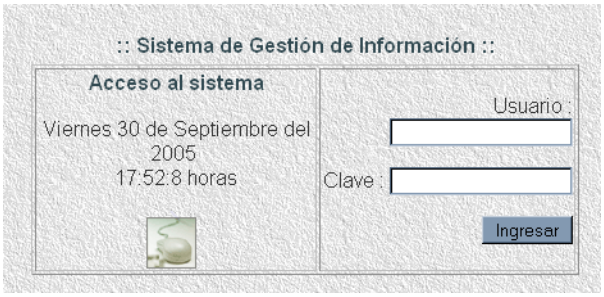
PANTALLA DE UBICACIÓN EN LA INTRANET

La aplicación consiste de 2 módulos:

- El **Administrativo** dónde se captura los catálogos ó sea los Proyectos y Usuarios. En este módulo existe un a/b/c para los catálogos
- El de la **Gestión de Información**, en dónde intervienen el Solicitador de envío, el Responsable del envío y el Almacenista

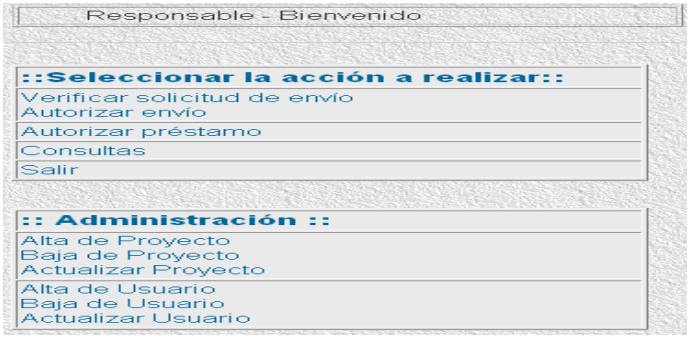
Al iniciar la sesión deberá digitar su cuenta de usuario y su clave de acceso al sistema el cual será validado para poder ejecutar la aplicación de acuerdo a su perfil autorizado.

- El usuario es el nombre del rol del empleado (a cinco caracteres) Ejemplo: test1.
- Una vez capturado su cuenta de usuario y su clave de acceso deberá digitar el botón Ingresar

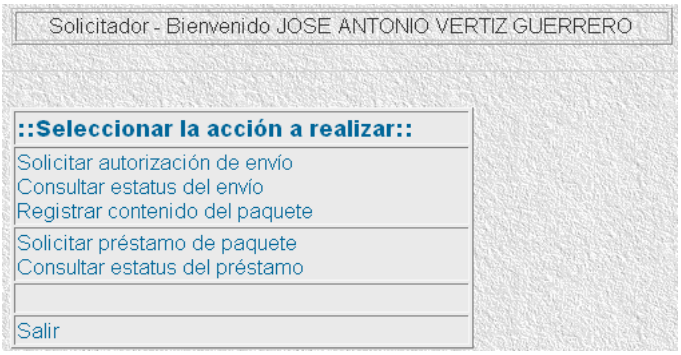


PANTALLA DE ACCESO AL SISTEMA A USUARIOS

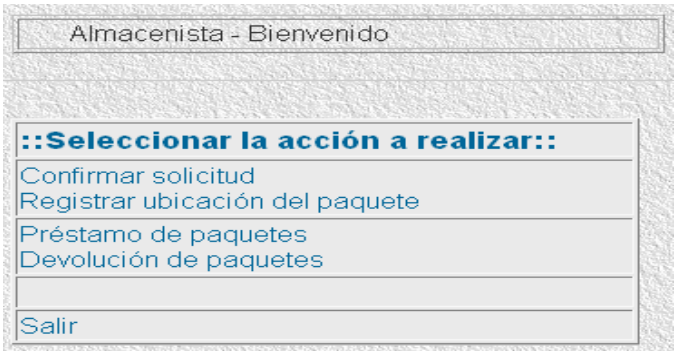
Una vez que se ha ingresado al Sistema de Gestión de Información le dará la bienvenida indicándole cual es su perfil de usuario. Para los tres tipos de roles que son el de responsable, almacenista y el solicitador según corresponda. Por ejemplo el usuario test1 su perfil correspondiente es el de responsable como se muestra a continuación:



PANTALLA DEL PERFIL DE RESPONSABLE



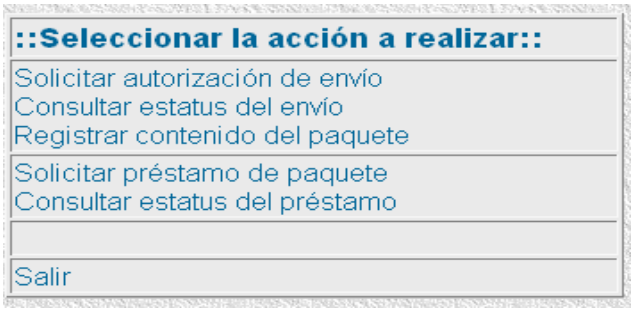
PANTALLA DEL PERFIL DE SOLICITADOR



PANTALLA DEL PERFIL DEL ALMACENISTA

Las opciones con las que cuenta el usuario de rol “Solicitador” son las siguientes:

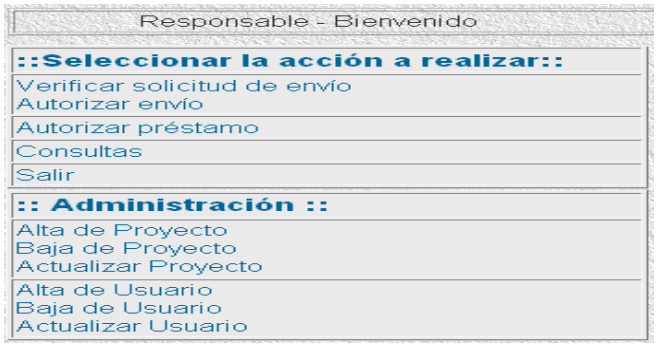
- Solicitar autorización de envío de paquete
- Consultar estatus del envío del paquete
- Registrar contenido del paquete
- Solicitar préstamo de paquete de proyecto previo
- Consultar estatus del préstamo del paquete
- Salir del Sistema de Gestión



PANTALLA MENÚ DE OPCIONES DEL SOLICITADOR

Las opciones con las que cuenta el usuario de rol “Responsable” son las siguientes:

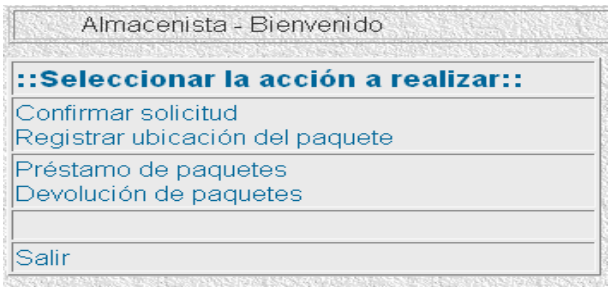
- Verificar solicitud de envío
- Autorizar envío
- Autorizar préstamo
- Consultas
- Salir
- Alta de proyecto
- Baja de proyecto
- Actualizar proyecto
- Alta de usuario
- Baja de usuario
- Alta de usuario



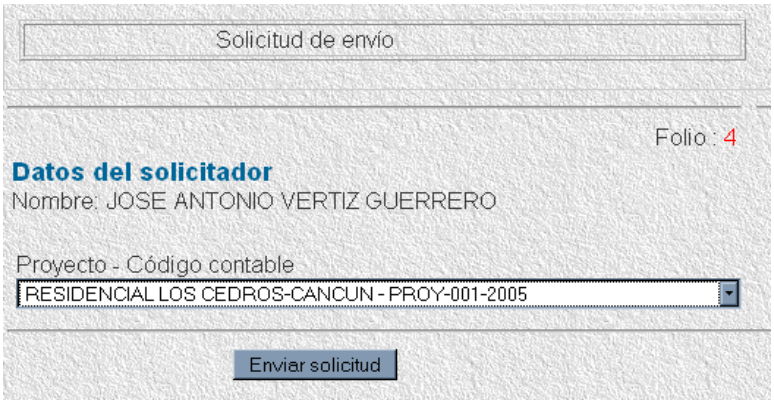
PANTALLA DEL MENÚ DE OPCIONES DEL RESPONSABLE

Las opciones con las que cuenta el usuario de rol “Almacenista” son las siguientes:

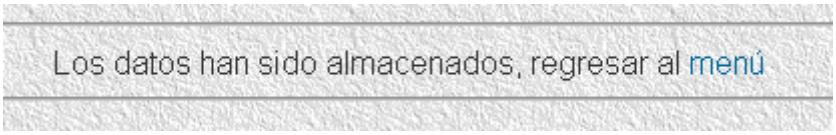
- Confirmar solicitud
- Registrar ubicación del paquete
- Préstamo de paquetes
- Devolución de paquetes
- Salir del sistema



Para una solicitud de envío de paquetes en el rol del “Solicitador” nos mostrará la siguiente pantalla; donde deberá elegirse el proyecto de acuerdo al catálogo y una vez seleccionado digitar Enviar solicitar. Obsérvese que el sistema nos indica el número de folio de la solicitud de envío.



PANTALLA SOLICITUD DE ENVÍO

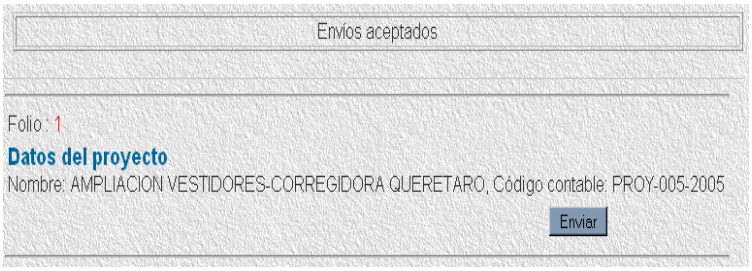


El sistema nos indica que se almacenaron los datos y nos indica regresar al menú principal.

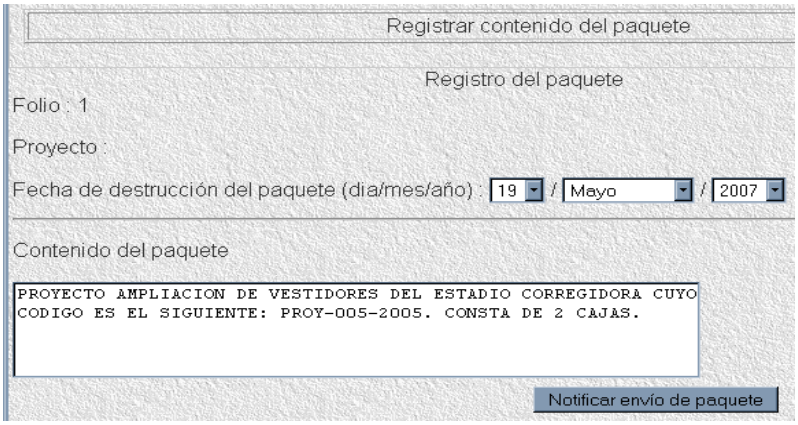
Si deseamos consultar el status del envío posteriormente nos mostrará la siguiente pantalla; para proseguir de acuerdo a los procedimientos establecidos.



Una vez que han sido aceptados los envíos aparecerá en la pantalla como se muestra a continuación:



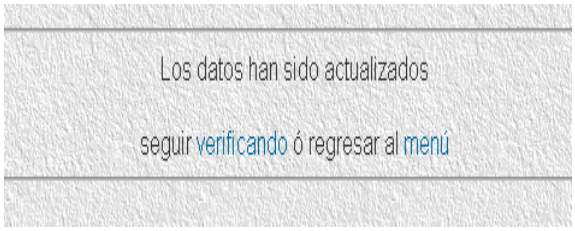
Al digitar el botón de enviar nos indicará que registremos fecha de destrucción del paquete así como el contenido del mismo.



Para el caso del rol de “Administrador” se mostrará como se muestra a continuación una vez que da seguimiento a las solicitudes de envío donde podrá indicar sí valida espacio o rechaza un envío así como escribir los comentarios del motivo de aceptación del espacio o rechazo para lo cual únicamente deberá clic en el botón de enviar.



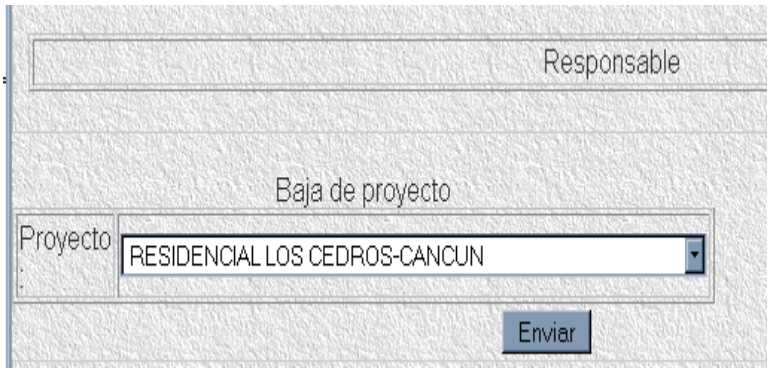
En cualquiera de los casos la confirmación será la siguiente



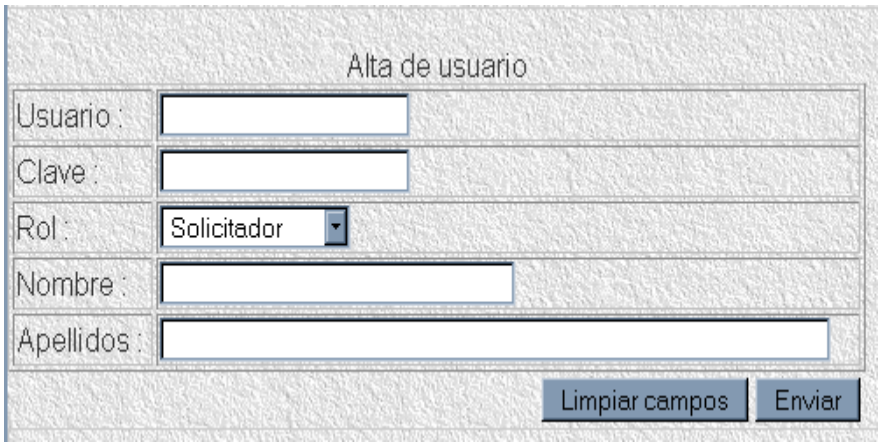
Para dar de alta un proyecto el responsable una vez que elige en el menú esa opción tendrá que capturar la descripción del proyecto y asignar el código contable conforme al catálogo. Dar clic en el botón Enviar donde el sistema nos mostrará que han sido almacenados los datos del proyecto. Puede limpiar los campos en caso de error de captura.



Para que el Administrador de una baja de algún proyecto bastará con indicar el nombre del proyecto bajo el catálogo y dar clic en Enviar, donde el sistema nos indicará que ha sido de baja.



Se podrá dar de alta usuarios bajo el rol correspondiente de la siguiente manera, capturar cuenta de usuario y su clave correspondiente; indicar bajo catálogo su rol indicando nombre y apellidos el sistema le permite limpiar campos en caso de error de captura y una vez finalizada la captura dar clic en el botón Enviar.



Podrá el responsable actualizar los datos de los usuarios en el menú Actualizar y bastará una vez corregidos los datos con dar clic en el botón Enviar, para lo cual el sistema confirmará dicha actualización.

Para el rol de “Almacenista” en la opción de confirmar solicitud bastará con validar el espacio correcto o asignar en espera los envíos y dar clic en el botón de Enviar como se muestra a continuación.



De igual manera el “Almacenista” indicará el préstamo o devolución de paquetes.

BIBLIOGRAFÍA

Utilización del UML en ingeniería de software con objetos y componentes

Perdita Stevens, Rob Pooley// Addison Wesley 2002

Análisis y Diseño de Sistemas

Kendall y Kendall // Prentice Hall

Edición 1995

Procedimientos Informáticos en Sistemas Empresariales

Frank J. Clark y Robert Gray // Prentice Hall International

Edición 1972

An Introduction to Programming and Object-Oriented Design Using Java

Jaime Niño y Frederick A. Hosch // Editorial Wiley

Second Edition 2002

El Arte de Probar el Software (The Art of Software Testing)

Glenford J. Myers // El Ateneo

1983 (John Wiley & Sons, Inc. 1979)

Software Engineering with Student Project Guidance

Barbee Teasley Mynatt // Prentice-Hall International Editions

1990

Software Testing Techniques

Boris Beizer y Van Nostrand Reinhold (N.Y.) Roger S. Pressman

Segunda edición 1990

Software Engineering: A Practitioner's Approach

McGraw-Hill Intl. Eds.

1987

Elementos de Administración

José Antonio Fernández Arena // editorial Diana, México

1986

Organización de empresas, análisis, diseño y estructura

Benjamín Franklin Finkowsky, Mc Graw Hill, México

El círculo de comunicaciones de Token-Ring
Ing. Jorge Mondragón // PC/TIPS
Año 2 Número 16, 15/May/1989

LAN Manager: La nueva onda de comunicación
Ing. Jorge Mondragón // PC/TIPS
Año 2 Número 18, 15/Jul/1989

Tipos de redes
José Daniel Sánchez Navarro // Serie Enter: El camino fácil a Internet
Capítulo 1, pp. 3-7, 14
Editorial McGraw Hill
Feb/1996

Sistemas de comunicaciones electrónicas
Wyne Tomáis // Prentice Hall
Segunda edición, 1996
Redes de computadoras
Andrew Tanenbaum // Prentice Hall
Tercera edición, 1997
Business Data Communication; W. Stallings // Prentice Hall
Cuarta edición, 2000
Backbone applications guide
ADC Fibermux // Fibermux Corp.
Primera edición, 1992
Fundamentos del diseño y la programación orientada a objetos
Sergio M Fernández Sastre, editorial McGraw Hill
2003

Organización de empresas, análisis, diseño y estructura, Benjamín Franklin Finkowsky, Mc Graw Hill, México.

Logística Empresarial. Control y planificación. RONALD H. Ed. Díaz de Santos, S.A., Madrid, 1991. pp. 59-87.

La integración cliente - empresa: las cinco mejores estrategias para lograr el éxito en los negocios con los clientes, WHITELEY, RICHARD, México. Editorial Prentice-Hall, 1996.

Ingeniería del Software. Un enfoque practico. Presuman, Roger S. Editorial McGraw_Hill. México 1997.

Introducción a los Sistemas Operativos. Harry M. Deitel Editorial Addisson Wesley Iberoamericana, edición especial 1999.

Direcciones electrónicas consultadas

http://www.mozilla.org/products/firefox/

http://www.jcreator.com/index.htm

http://java.sun.com/j2ee/index.jsp

http://jakarta.apache.org/tomcat/index.html

http://www.integraconsulting.com.mx/pestrategica.htm

http://www.itlp.edu.mx/publica/tutors.htm

http://ricoverimarketing.americas.tripod.com/RicoveriMarketing/id35.html

http://www.coworker.com.mx/nota.asp

http://www.elguille.info/colabora/puntoNET/canchala_FundamentosPOO

http://programarenc.webcindario.com

http://www.ilustrados.com/publicaciones

http://www.soyentrepreneur.com/serviciocliente/serviciocliente.shtmlBALLOU,

http://www.omg.org/uml/

http://www.elguille.info/colabora/puntoNET/canchala_FundamentosPOO

http://orbita.starmedia.com/~unamosapuntes/organigramas/organigramas.htm

http://www.sba.gov/espanol/Primeros_Pasos/proyectandoelfuturo.html

http://www.geocities.com/svg88/plan2.html#misi